
ISC


BIND 9

# BIND 9 Administrator Reference Manual

***Release 9.20.29***

**Internet Systems Consortium**

**Sep 16, 2026**

# CONTENTS


**1 Introduction to DNS and BIND 9** **1**
- 1.1 Scope of Document . . . 1
- 1.2 Organization of This Document . . . 1
- 1.3 Conventions Used in This Document . . . 2
- 1.4 The Domain Name System (DNS) . . . 2
- 1.5 DNS Security Overview . . . 7

**2 Resource Requirements** **9**
- 2.1 Hardware Requirements . . . 9
- 2.2 CPU Requirements . . . 9
- 2.3 Memory Requirements . . . 9
- 2.4 Name Server-Intensive Environment Issues . . . 9
- 2.5 Supported Platforms . . . 9
- 2.6 Unsupported Platforms . . . 11
- 2.7 Installing BIND 9 . . . 11

**3 Configurations and Zone Files** **13**
- 3.1 Introduction . . . 13
- 3.2 Authoritative Name Servers . . . 16
- 3.3 Resolver (Caching Name Servers) . . . 21
- 3.4 Load Balancing . . . 29
- 3.5 Zone File . . . 30

**4 Name Server Operations** **37**
- 4.1 Tools for Use With the Name Server Daemon . . . 37
- 4.2 Signals . . . 41
- 4.3 Plugins . . . 41
- 4.4 Configuring Plugins . . . 41
- 4.5 Developing Plugins . . . 42

**5 DNSSEC** **43**
- 5.1 Zone Signing . . . 43
- 5.2 Secure Delegation . . . 46
- 5.3 DNSSEC Validation . . . 47
- 5.4 Dynamic Trust Anchor Management . . . 48
- 5.5 PKCS#11 (Cryptoki) Support . . . 49

**6 Advanced Configurations** **55**
- 6.1 Dynamic Update . . . 55
- 6.2 NOTIFY . . . 56
- 6.3 Incremental Zone Transfers (IXFR) . . . 56

6.4 Split DNS . . . 56
6.5 IPv6 Support in BIND 9 . . . 59
6.6 Dynamically Loadable Zones (DLZ) . . . 60
6.7 Dynamic Database (DynDB) . . . 62
6.8 Catalog Zones . . . 63
6.9 DNS Firewalls and Response Policy Zones . . . 67

**7 Security Configurations** **77**
7.1 Security Assumptions . . . 77
7.2 Access Control Lists . . . 78
7.3 `Chroot` and `Setuid` . . . 79
7.4 Dynamic Update Security . . . 80
7.5 TSIG . . . 80
7.6 SIG(0) . . . 82

**8 Configuration Reference** **83**
8.1 Configuration File (named.conf) . . . 83
8.2 Blocks . . . 88
8.3 Statements . . . 257
8.4 Statements by Tag . . . 285
8.5 BIND 9 Statistics . . . 311

**9 Troubleshooting** **321**
9.1 Common Problems . . . 321
9.2 Incrementing and Changing the Serial Number . . . 322
9.3 Where Can I Get Help? . . . 323

**10 Building BIND 9** **325**
10.1 Required Libraries . . . 325
10.2 Optional Features . . . 326
10.3 macOS . . . 327

**11 DNSSEC Guide** **329**
11.1 Preface . . . 329
11.2 Introduction . . . 330
11.3 Getting Started . . . 335
11.4 Validation . . . 336
11.5 Signing . . . 348
11.6 Basic DNSSEC Troubleshooting . . . 369
11.7 Advanced Discussions . . . 377
11.8 Recipes . . . 389
11.9 Commonly Asked Questions . . . 408

**12 Manual Pages** **411**
12.1 arpaname - translate IP addresses to the corresponding ARPA names . . . 411
12.2 ddns-confgen - TSIG key generation tool . . . 411
12.3 delv - DNS lookup and validation utility . . . 412
12.4 dig - DNS lookup utility . . . 417
12.5 dnssec-cds - change DS records for a child zone based on CDS/CDNSKEY . . . 427
12.6 dnssec-dsfromkey - DNSSEC DS RR generation tool . . . 430
12.7 dnssec-importkey - import DNSKEY records from external systems so they can be managed . . . 432
12.8 dnssec-keyfromlabel - DNSSEC key generation tool . . . 433
12.9 dnssec-keygen: DNSSEC key generation tool . . . 437
12.10 dnssec-ksr - Create signed key response (SKR) files for offline KSK setups . . . 441
12.11 dnssec-revoke - set the REVOKED bit on a DNSSEC key . . . 443

12.12 dnssec-settime: set the key timing metadata for a DNSSEC key . . . 444
12.13 dnssec-signzone - DNSSEC zone signing tool . . . 447
12.14 dnssec-verify - DNSSEC zone verification tool . . . 453
12.15 dnstap-read - print dnstap data in human-readable form . . . 454
12.16 filter-aaaa.so - filter AAAA in DNS responses when A is present . . . 455
12.17 host - DNS lookup utility . . . 456
12.18 mdig - DNS pipelined lookup utility . . . 458
12.19 named-checkconf - named configuration file syntax checking tool . . . 462
12.20 named-checkzone - zone file validation tool . . . 463
12.21 named-compilezone - zone file converting tool . . . 466
12.22 named-journalprint - print zone journal in human-readable form . . . 469
12.23 named-nzd2nzf - convert an NZD database to NZF text format . . . 470
12.24 named-rrchecker - syntax checker for individual DNS resource records . . . 470
12.25 named.conf - configuration file for **named** . . . 474
12.26 named - Internet domain name server . . . 494
12.27 nsec3hash - generate NSEC3 hash . . . 497
12.28 nslookup - query Internet name servers interactively . . . 497
12.29 nsupdate - dynamic DNS update utility . . . 500
12.30 rndc-confgen - rndc key generation tool . . . 506
12.31 rndc.conf - rndc configuration file . . . 507
12.32 rndc - name server control utility . . . 509
12.33 tsig-keygen - TSIG key generation tool . . . 517

**13 General DNS Reference Information** **519**
13.1 Requests for Comment (RFCs) . . . 519
13.2 Notes . . . 524
13.3 Internet Drafts . . . 524

**14 Release Notes** **525**
14.1 Introduction . . . 528
14.2 Supported Platforms . . . 528
14.3 Download . . . 529
14.4 Known Issues . . . 529
14.5 Notes for BIND 9.20.29 . . . 529
14.6 Notes for BIND 9.20.28 . . . 533
14.7 Notes for BIND 9.20.27 . . . 533
14.8 Notes for BIND 9.20.26 . . . 534
14.9 Notes for BIND 9.20.25 . . . 538
14.10 Notes for BIND 9.20.24 . . . 538
14.11 Notes for BIND 9.20.23 . . . 539
14.12 Notes for BIND 9.20.22 . . . 542
14.13 Notes for BIND 9.20.21 . . . 543
14.14 Notes for BIND 9.20.20 . . . 544
14.15 Notes for BIND 9.20.19 . . . 545
14.16 Notes for BIND 9.20.18 . . . 546
14.17 Notes for BIND 9.20.17 . . . 547
14.18 Notes for BIND 9.20.16 . . . 547
14.19 Notes for BIND 9.20.15 . . . 548
14.20 Notes for BIND 9.20.14 . . . 549
14.21 Notes for BIND 9.20.13 . . . 549
14.22 Notes for BIND 9.20.12 . . . 550
14.23 Notes for BIND 9.20.11 . . . 551
14.24 Notes for BIND 9.20.10 . . . 552
14.25 Notes for BIND 9.20.9 . . . 552

14.26 Notes for BIND 9.20.8 . . . 553
14.27 Notes for BIND 9.20.7 . . . 555
14.28 Notes for BIND 9.20.6 . . . 557
14.29 Notes for BIND 9.20.5 . . . 558
14.30 Notes for BIND 9.20.4 . . . 559
14.31 Notes for BIND 9.20.3 . . . 561
14.32 Notes for BIND 9.20.2 . . . 562
14.33 Notes for BIND 9.20.1 . . . 564
14.34 Notes for BIND 9.20.0 . . . 565
14.35 License . . . 571
14.36 End of Life . . . 571
14.37 Thank You . . . 571

**15 Changelog** **573**
15.1 BIND 9.20.29 . . . 573
15.2 BIND 9.20.28 . . . 578
15.3 BIND 9.20.27 . . . 578
15.4 BIND 9.20.26 . . . 580
15.5 BIND 9.20.25 . . . 586
15.6 BIND 9.20.24 . . . 586
15.7 BIND 9.20.23 . . . 589
15.8 BIND 9.20.22 . . . 593
15.9 BIND 9.20.21 . . . 596
15.10 BIND 9.20.20 . . . 596
15.11 BIND 9.20.19 . . . 598
15.12 BIND 9.20.18 . . . 599
15.13 BIND 9.20.17 . . . 600
15.14 BIND 9.20.16 . . . 601
15.15 BIND 9.20.15 . . . 602
15.16 BIND 9.20.14 . . . 604
15.17 BIND 9.20.13 . . . 604
15.18 BIND 9.20.12 . . . 605
15.19 BIND 9.20.11 . . . 607
15.20 BIND 9.20.10 . . . 608
15.21 BIND 9.20.9 . . . 608
15.22 BIND 9.20.8 . . . 610
15.23 BIND 9.20.7 . . . 612
15.24 BIND 9.20.6 . . . 616
15.25 BIND 9.20.5 . . . 617
15.26 BIND 9.20.4 . . . 620
15.27 BIND 9.20.3 . . . 623
15.28 BIND 9.20.2 . . . 626
15.29 BIND 9.20.1 . . . 628
15.30 Changes prior to 9.20.1 . . . 631

**16 A Brief History of the DNS and BIND** **1057**

CHAPTER

ONE

# INTRODUCTION TO DNS AND BIND 9

The Internet Domain Name System (DNS) consists of:

- the syntax to specify the names of entities in the Internet in a hierarchical manner,
- the rules used for delegating authority over names, and
- the system implementation that actually maps names to Internet addresses.

DNS data is maintained in a group of distributed hierarchical databases.

## 1.1 Scope of Document

The Berkeley Internet Name Domain (BIND) software implements a domain name server for a number of operating systems. This document provides basic information about the installation and maintenance of Internet Systems Consortium (ISC) BIND version 9 software package for system administrators.

This manual covers BIND version 9.20.29.

## 1.2 Organization of This Document

*Introduction to DNS and BIND 9* introduces the basic DNS and BIND concepts. Some tutorial material on *The Domain Name System (DNS)* is presented for those unfamiliar with DNS. A *DNS Security Overview* is provided to allow BIND operators to implement appropriate security for their operational environment.

*Resource Requirements* describes the hardware and environment requirements for BIND 9 and lists both the supported and unsupported platforms.

*Configurations and Zone Files* is intended as a quickstart guide for newer users. Sample files are included for *Authoritative Name Servers* (both *primary* and *secondary*), as well as a simple *Resolver (Caching Name Servers)* and a *Forwarding Resolver Configuration*. Some reference material on the *Zone File* is included.

*Name Server Operations* covers basic BIND 9 software and DNS operations, including some useful tools, Unix signals, and plugins.

*Advanced Configurations* builds on the configurations of *Configurations and Zone Files*, adding functions and features the system administrator may need.

*Security Configurations* covers most aspects of BIND 9 security, including file permissions, running BIND 9 in a "jail," and securing file transfers and dynamic updates.

*DNSSEC* describes the theory and practice of cryptographic authentication of DNS information. The *DNSSEC Guide* is a practical guide to implementing DNSSEC.

*Configuration Reference* gives exhaustive descriptions of all supported blocks, statements, and grammars used in BIND 9's `named.conf` configuration file.

*Troubleshooting* provides information on identifying and solving BIND 9 and DNS problems. Information about bug-reporting procedures is also provided.

*Building BIND 9* is a definitive guide for those occasions where the user requires special options not provided in the standard Linux or Unix distributions.

The **Appendices** contain useful reference information, such as a bibliography and historic information related to BIND and the Domain Name System, as well as the current *man* pages for all the published tools.

## 1.3 Conventions Used in This Document

In this document, we generally use `fixed-width` text to indicate the following types of information:

- pathnames
- filenames
- URLs
- hostnames
- mailing list names
- new terms or concepts
- literal user input
- program output
- keywords
- variables

Text in "quotes," **bold text**, or *italics* is also used for emphasis or clarity.

## 1.4 The Domain Name System (DNS)

This is a brief description of the functionality and organization of the Domain Name System (DNS). It is provided to familiarize users with the concepts involved, the (often confusing) terminology used, and how all the parts fit together to form an operational system.

All network systems operate with network addresses, such as IPv4 and IPv6. The vast majority of humans find it easier to work with names rather than seemingly endless strings of network address digits. The earliest ARPANET systems (from which the Internet evolved) mapped names to addresses using a **hosts** file that was distributed to all entities whenever changes occurred. Operationally, such a system became rapidly unsustainable once there were more than 100 networked entities, which led to the specification and implementation of the Domain Name System that we use today.

### 1.4.1 DNS Fundamentals

The DNS naming system is organized as a tree structure comprised of multiple levels and thus it naturally creates a distributed system. Each node in the tree is given a label which defines its **Domain** (its area or zone) of **Authority**. The topmost node in the tree is the **Root Domain**; it delegates to **Domains** at the next level which are generically known as the **Top-Level Domains (TLDs)**. They in turn delegate to **Second-Level Domains (SLDs)**, and so on. The Top-Level Domains (TLDs) include a special group of TLDs called the **Country Code Top-Level Domains (ccTLDs)**, in which every country is assigned a unique two-character country code from ISO 3166 as its domain.

**Note**

The Domain Name System is controlled by ICANN (https://www.icann.org) (a 501c non-profit entity); their current policy is that any new TLD, consisting of three or more characters, may be proposed by any group of commercial sponsors and if it meets ICANN's criteria will be added to the TLDs.

The concept of delegation and authority flows down the DNS tree (the DNS hierarchy) as shown:

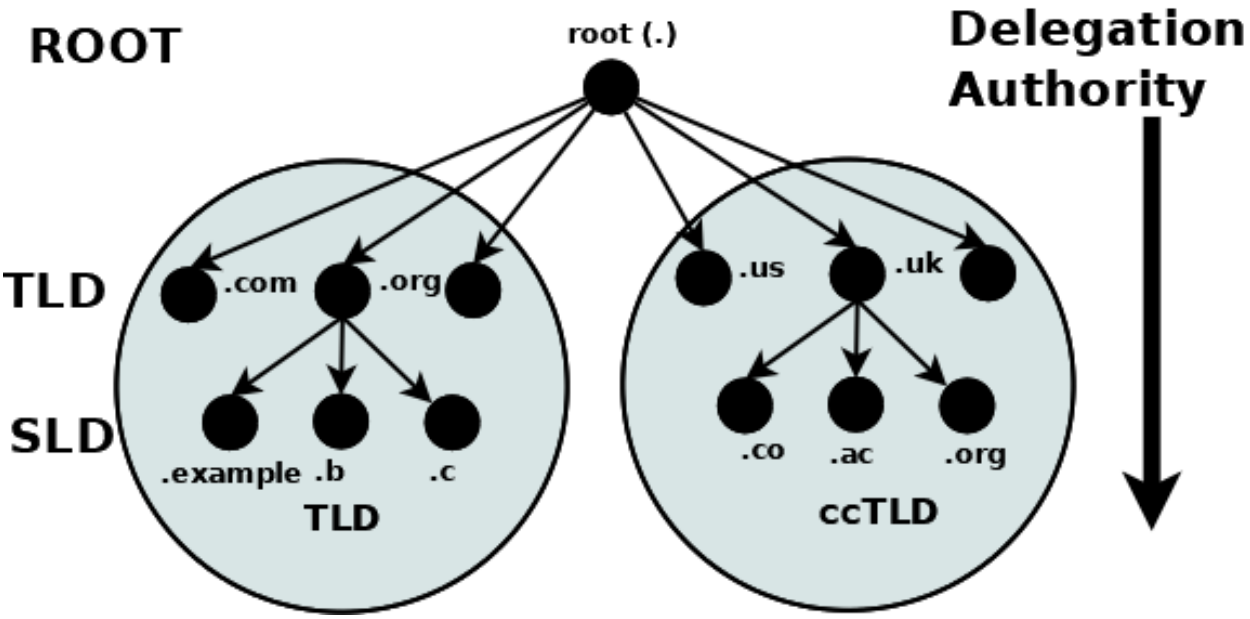



Fig. 1: Delegation and Authority in the DNS Name Space

A domain is the label of a node in the tree. A **domain name** uniquely identifies any node in the DNS tree and is written, left to right, by combining all the domain labels (each of which are unique within their parent's zone or domain of authority), with a dot separating each component, up to the root domain. In the above diagram the following are all domain names:

```
example.com
b.com
ac.uk
us
org
```

The root has a unique label of "." (dot), which is normally omitted when it is written as a domain name, but when it is written as a **Fully Qualified Domain Name (FQDN)** the dot must be present. Thus:

```
example.com      # domain name
example.com.     # FQDN
```

### 1.4.2 Authority and Delegation

Each domain (node) has been **delegated** the authority from its parent domain. The delegated authority includes specific responsibilities to ensure that every domain it delegates has a unique name or label within its zone or domain of authority, and that it maintains an **authoritative** list of its delegated domains. The responsibilities further include an operational requirement to operate two (or more) name servers (which may be contracted to a third party) which will contain the

authoritative data for all the domain labels within its zone of authority in a *zone file*. Again, the tree structure ensures that the DNS name space is naturally distributed.

The following diagram illustrates that **Authoritative Name Servers** exist for every level and every domain in the DNS name space:

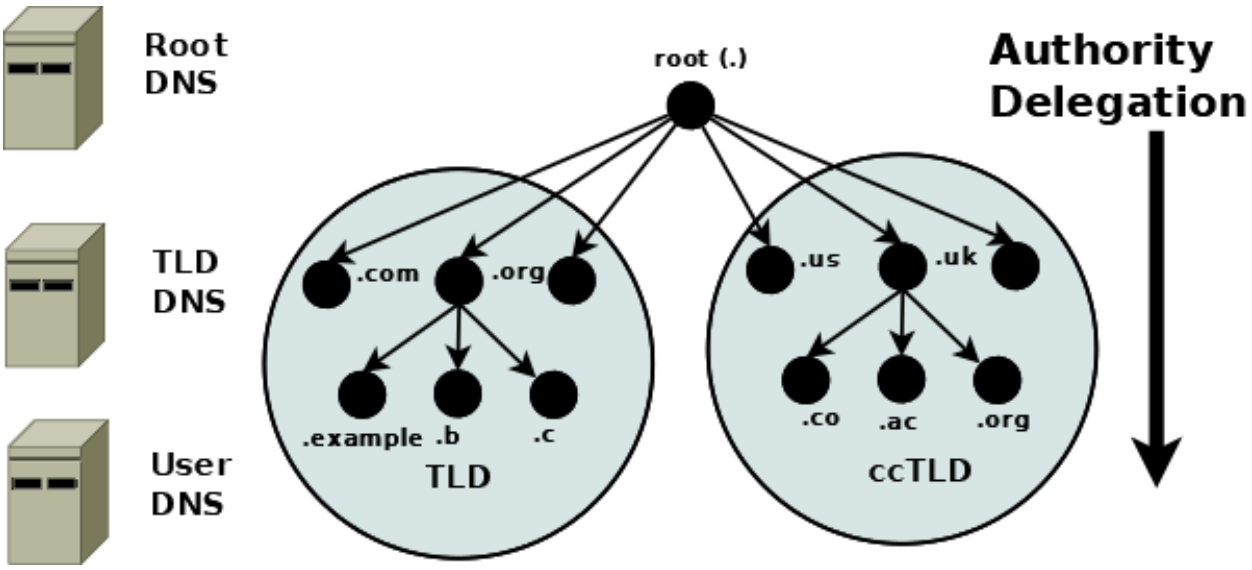



Fig. 2: Authoritative Name Servers in the DNS Name Space

**Note**

The difference between a domain and a zone can appear confusing. Practically, the terms are generally used synonymously in the DNS. If, however, you are into directed graphs and tree structure theory or similar exotica, a zone can be considered as an arc through any node (or domain) with the domain at its apex. The zone therefore encompasses all the name space below the domain. This can, however, lead to the concept of subzones and these were indeed defined in the original DNS specifications. Thankfully the term subzone has been lost in the mists of time.

### 1.4.3 Root Servers

The **root servers** are a critical part of the DNS authoritative infrastructure. There are 13 root servers (*a.root-servers.net* to *m.root-servers.net*). The number 13 is historically based on the maximum amount of name and IPv4 data that could be packed into a 512-byte UDP message, and not a perverse affinity for a number that certain cultures treat as unlucky. The 512-byte UDP data limit is no longer a limiting factor and all root servers now support both IPv4 and IPv6. In addition, almost all the root servers use **anycast**, with well over 300 instances of the root servers now providing service worldwide (see further information at https://root-servers.org). The root servers are the starting point for all **name resolution** within the DNS.

### 1.4.4 Name Resolution

So far all the emphasis has been on how the DNS stores its authoritative domain (zone) data. End-user systems use names (an email address or a web address) and need to access this authoritative data to obtain an IP address, which they use to contact the required network resources such as web, FTP, or mail servers. The process of converting a domain name to a result (typically an IP address, though other types of data may be obtained) is generically called **name resolution**, and is handled by **resolvers** (also known as **caching name servers** and many other terms). The following diagram shows the typical name resolution process:

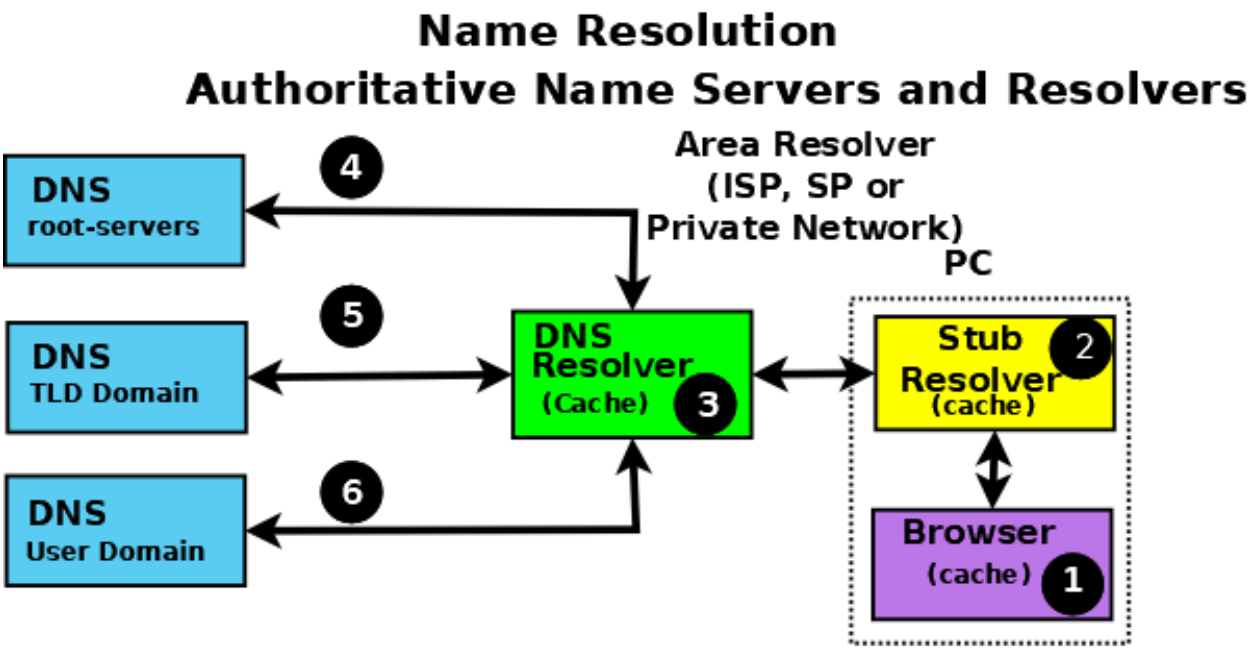



Fig. 3: Authoritative Name Servers and Name Resolution

An end-user application, such as a browser (1), when needing to resolve a name such as **www.example.com**, makes an internal system call to a minimal function resolution entity called a **stub resolver** (2). The stub resolver (using stored IP addresses) contacts a resolver (a caching name server or full-service resolver) (3), which in turn contacts all the necessary authoritative name servers (4, 5, and 6) to provide the answer that it then returns to the user (2, 1). To improve performance, all resolvers (including most stub resolvers) cache (store) their results such that a subsequent request for the same data is taken from the resolver's cache, removing the need to repeat the name resolution process and use time-consuming resources. All communication between the stub resolver, the resolver, and the authoritative name servers uses the DNS protocol's query and response message pair.

### 1.4.5 DNS Protocol and Queries

DNS **queries** use the UDP protocol over the reserved port 53 (but both TCP and TLS can optionally be used in some parts of the network).

The following diagram shows the name resolution process expressed in terms of DNS queries and responses.

The stub resolver sends a **recursive query** message (with the required domain name in the QUESTION section of the query) (2) to the resolver. A **recursive** query simply requests the resolver to find the complete answer. A stub resolver only ever sends recursive queries and always needs the service of a resolver. The response to a recursive query can be:

1. The answer to the user's QUESTION in the ANSWER section of the query response.
2. An error (such as NXDOMAIN - the name does not exist).

The resolver, on receipt of the user's recursive query, either responds immediately, if the ANSWER is in its cache, or accesses the DNS hierarchy to obtain the answer. The resolver always starts with root servers and sends an **iterative query** (4, 5, and 6). The response to an iterative query can be:

1. The answer to the resolver's QUESTION in the ANSWER section of the query response.

2. A **referral** (indicated by an empty ANSWER section but data in the AUTHORITY section, and typically IP addresses in the ADDITIONAL section of the response).

3. An error (such as NXDOMAIN - the name does not exist).

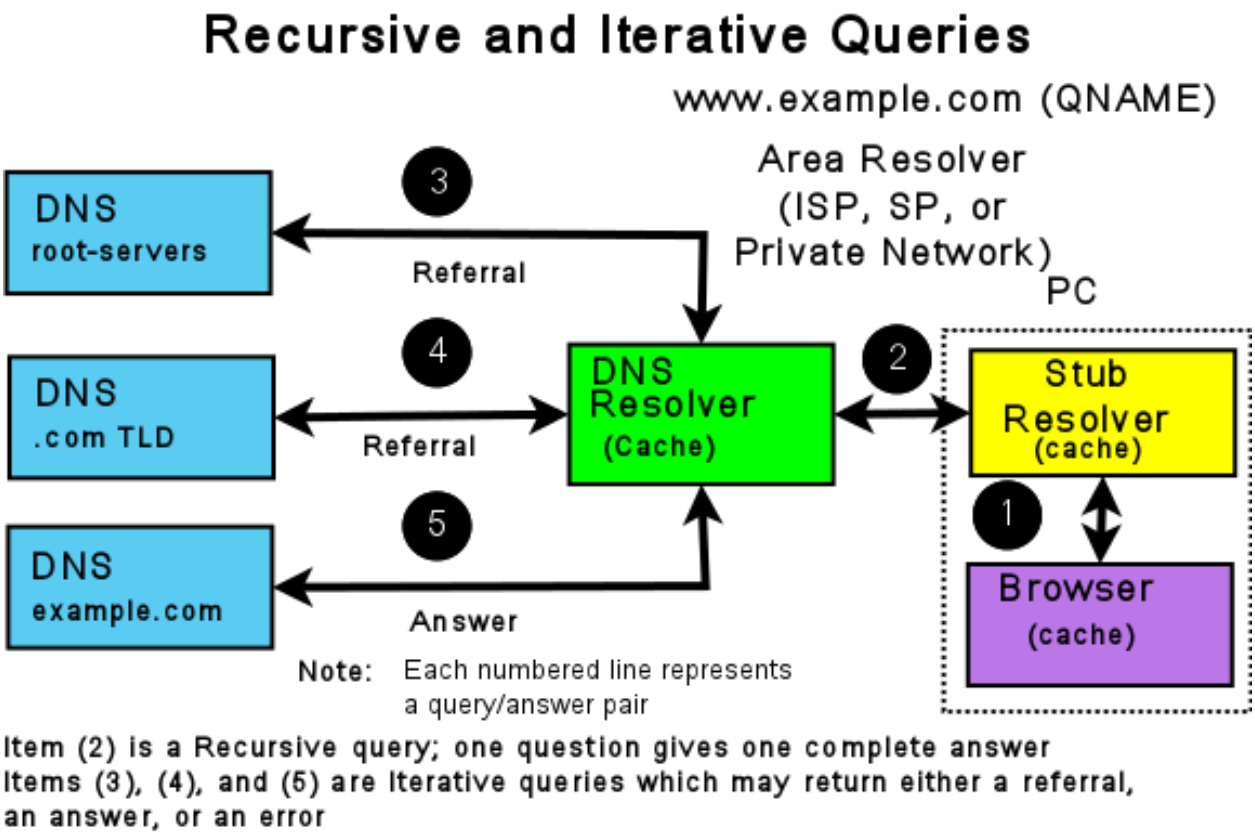



Fig. 4: Resolvers and Queries

If the response is either an answer or an error, these are returned immediately to the user (and cached for future use). If the response is a referral, the resolver needs to take additional action to respond to the user's recursive query.

A referral, in essence, indicates that the queried server does not know the answer (the ANSWER section of the response is empty), but it refers the resolver to the authoritative name servers (in the AUTHORITY section of the response) which it knows about in the domain name supplied in the QUESTION section of the query. Thus, if the QUESTION is for the domain name **www.example.com**, the root server to which the iterative query was sent adds a list of the **.com authoritative name servers** in the AUTHORITY section. The resolver selects one of the servers from the AUTHORITY section and sends an iterative query to it. Similarly, the .com authoritative name servers send a referral containing a list of the **example.com** authoritative name servers. This process continues down the DNS hierarchy until either an ANSWER or an error is received, at which point the user's original recursive query is sent a response.

> **Note**
>
> The DNS hierarchy is always accessed starting at the root servers and working down; there is no concept of "up" in the DNS hierarchy. Clearly, if the resolver has already cached the list of .com authoritative name servers and the user's recursive query QUESTION contains a domain name ending in .com, it can omit access to the root servers. However, that is simply an artifact (in this case a performance benefit) of caching and does not change the concept of top-down access within the DNS hierarchy.

The insatiably curious may find reading **RFC 1034** and **RFC 1035** a useful starting point for further information.

### 1.4.6 DNS and BIND 9

BIND 9 is a complete implementation of the DNS protocol. BIND 9 can be configured (using its `named.conf` file) as an authoritative name server, a resolver, and, on supported hosts, a stub resolver. While large operators usually dedicate DNS servers to a single function per system, smaller operators will find that BIND 9's flexible configuration features support multiple functions, such as a single DNS server acting as both an authoritative name server and a resolver.

Example configurations of basic *authoritative name servers* and *resolvers and forwarding resolvers*, as well as *advanced configurations* and *secure configurations*, are provided.

## 1.5 DNS Security Overview

DNS is a communications protocol. All communications protocols are potentially vulnerable to both subversion and eavesdropping. It is important for users to audit their exposure to the various threats within their operational environment and implement the appropriate solutions. BIND 9, a specific implementation of the DNS protocol, provides an extensive set of security features. The purpose of this section is to help users to select from the range of available security features those required for their specific user environment.

A generic DNS network is shown below, followed by text descriptions. In general, the further one goes from the left-hand side of the diagram, the more complex the implementation.

> **Note**
>
> Historically, DNS data was regarded as public and security was concerned, primarily, with ensuring the integrity of DNS data. DNS data privacy is increasingly regarded as an important dimension of overall security, specifically *DNS over TLS*.

The following notes refer to the numbered elements in the above diagram.

1. A variety of system administration techniques and methods may be used to secure BIND 9's local environment, including *file permissions*, running BIND 9 in a *jail*, and the use of *Access Control Lists*.

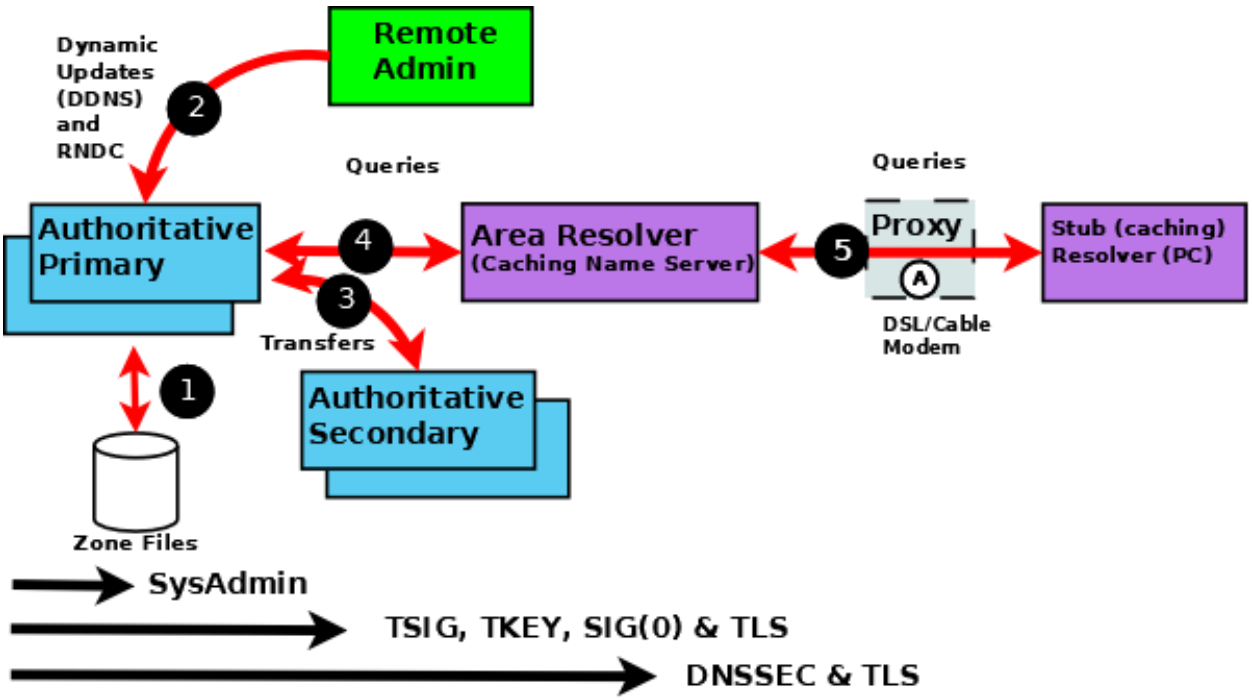



Fig. 5: BIND 9 Security Overview

2. The remote name daemon control (*rndc*) program allows the system administrator to control the operation of a name server. The majority of BIND 9 packages or ports come preconfigured with local (loopback address) security preconfigured. If `rndc` is being invoked from a remote host, further configuration is required. The `nsupdate` tool uses **Dynamic DNS (DDNS)** features and allows users to dynamically change the contents of the zone file(s). `nsupdate` access and security may be controlled using `named.conf` *statements or using TSIG or SIG(0) cryptographic methods*. Clearly, if the remote hosts used for either `rndc` or DDNS lie within a network entirely under the user's control, the security threat may be regarded as non-existent. Any implementation requirements, therefore, depend on the site's security policy.

3. Zone transfer from a **primary** to one or more **secondary** authoritative name servers across a public network carries risk. The zone transfer may be secured using `named.conf` *statements, TSIG cryptographic methods, or TLS*. Clearly, if the secondary authoritative name server(s) all lie within a network entirely under the user's control, the security threat may be regarded as non-existent. Any implementation requirements again depend on the site's security policy.

4. If the operator of an authoritative name server (primary or secondary) wishes to ensure that DNS responses to user-initiated queries about the zone(s) for which they are responsible can only have come from their server, that the data received by the user is the same as that sent, and that non-existent names are genuine, then *DNSSEC* is the only solution. DNSSEC requires configuration and operational changes both to the authoritative name servers and to any resolver which accesses those servers.

5. The typical Internet-connected end-user device (PCs, laptops, and even mobile phones) either has a stub resolver or operates via a DNS proxy. A stub resolver requires the services of an area or full-service resolver to completely answer user queries. Stub resolvers on the majority of PCs and laptops typically have a caching capability to increase performance. At this time there are no standard stub resolvers or proxy DNS tools that implement DNSSEC. BIND 9 may be configured to provide such capability on supported Linux or Unix platforms. *DNS over TLS* may be configured to verify the integrity of the data between the stub resolver and area (or full-service) resolver. However, unless the resolver and the authoritative name server implements DNSSEC, end-to-end integrity (from authoritative name server to stub resolver) cannot be guaranteed.

CHAPTER

TWO

# RESOURCE REQUIREMENTS

## 2.1 Hardware Requirements

DNS hardware requirements have traditionally been quite modest. For many installations, servers that have been retired from active duty have performed admirably as DNS servers.

However, the DNSSEC features of BIND 9 may be quite CPU-intensive, so organizations that make heavy use of these features may wish to consider larger systems for these applications. BIND 9 is fully multithreaded, allowing full utilization of multiprocessor systems for installations that need it.

## 2.2 CPU Requirements

CPU requirements for BIND 9 range from i386-class machines, for serving static zones without caching, to enterprise-class machines to process many dynamic updates and DNSSEC-signed zones, serving many thousands of queries per second.

## 2.3 Memory Requirements

Server memory must be sufficient to hold both the cache and the zones loaded from disk. The `max-cache-size` option can limit the amount of memory used by the cache, at the expense of reducing cache hit rates and causing more DNS traffic. It is still good practice to have enough memory to load all zone and cache data into memory; unfortunately, the best way to determine this for a given installation is to watch the name server in operation. After a few weeks, the server process should reach a relatively stable size where entries are expiring from the cache as fast as they are being inserted.

## 2.4 Name Server-Intensive Environment Issues

For name server-intensive environments, there are two configurations that may be used. The first is one where clients and any second-level internal name servers query the main name server, which has enough memory to build a large cache; this approach minimizes the bandwidth used by external name lookups. The second alternative is to set up second-level internal name servers to make queries independently. In this configuration, none of the individual machines need to have as much memory or CPU power as in the first alternative, but this has the disadvantage of making many more external queries, as none of the name servers share their cached data.

## 2.5 Supported Platforms

The current support status of BIND 9 versions across various platforms can be found in the ISC Knowledgebase:

https://kb.isc.org/docs/supported-platforms

In general, this version of BIND will build and run on any POSIX-compliant system with a modern C11 (or better) compiler, BSD-style sockets with RFC-compliant IPv6 support, POSIX-compliant threads, and the *required libraries*.

The following C11 features are required to compile BIND 9:

- Atomic operations support defined in <stdatomic.h>
- Thread Local Storage support defined in <threads.h>

Where it makes sense, BIND 9 uses C-standard fixes introduced by the C17 update of the C11 standard.

ISC regularly tests BIND on many operating systems and architectures, but lacks the resources to test all of them. Consequently, ISC is only able to offer support on a "best-effort" basis for some.

### 2.5.1 Regularly Tested Platforms

Current versions of BIND 9 are fully supported and regularly tested on the following systems:

- Debian 12, 13
- Ubuntu LTS 22.04, 24.04, 26.04
- Fedora 44
- Red Hat Enterprise Linux / CentOS / AlmaLinux 8, 9, 10
- FreeBSD 14, 15
- Alpine Linux 3.24

The amd64 CPU architecture is fully supported and regularly tested.

### 2.5.2 Best-Effort

The following are platforms on which BIND is known to build and run. ISC makes every effort to fix bugs on these platforms, but may be unable to do so quickly due to lack of hardware, less familiarity on the part of engineering staff, and other constraints. None of these are tested regularly by ISC.

- macOS 10.12+
- Solaris 11
- NetBSD
- OpenBSD
- Other Linux distributions still supported by their vendors, such as:
  - Ubuntu 22.10+
  - Gentoo
  - Arch Linux
- OpenWRT/LEDE 17.01+
- Other CPU architectures (arm, arm64, mips64, ppc64, s390x)

### 2.5.3 Community-Maintained

These systems may not all have the required dependencies for building BIND easily available, although it is possible in many cases to compile those directly from source. The community and interested parties may wish to help with maintenance, and we welcome patch contributions, although we cannot guarantee that we will accept them. All contributions will be assessed against the risk of adverse effect on officially supported platforms.

- Platforms past or close to their respective EOL dates, such as:

  - Ubuntu 14.04, 16.04, 18.04, 20.04 (Ubuntu ESM releases are not supported)
  - Red Hat Enterprise Linux / CentOS / Oracle Linux 6, 7
  - Debian 8 Jessie, 9 Stretch, 10 Buster, 11 Bullseye
  - FreeBSD 10.x, 11.x, 12.x, 13.x
- Less common CPU architectures (i386, i686, mips, mipsel, sparc, ppc, and others)

## 2.6 Unsupported Platforms

These are platforms on which current versions of BIND 9 are known *not* to build or run:

- Platforms without at least OpenSSL 1.0.2
- Windows
- Solaris 10 and older
- Platforms that do not support IPv6 Advanced Socket API (**RFC 3542**)
- Platforms that do not support atomic operations (via compiler or library)
- Linux without NPTL (Native POSIX Thread Library)
- Platforms on which **libuv >= 1.34** cannot be compiled or is not available

## 2.7 Installing BIND 9

*Building BIND 9* contains complete instructions for how to build BIND 9.

The ISC Knowledgebase contains many useful articles about installing BIND 9 on specific platforms.

CHAPTER

THREE

# CONFIGURATIONS AND ZONE FILES

## 3.1 Introduction

BIND 9 uses a single configuration file called *named.conf*. which is typically located in either /etc/namedb or /usr/local/etc/namedb.

> **Note**
>
> If *rndc* is being used locally (on the same host as BIND 9) then an additional file `rndc.conf` may be present, though `rndc` operates without this file. If `rndc` is being run from a remote host then an `rndc.conf` file must be present as it defines the link characteristics and properties.

Depending on the functionality of the system, one or more zone files is required.

The samples given throughout this and subsequent chapters use a standard base format for both the `named.conf` and the zone files for **example.com**. The intent is for the reader to see the evolution from a common base as features are added or removed.

### 3.1.1 `named.conf` Base File

This file illustrates the typical format and layout style used for `named.conf` and provides a basic logging service, which may be extended as required by the user.

```
// base named.conf file
// Recommended that you always maintain a change log in this file as shown here
// options clause defining the server-wide properties
options {
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
};

// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
```

(continues on next page)

(continued from previous page)

```
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
  };
  category default {
    example_log;
  };
};
```

The *logging* and *options* blocks and *category*, *channel*, *directory*, *file*, and *severity* statements are all described further in the appropriate sections of this ARM.

### 3.1.2 example.com base zone file

The following is a complete zone file for the domain **example.com**, which illustrates a number of common features. Comments in the file explain these features where appropriate. Zone files consist of *Resource Records (RR)*, which describe the zone's characteristics or properties.

```
; base zone file for example.com
$TTL 2d    ; default TTL for zone
$ORIGIN example.com. ; base domain-name
; Start of Authority RR defining the key characteristics of the zone (domain)
@         IN      SOA   ns1.example.com. hostmaster.example.com. (
                                2003080800 ; serial number
                                12h        ; refresh
                                15m        ; update retry
                                4d         ; expiry
                                2h         ; minimum
                                )
; name server RR for the domain
           IN      NS      ns1.example.com.
; the second name server is external to this zone (domain)
           IN      NS      ns2.example.net.
; mail server RRs for the zone (domain)
        3w IN      MX  10  mail.example.com.
; the second  mail servers is  external to the zone (domain)
           IN      MX  20  mail.example.net.
; domain hosts includes NS and MX records defined above
; plus any others required
; for instance a user query for the A RR of joe.example.com will
; return the IPv4 address 192.168.254.6 from this zone file
ns1        IN      A       192.168.254.2
mail       IN      A       192.168.254.4
joe        IN      A       192.168.254.6
www        IN      A       192.168.254.7
; aliases ftp (ftp server) to an external domain
ftp        IN      CNAME   ftp.example.net.
```

This type of zone file is frequently referred to as a **forward-mapped zone file**, since it maps domain names to some other value, while a *reverse-mapped zone file* maps an IP address to a domain name. The zone file is called **example.com**

for no good reason except that it is the domain name of the zone it describes; as always, users are free to use whatever file-naming convention is appropriate to their needs.

### 3.1.3 Other Zone Files

Depending on the configuration additional zone files may or should be present. Their format and functionality are briefly described here.

### 3.1.4 localhost Zone File

All end-user systems are shipped with a `hosts` file (usually located in /etc). This file is normally configured to map the name **localhost** (the name used by applications when they run locally) to the loopback address. It is argued, reasonably, that a forward-mapped zone file for **localhost** is therefore not strictly required. This manual does use the BIND 9 distribution file `localhost-forward.db` (normally in /etc/namedb/master or /usr/local/etc/namedb/master) in all configuration samples for the following reasons:

1. Many users elect to delete the `hosts` file for security reasons (it is a potential target of serious domain name redirection/poisoning attacks).
2. Systems normally lookup any name (including domain names) using the `hosts` file first (if present), followed by DNS. However, the `nsswitch.conf` file (typically in /etc) controls this order (normally **hosts: file dns**), allowing the order to be changed or the **file** value to be deleted entirely depending on local needs. Unless the BIND administrator controls this file and knows its values, it is unsafe to assume that **localhost** is forward-mapped correctly.
3. As a reminder to users that unnecessary queries for **localhost** form a non-trivial volume of DNS queries on the public network, which affects DNS performance for all users.

Users may, however, elect at their discretion not to implement this file since, depending on the operational environment, it may not be essential.

The BIND 9 distribution file `localhost-forward.db` format is shown for completeness and provides for both IPv4 and IPv6 localhost resolution. The zone (domain) name is **localhost.**

```
$TTL 3h
localhost.  SOA      localhost.  nobody.localhost. 42  1d  12h  1w  3h
            NS       localhost.
            A        127.0.0.1
            AAAA     ::1
```

**Note**

Readers of a certain age or disposition may note the reference in this file to the late, lamented Douglas Noel Adams.

### 3.1.5 localhost Reverse-Mapped Zone File

This zone file allows any query requesting the name associated with the loopback IP (127.0.0.1). This file is required to prevent unnecessary queries from reaching the public DNS hierarchy. The BIND 9 distribution file `localhost.rev` is shown for completeness:

```
$TTL 1D
@        IN        SOA  localhost. root.localhost. (
                        2007091701 ; serial
                        30800      ; refresh
                        7200       ; retry
```

(continues on next page)

(continued from previous page)

```
                        604800     ; expire
                        300 )      ; minimum
        IN        NS    localhost.
1       IN        PTR   localhost.
```

## 3.2 Authoritative Name Servers

These provide authoritative answers to user queries for the zones they support: for instance, the zone data describing the domain name **example.com**. An authoritative name server may support one or many zones.

Each zone may be defined as either a **primary** or a **secondary**. A primary zone reads its zone data directly from a file system. A secondary zone obtains its zone data from the primary zone using a process called **zone transfer**. Both the primary and the secondary zones provide authoritative data for their zone; there is no difference in the answer to a query from a primary or a secondary zone. An authoritative name server may support any combination of primary and secondary zones.

**Note**

The terms **primary** and **secondary** do not imply any access priority. Resolvers (name servers that provide the complete answers to user queries) are not aware of (and cannot find out) whether an authoritative answer comes from the primary or secondary name server. Instead, the resolver uses the list of authoritative servers for the zone (there must be at least two) and maintains a Round Trip Time (RTT) - the time taken to respond to the query - for each server in the list. The resolver uses the lowest-value server (the fastest) as its preferred server for the zone and continues to do so until its RTT becomes higher than the next slowest in its list, at which time that one becomes the preferred server.

For reasons of backward compatibility BIND 9 treats "primary" and "master" as synonyms, as well as "secondary" and "slave."

The following diagram shows the relationship between the primary and secondary name servers. The text below explains the process in detail.

The numbers in parentheses in the following text refer to the numbered items in the diagram above.

1. The authoritative primary name server always loads (or reloads) its zone files from (1) a local or networked filestore.
2. The authoritative secondary name server always loads its zone data from a primary via a **zone transfer** operation. Zone transfer may use **AXFR** (complete zone transfer) or **IXFR** (incremental zone transfer), but only if both primary and secondary name servers support the service. The zone transfer process (either AXFR or IXFR) works as follows:
   a) The secondary name server for the zone reads (3 and 4) the *SOA RR* periodically. The interval is defined by the **refresh** parameter of the Start of Authority (SOA) RR.
   b) The secondary compares the **serial number** parameter of the SOA RR received from the primary with the serial number in the SOA RR of its current zone data.
   c) If the received serial number is arithmetically greater (higher) than the current one, the secondary initiates a zone transfer (5) using AXFR or IXFR (depending on the primary and secondary configuration), using TCP over port 53 (6).
3. The typically recommended zone refresh times for the SOA RR (the time interval when the secondary reads or polls the primary for the zone SOA RR) are multiples of hours to reduce traffic loads. Worst-case zone change propagation can therefore take extended periods.
4. The optional NOTIFY (**RFC 1996**) feature (2) is automatically configured; use the `notify` statement to turn off the feature. Whenever the primary loads or reloads a zone, it sends a NOTIFY message to the configured

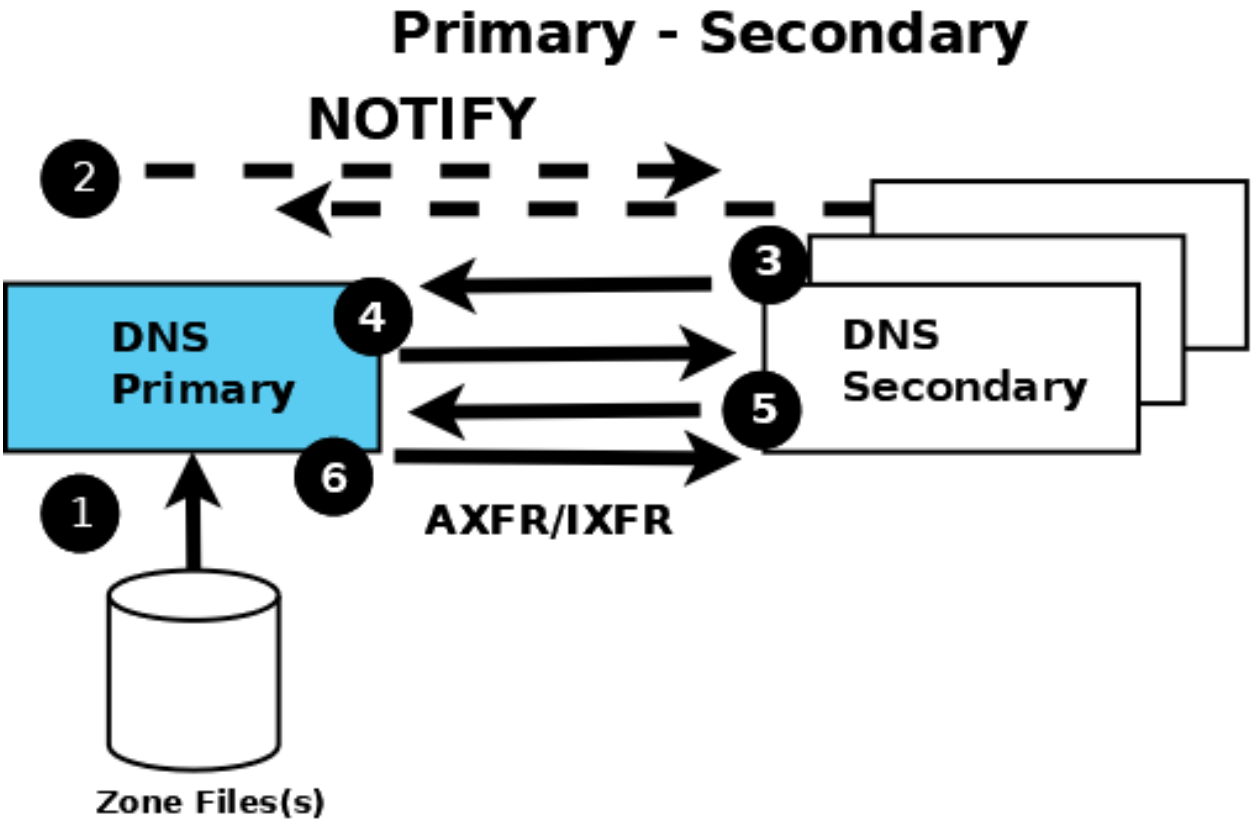



Fig. 1: Authoritative Primary and Secondary Name Servers

secondary (or secondaries) and may optionally be configured to send the NOTIFY message to other hosts using the *also-notify* statement. The NOTIFY message simply indicates to the secondary that the primary has loaded or reloaded the zone. On receipt of the NOTIFY message, the secondary respons to indicate it has received the NOTIFY and immediately reads the SOA RR from the primary (as described in section 2 a. above). If the zone file has changed, propagation is practically immediate.

The authoritative samples all use NOTIFY but identify the statements used, so that they can be removed if not required.

### 3.2.1 Primary Authoritative Name Server

The zone files are unmodified *from the base samples* but the *named.conf* file has been modified as shown:

```
// authoritative primary named.conf file
// options clause defining the server-wide properties
options {
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
  // This is the default - allows user queries from any IP
  allow-query { any; };
  // normal server operations may place items in the cache
  // this prevents any user query from accessing these items
  // only authoritative zone data will be returned
  allow-query-cache { none; };
  // Do not provide recursive service to user queries
  recursion no;
};
// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
  };
  category default {
    example_log;
  };
};
// Provide forward mapping zone for localhost
// (optional)
zone "localhost" {
  type primary;
  file "master/localhost-forward.db";
  notify no;
};
```

(continues on next page)

(continued from previous page)

```
// Provide reverse mapping zone for the loopback
// address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
  type primary;
  file "localhost.rev";
  notify no;
};
// We are the primary server for example.com
zone "example.com" {
  // this is the primary name server for the zone
  type primary;
  file "example.com";
  // this is the default
  notify yes;
  // IP addresses of secondary servers allowed to
  // transfer example.com from this server
  allow-transfer {
    192.168.4.14;
    192.168.5.53;
  };
};
```

The added statements and blocks are commented in the above file.

The *zone* block, and *allow-query*, *allow-query-cache*, *allow-transfer*, *file*, *notify*, *recursion*, and *type* statements are described in detail in the appropriate sections.

### 3.2.2 Secondary Authoritative Name Server

The zone files `local-host-forward.db` and `localhost.rev` are unmodified *from the base samples*. The **example.com** zone file is not required (the zone file is obtained from the primary via zone transfer). The *named.conf* file has been modified as shown:

```
// authoritative secondary named.conf file
// options clause defining the server-wide properties
options {
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
  // This is the default - allows user queries from any IP
  allow-query { any; };
  // normal server operations may place items in the cache
  // this prevents any user query from accessing these items
  // only authoritative zone data will be returned
  allow-query-cache { none; };
  // Do not provide recursive service to user queries
  recursion no;
};
// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
```

(continues on next page)

(continued from previous page)

```
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
   };
   category default {
     example_log;
  };
};
// Provide forward mapping zone for localhost
// (optional)
zone "localhost" {
  type primary;
  file "master/localhost-forward.db";
  notify no;
};
// Provide reverse mapping zone for the loopback
// address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
  type primary;
  file "localhost.rev";
  notify no;
};
// We are the secondary server for example.com
zone "example.com" {
  // this is a secondary server for the zone
  type secondary;
  // the file statement here allows the secondary to save
  // each zone transfer so that in the event of a program restart
  // the zone can be loaded immediately and the server can start
  // to respond to queries without waiting for a zone transfer
  file "example.com.saved";
  // IP address of example.com primary server
  primaries { 192.168.254.2; };
};
```

The statements and blocks added are all commented in the above file.

The *zone* block, and *allow-query*, *allow-query-cache*, *allow-transfer*, *file*, *primaries*, *recursion*, and *type* statements are described in detail in the appropriate sections.

If NOTIFY is not being used, no changes are required in this *named.conf* file, since it is the primary that initiates the NOTIFY message.

**Note**

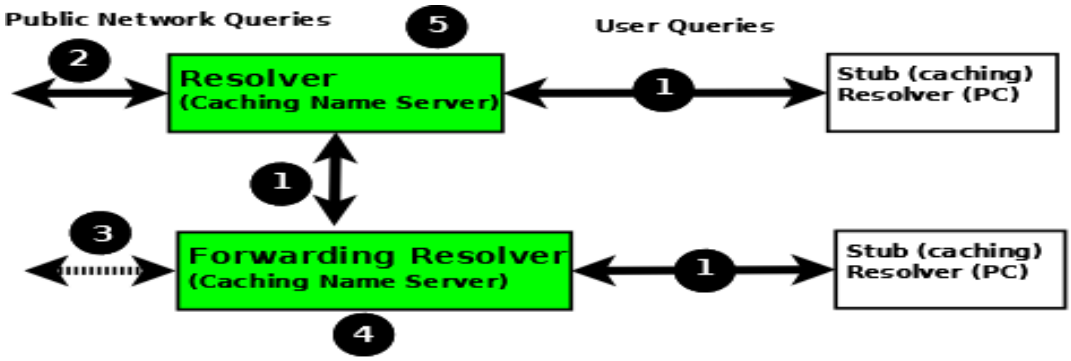
Just when the reader thought they understood primary and secondary, things can get more complicated. A secondary zone can also be a primary to other secondaries: `named`, by default, sends NOTIFY messages for every zone it loads. Specifying *notify primary-only;* in the `zone` block for the secondary causes `named` to only send NOTIFY messages for primary zones that it loads.

## 3.3 Resolver (Caching Name Servers)

Resolvers handle *recursive user queries* and provide complete answers; that is, they issue one or more *iterative queries* to the DNS hierarchy. Having obtained a complete answer (or an error), a resolver passes the answer to the user and places it in its cache. Subsequent user requests for the same query will be answered from the resolver's cache until the *TTL* of the cached answer has expired, when it will be flushed from the cache; the next user query that requests the same information results in a new series of queries to the DNS hierarchy.

Resolvers are frequently referred to by a bewildering variety of names, including caching name servers, recursive name servers, forwarding resolvers, area resolvers, and full-service resolvers.

The following diagram shows how resolvers can function in a typical networked environment:




Resolver and Forwarding Resolver

1. End-user systems are all distributed with a local **stub resolver** as a standard feature. Today, the majority of stub resolvers also provide a local cache service to speed up user response times.

2. A stub resolver has limited functionality; specifically, it cannot follow *referrals*. When a stub resolver receives a request for a name from a local program, such as a browser, and the answer is not in its local cache, it sends a *recursive user query* (1) to a locally configured resolver (5), which may have the answer available in its cache. If it does not, it issues *iterative queries* (2) to the DNS hierarchy to obtain the answer. The resolver to which the local system sends the user query is configured, for Linux and Unix hosts, in `/etc/resolv.conf`; for Windows users it is configured or changed via the Control Panel or Settings interface.

3. Alternatively, the user query can be sent to a **forwarding resolver** (4). Forwarding resolvers on first glance look fairly pointless, since they appear to be acting as a simple pass-though and, like the stub resolver, require a full-service resolver (5). However, forwarding resolvers can be very powerful additions to a network for the following reasons:

    a) Cost and Performance. Each **recursive user query** (1) at the forwarding resolver (4) results in two messages - the query and its answer. The resolver (5) may have to issue three, four, or more query pairs (2) to get the required answer. Traffic is reduced dramatically, increasing performance or reducing cost (if the link is tariffed). Additionally, since the forwarding resolver is typically shared across multiple hosts, its cache is more likely to contain answers, again improving user performance.

b) Network Maintenance. Forwarding resolvers (4) can be used to ease the burden of local administration by providing a single point at which changes to remote name servers can be managed, rather than having to update all hosts. Thus, all hosts in a particular network section or area can be configured to point to a forwarding resolver, which can be configured to stream DNS traffic as desired and changed over time with minimal effort.

c) Sanitizing Traffic. Especially in larger private networks it may be sensible to stream DNS traffic using a forwarding resolver structure. The forwarding resolver (4) may be configured, for example, to handle all in-domain traffic (relatively safe) and forward all external traffic to a **hardened** resolver (5).

d) Stealth Networks. Forwarding resolvers are extensively used in *stealth or split networks*.

4. Forwarding resolvers (4) can be configured to forward all traffic to a resolver (5), or to only forward selective traffic (5) while directly resolving other traffic (3).

**Attention**

While the diagram above shows **recursive user queries** arriving via interface (1), there is nothing to stop them from arriving via interface (2) via the public network. If no limits are placed on the source IPs that can send such queries, the resolver is termed an **open resolver**. Indeed, when the world was young this was the way things worked on the Internet. Much has changed and what seems to be a friendly, generous action can be used by rogue actors to cause all kinds of problems including **Denial of Service (DoS)** attacks. Resolvers should always be configured to limit the IP addresses that can use their services. BIND 9 provides a number of statements and blocks to simplify defining these IP limits and configuring a **closed resolver**. The resolver samples given here all configure closed resolvers using a variety of techniques.

### 3.3.1 Additional Zone Files

#### Root Servers (Hint) Zone File

Resolvers (although not necessarily forwarding resolvers) need to access the DNS hierarchy. To do this, they need to know the addresses (IPv4 and/or IPv6) of the 13 *root servers*. This is done by the provision of a root server zone file, which is contained in the standard BIND 9 distribution as the file `named.root` (normally found in /etc/namedb or /usr/local/namedb). This file may also be obtained from the IANA website (https://www.iana.org/domains/root/files).

**Note**

Many distributions rename this file for historical reasons. Consult the appropriate distribution documentation for the actual file name.

The hint zone file is referenced using the `type hint` statement and a zone (domain) name of "." (the generally silent dot).

> **Note**
>
> The root server IP addresses have been stable for a number of years and are likely to remain stable for the near future. BIND 9 has a root-server list in its executable such that even if this file is omitted, out-of-date, or corrupt BIND 9 can still function. For this reason, many sample configurations omit the hints file. All the samples given here include the hints file primarily as a reminder of the functionality of the configuration, rather than as an absolute necessity.

#### Private IP Reverse Map Zone Files

Resolvers are configured to send *iterative queries* to the public DNS hierarchy when the information requested is not in their cache or not defined in any local zone file. Many networks make extensive use of private IP addresses (defined by **RFC 1918**, **RFC 2193**, **RFC 5737**, and **RFC 6598**). By their nature these IP addresses are forward-mapped in various user zone files. However, certain applications may issue **reverse map** queries (mapping an IP address to a name). If the private IP addresses are not defined in one or more reverse-mapped zone file(s), the resolver sends them to the DNS hierarchy where they are simply useless traffic, slowing down DNS responses for all users.

Private IP addresses may be defined using standard *reverse-mapping techniques* or using the `empty-zones-enable` statement. By default this statement is set to `empty-zones-enable yes;` and thus automatically prevents unnecessary DNS traffic by sending an NXDOMAIN error response (indicating the name does not exist) to any request. However, some applications may require a genuine answer to such reverse-mapped requests or they will fail to function. Mail systems in particular perform reverse DNS queries as a first-line spam check; in this case a reverse-mapped zone file is essential. The sample configuration files given here for both the resolver and the forwarding resolver provide a reverse-mapping zone file for the private IP address 192.168.254.4, which is the mail server address in the *base zone file*, as an illustration of the reverse-map technique. The file is named `192.168.254.rev` and has a zone name of **254.168.192.in-addr.arpa**.

```
; reverse map zone file for 192.168.254.4 only
$TTL 2d  ; 172800 seconds
$ORIGIN 254.168.192.IN-ADDR.ARPA.
@     IN      SOA   ns1.example.com. hostmaster.example.com. (
                                    2003080800 ; serial number
                                    3h         ; refresh
                                    15m        ; update retry
                                    3w         ; expiry
                                    3h         ; nx = nxdomain ttl
                                    )
; only one NS is required for this local file
; and is an out of zone name
      IN      NS      ns1.example.com.
; other IP addresses can be added as required
; this maps 192.168.254.4 as shown
4     IN      PTR     mail.example.com. ; fully qualified domain name (FQDN)
```

### 3.3.2 Resolver Configuration

The resolver provides *recursive query support* to a defined set of IP addresses. It is therefore a **closed** resolver and cannot be used in wider network attacks.

```
// resolver named.conf file
// Two corporate subnets we wish to allow queries from
// defined in an acl clause
acl corpnets {
  192.168.4.0/24;
```

(continues on next page)

(continued from previous page)

```
  192.168.7.0/24;
};

// options clause defining the server-wide properties
options {
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
  // this is the default
  recursion yes;
  // recursive queries only allowed from these ips
  // and references the acl clause
  allow-query { corpnets; };
  // this ensures that any reverse map for private IPs
  // not defined in a zone file will *not* be passed to the public network
  // it is the default value
  empty-zones-enable yes;
};

// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
   };
   category default {
     example_log;
  };
};

// zone file for the root servers
// discretionary zone (see root server discussion above)
zone "." {
  type hint;
  file "named.root";
};

// zone file for the localhost forward map
// discretionary zone depending on hosts file (see discussion)
zone "localhost" {
  type primary;
  file "masters/localhost-forward.db";
```

(continues on next page)

(continued from previous page)

```
  notify no;
};

// zone file for the loopback address
// necessary zone
zone "0.0.127.in-addr.arpa" {
  type primary;
  file "localhost.rev";
  notify no;
};

// zone file for local IP reverse map
// discretionary file depending on requirements
zone "254.168.192.in-addr.arpa" {
  type primary;
  file "192.168.254.rev";
  notify no;
};
```

The *zone* and *acl* blocks, and the *allow-query*, *empty-zones-enable*, *file*, *notify*, *recursion*, and *type* statements are described in detail in the appropriate sections.

As a reminder, the configuration of this resolver does **not** access the DNS hierarchy (does not use the public network) for any recursive query for which:

1. The answer is already in the cache.
2. The domain name is **localhost** (zone localhost).
3. Is a reverse-map query for 127.0.0.1 (zone 0.0.127.in-addr.arpa).
4. Is a reverse-map query for 192.168.254/24 (zone 254.168.192.in-addr.arpa).
5. Is a reverse-map query for any local IP (*empty-zones-enable* statement).

All other recursive queries will result in access to the DNS hierarchy to resolve the query.

### 3.3.3 Forwarding Resolver Configuration

This forwarding resolver configuration forwards all recursive queries, other than those for the defined zones and those for which the answer is already in its cache, to a full-service resolver at the IP address 192.168.250.3, with an alternative at 192.168.230.27. The forwarding resolver will cache all responses from these servers. The configuration is closed, in that it defines those IPs from which it will accept recursive queries.

A second configuration in which selective forwarding occurs *is also provided*.

```
// forwarding named.conf file
// Two corporate subnets we wish to allow queries from
// defined in an acl clause
acl corpnets {
  192.168.4.0/24;
  192.168.7.0/24;
};

// options clause defining the server-wide properties
options {
```

(continues on next page)

(continued from previous page)

```
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
  // this is the default
  recursion yes;
  // recursive queries only allowed from these ips
  // and references the acl clause
  allow-query { corpnets; };
  // this ensures that any reverse map for private IPs
  // not defined in a zone file will *not* be passed to the public network
  // it is the default value
  empty-zones-enable yes;
  // this defines the addresses of the resolvers to which queries will be forwarded
  forwarders {
    192.168.250.3;
    192.168.230.27;
  };
  // indicates all queries will be forwarded other than for defined zones
  forward only;
};

// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
  };
  category default {
    example_log;
  };
};

// hints zone file is not required

// zone file for the localhost forward map
// discretionary zone depending on hosts file (see discussion)
zone "localhost" {
  type primary;
  file "masters/localhost-forward.db";
  notify no;
};
```

(continues on next page)

(continued from previous page)

```
// zone file for the loopback address
// necessary zone
zone "0.0.127.in-addr.arpa" {
  type primary;
  file "localhost.rev";
  notify no;
};

// zone file for local IP reverse map
// discretionary file depending on requirements
zone "254.168.192.in-addr.arpa" {
  type primary;
  file "192.168.254.rev";
  notify no;
};
```

The *zone* and *acl* blocks, and the *allow-query*, *empty-zones-enable*, *file*, *forward*, *forwarders*, *notify*, *recursion*, and *type* statements are described in detail in the appropriate sections.

As a reminder, the configuration of this forwarding resolver does **not** forward any recursive query for which:

1. The answer is already in the cache.
2. The domain name is **localhost** (zone localhost).
3. Is a reverse-map query for 127.0.0.1 (zone 0.0.127.in-addr.arpa).
4. Is a reverse-map query for 192.168.254/24 (zone 254.168.192.in-addr.arpa).
5. Is a reverse-map query for any local IP (*empty-zones-enable* statement).

All other recursive queries will be forwarded to resolve the query.

### 3.3.4 Selective Forwarding Resolver Configuration

This forwarding resolver configuration only forwards recursive queries for the zone **example.com** to the resolvers at 192.168.250.3 and 192.168.230.27. All other recursive queries, other than those for the defined zones and those for which the answer is already in its cache, are handled by this resolver. The forwarding resolver will cache all responses from both the public network and from the forwarded resolvers. The configuration is closed, in that it defines those IPs from which it will accept recursive queries.

```
// selective forwarding named.conf file
// Two corporate subnets we wish to allow queries from
// defined in an acl clause
acl corpnets {
  192.168.4.0/24;
  192.168.7.0/24;
};

// options clause defining the server-wide properties
options {
  // all relative paths use this directory as a base
  directory "/var";
  // version statement for security to avoid hacking known weaknesses
  // if the real version number is revealed
  version "not currently available";
```

(continues on next page)

(continued from previous page)

```
  // this is the default
  recursion yes;
  // recursive queries only allowed from these ips
  // and references the acl clause
  allow-query { corpnets; };
  // this ensures that any reverse map for private IPs
  // not defined in a zone file will *not* be passed to the public network
  // it is the default value
  empty-zones-enable yes;

  // forwarding is not global but selective by zone in this configuration
};

// logging clause
// log to /var/log/named/example.log all events from info UP in severity (no debug)
// uses 3 files in rotation swaps files when size reaches 250K
// failure messages that occur before logging is established are
// in syslog (/var/log/messages)
//
logging {
  channel example_log {
    // uses a relative path name and the directory statement to
    // expand to /var/log/named/example.log
    file "log/named/example.log" versions 3 size 250k;
    // only log info and up messages - all others discarded
    severity info;
   };
   category default {
     example_log;
  };
};

// zone file for the root servers
// discretionary zone (see root server discussion above)
zone "." {
  type hint;
  file "named.root";
};

// zone file for the localhost forward map
// discretionary zone depending on hosts file (see discussion)
zone "localhost" {
  type primary;
  file "masters/localhost-forward.db";
  notify no;
};

// zone file for the loopback address
// necessary zone
zone "0.0.127.in-addr.arpa" {
  type primary;
  file "localhost.rev";
```

(continues on next page)

(continued from previous page)

```
  notify no;
};

// zone file for local IP reverse map
// discretionary file depending on requirements
zone "254.168.192.in-addr.arpa" {
  type primary;
  file "192.168.254.rev";
  notify no;
};
// zone file forwarded example.com
zone "example.com" {
  type forward;
  // this defines the addresses of the resolvers to
  // which queries for this zone will be forwarded
  forwarders {
    192.168.250.3;
    192.168.230.27;
  };
  // indicates all queries for this zone will be forwarded
  forward only;
};
```

The *zone* and *acl* blocks, and the *allow-query*, *empty-zones-enable*, *file*, *forward*, *forwarders*, *notify*, *recursion*, and *type* statements are described in detail in the appropriate sections.

As a reminder, the configuration of this resolver does **not** access the DNS hierarchy (does not use the public network) for any recursive query for which:

1. The answer is already in the cache.
2. The domain name is **localhost** (zone localhost).
3. Is a reverse-map query for 127.0.0.1 (zone 0.0.127.in-addr.arpa).
4. Is a reverse-map query for 192.168.254/24 (zone 254.168.192.in-addr.arpa).
5. Is a reverse-map query for any local IP (empty-zones-enable statement).
6. Is a query for the domain name **example.com**, in which case it will be forwarded to either 192.168.250.3 or 192.168.230.27 (zone example.com).

All other recursive queries will result in access to the DNS hierarchy to resolve the query.

## 3.4 Load Balancing

A primitive form of load balancing can be achieved in the DNS by using multiple resource records (RRs) in a *zone file* (such as multiple A records) for one name.

For example, assuming three HTTP servers with network addresses of 10.0.0.1, 10.0.0.2, and 10.0.0.3, a set of records such as the following means that clients will connect to each machine one-third of the time:

| Name | TTL | CLASS | TYPE | Resource Record (RR) Data |
|---|---|---|---|---|
| www | 600 | IN | A | 10.0.0.1 |
| | 600 | IN | A | 10.0.0.2 |
| | 600 | IN | A | 10.0.0.3 |

When a resolver queries for these records, BIND rotates them and responds to the query with the records in a random order. In the example above, clients randomly receive records in the order 1, 2, 3; 2, 3, 1; and 3, 1, 2. Most clients use the first record returned and discard the rest.

For more detail on ordering responses, refer to the *rrset-order* statement in the `options` block.

## 3.5 Zone File

This section, largely borrowed from **RFC 1034**, describes the concept of a Resource Record (RR) and explains how to use them.

### 3.5.1 Resource Records

A domain name identifies a node in the DNS tree namespace. Each node has a set of resource information, which may be empty. The set of resource information associated with a particular name is composed of separate RRs. The order of RRs in a set is not significant and need not be preserved by name servers, resolvers, or other parts of the DNS. However, sorting of multiple RRs is permitted for optimization purposes: for example, to specify that a particular nearby server be tried first. See `sortlist` and *RRset Ordering*.

The components of a Resource Record are:

**owner name**
: The domain name where the RR is found.

**RR type**
: An encoded 16-bit value that specifies the type of the resource record. For a list of *types* of valid RRs, including those that have been obsoleted, please refer to *https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4*.

**TTL**
: The time-to-live of the RR. This field is a 32-bit integer in units of seconds, and is primarily used by resolvers when they cache RRs. The TTL describes how long a RR can be cached before it should be discarded.

**class**
: An encoded 16-bit value that identifies a protocol family or an instance of a protocol.

**RDATA**
: The resource data. The format of the data is type- and sometimes class-specific.

The following *classes* of resource records are currently valid in the DNS:

**IN**
: The Internet. The only widely *class* used today.

**CH**
: Chaosnet, a LAN protocol created at MIT in the mid-1970s. It was rarely used for its historical purpose, but was reused for BIND's built-in server information zones, e.g., **version.bind**.

**HS**
: Hesiod, an information service developed by MIT's Project Athena. It was used to share information about various systems databases, such as users, groups, printers, etc.

The *owner name* is often implicit, rather than forming an integral part of the RR. For example, many name servers internally form tree or hash structures for the name space, and chain RRs off nodes. The remaining RR parts are the fixed header (type, class, TTL), which is consistent for all RRs, and a variable part (RDATA) that fits the needs of the resource being described.

The TTL field is a time limit on how long an RR can be kept in a cache. This limit does not apply to authoritative data in zones; that also times out, but follows the refreshing policies for the zone. The TTL is assigned by the administrator for the zone where the data originates. While short TTLs can be used to minimize caching, and a zero TTL prohibits

caching, the realities of Internet performance suggest that these times should be on the order of days for the typical host. If a change is anticipated, the TTL can be reduced prior to the change to minimize inconsistency, and then increased back to its former value following the change.

The data in the RDATA section of RRs is carried as a combination of binary strings and domain names. The domain names are frequently used as "pointers" to other data in the DNS.

#### Textual Expression of RRs

RRs are represented in binary form in the packets of the DNS protocol, and are usually represented in highly encoded form when stored in a name server or resolver. In the examples provided in **RFC 1034**, a style similar to that used in primary files was employed in order to show the contents of RRs. In this format, most RRs are shown on a single line, although continuation lines are possible using parentheses.

The start of the line gives the owner of the RR. If a line begins with a blank, then the owner is assumed to be the same as that of the previous RR. Blank lines are often included for readability.

Following the owner are listed the TTL, type, and class of the RR. Class and type use the mnemonics defined above, and TTL is an integer before the type field. To avoid ambiguity in parsing, type and class mnemonics are disjoint, TTLs are integers, and the type mnemonic is always last. The IN class and TTL values are often omitted from examples in the interest of clarity.

The resource data or RDATA section of the RR is given using knowledge of the typical representation for the data.

For example, the RRs carried in a message might be shown as:

| | | |
|---|---|---|
| **ISI.EDU.** | **MX** | **10 VENERA.ISI.EDU.** |
| | **MX** | **10 VAXA.ISI.EDU** |
| **VENERA.ISI.EDU** | **A** | **128.9.0.32** |
| | **A** | **10.1.0.52** |
| **VAXA.ISI.EDU** | **A** | **10.2.0.27** |
| | **A** | **128.9.0.33** |

The MX RRs have an RDATA section which consists of a 16-bit number followed by a domain name. The address RRs use a standard IP address format to contain a 32-bit Internet address.

The above example shows six RRs, with two RRs at each of three domain names.

Here is another possible example:

| | | |
|---|---|---|
| **XX.LCS.MIT.EDU.** | **IN A** | **10.0.0.44** |
| | **CH A** | **MIT.EDU. 2420** |

This shows two addresses for **XX.LCS.MIT.EDU**, each of a different class.

### 3.5.2 Discussion of MX Records

As described above, domain servers store information as a series of resource records, each of which contains a particular piece of information about a given domain name (which is usually, but not always, a host). The simplest way to think of an RR is as a typed pair of data, a domain name matched with a relevant datum and stored with some additional type information, to help systems determine when the RR is relevant.

MX records are used to control delivery of email. The data specified in the record is a priority and a domain name. The priority controls the order in which email delivery is attempted, with the lowest number first. If two priorities are the same, a server is chosen randomly. If no servers at a given priority are responding, the mail transport agent falls back to the next largest priority. Priority numbers do not have any absolute meaning; they are relevant only respective to other

MX records for that domain name. The domain name given is the machine to which the mail is delivered. It *must* have an associated address record (A or AAAA); CNAME is not sufficient.

For a given domain, if there is both a CNAME record and an MX record, the MX record is in error and is ignored. Instead, the mail is delivered to the server specified in the MX record pointed to by the CNAME. For example:

| | | | | |
|---|---|---|---|---|
| **example.com.** | **IN** | **MX** | **10** | **mail.example.com.** |
| | **IN** | **MX** | **10** | **mail2.example.com.** |
| | **IN** | **MX** | **20** | **mail.backup.org.** |
| **mail.example.com.** | **IN** | **A** | **10.0.0.1** | |
| **mail2.example.com.** | **IN** | **A** | **10.0.0.2** | |

Mail delivery is attempted to **mail.example.com** and **mail2.example.com** (in any order); if neither of those succeeds, delivery to **mail.backup.org** is attempted.

### 3.5.3 Setting TTLs

The time-to-live (TTL) of the RR field is a 32-bit integer represented in units of seconds, and is primarily used by resolvers when they cache RRs. The TTL describes how long an RR can be cached before it should be discarded. The following three types of TTLs are currently used in a zone file.

**SOA minimum**
: The last field in the SOA is the negative caching TTL. This controls how long other servers cache no-such-domain (NXDOMAIN) responses from this server. Further details can be found in **RFC 2308**.

  The maximum time for negative caching is 3 hours (3h).

**$TTL**
: The $TTL directive at the top of the zone file (before the SOA) gives a default TTL for every RR without a specific TTL set.

**RR TTLs**
: Each RR can have a TTL as the second field in the RR, which controls how long other servers can cache it.

All of these TTLs default to units of seconds, though units can be explicitly specified: for example, **1h30m**.

### 3.5.4 Inverse Mapping in IPv4

Reverse name resolution (that is, translation from IP address to name) is achieved by means of the **in-addr.arpa** domain and PTR records. Entries in the in-addr.arpa domain are made in least-to-most significant order, read left to right. This is the opposite order to the way IP addresses are usually written. Thus, a machine with an IP address of 10.1.2.3 would have a corresponding in-addr.arpa name of 3.2.1.10.in-addr.arpa. This name should have a PTR resource record whose data field is the name of the machine or, optionally, multiple PTR records if the machine has more than one name. For example, in the **example.com** domain:

| | |
|---|---|
| **$ORIGIN** | **2.1.10.in-addr.arpa** |
| **3** | **IN PTR foo.example.com.** |

**Note**

The **$ORIGIN** line in this example is only to provide context; it does not necessarily appear in the actual usage. It is only used here to indicate that the example is relative to the listed origin.

### 3.5.5 Other Zone File Directives

The DNS "master file" format was initially defined in RFC 1035 and has subsequently been extended. While the format itself is class-independent, all records in a zone file must be of the same class.

Master file directives include **$ORIGIN**, **$INCLUDE**, and **$TTL.**

#### The @ (at-sign)

When used in the label (or name) field, the asperand or at-sign (@) symbol represents the current origin. At the start of the zone file, it is the **<zone_name>**, followed by a trailing dot (.).

#### The $ORIGIN Directive

Syntax: **$ORIGIN** domain-name [comment]

**$ORIGIN** sets the domain name that is appended to any unqualified records. When a zone is first read, there is an implicit `$ORIGIN <zone_name>.`; note the trailing dot. The current **$ORIGIN** is appended to the domain specified in the **$ORIGIN** argument if it is not absolute.

```
$ORIGIN example.com.
WWW     CNAME   MAIN-SERVER
```

is equivalent to

```
WWW.EXAMPLE.COM. CNAME MAIN-SERVER.EXAMPLE.COM.
```

#### The $INCLUDE Directive

Syntax: **$INCLUDE** filename [origin] [comment]

This reads and processes the file **filename** as if it were included in the file at this point. The **filename** can be an absolute path, or a relative path. In the latter case it is read from *named*'s working directory. If **origin** is specified, the file is processed with **$ORIGIN** set to that value; otherwise, the current **$ORIGIN** is used.

The origin and the current domain name revert to the values they had prior to the **$INCLUDE** once the file has been read.

> **Note**
>
> RFC 1035 specifies that the current origin should be restored after an **$INCLUDE**, but it is silent on whether the current domain name should also be restored. BIND 9 restores both of them. This could be construed as a deviation from RFC 1035, a feature, or both.

#### The $TTL Directive

Syntax: **$TTL** default-ttl [comment]

This sets the default Time-To-Live (TTL) for subsequent records with undefined TTLs. Valid TTLs are of the range 0-2147483647 seconds.

**$TTL** is defined in RFC 2308.

### 3.5.6 BIND Primary File Extension: the $GENERATE Directive

Syntax: **$GENERATE** range owner [ttl] [class] type rdata [comment]

**$GENERATE** is used to create a series of resource records that only differ from each other by an iterator.

**range**
: This can be one of two forms: start-stop or start-stop/step. If the first form is used, then step is set to 1. "start", "stop", and "step" must be positive integers between 0 and (2^31)-1. "start" must not be larger than "stop".

**owner**
: This describes the owner name of the resource records to be created.

  The **owner** string may include one or more **$** (dollar sign) symbols, which will be replaced with the iterator value when generating records; see below for details.

**ttl**
: This specifies the time-to-live of the generated records. If not specified, this is inherited using the normal TTL inheritance rules.

  **class** and **ttl** can be entered in either order.

**class**
: This specifies the class of the generated records. This must match the zone class if it is specified.

  **class** and **ttl** can be entered in either order.

**type**
: This can be any valid type.

**rdata**
: This is a string containing the RDATA of the resource record to be created. As with **owner**, the **rdata** string may include one or more **$** symbols, which are replaced with the iterator value. **rdata** may be quoted if there are spaces in the string; the quotation marks do not appear in the generated record.

  Any single **$** (dollar sign) symbols within the **owner** or **rdata** strings are replaced by the iterator value. To get a **$** in the output, escape the **$** using a backslash ****, e.g., `\$`. (For compatibility with earlier versions, **$$** is also recognized as indicating a literal **$** in the output.)

  The **$** may optionally be followed by modifiers which change the offset from the iterator, field width, and base. Modifiers are introduced by a **{** (left brace) immediately following the **$**, as in **${offset[,width[,base]]}**. For example, **${-20,3,d}** subtracts 20 from the current value and prints the result as a decimal in a zero-padded field of width 3. Available output forms are decimal (**d**), octal (**o**), hexadecimal (**x** or **X** for uppercase), and nibble (**n** or **N** for uppercase). The modfiier cannot contain whitespace or newlines.

  The default modifier is **${0,0,d}**. If the **owner** is not absolute, the current **$ORIGIN** is appended to the name.

  In nibble mode, the value is treated as if it were a reversed hexadecimal string, with each hexadecimal digit as a separate label. The width field includes the label separator.

Examples:

**$GENERATE** can be used to easily generate the sets of records required to support sub-/24 reverse delegations described in **RFC 2317**:

```
$ORIGIN 0.0.192.IN-ADDR.ARPA.
$GENERATE 1-2 @ NS SERVER$.EXAMPLE.
$GENERATE 1-127 $ CNAME $.0
```

is equivalent to

```
0.0.0.192.IN-ADDR.ARPA. NS SERVER1.EXAMPLE.
0.0.0.192.IN-ADDR.ARPA. NS SERVER2.EXAMPLE.
1.0.0.192.IN-ADDR.ARPA. CNAME 1.0.0.0.192.IN-ADDR.ARPA.
2.0.0.192.IN-ADDR.ARPA. CNAME 2.0.0.0.192.IN-ADDR.ARPA.
...
127.0.0.192.IN-ADDR.ARPA. CNAME 127.0.0.0.192.IN-ADDR.ARPA.
```

This example creates a set of A and MX records. Note the MX's **rdata** is a quoted string; the quotes are stripped when **$GENERATE** is processed:

```
$ORIGIN EXAMPLE.
$GENERATE 1-127 HOST-$ A 1.2.3.$
$GENERATE 1-127 HOST-$ MX "0 ."
```

is equivalent to

```
HOST-1.EXAMPLE.   A  1.2.3.1
HOST-1.EXAMPLE.   MX 0 .
HOST-2.EXAMPLE.   A  1.2.3.2
HOST-2.EXAMPLE.   MX 0 .
HOST-3.EXAMPLE.   A  1.2.3.3
HOST-3.EXAMPLE.   MX 0 .
...
HOST-127.EXAMPLE. A  1.2.3.127
HOST-127.EXAMPLE. MX 0 .
```

This example generates A and AAAA records using modifiers; the AAAA **owner** names are generated using nibble mode:

```
$ORIGIN EXAMPLE.
$GENERATE 0-2 HOST-${0,4,d} A 1.2.3.${1,0,d}
$GENERATE 1024-1026 ${0,3,n} AAAA 2001:db8::${0,4,x}
```

is equivalent to:

```
HOST-0000.EXAMPLE.   A      1.2.3.1
HOST-0001.EXAMPLE.   A      1.2.3.2
HOST-0002.EXAMPLE.   A      1.2.3.3
0.0.4.EXAMPLE.       AAAA   2001:db8::400
1.0.4.EXAMPLE.       AAAA   2001:db8::401
2.0.4.EXAMPLE.       AAAA   2001:db8::402
```

The **$GENERATE** directive is a BIND extension and not part of the standard zone file format.

### 3.5.7 Additional File Formats

In addition to the standard text format, BIND 9 supports the ability to read or dump to zone files in other formats.

The **raw** format is a binary representation of zone data in a manner similar to that used in zone transfers. Since it does not require parsing text, load time is significantly reduced.

For a primary server, a zone file in **raw** format is expected to be generated from a text zone file by the *named-compilezone* command. For a secondary server or a dynamic zone, the zone file is automatically generated when *named* dumps the zone contents after zone transfer or when applying prior updates, if one of these formats is specified by the **masterfile-format** option.

If a zone file in **raw** format needs manual modification, it first must be converted to **text** format by the *named-compilezone* command, then converted back after editing. For example:

```
named-compilezone -f raw -F text -o zonefile.text <origin> zonefile.raw
[edit zonefile.text]
named-compilezone -f text -F raw -o zonefile.raw <origin> zonefile.text
```

CHAPTER

FOUR

# NAME SERVER OPERATIONS

## 4.1 Tools for Use With the Name Server Daemon

This section describes several indispensable diagnostic, administrative, and monitoring tools available to the system administrator for controlling and debugging the name server daemon.

### 4.1.1 Diagnostic Tools

The `dig`, `host`, and `nslookup` programs are all command-line tools for manually querying name servers. They differ in style and output format.

**`dig`**
: `dig` is the most versatile and complete of these lookup tools. It has two modes: simple interactive mode for a single query, and batch mode, which executes a query for each in a list of several query lines. All query options are accessible from the command line.

  For more information and a list of available commands and options, see *dig - DNS lookup utility*.

**`host`**
: The `host` utility emphasizes simplicity and ease of use. By default, it converts between host names and Internet addresses, but its functionality can be extended with the use of options.

  For more information and a list of available commands and options, see *host - DNS lookup utility*.

**`nslookup`**
: `nslookup` has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains, or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

  Due to its arcane user interface and frequently inconsistent behavior, we do not recommend the use of `nslookup`. Use `dig` instead.

### 4.1.2 Administrative Tools

Administrative tools play an integral part in the management of a server.

**`named-checkconf`**
: The `named-checkconf` program checks the syntax of a `named.conf` file.

  For more information and a list of available commands and options, see *named-checkconf - named configuration file syntax checking tool*.

**`named-checkzone`**
: The `named-checkzone` program checks a zone file for syntax and consistency.

  For more information and a list of available commands and options, see *named-checkzone - zone file validation tool*.

***named-compilezone***

This tool is similar to *named-checkzone* but it always dumps the zone content to a specified file (typically in a different format).

For more information and a list of available commands and options, see *named-compilezone - zone file converting tool*.

***rndc***

The remote name daemon control (*rndc*) program allows the system administrator to control the operation of a name server.

See *rndc - name server control utility* for details of the available *rndc* commands.

*rndc* requires a configuration file, since all communication with the server is authenticated with digital signatures that rely on a shared secret, and there is no way to provide that secret other than with a configuration file. The default location for the *rndc* configuration file is `/etc/rndc.conf`, but an alternate location can be specified with the *-c* option. If the configuration file is not found, *rndc* also looks in `/etc/rndc.key` (or whatever `sysconfdir` was defined when the BIND build was configured). The `rndc.key` file is generated by running *rndc-confgen -a* as described in *controls*.

The format of the configuration file is similar to that of *named.conf*, but is limited to only three blocks: the *options*, *key*, *server*, and the *include Directive*. These blocks are what associate the secret keys to the servers with which they are meant to be shared. The order of blocks is not significant.

**options**

**Grammar:**

```
options {
        default-key <string>;
        default-port <integer>;
        default-server <string>;
        default-source-address ( <ipv4_address> | * );
        default-source-address-v6 ( <ipv6_address> | * );
};
```

**Blocks:** topmost

**default-server**

**Grammar:** `default-server <string>;`

**Blocks:** options

*default-server* takes a host name or address argument and represents the server that is contacted if no *-s* option is provided on the command line.

**default-key**

**Grammar:** `default-key <string>;`

**Blocks:** options

*default-key* takes the name of a key as its argument, as defined by a *key* block.

**default-port**

**Grammar:** `default-port <integer>;`

**Blocks:** options

*default-port* specifies the port to which *rndc* should connect if no port is given on the command line or in a *server* block.

**default-source-address**

**Grammar:** `default-source-address ( <ipv4_address> | * );`

**Blocks:** options

**default-source-address-v6**

**Grammar:** `default-source-address-v6 ( <ipv6_address> | * );`

**Blocks:** options

*default-source-address* and *default-source-address-v6* specify the IPv4 and IPv6 source address used to communicate with the server if no address is given on the command line or in a *server* block.

**key**

**Grammar server:** `key <string>;`

**Grammar topmost:**

```
key <string> {
        algorithm <string>;
        secret <string>;
}; // may occur multiple times
```

**Blocks:** topmost, server

The *key* block defines a key to be used by *rndc* when authenticating with *named*. Its syntax is identical to the *key* statement in *named.conf*. The keyword *key* is followed by a key name, which must be a valid domain name, though it need not actually be hierarchical; thus, a string like `rndc_key` is a valid name. The *key* block has two statements: *algorithm* and *secret*.

**algorithm**

**Grammar:** `algorithm <string>;`

**Blocks:** key

While the configuration parser accepts any string as the argument to *algorithm*, currently only the strings `hmac-md5`, `hmac-sha1`, `hmac-sha224`, `hmac-sha256`, `hmac-sha384`, and `hmac-sha512` have any meaning.

**secret**

**Grammar:** `secret <string>;`

**Blocks:** key

The secret is a Base64-encoded string as specified in **RFC 3548**.

**server**

**Grammar:**

```
server <string> {
        addresses { ( <quoted_string> [ port <integer> ] | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ); ... };
        key <string>;
        port <integer>;
        source-address ( <ipv4_address> | * );
        source-address-v6 ( <ipv6_address> | * );
}; // may occur multiple times
```

**Blocks:** topmost

The *server* block specifies connection parameters for a given server. The server can be specified as a host name or address.

**addresses**

**Grammar:** `addresses { ( <quoted_string> [ port <integer> ] | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ); ... };`

**Blocks:** server

Specifies one or more addresses to use when communicating with this server.

***key***

Associates a key defined using the *key* statement with a server.

**port**

**Grammar:** `port <integer>;`

**Blocks:** server

Specifes the port *rndc* should connect to on the server.

**source-address**

**Grammar:** `source-address ( <ipv4_address> | * );`

**Blocks:** server

**source-address-v6**

**Grammar:** `source-address-v6 ( <ipv6_address> | * );`

**Blocks:** server

Overrides *default-source-address* and *default-source-address-v6* for this specific server.

A sample minimal configuration file is as follows:

```
key rndc_key {
     algorithm "hmac-sha256";
     secret
       "c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
};
options {
     default-server 127.0.0.1;
     default-key    rndc_key;
};
```

This file, if installed as `/etc/rndc.conf`, allows the command:

*rndc reload*

to connect to 127.0.0.1 port 953 and causes the name server to reload, if a name server on the local machine is running with the following controls statements:

```
controls {
    inet 127.0.0.1
        allow { localhost; } keys { rndc_key; };
};
```

and it has an identical key block for `rndc_key`.

Running the *rndc-confgen* program conveniently creates an *rndc.conf* file, and also displays the corresponding *controls* statement needed to add to *named.conf*. Alternatively, it is possible to run *rndc-confgen -a* to set up an rndc.key file and not modify *named.conf* at all.

## 4.2 Signals

Certain Unix signals cause the name server to take specific actions, as described in the following table. These signals can be sent using the kill command.

| | |
|---|---|
| SIGHUP | Causes the server to read *named.conf* and reload the database. |
| SIGTERM | Causes the server to clean up and exit. |
| SIGINT | Causes the server to clean up and exit. |

## 4.3 Plugins

Plugins are a mechanism to extend the functionality of *named* using dynamically loadable libraries. By using plugins, core server functionality can be kept simple for the majority of users; more complex code implementing optional features need only be installed by users that need those features.

The plugin interface is a work in progress, and is expected to evolve as more plugins are added. Currently, only "query plugins" are supported; these modify the name server query logic. Other plugin types may be added in the future.

The only plugin currently included in BIND is *filter-aaaa.so*, which replaces the filter-aaaa feature that previously existed natively as part of *named*. The code for this feature has been removed from *named* and can no longer be configured using standard *named.conf* syntax, but linking in the *filter-aaaa.so* plugin provides identical functionality.

## 4.4 Configuring Plugins

**plugin**

**Grammar:** plugin ( query ) <string> [ { <unspecified-text> } ]; // may occur multiple times

**Blocks:** topmost, view

**Tags:** server

Configures plugins in *named.conf*.

A plugin is configured with the *plugin* statement in *named.conf*:

```
plugin query "library.so" {
    parameters
};
```

In this example, file library.so is the plugin library. query indicates that this is a query plugin.

Multiple *plugin* statements can be specified, to load different plugins or multiple instances of the same plugin.

parameters are passed as an opaque string to the plugin's initialization routine. Configuration syntax differs depending on the module.

## 4.5 Developing Plugins

Each plugin implements four functions:

- `plugin_register` to allocate memory, configure a plugin instance, and attach to hook points within *named* ,
- `plugin_destroy` to tear down the plugin instance and free memory,
- `plugin_version` to check that the plugin is compatible with the current version of the plugin API,
- `plugin_check` to test syntactic correctness of the plugin parameters.

At various locations within the *named* source code, there are "hook points" at which a plugin may register itself. When a hook point is reached while *named* is running, it is checked to see whether any plugins have registered themselves there; if so, the associated "hook action" - a function within the plugin library - is called. Hook actions may examine the runtime state and make changes: for example, modifying the answers to be sent back to a client or forcing a query to be aborted. More details can be found in the file `lib/ns/include/ns/hooks.h`.

CHAPTER

FIVE

# DNSSEC

DNS Security Extensions (DNSSEC) provide reliable protection from cache poisoning attacks. At the same time these extensions also provide other benefits: they limit the impact of random subdomain attacks on resolver caches and authoritative servers, and provide the foundation for modern applications like authenticated and private e-mail transfer.

To achieve this goal, DNSSEC adds digital signatures to DNS records in authoritative DNS zones, and DNS resolvers verify the validity of the signatures on the received records. If the signatures match the received data, the resolver can be sure that the data was not modified in transit.

**Note**

DNSSEC and transport-level encryption are complementary! Unlike typical transport-level encryption like DNS-over-TLS, DNS-over-HTTPS, or VPN, DNSSEC makes DNS records verifiable at all points of the DNS resolution chain.

This section focuses on ways to deploy DNSSEC using BIND. For a more in-depth discussion of DNSSEC principles (e.g. *How Does DNSSEC Change DNS Lookup?*) please see *DNSSEC Guide*.

## 5.1 Zone Signing

BIND offers several ways to generate signatures and maintain their validity during the lifetime of a DNS zone:

- *Fully Automated (Key and Signing Policy)* - **strongly recommended**
- *Manual Signing* - discouraged, use only for debugging

### 5.1.1 Zone keys

Regardless of the *zone-signing* method in use, cryptographic keys are stored in files named like `Kdnssec.example.+013+12345.key` and `Kdnssec.example.+013+12345.private`. The private key (in the `.private` file) is used to generate signatures, and the public key (in the `.key` file) is used for signature verification. Additionally, the *Fully Automated (Key and Signing Policy)* method creates a third file, `Kdnssec.example+013+12345.state`, which is used to track DNSSEC key timings and to perform key rollovers safely.

These filenames contain:

- the key name, which always matches the zone name (`dnssec.example.`),
- the algorithm number (013 is ECDSAP256SHA256, 008 is RSASHA256, etc.),
- and the key tag, i.e. a non-unique key identifier (12345 in this case).

**Warning**

Private keys are required for full disaster recovery. Back up key files in a safe location and protect them from unauthorized access. Anyone with access to the private key can create fake but seemingly valid DNS data.

### 5.1.2 Fully Automated (Key and Signing Policy)

Key and Signing Policy (KASP) is a method of configuration that describes how to maintain DNSSEC signing keys and how to sign the zone.

This is the recommended, fully automated way to sign and maintain DNS zones. For most use cases users can simply use the built-in default policy, which applies up-to-date DNSSEC practices:

```
zone "dnssec.example" {
    type primary;
    file "dnssec.example.db";
    dnssec-policy default;
};
```

The *dnssec-policy* statement requires dynamic DNS to be set up, or *inline-signing* to be enabled. In the example above we use the latter, because the `default` policy uses *inline-signing*.

This is sufficient to create the necessary signing keys, and generate `DNSKEY`, `RRSIG`, and `NSEC` records for the zone. BIND also takes care of any DNSSEC maintenance for this zone, including replacing signatures that are about to expire and managing *Key Rollovers*.

**Note**

*dnssec-policy* needs write access to the zone. Please see *dnssec-policy* for more details about implications for zone storage.

The default policy creates one key that is used to sign the complete zone, and uses `NSEC` to enable authenticated denial of existence (a secure way to tell which records do not exist in a zone). This policy is recommended and typically does not need to be changed.

If needed, a custom policy can be defined by adding a *dnssec-policy* statement into the configuration:

```
dnssec-policy "custom" {
    dnskey-ttl 600;
    keys {
        ksk lifetime P1Y algorithm ecdsap384sha384;
        zsk lifetime 60d algorithm ecdsap384sha384;
    };
    nsec3param iterations 0 optout no salt-length 0;
};
```

This `custom` policy, for example:

- uses a very short `DNSKEY` TTL (600 seconds),
- uses two keys to sign the zone: a Key Signing Key (KSK) to sign the key related RRsets (`DNSKEY`, `CDS`, and `CDNSKEY`), and a Zone Signing Key (ZSK) to sign the rest of the zone. The KSK is automatically rotated after one year and the ZSK after 60 days.

**Also:**

- The configured keys have a lifetime set and use the ECDSAP384SHA384 algorithm.
- The last line instructs BIND to generate NSEC3 records for *Proof of Non-Existence*, using zero extra iterations and no salt. NSEC3 opt-out is disabled, meaning insecure delegations also get an NSEC3 record.

For more information about KASP configuration see *dnssec-policy*.

The *Advanced Discussions* section in the DNSSEC Guide discusses the various policy settings and may be useful for determining values for specific needs.

### Key Rollover

When using a *dnssec-policy*, a key lifetime can be set to trigger key rollovers. ZSK rollovers are fully automatic, but for KSK and CSK rollovers a DS record needs to be submitted to the parent. See *Secure Delegation* for possible ways to do so.

Once the DS is in the parent (and the DS of the predecessor key is withdrawn), BIND needs to be told that this event has happened. This can be done automatically by configuring parental agents:

```
zone "dnssec.example" {
    type primary;
    file "dnssec.example.db";
    dnssec-policy default;
    parental-agents { 192.0.2.1; };
    checkds explicit;
};
```

Here one server, `192.0.2.1`, is configured for BIND to send DS queries to, to check the DS RRset for `dnssec-example` during key rollovers. This needs to be a trusted server, because BIND does not validate the response. The `checkds` option makes BIND use the explicitly configured parental agents, rather than looking them up by querying for the parent NS records.

If setting up a parental agent is undesirable, it is also possible to tell BIND that the DS is published in the parent with: *rndc dnssec -checkds -key 12345 published dnssec.example.*. and the DS for the predecessor key has been removed with: *rndc dnssec -checkds -key 54321 withdrawn dnssec.example.*. where 12345 and 54321 are the key tags of the successor and predecessor key, respectively.

To roll a key sooner than scheduled, or to roll a key that has an unlimited lifetime, use: *rndc dnssec -rollover -key 12345 dnssec.example.*.

You can pregenerate keys and save them in the key directory. As long as the key has no timing metadata set, it may be selected as a successor in the upcoming key rollover. To pregenerate keys without setting key timing metadata, use the *-G* option: `dnssec-keygen -G dnssec.example.`.

To revert a signed zone back to an insecure zone, change the zone configuration to use the built-in "insecure" policy. Detailed instructions are described in *Reverting to Unsigned*.

### Multi-Signer Model

Dynamic zones provide the ability to sign a zone by multiple providers, meaning each provider signs and serves the same zone independently, as is described in **RFC 8901**. BIND 9 is able to support Model 2, where each provider has their own KSK and ZSK (or CSK). The keys from the other provider can be imported via Dynamic Update. For each active KSK there must be a corresponding DS record in the parent zone. Key rollovers require coordination in order to update the DS and DNSKEY RRset.

### 5.1.3 Manual Signing

There are several tools available to manually sign a zone.

> **Warning**
>
> Please note manual procedures are available mainly for backwards compatibility and should be used only by expert users with specific needs.

To set up a DNSSEC secure zone manually, a series of steps must be followed. Please see chapter *Manual Signing* in the *DNSSEC Guide* for more information.

### 5.1.4 Monitoring with Private Type Records

The state of the signing process is signaled by private type records (with a default type value of 65534). When signing is complete, those records with a non-zero initial octet have a non-zero value for the final octet.

If the first octet of a private type record is non-zero, the record indicates either that the zone needs to be signed with the key matching the record, or that all signatures that match the record should be removed. Here are the meanings of the different values of the first octet:

- algorithm (octet 1)
- key ID in network order (octet 2 and 3)
- removal flag (octet 4)
- complete flag (octet 5)

Only records flagged as "complete" can be removed via dynamic update; attempts to remove other private type records are silently ignored.

If the first octet is zero (this is a reserved algorithm number that should never appear in a `DNSKEY` record), the record indicates that changes to the `NSEC3` chains are in progress. The rest of the record contains an `NSEC3PARAM` record, while the flag field tells what operation to perform based on the flag bits:

0x01 OPTOUT

0x80 CREATE

0x40 REMOVE

0x20 NONSEC

## 5.2 Secure Delegation

Once a zone is signed on the authoritative servers, the last remaining step is to establish chain of trust[1] between the parent zone (`example.`) and the local zone (`dnssec.example.`).

Generally the procedure is:

- **Wait** for stale data to expire from caches. The amount of time required is equal to the maximum TTL value used in the zone before signing. This step ensures that unsigned data expire from caches and resolvers do not get confused by missing signatures.
- Insert/update DS records in the parent zone (`dnssec.example. DS` record).

[1] For further details on how the chain of trust is used in practice, see *The 12-Step DNSSEC Validation Process (Simplified)* in the *DNSSEC Guide*.

There are multiple ways to update DS records in the parent zone. Refer to the documentation for the parent zone to find out which options are applicable to a given case zone. Generally the options are, from most- to least-recommended:

- Automatically update the DS record in the parent zone using `CDS`/`CDNSKEY` records automatically generated by BIND. This requires support for **RFC 7344** in either parent zone, registry, or registrar. In that case, configure BIND to *monitor DS records in the parent zone* and everything will happen automatically at the right time.
- Query the zone for automatically generated `CDS` or `CDNSKEY` records using *dig*, and then insert these records into the parent zone using the method specified by the parent zone (web form, e-mail, API, ...).
- Generate DS records manually using the *dnssec-dsfromkey* utility on *zone keys*, and then insert them into the parent zone.

## 5.3 DNSSEC Validation

The BIND resolver validates answers from authoritative servers by default. This behavior is controlled by the configuration statement *dnssec-validation*.

By default a trust anchor for the DNS root zone is used. This trust anchor is provided as part of BIND and is kept up-to-date using *Dynamic Trust Anchor Management*.

**Note**

DNSSEC validation works "out of the box" and does not require additional configuration. Additional configuration options are intended only for special cases.

To validate answers, the resolver needs at least one trusted starting point, a "trust anchor." Essentially, trust anchors are copies of `DNSKEY` RRs for zones that are used to form the first link in the cryptographic chain of trust. Alternative trust anchors can be specified using *trust-anchors*, but this setup is very unusual and is recommended only for expert use. For more information, see *Trust Anchors* in the *DNSSEC Guide*.

The BIND authoritative server does not verify signatures on load, so zone keys for authoritative zones do not need to be specified in the configuration file.

### 5.3.1 Validation Failures

When DNSSEC validation is configured, the resolver rejects any answers from signed, secure zones which fail to validate, and returns SERVFAIL to the client.

Responses may fail to validate for any of several reasons, including missing, expired, or invalid signatures; a key which does not match the DS RRset in the parent zone; or an insecure response from a zone which, according to its parent, should have been secure.

For more information see *Basic DNSSEC Troubleshooting*.

### 5.3.2 Coexistence With Unsigned (Insecure) Zones

Zones not protected by DNSSEC are called "insecure," and these zones seamlessly coexist with signed zones.

When the validator receives a response from an unsigned zone that has a signed parent, it must confirm with the parent that the zone was intentionally left unsigned. It does this by verifying, via signed and validated *NSEC/NSEC3 records*, that the parent zone contains no DS records for the child.

If the validator *can* prove that the zone is insecure, then the response is accepted. However, if it cannot, the validator must assume an insecure response to be a forgery; it rejects the response and logs an error.

The logged error reads "insecurity proof failed" and "got insecure response; parent indicates it should be secure."

## 5.4 Dynamic Trust Anchor Management

BIND is able to maintain DNSSEC trust anchors using **RFC 5011** key management. This feature allows *named* to keep track of changes to critical DNSSEC keys without any need for the operator to make changes to configuration files.

### 5.4.1 Validating Resolver

To configure a validating resolver to use **RFC 5011** to maintain a trust anchor, configure the trust anchor using a *trust-anchors* statement and the `initial-key` keyword. Information about this can be found in the *trust-anchors* statement description.

### 5.4.2 Authoritative Server

To set up an authoritative zone for **RFC 5011** trust anchor maintenance, generate two (or more) key signing keys (KSKs) for the zone. Sign the zone with one of them; this is the "active" KSK. All KSKs which do not sign the zone are "stand-by" keys.

Any validating resolver which is configured to use the active KSK as an **RFC 5011**-managed trust anchor takes note of the stand-by KSKs in the zone's DNSKEY RRset, and stores them for future reference. The resolver rechecks the zone periodically; after 30 days, if the new key is still there, the key is accepted by the resolver as a valid trust anchor for the zone. Anytime after this 30-day acceptance timer has completed, the active KSK can be revoked, and the zone can be "rolled over" to the newly accepted key.

The easiest way to place a stand-by key in a zone is to use the "smart signing" features of *dnssec-keygen* and *dnssec-signzone*. If a key exists with a publication date in the past, but an activation date which is unset or in the future, *dnssec-signzone -S* includes the DNSKEY record in the zone but does not sign with it:

```
$ dnssec-keygen -K keys -f KSK -P now -A now+2y example.net
$ dnssec-signzone -S -K keys example.net
```

To revoke a key, use the command *dnssec-revoke*. This adds the REVOKED bit to the key flags and regenerates the `K*.key` and `K*.private` files.

After revoking the active key, the zone must be signed with both the revoked KSK and the new active KSK. Smart signing takes care of this automatically.

Once a key has been revoked and used to sign the DNSKEY RRset in which it appears, that key is never again accepted as a valid trust anchor by the resolver. However, validation can proceed using the new active key, which was accepted by the resolver when it was a stand-by key.

See **RFC 5011** for more details on key rollover scenarios.

When a key has been revoked, its key ID changes, increasing by 128 and wrapping around at 65535. So, for example, the key "`Kexample.com.+005+10000`" becomes "`Kexample.com.+005+10128`".

If two keys have IDs exactly 128 apart and one is revoked, the two key IDs will collide, causing several problems. To prevent this, *dnssec-keygen* does not generate a new key if another key which may collide is present. This checking only occurs if the new keys are written to the same directory that holds all other keys in use for that zone.

Older versions of BIND 9 did not have this protection. Exercise caution if using key revocation on keys that were generated by previous releases, or if using keys stored in multiple directories or on multiple machines.

It is expected that a future release of BIND 9 will address this problem in a different way, by storing revoked keys with their original unrevoked key IDs.

## 5.5 PKCS#11 (Cryptoki) Support

Public Key Cryptography Standard #11 (PKCS#11) defines a platform-independent API for the control of hardware security modules (HSMs) and other cryptographic support devices.

PKCS#11 uses a "provider library": a dynamically loadable library which provides a low-level PKCS#11 interface to drive the HSM hardware. The PKCS#11 provider library comes from the HSM vendor, and it is specific to the HSM to be controlled.

BIND 9 accesses PKCS#11 libraries via OpenSSL extensions. The extension for OpenSSL 3 and newer is pkcs11-provider; for older OpenSSL versions, engine_pkcs11 from the OpenSC project can be used.

In both cases the extension is dynamically loaded into OpenSSL and the HSM is operated indirectly; any cryptographic operations not supported by the HSM can be carried out by OpenSSL instead.

### 5.5.1 Prerequisites

See the documentation provided by the HSM vendor for information about installing, initializing, testing, and troubleshooting the HSM.

### 5.5.2 Building SoftHSMv2

SoftHSMv2, the latest development version of SoftHSM, is available from https://github.com/softhsm/SoftHSMv2. It is a software library developed by the OpenDNSSEC project (https://www.opendnssec.org) which provides a PKCS#11 interface to a virtual HSM, implemented in the form of an SQLite3 database on the local filesystem. It provides less security than a true HSM, but it allows users to experiment with native PKCS#11 when an HSM is not available. SoftHSMv2 can be configured to use either OpenSSL or the Botan library to perform cryptographic functions, but when using it for native PKCS#11 in BIND, OpenSSL is required.

By default, the SoftHSMv2 configuration file is `prefix/etc/softhsm2.conf` (where `prefix` is configured at compile time). This location can be overridden by the SOFTHSM2_CONF environment variable. The SoftHSMv2 cryptographic store must be installed and initialized before using it with BIND.

```
$ cd SoftHSMv2
$ configure --with-crypto-backend=openssl --prefix=/opt/pkcs11/usr
$ make
$ make install
$ /opt/pkcs11/usr/bin/softhsm-util --init-token 0 --slot 0 --label softhsmv2
```

### 5.5.3 OpenSSL 1.x.x With engine_pkcs11

OpenSSL engine-based PKCS#11 uses the engine_pkcs11 OpenSSL engine from the libp11 project.

engine_pkcs11 tries to fit the PKCS#11 API within the engine API of OpenSSL. That is, it provides a gateway between PKCS#11 modules and the OpenSSL engine API. One has to register the engine with OpenSSL and one has to provide the path to the PKCS#11 module which should be gatewayed to. This can be done by editing the OpenSSL configuration file, by engine specific controls, or by using the p11-kit proxy module.

It is recommended, that libp11 >= 0.4.12 is used.

For more detailed instructions, including examples, we recommend reading:

https://gitlab.isc.org/isc-projects/bind9/-/wikis/BIND-9-PKCS11

When using engine_pkcs11, be sure to pass the *-E pkcs11* argument to all BIND binaries that potentially use the keys, to activate the engine support.

Even though OpenSSL 3 has compatibility support for Engine API, its use is not recommended due to bugs in OpenSSL and libp11.

It is not possible to generate new keys via engine_pkcs11, so its use is not recommended in a `dnssec-policy` setup. However, it is possible to put previously generated keys in the `key-directory` and let the key manager select those keys when a key rollover is started.

### 5.5.4 Configuring engine_pkcs11

The canonical documentation for configuring engine_pkcs11 is in the libp11/README.md file, but a sample working configuration is included here for the user's convenience:

In our example, we use a custom copy of OpenSSL configuration, driven by an environment variable called OPENSSL_CONF. First, copy the global OpenSSL configuration (often found in `etc/ssl/openssl.conf`) and customize it to use engine_pkcs11.

```
cp /etc/ssl/openssl.cnf /opt/bind9/etc/openssl.cnf
```

Then, export the environment variable:

```
export OPENSSL_CONF=/opt/bind9/etc/openssl.cnf
```

Then add the following line at the top of the file, before any sections (in square brackets) are defined:

```
openssl_conf = openssl_init
```

Make sure there are no other 'openssl_conf = ...' lines in the file.

Add the following lines at the bottom of the file:

```
[openssl_init]
engines=engine_section

[engine_section]
pkcs11 = pkcs11_section

[pkcs11_section]
engine_id = pkcs11
dynamic_path = <PATHTO>/pkcs11.so
MODULE_PATH = <FULL_PATH_TO_HSM_MODULE>
# if automatic logging to the token is needed, PIN can be specified as below
#PIN = 1234
init = 0
```

### 5.5.5 Enabling the OpenSSL Engine in BIND Commands

When using OpenSSL Engine-based PKCS#11, the "engine" to be used by OpenSSL can be specified in *named* and in all of the BIND `dnssec-*` tools by using the `-E <engine>` command-line option. This engine name matches the `engine_id` in the `openssl.cnf` created in previous section.

The zone signing commences as usual, with only one small difference: we need to provide the name of the OpenSSL engine using the -E command-line option.

```
dnssec-signzone -E pkcs11 -S -o example.net example.net
```

### 5.5.6 OpenSSL 3 With pkcs11-provider

OpenSSL provider-based PKCS#11 uses the pkcs11-provider project.

pkcs11-provider tries to fit the PKCS#11 API within the Provider API of OpenSSL; that is, it provides a gateway between PKCS#11 modules and the OpenSSL Provider API. The engine must be registered with OpenSSL and the path to the PKCS#11 module gateway must be provided. This can be done by editing the OpenSSL configuration file, by engine-specific controls, or by using the p11-kit proxy module.

The pkcs11-provider git commit 2e8c26b4157fd21422c66f0b4d7b26cf8c320570 from October 2, 2023 or later must be used.

BIND support for pkcs11-provider is built in; with pcks11-provider, the -E command-line option explained above should not be used.

### 5.5.7 Configuring pkcs11-provider

The canonical documentation for configuring pkcs11-provider is in the provider-pkcs11.7 manual page, but a copy of a working configuration is provided here for convenience:

In this example, we use a custom copy of OpenSSL configuration, driven by an environment variable called OPENSSL_CONF. First, copy the global OpenSSL configuration (often found in `etc/ssl/openssl.conf`) and customize it to use pkcs11-provider.

```
cp /etc/ssl/openssl.cnf /opt/bind9/etc/openssl.cnf
```

Next, export the environment variable:

```
export OPENSSL_CONF=/opt/bind9/etc/openssl.cnf
```

Then add the following line at the top of the file, before any sections (in square brackets) are defined:

```
openssl_conf = openssl_init
```

Make sure there are no other 'openssl_conf = ...' lines in the file.

Add the following lines at the bottom of the file:

```
[openssl_init]
providers = provider_init

[provider_init]
default = default_init
pkcs11 = pkcs11_init

[default_init]
activate = 1

[pkcs11_init]
module = <PATHTO>/pkcs11.so
pkcs11-module-path = <FULL_PATH_TO_HSM_MODULE>
# bind uses the digest+sign api. this is broken with the default load behaviour,
# but works with early load. see:␣
↪https://github.com/openssl-projects/pkcs11-provider/issues/266
pkcs11-module-load-behavior = early
# no-deinit quirk is needed if you use softhsm2
#pkcs11-module-quirks = no-deinit
```

(continues on next page)

(continued from previous page)

```
# if automatic logging to the token is needed, PIN can be specified as below
# the file referenced should contain just the PIN
#pkcs11-module-token-pin = file:/etc/pki/pin.txt
activate = 1
```

### 5.5.8 Key Generation

HSM keys can now be created and used. We are assuming that BIND 9 is already installed, either from a package or from the sources, and the tools are readily available in the `$PATH`.

For generating the keys, we are going to use `pkcs11-tool` available from the OpenSC suite. On both DEB-based and RPM-based distributions, the package is called opensc.

We need to generate at least two RSA keys:

```
pkcs11-tool --module <FULL_PATH_TO_HSM_MODULE> -l -k --key-type rsa:2048 --label␣
↪example.net-ksk --pin <PIN>
pkcs11-tool --module <FULL_PATH_TO_HSM_MODULE> -l -k --key-type rsa:2048 --label␣
↪example.net-zsk --pin <PIN>
```

Remember that each key should have unique label and we are going to use that label to reference the private key.

Convert the RSA keys stored in the HSM into a format that BIND 9 understands. The *dnssec-keyfromlabel* tool from BIND 9 can link the raw keys stored in the HSM with the `K<zone>+<alg>+<id>` files.

The OpenSSL engine name (`pkcs11`) must be provided if using the engine and the algorithm (`RSASHA256`). The key is referenced with the PKCS#11 URI scheme; it can contain the PKCS#11 token label (we assume that it has been initialized as bind9), the PKCS#11 object label (called "label" when generating the keys using `pkcs11-tool`), and the HSM PIN. Refer to **RFC 7512** for the full PKCS#11 URI specification.

Convert the KSK:

```
dnssec-keyfromlabel -E pkcs11 -a RSASHA256 -l
↪"pkcs11:token=bind9;object=example.net-ksk;pin-value=0000" -f KSK example.net
```

and ZSK:

```
dnssec-keyfromlabel -E pkcs11 -a RSASHA256 -l
↪"pkcs11:token=bind9;object=example.net-zsk;pin-value=0000" example.net
```

NOTE: a PIN stored on disk can be used by specifying `pin-source=<path_to>/<file>`, e.g:

```
(umask 0700 && echo -n 0000 > /opt/bind9/etc/pin.txt)
```

and then use in the label specification:

```
pin-source=/opt/bind9/etc/pin.txt
```

Confirm that there is one KSK and one ZSK present in the current directory:

```
ls -l K*
```

The output should look like this (the second number will be different):

```
Kexample.net.+008+31729.key
Kexample.net.+008+31729.private
Kexample.net.+008+42231.key
Kexample.net.+008+42231.private
```

A note on generating ECDSA keys: there is a bug in libp11 when looking up a key. That function compares keys only on their ID, not the label, so when looking up a key it returns the first key, rather than the matching key. To work around this when creating ECDSA keys, specify a unique ID:

```
ksk=$(echo "example.net-ksk" | openssl sha1 -r | awk '{print $1}')
zsk=$(echo "example.net-zsk" | openssl sha1 -r | awk '{print $1}')
pkcs11-tool --module <FULL_PATH_TO_HSM_MODULE> -l -k --key-type EC:prime256v1 --id
↪$ksk --label example.net-ksk --pin <PIN>
pkcs11-tool --module <FULL_PATH_TO_HSM_MODULE> -l -k --key-type EC:prime256v1 --id
↪$zsk --label example.net-zsk --pin <PIN>
```

### 5.5.9 Running *named* With Automatic Zone Re-signing

The zone can also be signed automatically by *named*. Again, we need to provide the name of the OpenSSL engine using the *-E* command-line option, if using OpenSSL 1.x.x with engine_pkcs11; this is not needed when using OpenSSL 3.x.x providers.

```
named -E pkcs11 -c named.conf
```

The logs should have lines like:

```
Fetching example.net/RSASHA256/31729 (KSK) from key repository.
DNSKEY example.net/RSASHA256/31729 (KSK) is now published
DNSKEY example.net/RSA256SHA256/31729 (KSK) is now active
Fetching example.net/RSASHA256/42231 (ZSK) from key repository.
DNSKEY example.net/RSASHA256/42231 (ZSK) is now published
DNSKEY example.net/RSA256SHA256/42231 (ZSK) is now active
```

For *named* to dynamically re-sign zones using HSM keys, and/or to sign new records inserted via nsupdate, *named* must have access to the HSM PIN. In OpenSSL-based PKCS#11, this is accomplished by placing the PIN into the `openssl.cnf` file (in the above examples, `/opt/pkcs11/usr/ssl/openssl.cnf`).

See OpenSSL extension-specific documentation for instructions on configuring the PIN on the global level; doing so allows the `dnssec-\*` tools to access the HSM without PIN entry. (The `pkcs11-\*` tools access the HSM directly, not via OpenSSL, so a PIN is still required to use them.)

CHAPTER

SIX

# ADVANCED CONFIGURATIONS

## 6.1 Dynamic Update

Dynamic update is a method for adding, replacing, or deleting records in a primary server by sending it a special form of DNS messages. The format and meaning of these messages is specified in **RFC 2136**.

Dynamic update is enabled by including an *allow-update* or an *update-policy* clause in the *zone* statement.

If the zone's *update-policy* is set to `local`, updates to the zone are permitted for the key `local-ddns`, which is generated by *named* at startup. See *Dynamic Update Policies* for more details.

Dynamic updates using Kerberos-signed requests can be made using the TKEY/GSS protocol, by setting the *tkey-gssapi-keytab* option. Once enabled, Kerberos-signed requests are matched against the update policies for the zone, using the Kerberos principal as the signer for the request.

Updating of secure zones (zones using DNSSEC) follows **RFC 3007**: RRSIG, NSEC, and NSEC3 records affected by updates are automatically regenerated by the server using an online zone key. Update authorization is based on transaction signatures and an explicit server policy.

### 6.1.1 The Journal File

All changes made to a zone using dynamic update are stored in the zone's journal file. This file is automatically created by the server when the first dynamic update takes place. The name of the journal file is formed by appending the extension `.jnl` to the name of the corresponding zone file unless specifically overridden. The journal file is in a binary format and should not be edited manually.

The server also occasionally writes ("dumps") the complete contents of the updated zone to its zone file. This is not done immediately after each dynamic update because that would be too slow when a large zone is updated frequently. Instead, the dump is delayed by up to 15 minutes, allowing additional updates to take place. During the dump process, transient files are created with the extensions `.jnw` and `.jbk`; under ordinary circumstances, these are removed when the dump is complete, and can be safely ignored.

When a server is restarted after a shutdown or crash, it replays the journal file to incorporate into the zone any updates that took place after the last zone dump.

Changes that result from incoming incremental zone transfers are also journaled in a similar way.

The zone files of dynamic zones cannot normally be edited by hand because they are not guaranteed to contain the most recent dynamic changes; those are only in the journal file. The only way to ensure that the zone file of a dynamic zone is up-to-date is to run *rndc stop*.

To make changes to a dynamic zone manually, follow these steps: first, disable dynamic updates to the zone using *rndc freeze zone*. This updates the zone file with the changes stored in its `.jnl` file. Then, edit the zone file. Finally, run *rndc thaw zone* to reload the changed zone and re-enable dynamic updates.

*rndc sync zone* updates the zone file with changes from the journal file without stopping dynamic updates; this may be useful for viewing the current zone state. To remove the `.jnl` file after updating the zone file, use *rndc sync -clean*.

## 6.2 NOTIFY

DNS NOTIFY is a mechanism that allows primary servers to notify their secondary servers of changes to a zone's data. In response to a NOTIFY message from a primary server, the secondary checks to see that its version of the zone is the current version and, if not, initiates a zone transfer.

For more information about DNS NOTIFY, see the description of the *notify* and *also-notify* statements. The NOTIFY protocol is specified in **RFC 1996**.

> **Note**
>
> As a secondary zone can also be a primary to other secondaries, *named*, by default, sends NOTIFY messages for every zone it loads.

## 6.3 Incremental Zone Transfers (IXFR)

The incremental zone transfer (IXFR) protocol is a way for secondary servers to transfer only changed data, instead of having to transfer an entire zone. The IXFR protocol is specified in **RFC 1995**.

When acting as a primary server, BIND 9 supports IXFR for those zones where the necessary change history information is available. These include primary zones maintained by dynamic update and secondary zones whose data was obtained by IXFR. For manually maintained primary zones, and for secondary zones obtained by performing a full zone transfer (AXFR), IXFR is supported only if the option *ixfr-from-differences* is set to `yes`.

When acting as a secondary server, BIND 9 attempts to use IXFR unless it is explicitly disabled. For more information about disabling IXFR, see the description of the *request-ixfr* clause of the *server* statement.

When a secondary server receives a zone via AXFR, it creates a new copy of the zone database and then swaps it into place; during the loading process, queries continue to be served from the old database with no interference. When receiving a zone via IXFR, however, changes are applied to the running zone, which may degrade query performance during the transfer. If a server receiving an IXFR request determines that the response size would be similar in size to an AXFR response, it may wish to send AXFR instead. The threshold at which this determination is made can be configured using the *max-ixfr-ratio* option.

## 6.4 Split DNS

Setting up different views of the DNS space to internal and external resolvers is usually referred to as a *split DNS* setup. There are several reasons an organization might want to set up its DNS this way.

One common reason to use split DNS is to hide "internal" DNS information from "external" clients on the Internet. There is some debate as to whether this is actually useful. Internal DNS information leaks out in many ways (via email headers, for example) and most savvy "attackers" can find the information they need using other means. However, since listing addresses of internal servers that external clients cannot possibly reach can result in connection delays and other annoyances, an organization may choose to use split DNS to present a consistent view of itself to the outside world.

Another common reason for setting up a split DNS system is to allow internal networks that are behind filters or in **RFC 1918** space (reserved IP space, as documented in **RFC 1918**) to resolve DNS on the Internet. Split DNS can also be used to allow mail from outside back into the internal network.

### 6.4.1 Example Split DNS Setup

Let's say a company named *Example, Inc.* (`example.com`) has several corporate sites that have an internal network with reserved Internet Protocol (IP) space and an external demilitarized zone (DMZ), or "outside" section of a network, that is available to the public.

Example, Inc. wants its internal clients to be able to resolve external hostnames and to exchange mail with people on the outside. The company also wants its internal resolvers to have access to certain internal-only zones that are not available at all outside of the internal network.

To accomplish this, the company sets up two sets of name servers. One set is on the inside network (in the reserved IP space) and the other set is on bastion hosts, which are "proxy" hosts in the DMZ that can talk to both sides of its network.

The internal servers are configured to forward all queries, except queries for `site1.internal`, `site2.internal`, `site1.example.com`, and `site2.example.com`, to the servers in the DMZ. These internal servers have complete sets of information for `site1.example.com`, `site2.example.com`, `site1.internal`, and `site2.internal`.

To protect the `site1.internal` and `site2.internal` domains, the internal name servers must be configured to disallow all queries to these domains from any external hosts, including the bastion hosts.

The external servers, which are on the bastion hosts, are configured to serve the "public" version of the `site1.example.com` and `site2.example.com` zones. This could include things such as the host records for public servers (`www.example.com` and `ftp.example.com`) and mail exchange (MX) records (`a.mx.example.com` and `b.mx.example.com`).

In addition, the public `site1.example.com` and `site2.example.com` zones should have special MX records that contain wildcard (`*`) records pointing to the bastion hosts. This is needed because external mail servers have no other way of determining how to deliver mail to those internal hosts. With the wildcard records, the mail is delivered to the bastion host, which can then forward it on to internal hosts.

Here's an example of a wildcard MX record:

```
*   IN MX 10 external1.example.com.
```

Now that they accept mail on behalf of anything in the internal network, the bastion hosts need to know how to deliver mail to internal hosts. The resolvers on the bastion hosts need to be configured to point to the internal name servers for DNS resolution.

Queries for internal hostnames are answered by the internal servers, and queries for external hostnames are forwarded back out to the DNS servers on the bastion hosts.

For all of this to work properly, internal clients need to be configured to query *only* the internal name servers for DNS queries. This could also be enforced via selective filtering on the network.

If everything has been set properly, Example, Inc.'s internal clients are now able to:

- Look up any hostnames in the `site1.example.com` and `site2.example.com` zones.
- Look up any hostnames in the `site1.internal` and `site2.internal` domains.
- Look up any hostnames on the Internet.
- Exchange mail with both internal and external users.

Hosts on the Internet are able to:

- Look up any hostnames in the `site1.example.com` and `site2.example.com` zones.
- Exchange mail with anyone in the `site1.example.com` and `site2.example.com` zones.

Here is an example configuration for the setup just described above. Note that this is only configuration information; for information on how to configure the zone files, see *Configurations and Zone Files*.

Internal DNS server config:

```
acl internals { 172.16.72.0/24; 192.168.1.0/24; };

acl externals { bastion-ips-go-here; };
```

(continues on next page)

(continued from previous page)

```
options {
    ...
    ...
    forward only;
    // forward to external servers
    forwarders {
    bastion-ips-go-here;
    };
    // sample allow-transfer (no one)
    allow-transfer { none; };
    // restrict query access
    allow-query { internals; externals; };
    // restrict recursion
    allow-recursion { internals; };
    ...
    ...
};

// sample primary zone
zone "site1.example.com" {
  type primary;
  file "m/site1.example.com";
  // do normal iterative resolution (do not forward)
  forwarders { };
  allow-query { internals; externals; };
  allow-transfer { internals; };
};

// sample secondary zone
zone "site2.example.com" {
  type secondary;
  file "s/site2.example.com";
  primaries { 172.16.72.3; };
  forwarders { };
  allow-query { internals; externals; };
  allow-transfer { internals; };
};

zone "site1.internal" {
  type primary;
  file "m/site1.internal";
  forwarders { };
  allow-query { internals; };
  allow-transfer { internals; }
};

zone "site2.internal" {
  type secondary;
  file "s/site2.internal";
  primaries { 172.16.72.3; };
  forwarders { };
  allow-query { internals };
```

(continues on next page)

(continued from previous page)

```
  allow-transfer { internals; }
};
```

External (bastion host) DNS server configuration:

```
acl internals { 172.16.72.0/24; 192.168.1.0/24; };

acl externals { bastion-ips-go-here; };

options {
  ...
  ...
  // sample allow-transfer (no one)
  allow-transfer { none; };
  // default query access
  allow-query { any; };
  // restrict cache access
  allow-query-cache { internals; externals; };
  // restrict recursion
  allow-recursion { internals; externals; };
  ...
  ...
};

// sample secondary zone
zone "site1.example.com" {
  type primary;
  file "m/site1.foo.com";
  allow-transfer { internals; externals; };
};

zone "site2.example.com" {
  type secondary;
  file "s/site2.foo.com";
  primaries { another_bastion_host_maybe; };
  allow-transfer { internals; externals; }
};
```

In the `resolv.conf` (or equivalent) on the bastion host(s):

```
search ...
nameserver 172.16.72.2
nameserver 172.16.72.3
nameserver 172.16.72.4
```

## 6.5 IPv6 Support in BIND 9

BIND 9 fully supports all currently defined forms of IPv6 name-to-address and address-to-name lookups. It also uses IPv6 addresses to make queries when running on an IPv6-capable system.

For forward lookups, BIND 9 supports only AAAA records. **RFC 3363** deprecated the use of A6 records, and client-side support for A6 records was accordingly removed from BIND 9. However, authoritative BIND 9 name servers still load

zone files containing A6 records correctly, answer queries for A6 records, and accept zone transfer for a zone containing A6 records.

For IPv6 reverse lookups, BIND 9 supports the traditional "nibble" format used in the `ip6.arpa` domain, as well as the older, deprecated `ip6.int` domain. Older versions of BIND 9 supported the "binary label" (also known as "bitstring") format, but support of binary labels has been completely removed per **RFC 3363**. Many applications in BIND 9 do not understand the binary label format at all anymore, and return an error if one is given. In particular, an authoritative BIND 9 name server will not load a zone file containing binary labels.

### 6.5.1 Address Lookups Using AAAA Records

The IPv6 AAAA record is a parallel to the IPv4 A record, and, unlike the deprecated A6 record, specifies the entire IPv6 address in a single record. For example:

```
$ORIGIN example.com.
host            3600    IN      AAAA    2001:db8::1
```

Use of IPv4-in-IPv6 mapped addresses is not recommended. If a host has an IPv4 address, use an A record, not a AAAA, with `::ffff:192.168.42.1` as the address.

### 6.5.2 Address-to-Name Lookups Using Nibble Format

When looking up an address in nibble format, the address components are simply reversed, just as in IPv4, and `ip6.arpa.` is appended to the resulting name. For example, the following commands produce a reverse name lookup for a host with address `2001:db8::1`:

```
$ORIGIN 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  14400   IN    PTR    (
                    host.example.com. )
```

## 6.6 Dynamically Loadable Zones (DLZ)

Dynamically Loadable Zones (DLZ) are an extension to BIND 9 that allows zone data to be retrieved directly from an external database. There is no required format or schema.

There are number of contributed DLZ modules for several different database backends, including MySQL and LDAP, but they are not actively maintained.

The DLZ module provides data to *named* in text format, which is then converted to DNS wire format by *named*. This conversion, and the lack of any internal caching, places significant limits on the query performance of DLZ modules. Consequently, DLZ is not recommended for use on high-volume servers. However, it can be used in a hidden primary configuration, with secondaries retrieving zone updates via AXFR. Note, however, that DLZ has no built-in support for DNS notify; secondary servers are not automatically informed of changes to the zones in the database.

### 6.6.1 Configuring DLZ

**dlz**

**Grammar zone (primary, redirect, secondary):** `dlz <string>;`

**Grammar topmost, view:**

```
dlz <string> {
        database <string>;
        search <boolean>;
}; // may occur multiple times
```

**Blocks:** topmost, view, zone (primary, redirect, secondary)

**Tags:** zone

Configures a Dynamically Loadable Zone (DLZ) database in *named.conf*.

A DLZ database is configured with a *dlz* statement in *named.conf*:

```
dlz example {
database "dlopen driver.so args";
search yes;
};
```

This specifies a DLZ module to search when answering queries; the module is implemented in `driver.so` and is loaded at runtime by the dlopen DLZ driver. Multiple *dlz* statements can be specified.

**search**

**Grammar:** `search <boolean>;`

**Blocks:** dlz, view.dlz

**Tags:** query

Specifies whether a Dynamically Loadable Zone (DLZ) module is queried for an answer to a query name.

When answering a query, all DLZ modules with *search* set to `yes` are queried to see whether they contain an answer for the query name. The best available answer is returned to the client.

The *search* option in the above example can be omitted, because `yes` is the default value.

If *search* is set to `no`, this DLZ module is *not* searched for the best match when a query is received. Instead, zones in this DLZ must be separately specified in a zone statement. This allows users to configure a zone normally using standard zone-option semantics, but specify a different database backend for storage of the zone's data. For example, to implement NXDOMAIN redirection using a DLZ module for backend storage of redirection rules:

```
dlz other {
       database "dlopen driver.so args";
       search no;
};

zone "." {
       type redirect;
       dlz other;
};
```

### 6.6.2 Sample DLZ Module

For guidance in the implementation of DLZ modules, the `example` directory in the [gitlab.isc.org/isc-projects/dlz-modules](https://gitlab.isc.org/isc-projects/dlz-modules/-/tree/main/example?ref_type=heads) contains a basic dynamically linkable DLZ module - i.e., one which can be loaded at runtime by the "dlopen" DLZ driver. The example sets up a single zone, whose name is passed to the module as an argument in the *dlz* statement:

```
dlz other {
       database "dlopen driver.so example.nil";
};
```

In the above example, the module is configured to create a zone "example.nil", which can answer queries and AXFR requests and accept DDNS updates. At runtime, prior to any updates, the zone contains an SOA, NS, and a single A record at the apex:

```
example.nil.  3600    IN      SOA     example.nil. hostmaster.example.nil. (
                          123 900 600 86400 3600
                      )
example.nil.  3600    IN      NS      example.nil.
example.nil.  1800    IN      A       10.53.0.1
```

The sample driver can retrieve information about the querying client and alter its response on the basis of this information. To demonstrate this feature, the example driver responds to queries for "source-addr.``zonename``>/TXT" with the source address of the query. Note, however, that this record will *not* be included in AXFR or ANY responses. Normally, this feature is used to alter responses in some other fashion, e.g., by providing different address records for a particular name depending on the network from which the query arrived.

Documentation of the DLZ module API can be found in [README](https://gitlab.isc.org/isc-projects/dlz-modules/-/raw/main/example/README). This repository also contains the header file [dlz_minimal.h](https://gitlab.isc.org/isc-projects/dlz-modules/-/raw/main/modules/include/dlz_minimal.h), which defines the API and should be included by any dynamically linkable DLZ module.

## 6.7 Dynamic Database (DynDB)

Dynamic Database, or DynDB, is an extension to BIND 9 which, like DLZ (see *Dynamically Loadable Zones (DLZ)*), allows zone data to be retrieved from an external database. Unlike DLZ, a DynDB module provides a full-featured BIND zone database interface. Where DLZ translates DNS queries into real-time database lookups, resulting in relatively poor query performance, and is unable to handle DNSSEC-signed data due to its limited API, a DynDB module can pre-load an in-memory database from the external data source, providing the same performance and functionality as zones served natively by BIND.

A DynDB module supporting LDAP has been created by Red Hat and is available from https://pagure.io/bind-dyndb-ldap.

A sample DynDB module for testing and developer guidance is included with the BIND source code, in the directory `bin/tests/system/dyndb/driver`.

### 6.7.1 Configuring DynDB

**dyndb**

**Grammar:** `dyndb <string> <quoted_string> { <unspecified-text> }; // may occur multiple times`

**Blocks:** topmost, view

**Tags:** zone

Configures a DynDB database in *named.conf*.

A DynDB database is configured with a *dyndb* statement in *named.conf*:

```
dyndb example "driver.so" {
    parameters
};
```

The file `driver.so` is a DynDB module which implements the full DNS database API. Multiple *dyndb* statements can be specified, to load different drivers or multiple instances of the same driver. Zones provided by a DynDB module are added to the view's zone table, and are treated as normal authoritative zones when BIND responds to queries. Zone configuration is handled internally by the DynDB module.

The parameters are passed as an opaque string to the DynDB module's initialization routine. Configuration syntax differs depending on the driver.

### 6.7.2 Sample DynDB Module

For guidance in the implementation of DynDB modules, the directory `bin/tests/system/dyndb/driver` contains a basic DynDB module. The example sets up two zones, whose names are passed to the module as arguments in the *dyndb* statement:

```
dyndb sample "sample.so" { example.nil. arpa. };
```

In the above example, the module is configured to create a zone, "example.nil", which can answer queries and AXFR requests and accept DDNS updates. At runtime, prior to any updates, the zone contains an SOA, NS, and a single A record at the apex:

```
example.nil.  86400    IN      SOA     example.nil. example.nil. (
                                              0 28800 7200 604800 86400
                                      )
example.nil.  86400    IN      NS      example.nil.
example.nil.  86400    IN      A       127.0.0.1
```

When the zone is updated dynamically, the DynDB module determines whether the updated RR is an address (i.e., type A or AAAA); if so, it automatically updates the corresponding PTR record in a reverse zone. Note that updates are not stored permanently; all updates are lost when the server is restarted.

## 6.8 Catalog Zones

A "catalog zone" is a special DNS zone that contains a list of other zones to be served, along with their configuration parameters. Zones listed in a catalog zone are called "member zones." When a catalog zone is loaded or transferred to a secondary server which supports this functionality, the secondary server creates the member zones automatically. When the catalog zone is updated (for example, to add or delete member zones, or change their configuration parameters), those changes are immediately put into effect. Because the catalog zone is a normal DNS zone, these configuration changes can be propagated using the standard AXFR/IXFR zone transfer mechanism.

The format and behavior of catalog zones are specified in **RFC 9432**.

### 6.8.1 Principle of Operation

Normally, if a zone is to be served by a secondary server, the *named.conf* file on the server must list the zone, or the zone must be added using *rndc addzone*. In environments with a large number of secondary servers, and/or where the zones being served are changing frequently, the overhead involved in maintaining consistent zone configuration on all the secondary servers can be significant.

A catalog zone is a way to ease this administrative burden: it is a DNS zone that lists member zones that should be served by secondary servers. When a secondary server receives an update to the catalog zone, it adds, removes, or reconfigures member zones based on the data received.

To use a catalog zone, it must first be set up as a normal zone on both the primary and secondary servers that are configured to use it. It must also be added to a *catalog-zones* list in the *options* or *view* statement in *named.conf*. This is comparable to the way a policy zone is configured as a normal zone and also listed in a *response-policy* statement.

To use the catalog zone feature to serve a new member zone:

- Set up the member zone to be served on the primary as normal. This can be done by editing *named.conf* or by running *rndc addzone*.
- Add an entry to the catalog zone for the new member zone. This can be done by editing the catalog zone's zone file and running *rndc reload*, or by updating the zone using *nsupdate*.

The change to the catalog zone is propagated from the primary to all secondaries using the normal AXFR/IXFR mechanism. When the secondary receives the update to the catalog zone, it detects the entry for the new member zone, creates an instance of that zone on the secondary server, and points that instance to the *primaries* specified in the catalog zone data. The newly created member zone is a normal secondary zone, so BIND immediately initiates a transfer of zone contents from the primary. Once complete, the secondary starts serving the member zone.

Removing a member zone from a secondary server requires only deleting the member zone's entry in the catalog zone; the change to the catalog zone is propagated to the secondary server using the normal AXFR/IXFR transfer mechanism. The secondary server, on processing the update, notices that the member zone has been removed, stops serving the zone, and removes it from its list of configured zones. However, removing the member zone from the primary server must be done by editing the configuration file or running *rndc delzone*.

### 6.8.2 Configuring Catalog Zones

**catalog-zones**

**Grammar:** `catalog-zones { zone <string> [ default-primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... } ] [ zone-directory <quoted_string> ] [ in-memory <boolean> ] [ min-update-interval <duration> ]; ... };`

**Blocks:** options, view

**Tags:** zone

Configures catalog zones in *named.conf*.

Catalog zones are configured with a *catalog-zones* statement in the *options* or *view* section of *named.conf*. For example:

```
catalog-zones {
    zone "catalog.example"
         default-primaries { 10.53.0.1; }
         in-memory no
         zone-directory "catzones"
         min-update-interval 10;
};
```

This statement specifies that the zone `catalog.example` is a catalog zone. This zone must be properly configured in the same view. In most configurations, it would be a secondary zone.

The options following the zone name are not required, and may be specified in any order.

**default-masters**
Synonym for `default-primaries`.

**default-primaries**
This option defines the default primaries for member zones listed in a catalog zone, and can be overridden by options within a catalog zone. If no such options are included, then member zones transfer their contents from the servers listed in this option.

**in-memory**
: This option, if set to `yes`, causes member zones to be stored only in memory. This is functionally equivalent to configuring a secondary zone without a *`file`* option. The default is `no`; member zones' content is stored locally in a file whose name is automatically generated from the view name, catalog zone name, and member zone name.

**zone-directory**
: This option causes local copies of member zones' zone files to be stored in the specified directory, if `in-memory` is not set to `yes`. The default is to store zone files in the server's working directory. A non-absolute pathname in `zone-directory` is assumed to be relative to the working directory.

**min-update-interval**
: This option sets the minimum interval between updates to catalog zones, in seconds. If an update to a catalog zone (for example, via IXFR) happens less than `min-update-interval` seconds after the most recent update, the changes are not carried out until this interval has elapsed. The default is 5 seconds.

Catalog zones are defined on a per-view basis. Configuring a non-empty *`catalog-zones`* statement in a view automatically turns on *`allow-new-zones`* for that view. This means that *`rndc addzone`* and *`rndc delzone`* also work in any view that supports catalog zones.

### 6.8.3 Catalog Zone Format

A catalog zone is a regular DNS zone; therefore, it must have a single `SOA` and at least one `NS` record.

A record stating the version of the catalog zone format is also required. If the version number listed is not supported by the server, then a catalog zone may not be used by that server.

```
catalog.example.    IN SOA . . 2016022901 900 600 86400 1
catalog.example.    IN NS invalid.
version.catalog.example.    IN TXT "2"
```

Note that this record must have the domain name `version.catalog-zone-name`. The data stored in a catalog zone is indicated by the domain name label immediately before the catalog zone domain. Currently BIND supports catalog zone schema versions "1" and "2".

Also note that the catalog zone must have an NS record in order to be a valid DNS zone, and using the value "invalid." for NS is recommended.

A member zone is added by including a `PTR` resource record in the `zones` sub-domain of the catalog zone. The record label can be any unique label. The target of the PTR record is the member zone name. For example, to add member zones `domain.example` and `domain2.example`:

```
5960775ba382e7a4e09263fc06e7c00569b6a05c.zones.catalog.example. IN PTR domain.example.
uniquelabel.zones.catalog.example. IN PTR domain2.example.
```

The label is necessary to identify custom properties (see below) for a specific member zone. Also, the zone state can be reset by changing its label, in which case BIND will remove the member zone and add it back.

### 6.8.4 Catalog Zone Custom Properties

BIND uses catalog zones custom properties to define different properties which can be set either globally for the whole catalog zone or for a single member zone. Global custom properties override the settings in the configuration file, and member zone custom properties override global custom properties.

For the version "1" of the schema custom properties must be placed without a special suffix.

For the version "2" of the schema custom properties must be placed under the ".ext" suffix.

Global custom properties are set at the apex of the catalog zone, e.g.:

```
primaries.ext.catalog.example.    IN AAAA 2001:db8::1
```

BIND currently supports the following custom properties:

- A simple *primaries* definition:

  ```
  primaries.ext.catalog.example.    IN A 192.0.2.1
  ```

  This custom property defines a primary server for the member zones, which can be either an A or AAAA record. If multiple primaries are set, the order in which they are used is random.

  Note: `masters` can be used as a synonym for *primaries*.

- A *primaries* with a TSIG key defined:

  ```
  label.primaries.ext.catalog.example.     IN A 192.0.2.2
  label.primaries.ext.catalog.example.     IN TXT "tsig_key_name"
  ```

  This custom property defines a primary server for the member zone with a TSIG key set. The TSIG key must be configured in the configuration file. `label` can be any valid DNS label.

  Note: `masters` can be used as a synonym for *primaries*.

- *allow-query* and *allow-transfer* ACLs:

  ```
  allow-query.ext.catalog.example.   IN APL 1:10.0.0.1/24
  allow-transfer.ext.catalog.example.    IN APL !1:10.0.0.1/32 1:10.0.0.0/24
  ```

  These custom properties are the equivalents of *allow-query* and *allow-transfer* options in a zone declaration in the *named.conf* configuration file. The ACL is processed in order; if there is no match to any rule, the default policy is to deny access. For the syntax of the APL RR, see **RFC 3123**.

The member zone-specific custom properties are defined the same way as global custom properties, but in the member zone subdomain:

```
primaries.ext.5960775ba382e7a4e09263fc06e7c00569b6a05c.zones.catalog.example. IN A␣
↪192.0.2.2
label.primaries.ext.5960775ba382e7a4e09263fc06e7c00569b6a05c.zones.catalog.example.␣
↪IN AAAA 2001:db8::2
label.primaries.ext.5960775ba382e7a4e09263fc06e7c00569b6a05c.zones.catalog.example.␣
↪IN TXT "tsig_key_name"
allow-query.ext.5960775ba382e7a4e09263fc06e7c00569b6a05c.zones.catalog.example. IN␣
↪APL 1:10.0.0.0/24
primaries.ext.uniquelabel.zones.catalog.example. IN A 192.0.2.3
```

Custom properties defined for a specific zone override the global custom properties defined in the catalog zone. These in turn override the global options defined in the *catalog-zones* statement in the configuration file.

Note that none of the global records for a custom property are inherited if any records are defined for that custom property for the specific zone. For example, if the zone had a *primaries* record of type A but not AAAA, it would *not* inherit the type AAAA record from the global custom property or from the global option in the configuration file.

### 6.8.5 Change of Ownership (coo)

BIND supports the catalog zones "Change of Ownership" (coo) property. When the same entry which exists in one catalog zone is added into another catalog zone, the default behavior for BIND is to ignore it, and continue serving the zone using the catalog zone where it was originally existed, unless it is removed from there, then it can be added into the new one.

Using the `coo` property it is possible to gracefully move a zone from one catalog zone into another, by letting the catalog consumers know that it is permitted to do so. To do that, the original catalog zone should be updated with a new record with `coo` custom property:

```
uniquelabel.zones.catalog.example. IN PTR domain2.example.
coo.uniquelabel.zones.catalog.example. IN PTR catalog2.example.
```

Here, the `catalog.example` catalog zone gives permission for the member zone with label "uniquelabel" to be transferred into `catalog2.example` catalog zone. Catalog consumers which support the `coo` property will then take note, and when the zone is finally added into `catalog2.example` catalog zone, catalog consumers will change the ownership of the zone from `catalog.example` to `catalog2.example`. BIND's implementation simply deletes the zone from the old catalog zone and adds it back into the new catalog zone, which also means that all associated state for the just migrated zone will be reset, including when the unique label is the same.

The record with `coo` custom property can be later deleted by the catalog zone operator after confirming that all the consumers have received it and have successfully changed the ownership of the zone.

## 6.9 DNS Firewalls and Response Policy Zones

A DNS firewall examines DNS traffic and allows some responses to pass through while blocking others. This examination can be based on several criteria, including the name requested, the data (such as an IP address) associated with that name, or the name or IP address of the name server that is authoritative for the requested name. Based on these criteria, a DNS firewall can be configured to discard, modify, or replace the original response, allowing administrators more control over what systems can access or be accessed from their networks.

DNS Response Policy Zones (RPZ) are a form of DNS firewall in which the firewall rules are expressed within the DNS itself - encoded in an open, vendor-neutral format as records in specially constructed DNS zones.

Using DNS zones to configure policy allows policy to be shared from one server to another using the standard DNS zone transfer mechanism. This allows a DNS operator to maintain their own firewall policies and share them easily amongst their internal name servers, or to subscribe to external firewall policies such as commercial or cooperative "threat feeds," or both.

*named* can subscribe to up to 64 Response Policy Zones, each of which encodes a separate policy rule set. Each rule is stored in a DNS resource record set (RRset) within the RPZ, and consists of a **trigger** and an **action**. There are five types of triggers and six types of actions.

A response policy rule in a DNS RPZ can be triggered as follows:

- by the IP address of the client
- by the query name
- by an address which would be present in a truthful response
- by the name or address of an authoritative name server responsible for publishing the original response

A response policy action can be one of the following:

- to synthesize a "domain does not exist" (NXDOMAIN) response
- to synthesize a "name exists but there are no records of the requested type" (NODATA) response
- to drop the response
- to switch to TCP by sending a truncated UDP response that requires the DNS client to try again with TCP
- to replace/override the response's data with specific data (provided within the response policy zone)
- to exempt the response from further policy processing

The most common use of a DNS firewall is to "poison" a domain name, IP address, name server name, or name server IP address. Poisoning is usually done by forcing a synthetic "domain does not exist" (NXDOMAIN) response. This means that if an administrator maintains a list of known "phishing" domains, these names can be made unreachable by customers or end users just by adding a firewall policy into the recursive DNS server, with a trigger for each known "phishing" domain, and an action in every case forcing a synthetic NXDOMAIN response. It is also possible to use a data-replacement action such as answering for these known "phishing" domains with the name of a local web server that can display a warning page. Such a web server would be called a "walled garden."

**Note**

Authoritative name servers can be responsible for many different domains. If DNS RPZ is used to poison all domains served by some authoritative name server name or address, the effects can be quite far-reaching. Users are advised to ensure that such authoritative name servers do not also serve domains that should not be poisoned.

### 6.9.1 Why Use a DNS Firewall?

Criminal and network abuse traffic on the Internet often uses the Domain Name System (DNS), so protection against these threats should include DNS firewalling. A DNS firewall can selectively intercept DNS queries for known network assets including domain names, IP addresses, and name servers. Interception can mean rewriting a DNS response to direct a web browser to a "walled garden," or simply making any malicious network assets invisible and unreachable.

### 6.9.2 What Can a DNS Firewall Do?

Firewalls work by applying a set of rules to a traffic flow, where each rule consists of a trigger and an action. Triggers determine which messages within the traffic flow are handled specially, and actions determine what that special handling is. For a DNS firewall, the traffic flow to be controlled consists of responses sent by a recursive DNS server to its end-user clients. Some true responses are not safe for all clients, so the policy rules in a DNS firewall allow some responses to be intercepted and replaced with safer content.

### 6.9.3 Creating and Maintaining RPZ Rule Sets

In DNS RPZ, the DNS firewall policy rule set is stored in a DNS zone, which is maintained and synchronized using the same tools and methods as for any other DNS zone. The primary name server for a DNS RPZ may be an internal server, if an administrator is creating and maintaining their own DNS policy zone, or it may be an external name server (such as a security vendor's server), if importing a policy zone published externally. The primary copy of the DNS firewall policy can be a DNS "zone file" which is either edited by hand or generated from a database. A DNS zone can also be edited indirectly using DNS dynamic updates (for example, using the "nsupdate" shell level utility).

DNS RPZ allows firewall rules to be expressed in a DNS zone format and then carried to subscribers as DNS data. A recursive DNS server which is capable of processing DNS RPZ synchronizes these DNS firewall rules using the same standard DNS tools and protocols used for secondary name service. The DNS policy information is then promoted to the DNS control plane inside the customer's DNS resolver, making that server into a DNS firewall.

A security company whose products include threat intelligence feeds can use a DNS Response Policy Zone (RPZ) as a delivery channel to customers. Threats can be expressed as known-malicious IP addresses and subnets, known-malicious domain names, and known-malicious domain name servers. By feeding this threat information directly into customers' local DNS resolvers, providers can transform these DNS servers into a distributed DNS firewall.

When a customer's DNS resolver is connected by a realtime subscription to a threat intelligence feed, the provider can protect the customer's end users from malicious network elements (including IP addresses and subnets, domain names, and name servers) immediately as they are discovered. While it may take days or weeks to "take down" criminal and abusive infrastructure once reported, a distributed DNS firewall can respond instantly.

Other distributed TCP/IP firewalls have been on the market for many years, and enterprise users are now comfortable importing real-time threat intelligence from their security vendors directly into their firewalls. This intelligence can take

the form of known-malicious IP addresses or subnets, or of patterns which identify known-malicious email attachments, file downloads, or web addresses (URLs). In some products it is also possible to block DNS packets based on the names or addresses they carry.

### 6.9.4 Limitations of DNS RPZ

We're often asked if DNS RPZ could be used to set up redirection to a CDN. For example, if "mydomain.com" is a normal domain with SOA, NS, MX, TXT records etc., then if someone sends an A or AAAA query for "mydomain.com", can we use DNS RPZ on an authoritative nameserver to return "CNAME mydomain.com.my-cdn-provider.net"?

The problem with this suggestion is that there is no way to CNAME only A and AAAA queries, not even with RPZ.

The underlying reason is that if the authoritative server answers with a CNAME, the recursive server making that query will cache the response. Thereafter (while the CNAME is still in cache), it assumes that there are no records of any non-CNAME type for the name that was being queried, and directs subsequent queries for all other types directly to the target name of the CNAME record.

To be clear, this is not a limitation of RPZ; it is a function of the way the DNS protocol works. It's simply not possible to use "partial" CNAMES to help when setting up CDNs because doing this will break other functionality such as email routing.

Similarly, following the DNS protocol definition, wildcards in the form of `*.example` records might behave in unintuitive ways. For a detailed definition of wildcards in DNS, please see **RFC 4592**, especially section 2.

### 6.9.5 DNS Firewall Usage Examples

Here are some scenarios in which a DNS firewall might be useful.

Some known threats are based on an IP address or subnet (IP address range). For example, an analysis may show that all addresses in a "class C" network are used by a criminal gang for "phishing" web servers. With a DNS firewall based on DNS RPZ, a firewall policy can be created such as "if a DNS lookup would result in an address from this class C network, then answer instead with an NXDOMAIN indication." That simple rule would prevent any end users inside customers' networks from being able to look up any domain name used in this phishing attack – without having to know in advance what those names might be.

Other known threats are based on domain names. An analysis may determine that a certain domain name or set of domain names is being or will shortly be used for spamming, phishing, or other Internet-based attacks which all require working domain names. By adding name-triggered rules to a distributed DNS firewall, providers can protect customers' end users from any attacks which require them to be able to look up any of these malicious names. The names can be wildcards (for example, *.evil.com), and these wildcards can have exceptions if some domains are not as malicious as others (if *.evil.com is bad, then not.evil.com might be an exception).

Alongside growth in electronic crime has come growth of electronic criminal expertise. Many criminal gangs now maintain their own extensive DNS infrastructure to support a large number of domain names and a diverse set of IP addressing resources. Analyses show in many cases that the only truly fixed assets criminal organizations have are their name servers, which are by nature slightly less mobile than other network assets. In such cases, DNS administrators can anchor their DNS firewall policies in the abusive name server names or name server addresses, and thus protect their customers' end users from threats where neither the domain name nor the IP address of that threat is known in advance.

Electronic criminals rely on the full resiliency of DNS just as the rest of digital society does. By targeting criminal assets at the DNS level we can deny these criminals the resilience they need. A distributed DNS firewall can leverage the high skills of a security company to protect a large number of end users. DNS RPZ, as the first open and vendor-neutral distributed DNS firewall, can be an effective way to deliver threat intelligence to customers.

#### A Real-World Example of DNS RPZ's Value

The Conficker malware worm (https://en.wikipedia.org/wiki/Conficker) was first detected in 2008. Although it is no longer an active threat, the techniques described here can be applied to other DNS threats.

Conficker used a domain generation algorithm (DGA) to choose up to 50,000 command and control domains per day. It would be impractical to create an RPZ that contains so many domain names and changes so much on a daily basis. Instead, we can trigger RPZ rules based on the names of the name servers that are authoritative for the command and control domains, rather than trying to trigger on each of 50,000 different (daily) query names. Since the well-known name server names for Conficker's domain names are never used by nonmalicious domains, it is safe to poison all lookups that rely on these name servers. Here is an example that achieves this result:

```
$ORIGIN rpz.example.com.
ns.0xc0f1c3a5.com.rpz-nsdname  CNAME  *.walled-garden.example.com.
ns.0xc0f1c3a5.net.rpz-nsdname  CNAME  *.walled-garden.example.com.
ns.0xc0f1c3a5.org.rpz-nsdname  CNAME  *.walled-garden.example.com.
```

The `*` at the beginning of these CNAME target names is special, and it causes the original query name to be prepended to the CNAME target. So if a user tries to visit the Conficker command and control domain *racaldftn.com.ai* (which was a valid Conficker command and control domain name on 19-October-2011), the RPZ-configured recursive name server will send back this answer:

```
racaldftn.com.ai.     CNAME     racaldftn.com.ai.walled-garden.example.com.
racaldftn.com.ai.walled-garden.example.com.     A      192.168.50.3
```

This example presumes that the following DNS content has also been created, which is not part of the RPZ zone itself but is in another domain:

```
$ORIGIN walled-garden.example.com.
*     A     192.168.50.3
```

Assuming that we're running a web server listening on 192.168.50.3 that always displays a warning message no matter what uniform resource identifier (URI) is used, the above RPZ configuration will instruct the web browser of any infected end user to connect to a "server name" consisting of their original lookup name (racaldftn.com.ai) prepended to the walled garden domain name (walled-garden.example.com). This is the name that will appear in the web server's log file, and having the full name in that log file will facilitate an analysis as to which users are infected with what virus.

### 6.9.6 Keeping Firewall Policies Updated

It is vital for overall system performance that incremental zone transfers (see **RFC 1995**) and real-time change notification (see **RFC 1996**) be used to synchronize DNS firewall rule sets between the publisher's primary copy of the rule set and the subscribers' working copies of the rule set.

If DNS dynamic updates are used to maintain a DNS RPZ rule set, the name server automatically calculates a stream of deltas for use when sending incremental zone transfers to the subscribing name servers. Sending a stream of deltas – known as an "incremental zone transfer" or IXFR – is usually much faster than sending the full zone every time it changes, so it's worth the effort to use an editing method that makes such incremental transfers possible.

Administrators who edit or periodically regenerate a DNS RPZ rule set and whose primary name server uses BIND can enable the *ixfr-from-differences* option, which tells the primary name server to calculate the differences between each new zone and the preceding version, and to make these differences available as a stream of deltas for use in incremental zone transfers to the subscribing name servers. This will look something like the following:

```
options {
          // ...
          ixfr-from-differences yes;
```

(continues on next page)

(continued from previous page)

```
        // ...
};
```

As mentioned above, the simplest and most common use of a DNS firewall is to poison domain names known to be purely malicious, by simply making them disappear. All DNS RPZ rules are expressed as resource record sets (RRsets), and the way to express a "force a name-does-not-exist condition" is by adding a CNAME pointing to the root domain (`.`). In practice this looks like:

```
$ORIGIN rpz.example.com.
malicious1.org          CNAME .
*.malicious1.org        CNAME .
malicious2.org          CNAME .
*.malicious2.org        CNAME .
```

Two things are noteworthy in this example. First, the malicious names are made relative within the response policy zone. Since there is no trailing dot following ".org" in the above example, the actual RRsets created within this response policy zone are, after expansion:

```
malicious1.org.rpz.example.com.         CNAME .
*.malicious1.org.rpz.example.com.       CNAME .
malicious2.org.rpz.example.com.         CNAME .
*.malicious2.org.rpz.example.com.       CNAME .
```

Second, for each name being poisoned, a wildcard name is also listed. This is because a malicious domain name probably has or may potentially have malicious subdomains.

In the above example, the relative domain names *malicious1.org* and *malicious2.org* will match only the real domain names *malicious1.org* and *malicious2.org*, respectively. The relative domain names **.malicious1.org* and **.malicious2.org* will match any *subdomain.of.malicious1.org* or *subdomain.of.malicious2.org*, respectively.

This example forces an NXDOMAIN condition as its policy action, but other policy actions are also possible.

### 6.9.7 Performance and Scalability When Using Multiple RPZs

Since version 9.10, BIND can be configured to have different response policies depending on the identity of the querying client and the nature of the query. To configure BIND response policy, the information is placed into a zone file whose only purpose is conveying the policy information to BIND. A zone file containing response policy information is called a Response Policy Zone, or RPZ, and the mechanism in BIND that uses the information in those zones is called DNS RPZ.

It is possible to use as many as 64 separate RPZ files in a single instance of BIND, and BIND is not significantly slowed by such heavy use of RPZ.

(Note: by default, BIND 9.11 only supports up to 32 RPZ files, but this can be increased to 64 at compile time. All other supported versions of BIND support 64 by default.)

Each one of the policy zone files can specify policy for as many different domains as necessary. The limit of 64 is on the number of independently-specified policy collections and not the number of zones for which they specify policy.

Policy information from all of the policy zones together are stored in a special data structure allowing simultaneous lookups across all policy zones to be performed very rapidly. Looking up a policy rule is proportional to the logarithm of the number of rules in the largest single policy zone.

### 6.9.8 Practical Tips for DNS Firewalls and DNS RPZ

Administrators who subscribe to an externally published DNS policy zone and who have a large number of internal recursive name servers should create an internal name server called a "distribution master" (DM). The DM is a secondary (stealth secondary) name server from the publisher's point of view; that is, the DM is fetching zone content from the external server. The DM is also a primary name server from the internal recursive name servers' point of view: they fetch zone content from the DM. In this configuration the DM is acting as a gateway between the external publisher and the internal subscribers.

The primary server must know the unicast listener address of every subscribing recursive server, and must enumerate all of these addresses as destinations for real time zone change notification (as described in **RFC 1996**). So if an enterprise-wide RPZ is called "rpz.example.com" and if the unicast listener addresses of four of the subscribing recursive name servers are 192.0.200.1, 192.0.201.1, 192.0.202.1, and 192.0.203.1, the primary server's configuration looks like this:

```
zone "rpz.example.com" {
     type primary;
     file "primary/rpz.example.com";
     notify explicit;
     also-notify { 192.0.200.1;
                   192.0.201.1;
                   192.0.202.1;
                   192.0.203.1; };
     allow-transfer { 192.0.200.1;
                      192.0.201.1;
                      192.0.202.1;
                      192.0.203.1; };
     allow-query { localhost; };
};
```

Each recursive DNS server that subscribes to the policy zone must be configured as a secondary server for the zone, and must also be configured to use the policy zone for local response policy. To subscribe a recursive name server to a response policy zone where the unicast listener address of the primary server is 192.0.220.2, the server's configuration should look like this:

```
options {
     // ...
     response-policy {
          zone "rpz.example.com";
     };
     // ...
};

zone "rpz.example.com";
     type secondary;
     primaries { 192.0.222.2; };
     file "secondary/rpz.example.com";
     allow-query { localhost; };
     allow-transfer { none; };
};
```

Note that queries are restricted to "localhost," since query access is never used by DNS RPZ itself, but may be useful to DNS operators for use in debugging. Transfers should be disallowed to prevent policy information leaks.

If an organization's business continuity depends on full connectivity with another company whose ISP also serves some criminal or abusive customers, it's possible that one or more external RPZ providers – that is, security feed vendors – may eventually add some RPZ rules that could hurt a company's connectivity to its business partner. Users can protect

themselves from this risk by using an internal RPZ in addition to any external RPZs, and by putting into their internal RPZ some "pass-through" rules to prevent any policy action from affecting a DNS response that involves a business partner.

A recursive DNS server can be connected to more than one RPZ, and these are searched in order. Therefore, to protect a network from dangerous policies which may someday appear in external RPZ zones, administrators should list the internal RPZ zones first.

```
options {
     // ...
     response-policy {
          zone "rpz.example.com";
          zone "rpz.security-vendor-1.com";
          zone "rpz.security-vendor-2.com";
     };
     // ...
};
```

Within an internal RPZ, there need to be rules describing the network assets of business partners whose communications need to be protected. Although it is not generally possible to know what domain names they use, administrators will be aware of what address space they have and perhaps what name server names they use.

```
$ORIGIN rpz.example.com.
8.0.0.0.10.rpz-ip                CNAME   rpz-passthru.
16.0.0.45.128.rpz-nsip           CNAME   rpz-passthru.
ns.partner1.com.rpz-nsdname      CNAME   rpz-passthru.
ns.partner2.com.rpz-nsdname      CNAME   rpz-passthru.
```

Here, we know that answers in address block 10.0.0.0/8 indicate a business partner, as well as answers involving any name server whose address is in the 128.45.0.0/16 address block, and answers involving the name servers whose names are ns.partner1.com or ns.partner2.com.

The above example demonstrates that when matching by answer IP address (the .rpz-ip owner), or by name server IP address (the .rpz-nsip owner) or by name server domain name (the .rpz-nsdname owner), the special RPZ marker (.rpz-ip, .rpz-nsip, or .rpz-nsdname) does not appear as part of the CNAME target name.

By triggering these rules using the known network assets of a partner, and using the "pass-through" policy action, no later RPZ processing (which in the above example refers to the "rpz.security-vendor-1.com" and "rpz.security-vendor-2.com" policy zones) will have any effect on DNS responses for partner assets.

### 6.9.9 Creating a Simple Walled Garden Triggered by IP Address

It may be the case that the only thing known about an attacker is the IP address block they are using for their "phishing" web servers. If the domain names and name servers they use are unknown, but it is known that every one of their "phishing" web servers is within a small block of IP addresses, a response can be triggered on all answers that would include records in this address range, using RPZ rules that look like the following example:

```
$ORIGIN rpz.example.com.
22.0.212.94.109.rpz-ip           CNAME   drop.garden.example.com.
*.212.94.109.in-addr.arpa        CNAME   .
*.213.94.109.in-addr.arpa        CNAME   .
*.214.94.109.in-addr.arpa        CNAME   .
*.215.94.109.in-addr.arpa        CNAME   .
```

Here, if a truthful answer would include an A (address) RR (resource record) whose value were within the 109.94.212.0/22 address block, then a synthetic answer is sent instead of the truthful answer. Assuming the query is for www.malicious.net, the synthetic answer is:

```
www.malicious.net.              CNAME   drop.garden.example.com.
drop.garden.example.com.        A       192.168.7.89
```

This assumes that *drop.garden.example.com* has been created as real DNS content, outside of the RPZ:

```
$ORIGIN example.com.
drop.garden                     A       192.168.7.89
```

In this example, there is no "*" in the CNAME target name, so the original query name will not be present in the walled garden web server's log file. This is an undesirable loss of information, and is shown here for example purposes only.

The above example RPZ rules would also affect address-to-name (also known as "reverse DNS") lookups for the unwanted addresses. If a mail or web server receives a connection from an address in the example's 109.94.212.0/22 address block, it will perform a PTR record lookup to find the domain name associated with that IP address.

This kind of address-to-name translation is usually used for diagnostic or logging purposes, but it is also common for email servers to reject any email from IP addresses which have no address-to-name translation. Most mail from such IP addresses is spam, so the lack of a PTR record here has some predictive value. By using the "force name-does-not-exist" policy trigger on all lookups in the PTR name space associated with an address block, DNS administrators can give their servers a hint that these IP addresses are probably sending junk.

### 6.9.10 A Known Inconsistency in DNS RPZ's NSDNAME and NSIP Rules

Response Policy Zones define several possible triggers for each rule, and among these two are known to produce inconsistent results. This is not a bug; rather, it relates to inconsistencies in the DNS delegation model.

#### DNS Delegation

In DNS authority data, an NS RRset that is not at the apex of a DNS zone creates a sub-zone. That sub-zone's data is separate from the current (or "parent") zone, and it can have different authoritative name servers than the current zone. In this way, the root zone leads to COM, NET, ORG, and so on, each of which have their own name servers and their own way of managing their authoritative data. Similarly, ORG has delegations to ISC.ORG and to millions of other "dot-ORG" zones, each of which can have its own set of authoritative name servers. In the parlance of the protocol, these NS RRsets below the apex of a zone are called "delegation points." An NS RRset at a delegation point contains a list of authoritative servers to which the parent zone is delegating authority for all names at or below the delegation point.

At the apex of every zone there is also an NS RRset. Ideally, this so-called "apex NS RRset" should be identical to the "delegation point NS RRset" in the parent zone, but this ideal is not always achieved. In the real DNS, it's almost always easier for a zone administrator to update one of these NS RRsets than the other, so that one will be correct and the other out of date. This inconsistency is so common that it's been necessarily rendered harmless: domains that are inconsistent in this way are less reliable and perhaps slower, but they still function as long as there is some overlap between each of the NS RRsets and the truth. ("Truth" in this case refers to the actual set of name servers that are authoritative for the zone.)

#### A Quick Review of DNS Iteration

In DNS recursive name servers, an incoming query that cannot be answered from the local cache is sent to the closest known delegation point for the query name. For example, if a server is looking up XYZZY.ISC.ORG and it the name servers for ISC.ORG, then it sends the query to those servers directly; however, if it has never heard of ISC.ORG before, it must first send the query to the name servers for ORG (or perhaps even to the root zone that is the parent of ORG).

When it asks one of the parent name servers, that server will not have an answer, so it sends a "referral" consisting only of the "delegation point NS RRset." Once the server receives this referral, it "iterates" by sending the same query again, but this time to name servers for a more specific part of the query name. Eventually this iteration terminates, usually by getting an answer or a "name error" (NXDOMAIN) from the query name's authoritative server, or by encountering some type of server failure.

When an authoritative server for the query name sends an answer, it has the option of including a copy of the zone's apex NS RRset. If this occurs, the recursive name server caches this NS RRset, replacing the delegation point NS RRset that it had received during the iteration process. In the parlance of the DNS, the delegation point NS RRset is "glue," meaning non-authoritative data, or more of a hint than a real truth. On the other hand, the apex NS RRset is authoritative data, coming as it does from the zone itself, and it is considered more credible than the "glue." For this reason, it's a little bit more important that the apex NS RRset be correct than that the delegation point NS RRset be correct, since the former will quickly replace the latter, and will be used more often for a longer total period of time.

Importantly, the authoritative name server need not include its apex NS RRset in any answers, and recursive name servers do not ordinarily query directly for this RRset. Therefore it is possible for the apex NS RRset to be completely wrong without any operational ill-effects, since the wrong data need not be exposed. Of course, if a query comes in for this NS RRset, most recursive name servers will forward the query to the zone's authority servers, since it's bad form to return "glue" data when asked a specific question. In these corner cases, bad apex NS RRset data can cause a zone to become unreachable unpredictably, according to what other queries the recursive name server has processed.

There is another kind of "glue," for name servers whose names are below delegation points. If ORG delegates ISC.ORG to NS-EXT.ISC.ORG, the ORG server needs to know an address for NS-EXT.ISC.ORG and return this address as part of the delegation response. However, the name-to-address binding for this name server is only authoritative inside the ISC.ORG zone; therefore, the A or AAAA RRset given out with the delegation is non-authoritative "glue," which is replaced by an authoritative RRset if one is seen. As with apex NS RRsets, the real A or AAAA RRset is not automatically queried for by the recursive name server, but is queried for if an incoming query asks for this RRset.

### Enter RPZ

RPZ has two trigger types that are intended to allow policy zone authors to target entire groups of domains based on those domains all being served by the same DNS servers: NSDNAME and NSIP. The NSDNAME and NSIP rules are matched against the name and IP address (respectively) of the nameservers of the zone the answer is in, and all of its parent zones. In its default configuration, BIND actively fetches any missing NS RRsets and address records. If, in the process of attempting to resolve the names of all of these delegated server names, BIND receives a SERVFAIL response for any of the queries, then it aborts the policy rule evaluation and returns SERVFAIL for the query. This is technically neither a match nor a non-match of the rule.

Every "." in a fully qualified domain name (FQDN) represents a potential delegation point. When BIND goes searching for parent zone NS RRsets (and, in the case of NSIP, their accompanying address records), it has to check every possible delegation point. This can become a problem for some specialized pseudo-domains, such as some domain name and network reputation systems, that have many "." characters in the names. It is further complicated if that system also has non-compliant DNS servers that silently drop queries for NS and SOA records. This forces BIND to wait for those queries to time out before it can finish evaluating the policy rule, even if this takes longer than a reasonable client typically waits for an answer (delays of over 60 seconds have been observed).

While both of these cases do involve configurations and/or servers that are technically "broken," they may still "work" outside of RPZ NSIP and NSDNAME rules because of redundancy and iteration optimizations.

There are two RPZ options, `nsip-wait-recurse` and `nsdname-wait-recurse`, that alter BIND's behavior by allowing it to use only those records that already exist in the cache when evaluating NSIP and NSDNAME rules, respectively.

Therefore NSDNAME and NSIP rules are unreliable. The rules may be matched against either the apex NS RRset or the "glue" NS RRset, each with their associated addresses (that also might or might not be "glue"). It's in the administrator's interests to discover both the delegation name server names and addresses, and the apex name server names and authoritative address records, to ensure correct use of NS and NSIP triggers in RPZ. Even then, there may be collateral damage to completely unrelated domains that otherwise "work," just by having NSIP and NSDNAME rules.

### 6.9.11 Example: Using RPZ to Disable Mozilla DoH-by-Default

Mozilla announced in September 2019 that they would enable DNS-over-HTTPS (DoH) for all US-based users of the Firefox browser, sending all their DNS queries to predefined DoH providers (Cloudflare's 1.1.1.1 service in particular). This is a concern for some network administrators who do not want their users' DNS queries to be rerouted unexpectedly. However, Mozilla provides a mechanism to disable the DoH-by-default setting: if the Mozilla-owned domain use-application-dns.net returns an NXDOMAIN response code, Firefox will not use DoH.

To accomplish this using RPZ:

1. Create a polizy zone file called `mozilla.rpz.db` configured so that NXDOMAIN will be returned for any query to `use-application-dns.net`:

```
$TTL  604800
$ORIGIN       mozilla.rpz.
@     IN      SOA     localhost. root.localhost. 1 604800 86400 2419200 604800
@     IN      NS      localhost.
use-application-dns.net CNAME .
```

2. Add the zone into the BIND configuration (usually *named.conf*):

```
zone mozilla.rpz {
    type primary;
    file "/<PATH_TO>/mozilla.rpz.db";
    allow-query { localhost; };
};
```

3. Enable use of the Response Policy Zone for all incoming queries by adding the *response-policy* directive into the `options {}` section:

```
options {
      response-policy { zone mozilla.rpz; } break-dnssec yes;
};
```

4. Reload the configuration and test whether the Response Policy Zone that was just added is in effect:

```
# rndc reload
# dig IN A use-application-dns.net @<IP_ADDRESS_OF_YOUR_RESOLVER>
# dig IN AAAA use-application-dns.net @<IP_ADDRESS_OF_YOUR_RESOLVER>
```

The response should return NXDOMAIN instead of the list of IP addresses, and the BIND 9 log should contain lines like this:

```
09-Sep-2019 18:50:49.439 client @0x7faf8e004a00 ::1#54175 (use-application-dns.net):␣
↪rpz QNAME NXDOMAIN rewrite use-application-dns.net/AAAA/IN via␣
↪use-application-dns.net.mozilla.rpz
09-Sep-2019 18:50:49.439 client @0x7faf8e007800 127.0.0.1#62915␣
↪(use-application-dns.net): rpz QNAME NXDOMAIN rewrite␣
↪use-application-dns.net/AAAA/IN via use-application-dns.net.mozilla.rpz
```

Note that this is the simplest possible configuration; specific configurations may be different, especially for administrators who are already using other response policy zones, or whose servers are configured with multiple views.

CHAPTER

SEVEN

# SECURITY CONFIGURATIONS

## 7.1 Security Assumptions

BIND 9's design assumes that access to the objects listed below is limited only to trusted parties. An incorrect deployment, which does not follow rules set by this section, cannot be the basis for CVE assignment or special security-sensitive handling of issues.

Unauthorized access can potentially disclose sensitive data, slow down server operation, etc. Unauthorized, unexpected, or incorrect writes to any of the following listed objects can potentially cause crashes, incorrect data handling, or corruption:

- All files stored on disk - including zone files, configuration files, key files, temporary files, etc.
- Clients communicating via the `controls` socket using configured keys
- Access to `statistics-channels` from untrusted clients
- Sockets used for `update-policy` type *external*

Problems that can only be triggered by configuring resources (views, zones, ...) with DNS classes other than Internet (IN), such as CHAOS (CH) or HESIOD (HS), cannot be the basis for CVE assignment or special security-sensitive handling of issues.

Certain aspects of the DNS protocol are left unspecified, such as the handling of responses from DNS servers which do not fully conform to the DNS protocol. For such a situation, BIND implements its own safety checks and limits which are subject to change as the protocol and deployment evolve.

### 7.1.1 Authoritative Servers

By default, zones use intentionally lenient limits (unlimited size, long transfer timeouts, etc.). These defaults can be misused by the source of data (zone transfers or UPDATEs) to exhaust resources on the receiving side.

The impact of malicious zone changes can be limited, to an extent, using configuration options listed in sections *Server Resource Limits* and *Zone Transfers*. Limits should also be applied to zones where malicious clients may potentially be authorized to use *Dynamic Update*.

### 7.1.2 DNS Resolvers

By definition, DNS resolvers act as traffic amplifiers; during normal operation, a DNS resolver can legitimately generate more outgoing traffic (counted in packets or bytes) than the incoming client traffic that triggered it. The DNS protocol specification does not currently specify limits for this amplification, but BIND implements its own limits to balance interoperability and safety. As a general rule, if a traffic amplification factor for any given scenario is lower than 100 packets, ISC does not handle the given scenario as a security issue. These limits are subject to change as DNS deployment evolves.

All DNS answers received by the DNS resolver are treated as untrusted input and are subject to safety and correctness checks. However, protocol non-conformity might cause unexpected behavior. If such unexpected behavior is limited to DNS domains hosted on non-conformant servers, it is not deemed a security issue *in BIND*.

## 7.2 Access Control Lists

Access Control Lists (ACLs) are address match lists that can be set up and nicknamed for future use in *allow-notify*, *allow-query*, *allow-query-on*, *allow-recursion*, *blackhole*, *allow-transfer*, *match-clients*, etc.

ACLs give users finer control over who can access the name server, without cluttering up configuration files with huge lists of IP addresses.

It is a *good idea* to use ACLs and to control access. Limiting access to the server by outside parties can help prevent spoofing and denial-of-service (DoS) attacks against the server.

ACLs match clients on the basis of up to three characteristics: 1) The client's IP address; 2) the TSIG or SIG(0) key that was used to sign the request, if any; and 3) an address prefix encoded in an EDNS Client-Subnet option, if any.

Here is an example of ACLs based on client addresses:

```
// Set up an ACL named "bogusnets" that blocks
// RFC1918 space and some reserved space, which is
// commonly used in spoofing attacks.
acl bogusnets {
    0.0.0.0/8;  192.0.2.0/24; 224.0.0.0/3;
    10.0.0.0/8; 172.16.0.0/12; 192.168.0.0/16;
};

// Set up an ACL called our-nets. Replace this with the
// real IP numbers.
acl our-nets { x.x.x.x/24; x.x.x.x/21; };
options {
  ...
  ...
  allow-query { our-nets; };
  allow-recursion { our-nets; };
  ...
  blackhole { bogusnets; };
  ...
};

zone "example.com" {
  type primary;
  file "m/example.com";
  allow-query { any; };
};
```

This allows authoritative queries for `example.com` from any address, but recursive queries only from the networks specified in `our-nets`, and no queries at all from the networks specified in `bogusnets`.

In addition to network addresses and prefixes, which are matched against the source address of the DNS request, ACLs may include `key` elements, which specify the name of a TSIG or SIG(0) key.

When BIND 9 is built with GeoIP support, ACLs can also be used for geographic access restrictions. This is done by specifying an ACL element of the form: `geoip db database field value`.

The `field` parameter indicates which field to search for a match. Available fields are `country`, `region`, `city`, `continent`, `postal` (postal code), `metro` (metro code), `area` (area code), `tz` (timezone), `isp`, `asnum`, and `domain`.

`value` is the value to search for within the database. A string may be quoted if it contains spaces or other special characters. An `asnum` search for autonomous system number can be specified using the string "ASNNNN" or the integer NNNN. If a `country` search is specified with a string that is two characters long, it must be a standard ISO-3166-1 two-letter country code; otherwise, it is interpreted as the full name of the country. Similarly, if `region` is the search term and the string is two characters long, it is treated as a standard two-letter state or province abbreviation; otherwise, it is treated as the full name of the state or province.

The *`database`* field indicates which GeoIP database to search for a match. In most cases this is unnecessary, because most search fields can only be found in a single database. However, searches for `continent` or `country` can be answered from either the `city` or `country` databases, so for these search types, specifying a *`database`* forces the query to be answered from that database and no other. If a *`database`* is not specified, these queries are first answered from the `city` database if it is installed, and then from the `country` database if it is installed. Valid database names are `country`, `city`, `asnum`, `isp`, and `domain`.

Some example GeoIP ACLs:

```
geoip country US;
geoip country JP;
geoip db country country Canada;
geoip region WA;
geoip city "San Francisco";
geoip region Oklahoma;
geoip postal 95062;
geoip tz "America/Los_Angeles";
geoip org "Internet Systems Consortium";
```

ACLs use a "first-match" logic rather than "best-match"; if an address prefix matches an ACL element, then that ACL is considered to have matched even if a later element would have matched more specifically. For example, the ACL `{ 10/8; !10.0.0.1; }` would actually match a query from 10.0.0.1, because the first element indicates that the query should be accepted, and the second element is ignored.

When using "nested" ACLs (that is, ACLs included or referenced within other ACLs), a negative match of a nested ACL tells the containing ACL to continue looking for matches. This enables complex ACLs to be constructed, in which multiple client characteristics can be checked at the same time. For example, to construct an ACL which allows a query only when it originates from a particular network *and* only when it is signed with a particular key, use:

```
allow-query { !{ !10/8; any; }; key example; };
```

Within the nested ACL, any address that is *not* in the 10/8 network prefix is rejected, which terminates the processing of the ACL. Any address that *is* in the 10/8 network prefix is accepted, but this causes a negative match of the nested ACL, so the containing ACL continues processing. The query is accepted if it is signed by the key `example`, and rejected otherwise. The ACL, then, only matches when *both* conditions are true.

## 7.3 `Chroot` and `Setuid`

On Unix servers, it is possible to run BIND in a *chrooted* environment (using the `chroot()` function) by specifying the *`-t`* option for *`named`*. This can help improve system security by placing BIND in a "sandbox," which limits the damage done if a server is compromised.

Another useful feature in the Unix version of BIND is the ability to run the daemon as an unprivileged user (*`-u`* user). We suggest running as an unprivileged user when using the `chroot` feature.

Here is an example command line to load BIND in a `chroot` sandbox, `/var/named`, and to run *`named`* `setuid` to user 202:

```
/usr/local/sbin/named -u 202 -t /var/named
```

### 7.3.1 The `chroot` Environment

For a `chroot` environment to work properly in a particular directory (for example, `/var/named`), the environment must include everything BIND needs to run. From BIND's point of view, `/var/named` is the root of the filesystem; the values of options like *`directory`* and *`pid-file`* must be adjusted to account for this.

Unlike with earlier versions of BIND, *`named`* does *not* typically need to be compiled statically, nor do shared libraries need to be installed under the new root. However, depending on the operating system, it may be necessary to set up locations such as `/dev/zero`, `/dev/random`, `/dev/log`, and `/etc/localtime`.

### 7.3.2 Using the `setuid` Function

Prior to running the *`named`* daemon, use the `touch` utility (to change file access and modification times) or the `chown` utility (to set the user id and/or group id) on files where BIND should write.

> **Note**
>
> If the *`named`* daemon is running as an unprivileged user, it cannot bind to new restricted ports if the server is reloaded.

## 7.4 Dynamic Update Security

Access to the dynamic update facility should be strictly limited. In earlier versions of BIND, the only way to do this was based on the IP address of the host requesting the update, by listing an IP address or network prefix in the *`allow-update`* zone option. This method is insecure, since the source address of the update UDP packet is easily forged. Also note that if the IP addresses allowed by the *`allow-update`* option include the address of a secondary server which performs forwarding of dynamic updates, the primary can be trivially attacked by sending the update to the secondary, which forwards it to the primary with its own source IP address - causing the primary to approve it without question.

For these reasons, we strongly recommend that updates be cryptographically authenticated by means of transaction signatures (TSIG). That is, the *`allow-update`* option should list only TSIG key names, not IP addresses or network prefixes. Alternatively, the *`update-policy`* option can be used.

Some sites choose to keep all dynamically updated DNS data in a subdomain and delegate that subdomain to a separate zone. This way, the top-level zone containing critical data, such as the IP addresses of public web and mail servers, need not allow dynamic updates at all.

## 7.5 TSIG

TSIG (Transaction SIGnatures) is a mechanism for authenticating DNS messages, originally specified in **RFC 2845**. It allows DNS messages to be cryptographically signed using a shared secret. TSIG can be used in any DNS transaction, as a way to restrict access to certain server functions (e.g., recursive queries) to authorized clients when IP-based access control is insufficient or needs to be overridden, or as a way to ensure message authenticity when it is critical to the integrity of the server, such as with dynamic UPDATE messages or zone transfers from a primary to a secondary server.

This section is a guide to setting up TSIG in BIND. It describes the configuration syntax and the process of creating TSIG keys.

*`named`* supports TSIG for server-to-server communication, and some of the tools included with BIND support it for sending messages to *`named`*:

- *nsupdate - dynamic DNS update utility* supports TSIG via the *`-k`*, *`-l`*, and *`-y`* command-line options, or via the `key` command when running interactively.

- *dig - DNS lookup utility* supports TSIG via the `-k` and `-y` command-line options.

### 7.5.1 Generating a Shared Key

TSIG keys can be generated using the `tsig-keygen` command; the output of the command is a `key` directive suitable for inclusion in `named.conf`. The key name, algorithm, and size can be specified by command-line parameters; the defaults are "tsig-key", HMAC-SHA256, and 256 bits, respectively.

Any string which is a valid DNS name can be used as a key name. For example, a key to be shared between servers called `host1` and `host2` could be called "host1-host2.", and this key can be generated using:

```
$ tsig-keygen host1-host2. > host1-host2.key
```

This key may then be copied to both hosts. The key name and secret must be identical on both hosts. (Note: copying a shared secret from one server to another is beyond the scope of the DNS. A secure transport mechanism should be used: secure FTP, SSL, ssh, telephone, encrypted email, etc.)

`tsig-keygen` can also be run as `ddns-confgen`, in which case its output includes additional configuration text for setting up dynamic DNS in `named`. See *ddns-confgen - TSIG key generation tool* for details.

### 7.5.2 Loading a New Key

For a key shared between servers called `host1` and `host2`, the following could be added to each server's `named.conf` file:

```
key "host1-host2." {
    algorithm hmac-sha256;
    secret "DAopyf1mhCbFVZw7pgmNPBoLUq8wEUT7UuPoLENP2HY=";
};
```

(This is the same key generated above using `tsig-keygen`.)

Since this text contains a secret, it is recommended that either `named.conf` not be world-readable, or that the `key` directive be stored in a file which is not world-readable and which is included in `named.conf` via the `include` directive.

Once a key has been added to `named.conf` and the server has been restarted or reconfigured, the server can recognize the key. If the server receives a message signed by the key, it is able to verify the signature. If the signature is valid, the response is signed using the same key.

### 7.5.3 Instructing the Server to Use a Key

A server sending a request to another server must be told whether to use a key, and if so, which key to use.

For example, a key may be specified for each server in the `primaries` statement in the definition of a secondary zone; in this case, all SOA QUERY messages, NOTIFY messages, and zone transfer requests (AXFR or IXFR) are signed using the specified key. Keys may also be specified in the `also-notify` statement of a primary or secondary zone, causing NOTIFY messages to be signed using the specified key.

Keys can also be specified in a `server` directive. Adding the following on `host1`, if the IP address of `host2` is 10.1.2.3, would cause *all* requests from `host1` to `host2`, including normal DNS queries, to be signed using the `host1-host2.` key:

```
server 10.1.2.3 {
    keys { host1-host2. ;};
};
```

Multiple keys may be present in the `keys` statement, but only the first one is used. As this directive does not contain secrets, it can be used in a world-readable file.

Requests sent by `host2` to `host1` would *not* be signed, unless a similar `server` directive were in `host2`'s configuration file.

When any server sends a TSIG-signed DNS request, it expects the response to be signed with the same key. If a response is not signed, or if the signature is not valid, the response is rejected.

### 7.5.4 TSIG-Based Access Control

TSIG keys may be specified in ACL definitions and ACL directives such as *`allow-query`*, *`allow-transfer`*, and *`allow-update`*. The above key would be denoted in an ACL element as `key host1-host2.`

Here is an example of an *`allow-update`* directive using a TSIG key:

```
allow-update { !{ !localnets; any; }; key host1-host2. ;};
```

This allows dynamic updates to succeed only if the UPDATE request comes from an address in `localnets`, *and* if it is signed using the `host1-host2.` key.

See *Dynamic Update Policies* for a discussion of the more flexible *`update-policy`* statement.

### 7.5.5 Errors

Processing of TSIG-signed messages can result in several errors:

- If a TSIG-aware server receives a message signed by an unknown key, the response will be unsigned, with the TSIG extended error code set to BADKEY.
- If a TSIG-aware server receives a message from a known key but with an invalid signature, the response will be unsigned, with the TSIG extended error code set to BADSIG.
- If a TSIG-aware server receives a message with a time outside of the allowed range, the response will be signed but the TSIG extended error code set to BADTIME, and the time values will be adjusted so that the response can be successfully verified.

In all of the above cases, the server returns a response code of NOTAUTH (not authenticated).

## 7.6 SIG(0)

BIND partially supports DNSSEC SIG(0) transaction signatures as specified in **RFC 2535** and **RFC 2931**. SIG(0) uses public/private keys to authenticate messages. Access control is performed in the same manner as with TSIG keys; privileges can be granted or denied in ACL directives based on the key name.

When a SIG(0) signed message is received, it is only verified if the key is known and trusted by the server. The server does not attempt to recursively fetch or validate the key.

SIG(0) signing of multiple-message TCP streams is not supported.

The only tool shipped with BIND 9 that generates SIG(0) signed messages is *`nsupdate`*.

CHAPTER

EIGHT

# CONFIGURATION REFERENCE

The operational functionality of BIND 9 is defined using the file **named.conf**, which is typically located in /etc or /usr/local/etc/namedb, depending on the operating system or distribution. A further file **rndc.conf** will be present if **rndc** is being run from a remote host, but is not required if rndc is being run from **localhost** (the same system as BIND 9 is running on).

## 8.1 Configuration File (named.conf)

The file `named.conf` may contain three types of entities:

**Comment**
*Multiple comment formats are supported*.

**Block**
*Blocks* are containers for *statements* which either have common functionality - for example, the definition of a cryptographic key in a *key* block - or which define the scope of the statement - for example, a statement which appears in a *zone* block has scope only for that zone.

Blocks are organized hierarchically within `named.conf` and may have a number of different properties:

- Certain blocks cannot be nested inside other blocks and thus may be regarded as the *topmost-level* blocks: for example, the *options* block and the *logging* block.
- Certain blocks can appear multiple times, in which case they have an associated name to disambiguate them: for example, the *zone* block (`zone example.com { ... };`) or the *key* block (`key mykey { ... };`).
- Certain blocks may be "nested" within other blocks. For example, the *zone* block may appear within a *view* block.

The description of each block in this manual lists its permissible locations.

**Statement**

- Statements define and control specific BIND behaviors.
- Statements may have a single parameter (a **Value**) or multiple parameters (**Argument/Value** pairs). For example, the *recursion* statement takes a single value parameter - in this case, the string `yes` or `no` (`recursion yes;`) - while the *port* statement takes a numeric value defining the DNS port number (`port 53;`). More complex statements take one or more argument/value pairs. The *also-notify* statement may take a number of such argument/value pairs, such as `also-notify port 5353;`, where `port` is the argument and `5353` is the corresponding value.
- Statements can appear in a single *block* - for example, an *algorithm* statement can appear only in a *key* block - or in multiple blocks - for example, an *also-notify* statement can appear in an *options* block where it has global (server-wide) scope, in a *zone* block where it has scope only for the specific zone (and

overrides any global statement), or even in a *view* block where it has scope for only that view (and overrides any global statement).

The file `named.conf` may further contain one or more instances of the *include* **Directive**. This directive is provided for administrative convenience in assembling a complete `named.conf` file and plays no subsequent role in BIND 9 operational characteristics or functionality.

**Note**

Over a period of many years the BIND ARM acquired a bewildering array of terminology. Many of the terms used described similar concepts and served only to add a layer of complexity, possibly confusion, and perhaps mystique to BIND 9 configuration. The ARM now uses only the terms **Block**, **Statement**, **Argument**, **Value**, and **Directive** to describe all entities used in BIND 9 configuration.

### 8.1.1 Comment Syntax

The BIND 9 comment syntax allows comments to appear anywhere that whitespace may appear in a BIND configuration file. To appeal to programmers of all kinds, they can be written in the C, C++, or shell/Perl style.

#### Syntax

```
/* This is a BIND comment as in C */
```

```
// This is a BIND comment as in C++
```

```
# This is a BIND comment as in common Unix shells
# and Perl
```

#### Definition and Usage

Comments can be inserted anywhere that whitespace may appear in a BIND configuration file.

C-style comments start with the two characters /* (slash, star) and end with */ (star, slash). Because they are completely delimited with these characters, they can be used to comment only a portion of a line or to span multiple lines.

C-style comments cannot be nested. For example, the following is not valid because the entire comment ends with the first */:

```
/* This is the start of a comment.
   This is still part of the comment.
/* This is an incorrect attempt at nesting a comment. */
   This is no longer in any comment. */
```

C++-style comments start with the two characters // (slash, slash) and continue to the end of the physical line. They cannot be continued across multiple physical lines; to have one logical comment span multiple lines, each line must use the // pair. For example:

```
// This is the start of a comment.  The next line
// is a new comment, even though it is logically
// part of the previous comment.
```

Shell-style (or Perl-style) comments start with the character # (number/pound sign) and continue to the end of the physical line, as in C++ comments. For example:

```
# This is the start of a comment.  The next line
# is a new comment, even though it is logically
# part of the previous comment.
```

**Warning**

The semicolon (`;`) character cannot start a comment, unlike in a zone file. The semicolon indicates the end of a configuration statement.

### 8.1.2 Configuration Layout Styles

BIND is very picky about opening and closing brackets/braces, semicolons, and all the other separators defined in the formal syntaxes in later sections. There are many layout styles that can assist in minimizing errors, as shown in the following examples:

```
// dense single-line style
zone "example.com" in{type secondary; file "secondary.example.com"; primaries
↪{10.0.0.1;};};
//  single-statement-per-line style
zone "example.com" in{
        type secondary;
        file "secondary.example.com";
        primaries {10.0.0.1;};
};
// spot the difference
zone "example.com" in{
        type secondary;
file "sec.secondary.com";
primaries {10.0.0.1;}; };
```

### 8.1.3 `include` Directive

```
include filename;
```

#### `include` Directive Definition and Usage

The include directive inserts the specified file (or files if a valid glob expression is detected) at the point where the include directive is encountered. The include directive facilitates the administration of configuration files by permitting the reading or writing of some things but not others. For example, the statement could include private keys that are readable only by the name server.

### 8.1.4 Address Match Lists

#### Syntax

An address match list is a list of semicolon-separated *address_match_element* s.

```
{ <address_match_element>; ... };
```

Each element is then defined as:

**`address_match_element`**

```
[ ! ] ( <ip_address> | <netprefix> | key <server_key> | <acl_name> | {␣
↪address_match_list } )
```

### Definition and Usage

Address match lists are primarily used to determine access control for various server operations. They are also used in the *listen-on* and *sortlist* statements. The elements which constitute an address match list can be any of the following:

- *ip_address*: an IP address (IPv4 or IPv6)
- *netprefix*: an IP prefix (in / notation)
- *server_key*: a key ID, as defined by the key statement
- *acl_name*: the name of an address match list defined with the *acl* statement
- a nested address match list enclosed in braces

Elements can be negated with a leading exclamation mark (!), and the match list names "any", "none", "localhost", and "localnets" are predefined. More information on those names can be found in the description of the *acl* statement.

The addition of the key clause made the name of this syntactic element something of a misnomer, since security keys can be used to validate access without regard to a host or network address. Nonetheless, the term "address match list" is still used throughout the documentation.

When a given IP address or prefix is compared to an address match list, the comparison takes place in approximately O(1) time. However, key comparisons require that the list of keys be traversed until a matching key is found, and therefore may be somewhat slower.

The interpretation of a match depends on whether the list is being used for access control, defining *listen-on* ports, or in a *sortlist*, and whether the element was negated.

When used as an access control list, a non-negated match allows access and a negated match denies access. If there is no match, access is denied. The clauses *allow-notify*, *allow-recursion*, *allow-recursion-on*, *allow-query*, *allow-query-on*, *allow-query-cache*, *allow-query-cache-on*, *allow-transfer*, *allow-update*, *allow-update-forwarding*, and *blackhole* all use address match lists. Similarly, the *listen-on* option causes the server to refuse queries on any of the machine's addresses which do not match the list.

Order of insertion is significant. If more than one element in an ACL is found to match a given IP address or prefix, preference is given to the one that came *first* in the ACL definition. Because of this first-match behavior, an element that defines a subset of another element in the list should come before the broader element, regardless of whether either is negated. For example, in `1.2.3/24; ! 1.2.3.13;` the 1.2.3.13 element is completely useless because the algorithm matches any lookup for 1.2.3.13 to the 1.2.3/24 element. Using `! 1.2.3.13; 1.2.3/24` fixes that problem by blocking 1.2.3.13 via the negation, but all other 1.2.3.* hosts pass through.

## 8.1.5 Glossary of Terms Used

Following is a list of terms used throughout the BIND configuration file documentation:

**acl_name**
: The name of an *address_match_list* as defined by the *acl* statement.

**address_match_list**
: See *Address Match Lists*.

**boolean**
: Either yes or no. The words true and false are also accepted, as are the numbers 1 and 0.

**domain_name**
: A quoted string which is used as a DNS name; for example: my.test.domain.

**duration**
A duration in BIND 9 can be written in three ways: as a single number representing seconds, as a string of numbers with TTL-style time-unit suffixes, or in ISO 6801 duration format.

Allowed TTL time-unit suffixes are: "W" (week), "D" (day), "H" (hour), "M" (minute), and "S" (second). Examples: "1W" (1 week), "3d12h" (3 days, 12 hours).

ISO 8601 duration format consists of the letter "P", followed by an optional series of numbers with unit suffixes "Y" (year), "M" (month), "W" (week), and "D" (day); this may optionally be followed by the letter "T", and another series of numbers with unit suffixes "H" (hour), "M" (minute), and "S" (second). Examples: "P3M10D" (3 months, 10 days), "P2WT12H" (2 weeks, 12 hours), "pt15m" (15 minutes). For more information on ISO 8601 duration format, see **RFC 3339**, appendix A.

Both TTL-style and ISO 8601 duration formats are case-insensitive.

**fixedpoint**
A non-negative real number that can be specified to the nearest one-hundredth. Up to five digits can be specified before a decimal point, and up to two digits after, so the maximum value is 99999.99. Acceptable values might be further limited by the contexts in which they are used.

**integer**
A non-negative 32-bit integer (i.e., a number between 0 and 4294967295, inclusive). Its acceptable value might be further limited by the context in which it is used.

**ip_address**
An *ipv4_address* or *ipv6_address*.

**ipv4_address**
An IPv4 address with exactly four integer elements valued 0 through 255 and separated by dots (`.`), such as `192.168.1.1` (a "dotted-decimal" notation with all four elements present).

**ipv6_address**
An IPv6 address, such as `2001:db8::1234`. IPv6-scoped addresses that have ambiguity on their scope zones must be disambiguated by an appropriate zone ID with the percent character (`%`) as a delimiter. It is strongly recommended to use string zone names rather than numeric identifiers, to be robust against system configuration changes. However, since there is no standard mapping for such names and identifier values, only interface names as link identifiers are supported, assuming one-to-one mapping between interfaces and links. For example, a link-local address `fe80::1` on the link attached to the interface `ne0` can be specified as `fe80::1%ne0`. Note that on most systems link-local addresses always have ambiguity and need to be disambiguated.

**netprefix**
An IP network specified as an *ip_address*, followed by a slash (`/`) and then the number of bits in the netmask. Trailing zeros in an *ip_address* may be omitted. For example, `127/8` is the network `127.0.0.0` with netmask `255.0.0.0` and `1.2.3.0/28` is network `1.2.3.0` with netmask `255.255.255.240`. When specifying a prefix involving an IPv6-scoped address, the scope may be omitted. In that case, the prefix matches packets from any scope.

**percentage**
An integer value followed by `%` to represent percent.

**port**
An IP port *integer*. It is limited to 0 through 65535, with values below 1024 typically restricted to use by processes running as root. In some cases, an asterisk (`*`) character can be used as a placeholder to select a random high-numbered port.

**portrange**
A list of a *port* or a port range. A port range is specified in the form of `range` followed by two *port* s, `port_low` and `port_high`, which represents port numbers from `port_low` through `port_high`, inclusive. `port_low` must not be larger than `port_high`. For example, `range 1024 65535` represents ports from 1024 through 65535. The asterisk (`*`) character is not allowed as a valid *port* or as a port range boundary.

**server-list**
A named list of one or more *ip_address* es with optional *tls_id*, *server_key*, and/or *port*. A `server-list` list may include other `server-list` lists.

**server_key**
A *domain_name* representing the name of a shared key, to be used for *transaction security*. Keys are defined using *key* blocks.

**size**
**sizeval**
A 64-bit unsigned integer. Integers may take values 0 <= value <= 18446744073709551615, though certain parameters (such as *max-journal-size*) may use a more limited range within these extremes. In most cases, setting a value to 0 does not literally mean zero; it means "undefined" or "as big as possible," depending on the context. See the explanations of particular parameters that use `size` for details on how they interpret its use. Numeric values can optionally be followed by a scaling factor: `K` or `k` for kilobytes, `M` or `m` for megabytes, and `G` or `g` for gigabytes, which scale by 1024, 1024*1024, and 1024*1024*1024 respectively.

Some statements also accept the keywords `unlimited` or `default`: `unlimited` generally means "as big as possible," and is usually the best way to safely set a very large number. `default` uses the limit that was in force when the server was started.

**tls_id**
A named TLS configuration object which defines a TLS key and certificate. See *tls* block.

## 8.2 Blocks

A BIND 9 configuration consists of blocks, statements, and comments.

The following blocks are supported:

***acl***
Defines a named IP address matching list, for access control and other uses.

***controls***
Declares control channels to be used by the *rndc* utility.

***dnssec-policy***
Describes a DNSSEC key and signing policy for zones. See *dnssec-policy* for details.

***key***
Specifies key information for use in authentication and authorization using TSIG.

***key-store***
Describes a DNSSEC key store. See *key-store Grammar* for details.

***logging***
Specifies what information the server logs and where the log messages are sent.

***options***
Controls global server configuration options and sets defaults for other statements.

***remote-servers***
Defines a named list of servers for inclusion in various zone statements such as *parental-agents*, *primaries* or *also-notify* lists.

***server***
Sets certain configuration options on a per-server basis.

***statistics-channels***
Declares communication channels to get access to *named* statistics.

***tls***
Specifies configuration information for a TLS connection, including a *key-file*, *cert-file*, *ca-file*, *dhparam-file*, *remote-hostname*, *ciphers*, *protocols*, *prefer-server-ciphers*, and *session-tickets*.

***http***
Specifies configuration information for an HTTP connection, including *endpoints*, *listener-clients*, and *streams-per-connection*.

***trust-anchors***
Defines DNSSEC trust anchors: if used with the `initial-key` or `initial-ds` keyword, trust anchors are kept up-to-date using **RFC 5011** trust anchor maintenance; if used with `static-key` or `static-ds`, keys are permanent.

***managed-keys***
Is identical to *trust-anchors*; this option is deprecated in favor of *trust-anchors* with the `initial-key` keyword, and may be removed in a future release.

***trusted-keys***
Defines permanent trusted DNSSEC keys; this option is deprecated in favor of *trust-anchors* with the `static-key` keyword, and may be removed in a future release.

***view***
Defines a view.

***zone***
Defines a zone.

The *logging* and *options* statements may only occur once per configuration.

### 8.2.1 *acl* Block Grammar

**acl**

**Grammar:** `acl <string> { <address_match_element>; ... }; // may occur multiple times`

**Blocks:** topmost

**Tags:** server

Assigns a symbolic name to an address match list.

### 8.2.2 *acl* Block Definition and Usage

The *acl* statement assigns a symbolic name to an address match list. It gets its name from one of the primary uses of address match lists: Access Control Lists (ACLs).

The following ACLs are built-in:

**any**
Matches all hosts.

**none**
Matches no hosts.

**localhost**
Matches the IPv4 and IPv6 addresses of all network interfaces on the system. When addresses are added or removed, the `localhost` ACL element is updated to reflect the changes.

**localnets**
Matches any host on an IPv4 or IPv6 network for which the system has an interface. When addresses are added or removed, the `localnets` ACL element is updated to reflect the changes. Some systems do not provide a way to determine the prefix lengths of local IPv6 addresses; in such cases, `localnets` only matches the local IPv6 addresses, just like `localhost`.

### 8.2.3 *controls* Block Grammar

**controls**

**Grammar:**

```
controls {
        inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ]␣
↪allow { <address_match_element>; ... } [ keys { <string>; ... } ] [ read-only
↪<boolean> ]; // may occur multiple times
        unix <quoted_string> perm <integer> owner <integer> group <integer> [␣
↪keys { <string>; ... } ] [ read-only <boolean> ]; // may occur multiple times
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** server

Specifies control channels to be used to manage the name server.

### 8.2.4 *controls* Block Definition and Usage

The *controls* statement declares control channels to be used by system administrators to manage the operation of the name server. These control channels are used by the *rndc* utility to send commands to and retrieve non-DNS results from a name server.

**unix**

**Grammar:** `unix <quoted_string> perm <integer> owner <integer> group <integer> [ keys { <string>; ... } ] [ read-only <boolean> ]; // may occur multiple times`

**Blocks:** controls

**Tags:** obsolete

Specifies a Unix domain socket as a control channel.

This option has been removed and using it will cause a fatal error.

**inet**

**Grammar controls:** `inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ] allow { <address_match_element>; ... } [ keys { <string>; ... } ] [ read-only <boolean> ]; // may occur multiple times`

**Grammar statistics-channels:** `inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ] [ allow { <address_match_element>; ... } ]; // may occur multiple times`

**Blocks:** controls, statistics-channels

**Tags:** server

Specifies a TCP socket as a control channel.

An *inet* control channel is a TCP socket listening at the specified *port* on the specified *ip_address*, which can be an IPv4 or IPv6 address. An *ip_address* of * (asterisk) is interpreted as the IPv4 wildcard address; connections are accepted on any of the system's IPv4 addresses. To listen on the IPv6 wildcard address, use an *ip_address* of ::. If *rndc* is only used on the local host, using the loopback address (127.0.0.1 or ::1) is recommended for maximum security.

If no port is specified, port 953 is used. The asterisk * cannot be used for *port*.

The ability to issue commands over the control channel is restricted by the allow and *keys* clauses.

**allow**
Connections to the control channel are permitted based on the *address_match_list*. This is for simple IP address-based filtering only; any *server_key* elements of the *address_match_list* are ignored.

***keys***
The primary authorization mechanism of the command channel is the list of *server_key* s. Each listed *key* is authorized to execute commands over the control channel. See *Administrative Tools* for information about configuring keys in *rndc*.

**read-only**
If the read-only argument is on, the control channel is limited to the following set of read-only commands: nta -dump, *null*, status, showzone, testgen, and zonestatus. By default, read-only is not enabled and the control channel allows read-write access.

If no *controls* statement is present, *named* sets up a default control channel listening on the loopback address 127.0.0.1 and its IPv6 counterpart, ::1. In this case, and also when the *controls* statement is present but does not have a *keys* clause, *named* attempts to load the command channel key from the file /etc/rndc.key. To create an rndc.key file, run *rndc-confgen -a*.

To disable the command channel, use an empty *controls* statement: controls { };.

### 8.2.5 key Block Grammar

**key**

**Grammar:**

```
key <string> {
        algorithm <string>;
        secret <string>;
}; // may occur multiple times
```

**Blocks:** topmost, view

**Tags:** security

Defines a shared secret key for use with *TSIG* or the command channel.

### 8.2.6 key Block Definition and Usage

The key statement defines a shared secret key for use with TSIG (see *TSIG*) or the command channel (see *controls*).

The key statement can occur at the top level of the configuration file or inside a *view* statement. Keys defined in top-level key statements can be used in all views. Keys intended for use in a *controls* statement must be defined at the top level.

The *server_key*, also known as the key name, is a domain name that uniquely identifies the key. It can be used in a *server* statement to cause requests sent to that server to be signed with this key, or in address match lists to verify that incoming requests have been signed with a key matching this name, algorithm, and secret.

**algorithm**

**Grammar:** `algorithm <string>;`

**Blocks:** key, view.key

**Tags:** security

Defines the algorithm to be used in a key clause.

The `algorithm_id` is a string that specifies a security/authentication algorithm. The *named* server supports `hmac-md5`, `hmac-sha1`, `hmac-sha224`, `hmac-sha256`, `hmac-sha384`, and `hmac-sha512` TSIG authentication. Truncated hashes are supported by appending the minimum number of required bits preceded by a dash, e.g., `hmac-sha1-80`.

**secret**

**Grammar:** `secret <string>;`

**Blocks:** key, view.key

**Tags:** security

Defines a Base64-encoded string to be used as the secret by the algorithm.

The `secret_string` is the secret to be used by the algorithm, and is treated as a Base64-encoded string.

### 8.2.7 *key-store* Block Grammar

**key-store**

**Grammar:**

```
key-store <string> {
        directory <string>;
        pkcs11-uri <quoted_string>;
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** dnssec

Configures a DNSSEC key store.

### 8.2.8 `key-store` Block Definition and Usage

The `key-store` statement defines how DNSSEC keys should be stored.

There is one built-in key store named `key-directory`. Configuring keys to use `key-store "key-directory"` is identical to using `key-directory`.

The following options can be specified in a *key-store* statement:

**pkcs11-uri**

**Grammar:** `pkcs11-uri <quoted_string>;`

**Blocks:** key-store

**Tags:** dnssec, pkcs11

The `uri` is a string that specifies a PKCS#11 URI Scheme (defined in **RFC 7512**). When set, *named* tries to create keys inside the corresponding PKCS#11 token. This requires BIND to be built with OpenSSL 3, and to have a PKCS#11 provider configured.

### 8.2.9 *logging* Block Grammar

**logging**

**Grammar:**

```
logging {
        category <string> { <string>; ... }; // may occur multiple times
        channel <string> {
                buffered <boolean>;
                file <quoted_string> [ versions ( unlimited | <integer> ) ] [␣
↪size <size> ] [ suffix ( increment | timestamp ) ];
                null;
                print-category <boolean>;
                print-severity <boolean>;
                print-time ( iso8601 | iso8601-utc | local | <boolean> );
                severity <log_severity>;
                stderr;
                syslog [ <syslog_facility> ];
        }; // may occur multiple times
};
```

**Blocks:** topmost

**Tags:** logging

Configures logging options for the name server.

### 8.2.10 *logging* Block Definition and Usage

The *logging* statement configures a wide variety of logging options for the name server. Its *channel* phrase associates output methods, format options, and severity levels with a name that can then be used with the *category* phrase to select how various classes of messages are logged.

Only one *logging* statement is used to define as many channels and categories as desired. If there is no *logging* statement, the logging configuration is:

```
logging {
     category default { default_syslog; default_debug; };
     category unmatched { null; };
};
```

If *named* is started with the *-L* option, it logs to the specified file at startup, instead of using syslog. In this case the logging configuration is:

```
logging {
     category default { default_logfile; default_debug; };
     category unmatched { null; };
};
```

The logging configuration is only established when the entire configuration file has been parsed. When the server starts up, all logging messages regarding syntax errors in the configuration file go to the default channels, or to standard error if the *-g* option was specified.

### The *channel* Phrase

**channel**

**Grammar:**

```
channel <string> {
        buffered <boolean>;
        file <quoted_string> [ versions ( unlimited | <integer> ) ] [ size <size>␣
↪] [ suffix ( increment | timestamp ) ];
        null;
        print-category <boolean>;
        print-severity <boolean>;
        print-time ( iso8601 | iso8601-utc | local | <boolean> );
        severity <log_severity>;
        stderr;
        syslog [ <syslog_facility> ];
}; // may occur multiple times
```

**Blocks:** logging

**Tags:** logging

Defines a stream of data that can be independently logged.

All log output goes to one or more `channels`; there is no limit to the number of channels that can be created.

Every channel definition must include a destination clause that says whether messages selected for the channel go to a file, go to a particular syslog facility, go to the standard error stream, or are discarded. The definition can optionally also limit the message severity level that is accepted by the channel (the default is `info`), and whether to include a *named*-generated time stamp, the category name, and/or the severity level (the default is not to include any).

**null**

**Grammar:** `null;`

**Blocks:** logging.channel

**Tags:** logging

Causes all messages sent to the logging channel to be discarded.

The *null* destination clause causes all messages sent to the channel to be discarded; in that case, other options for the channel are meaningless.

**file**

The `file` destination clause directs the channel to a disk file. It can include additional arguments to specify how

large the file is allowed to become before it is rolled to a backup file (`size`), how many backup versions of the file are saved each time this happens (`versions`), and the format to use for naming backup versions (`suffix`).

The `size` option is used to limit log file growth. If the file ever exceeds the specified size, then *named* stops writing to the file unless it has a `versions` option associated with it. If backup versions are kept, the files are rolled as described below. If there is no `versions` option, no more data is written to the log until some out-of-band mechanism removes or truncates the log to less than the maximum size. The default behavior is not to limit the size of the file.

File rolling only occurs when the file exceeds the size specified with the `size` option. No backup versions are kept by default; any existing log file is simply appended. The `versions` option specifies how many backup versions of the file should be kept. If set to `unlimited`, there is no limit.

The `suffix` option can be set to either `increment` or `timestamp`. If set to `timestamp`, then when a log file is rolled, it is saved with the current timestamp as a file suffix. If set to `increment`, then backup files are saved with incrementing numbers as suffixes; older files are renamed when rolling. For example, if `versions` is set to 3 and `suffix` to `increment`, then when `filename.log` reaches the size specified by `size`, `filename.log.1` is renamed to `filename.log.2`, `filename.log.0` is renamed to `filename.log.1`, and `filename.log` is renamed to `filename.log.0`, whereupon a new `filename.log` is opened.

Here is an example using the `size`, `versions`, and `suffix` options:

```
channel an_example_channel {
    file "example.log" versions 3 size 20m suffix increment;
    print-time yes;
    print-category yes;
};
```

**syslog**

**Grammar:** `syslog [ <syslog_facility> ];`

**Blocks:** logging.channel

**Tags:** logging

Directs the logging channel to the system log.

The *syslog* destination clause directs the channel to the system log. Its argument is a syslog facility as described in the *syslog* man page. Known facilities are `kern`, `user`, `mail`, `daemon`, `auth`, *syslog*, `lpr`, `news`, `uucp`, `cron`, `authpriv`, `ftp`, `local0`, `local1`, `local2`, `local3`, `local4`, `local5`, `local6`, and `local7`; however, not all facilities are supported on all operating systems. How *syslog* handles messages sent to this facility is described in the `syslog.conf` man page. On a system which uses a very old version of *syslog*, which only uses two arguments to the `openlog()` function, this clause is silently ignored.

**severity**

**Grammar:** `severity <log_severity>;`

**Blocks:** logging.channel

**Tags:** logging

Defines the priority level of log messages.

The *severity* clause works like *syslog*'s "priorities," except that they can also be used when writing straight to a file rather than using *syslog*. Messages which are not at least of the severity level given are not selected for the channel; messages of higher severity levels are accepted.

When using *syslog*, the `syslog.conf` priorities also determine what eventually passes through. For example, defining a channel facility and severity as `daemon` and `debug`, but only logging `daemon.warning` via `syslog.conf`, causes messages of severity `info` and `notice` to be dropped. If the situation were reversed, with *named* writing messages of only `warning` or higher, then `syslogd` would print all messages it received from the channel.

**stderr**

**Grammar:** `stderr;`

**Blocks:** logging.channel

**Tags:** logging

Directs the logging channel output to the server's standard error stream.

The *stderr* destination clause directs the channel to the server's standard error stream. This is intended for use when the server is running as a foreground process, as when debugging a configuration, for example.

The server can supply extensive debugging information when it is in debugging mode. If the server's global debug level is greater than zero, debugging mode is active. The global debug level is set either by starting the *named* server with the *-d* flag followed by a positive integer, or by running *rndc trace*. The global debug level can be set to zero, and debugging mode turned off, by running `rndc notrace`. All debugging messages in the server have a debug level; higher debug levels give more detailed output. Channels that indicate a specific debug severity get debugging output of level 3 or less any time the server is in debugging mode, regardless of the global debugging level:

```
channel specific_debug_level {
    file "foo";
    severity debug 3;
};
```

Channels with `dynamic` severity use the server's global debug level to determine which messages to print.

**print-time**

**Grammar:** `print-time ( iso8601 | iso8601-utc | local | <boolean> );`

**Blocks:** logging.channel

**Tags:** logging

Specifies the time format for log messages.

*print-time* can be set to `yes`, `no`, or a time format specifier, which may be one of `local`, `iso8601`, or `iso8601-utc`. If set to `no`, the date and time are not logged. If set to `yes` or `local`, the date and time are logged in a human-readable format, using the local time zone. If set to `iso8601`, the local time is logged in ISO 8601 format. If set to `iso8601-utc`, the date and time are logged in ISO 8601 format, with time zone set to UTC. The default is `no`.

*print-time* may be specified for a *syslog* channel, but it is usually pointless since *syslog* also logs the date and time.

**print-category**

**Grammar:** `print-category <boolean>;`

**Blocks:** logging.channel

**Tags:** logging

Includes the category in log messages.

If *print-category* is requested, then the category of the message is logged as well.

**print-severity**

**Grammar:** `print-severity <boolean>;`

**Blocks:** logging.channel

**Tags:** logging

Includes the severity in log messages.

If *print-severity* is on, then the severity level of the message is logged. The `print-` options may be used in any combination, and are always printed in the following order: time, category, severity.

Here is an example where all three `print-` options are on:

```
28-Feb-2000 15:05:32.863 general: notice: running
```

**buffered**

**Grammar:** `buffered <boolean>;`

**Blocks:** logging.channel

**Tags:** logging

Controls flushing of log messages.

If *buffered* has been turned on, the output to files is not flushed after each log entry. By default all log messages are flushed.

There are four predefined channels that are used for *named*'s default logging, as follows. If *named* is started with the *-L* option, then a fifth channel, `default_logfile`, is added. How they are used is described in *category*.

```
channel default_syslog {
    // send to syslog's daemon facility
    syslog daemon;
    // only send priority info and higher
    severity info;
};

channel default_debug {
    // write to named.run in the working directory
    // Note: stderr is used instead of "named.run" if
    // the server is started with the '-g' option.
    file "named.run";
    // log at the server's current debug level
    severity dynamic;
};

channel default_stderr {
    // writes to stderr
    stderr;
    // only send priority info and higher
    severity info;
};
```

(continues on next page)

(continued from previous page)

```
channel null {
   // toss anything sent to this channel
   null;
};

channel default_logfile {
    // this channel is only present if named is
    // started with the -L option, whose argument
    // provides the file name
    file "...";
    // log at the server's current debug level
    severity dynamic;
};
```

The `default_debug` channel has the special property that it only produces output when the server's debug level is non-zero. It normally writes to a file called `named.run` in the server's working directory.

For security reasons, when the *-u* command-line option is used, the `named.run` file is created only after *named* has changed to the new UID, and any debug output generated while *named* is starting - and still running as root - is discarded. To capture this output, run the server with the *-L* option to specify a default logfile, or the *-g* option to log to standard error which can be redirected to a file.

Once a channel is defined, it cannot be redefined. The built-in channels cannot be altered directly, but the default logging can be modified by pointing categories at defined channels.

### The *category* Phrase

There are many categories, so desired logs can be sent anywhere while unwanted logs are ignored. If a list of channels is not specified for a category, log messages in that category are sent to the `default` category instead. If no default category is specified, the following "default default" is used:

```
category default { default_syslog; default_debug; };
```

If *named* is started with the *-L* option, the default category is:

```
category default { default_logfile; default_debug; };
```

As an example, let's say a user wants to log security events to a file, but also wants to keep the default logging behavior. They would specify the following:

```
channel my_security_channel {
    file "my_security_file";
    severity info;
};
category security {
    my_security_channel;
    default_syslog;
    default_debug;
};
```

To discard all messages in a category, specify the *null* channel:

```
category xfer-out { null; };
category notify { null; };
```

**category**

**Grammar:** `category <string> { <string>; ... }; // may occur multiple times`

**Blocks:** logging

**Tags:** logging

Specifies the type of data logged to a particular channel.

The following are the available categories and brief descriptions of the types of log information they contain. More categories may be added in future BIND releases.

**client**
Processing of client requests.

**cname**
Name servers that are skipped for being a CNAME rather than A/AAAA records.

**config**
Configuration file parsing and processing.

**database**
Messages relating to the databases used internally by the name server to store zone and cache data.

**default**
Logging options for those categories where no specific configuration has been defined.

**dispatch**
Dispatching of incoming packets to the server modules where they are to be processed.

**dnssec**
DNSSEC and TSIG protocol processing.

**dnstap**
The *dnstap* DNS traffic capture system.

**edns-disabled**
Log queries that have been forced to use plain DNS due to timeouts. This is often due to the remote servers not being **RFC 1034**-compliant (not always returning FORMERR or similar to EDNS queries and other extensions to the DNS when they are not understood). In other words, this is targeted at servers that fail to respond to DNS queries that they don't understand.

Note: the log message can also be due to packet loss. Before reporting servers for non-**RFC 1034** compliance they should be re-tested to determine the nature of the non-compliance. This testing should prevent or reduce the number of false-positive reports.

Note: eventually *named* will have to stop treating such timeouts as due to **RFC 1034** non-compliance and start treating it as plain packet loss. Falsely classifying packet loss as due to **RFC 1034** non-compliance impacts DNSSEC validation, which requires EDNS for the DNSSEC records to be returned.

**general**
A catch-all for many things that still are not classified into categories.

**lame-servers**
Misconfigurations in remote servers, discovered by BIND 9 when trying to query those servers during resolution.

**network**
Network operations.

**notify**
The NOTIFY protocol.

**nsid**
NSID options received from upstream servers.

**queries**
The locations where queries should be logged.

At startup, specifying the category `queries` also enables query logging unless the *querylog* option has been specified.

The query log entry first reports a client object identifier in @0x<hexadecimal-number> format. Next, it reports the client's IP address and port number, and the query name, class, and type. Next, it reports whether the Recursion Desired flag was set (+ if set, - if not set), whether the query was signed (S), whether EDNS was in use along with the EDNS version number (E(#)), whether TCP was used (T), whether DO (DNSSEC Ok) was set (D), whether CD (Checking Disabled) was set (C), whether a valid DNS Server COOKIE was received (V), and whether a DNS COOKIE option without a valid Server COOKIE was present (K). After this, the destination address the query was sent to is reported. Finally, if any CLIENT-SUBNET option was present in the client query, it is included in square brackets in the format [ECS address/source/scope].

```
client @0x7f91b8005490 127.0.0.1#62536 (www.example.com): query: www.example.com␣
↪IN AAAA +E(0)K (127.0.0.1)
client @0x7f91b4007400 ::1#62537 (www.example.net): query: www.example.net IN␣
↪AAAA +E(0)K (::1)
```

The first part of this log message, showing the client address/port number and query name, is repeated in all subsequent log messages related to the same query.

**query-errors**
Information about queries that resulted in some failure.

**rate-limit**
Start, periodic, and final notices of the rate limiting of a stream of responses that are logged at `info` severity in this category. These messages include a hash value of the domain name of the response and the name itself, except when there is insufficient memory to record the name for the final notice. The final notice is normally delayed until about one minute after rate limiting stops. A lack of memory can hurry the final notice, which is indicated by an initial asterisk (*). Various internal events are logged at debug level 1 and higher.

Rate limiting of individual requests is logged in the `query-errors` category.

**resolver**
DNS resolution, such as the recursive lookups performed on behalf of clients by a caching name server.

**responses**
The locations where query response summaries should be logged.

**rpz**
Information about errors in response policy zone files, rewritten responses, and, at the highest `debug` levels, mere rewriting attempts.

**rpz-passthru**
Information about RPZ PASSTHRU policy activity. This category allows pre-approved policy activity to be logged into a dedicated channel.

**security**
Approval and denial of requests.

**serve-stale**
Indication of whether a stale answer is used following a resolver failure.

**spill**
Queries that have been terminated, either by dropping or responding with SERVFAIL, as a result of a fetchlimit quota being exceeded.

**sslkeylog**
TLS pre-master secrets (for debugging purposes).

**trust-anchor-telemetry**
*trust-anchor-telemetry* requests received by *named*.

**unmatched**
Messages that *named* was unable to determine the class of, or for which there was no matching *view*. A one-line summary is also logged to the `client` category. This category is best sent to a file or stderr; by default it is sent to the *null* channel.

**update**
Dynamic updates.

**update-security**
Approval and denial of update requests.

**xfer-in**
Zone transfers the server is receiving.

**xfer-out**
Zone transfers the server is sending.

**zoneload**
Loading of zones and creation of automatic empty zones.

### The `query-errors` Category

The `query-errors` category is used to indicate why and how specific queries resulted in responses which indicate an error. Normally, these messages are logged at `debug` logging levels; note, however, that if query logging is active, some are logged at `info`. The logging levels are described below:

At `debug` level 1 or higher - or at `info` when query logging is active - each response with the rcode of SERVFAIL is logged as follows:

```
client 127.0.0.1#61502: query failed (SERVFAIL) for www.example.com/IN/AAAA at query.
c:3880
```

This means an error resulting in SERVFAIL was detected at line 3880 of source file `query.c`. Log messages of this level are particularly helpful in identifying the cause of SERVFAIL for an authoritative server.

At `debug` level 2 or higher, detailed context information about recursive resolutions that resulted in SERVFAIL is logged. The log message looks like this:

```
fetch completed at resolver.c:2970 for www.example.com/A
in 10.000183: timed out/success [domain:example.com,
referral:2,restart:7,qrysent:8,timeout:5,lame:0,quota:0,neterr:0,
badresp:1,adberr:0,findfail:0,valfail:0]
```

The first part before the colon shows that a recursive resolution for AAAA records of www.example.com completed in 10.000183 seconds, and the final result that led to the SERVFAIL was determined at line 2970 of source file `resolver.c`.

The next part shows the detected final result and the latest result of DNSSEC validation. The latter is always "success" when no validation attempt was made. In this example, this query probably resulted in SERVFAIL because all name servers are down or unreachable, leading to a timeout in 10 seconds. DNSSEC validation was probably not attempted.

The last part, enclosed in square brackets, shows statistics collected for this particular resolution attempt. The `domain` field shows the deepest zone that the resolver reached; it is the zone where the error was finally detected. The meaning of the other fields is summarized in the following list.

**referral**
The number of referrals the resolver received throughout the resolution process. In the above `example.com` there are two.

**restart**
The number of cycles that the resolver tried remote servers at the `domain` zone. In each cycle, the resolver sends one query (possibly resending it, depending on the response) to each known name server of the `domain` zone.

**qrysent**
The number of queries the resolver sent at the `domain` zone.

**timeout**
The number of timeouts the resolver received since the last response.

**lame**
The number of lame servers the resolver detected at the `domain` zone. A server is detected to be lame either by an invalid response or as a result of lookup in BIND 9's address database (ADB), where lame servers are cached.

**quota**
The number of times the resolver was unable to send a query because it had exceeded the permissible fetch quota for a server.

**neterr**
The number of erroneous results that the resolver encountered in sending queries at the `domain` zone. One common case is when the remote server is unreachable and the resolver receives an "ICMP unreachable" error message.

**badresp**
The number of unexpected responses (other than `lame`) to queries sent by the resolver at the `domain` zone.

**adberr**
Failures in finding remote server addresses of the``domain`` zone in the ADB. One common case of this is that the remote server's name does not have any address records.

**findfail**
Failures to resolve remote server addresses. This is a total number of failures throughout the resolution process.

**valfail**
Failures of DNSSEC validation. Validation failures are counted throughout the resolution process (not limited to the `domain` zone), but should only happen in `domain`.

At `debug` level 3 or higher, the same messages as those at `debug` level 1 are logged for errors other than SERVFAIL. Note that negative responses such as NXDOMAIN are not errors, and are not logged at this debug level.

At `debug` level 4 or higher, the detailed context information logged at `debug` level 2 is logged for errors other than SERVFAIL and for negative responses such as NXDOMAIN.

### 8.2.11 `remote-servers` Block Grammar

**remote-servers**

**Grammar:** `remote-servers <string> [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }; // may occur multiple times`

**Blocks:** topmost

**Tags:** server

Defines a list of servers to be used by primary and secondary zones.

This specifies a list that allows for a common set of servers to be easily used by multiple zones. The following options may reference to a list of remote servers: *parental-agents*, *primaries*, and *also-notify*.

A "parental agent" is a trusted DNS server that is queried to check whether DS records for a given zones are up-to-date.

A "primary server" is where a secondary server can request zone transfers from.

To force the zone transfer requests to be sent over TLS, use *tls* keyword, e.g. `primaries { 192.0.2.1 tls tls-configuration-name; };`, where `tls-configuration-name` refers to a previously defined *tls statement*.

**Warning**

Please note that TLS connections to primaries are **not authenticated** unless *remote-hostname* or *ca-file* are specified within the *tls statement* in use (see information on *Strict TLS* and *Mutual TLS* for more details). **Not authenticated mode** (*Opportunistic TLS*) provides protection from passive observers but does not protect from man-in-the-middle attacks on zone transfers.

### 8.2.12 `options` Block Grammar

**`options`**

**Grammar:**

```
options {
        allow-new-zones <boolean>;
        allow-notify { <address_match_element>; ... };
        allow-proxy { <address_match_element>; ... }; // experimental
        allow-proxy-on { <address_match_element>; ... }; // experimental
        allow-query { <address_match_element>; ... };
        allow-query-cache { <address_match_element>; ... };
        allow-query-cache-on { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-recursion { <address_match_element>; ... };
        allow-recursion-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
        answer-cookie <boolean>;
        attach-cache <string>;
        auth-nxdomain <boolean>;
        automatic-interface-scan <boolean>;
        avoid-v4-udp-ports { <portrange>; ... }; // deprecated
        avoid-v6-udp-ports { <portrange>; ... }; // deprecated
        bindkeys-file <quoted_string>; // test only
        blackhole { <address_match_element>; ... };
        catalog-zones { zone <string> [ default-primaries [ port <integer> ] [␣
↪source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { (
↪<server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port
↪<integer> ] ) [ key <string> ] [ tls <string> ]; ... } ] [ zone-directory
```

(continues on next page)

(continued from previous page)

```
↪<quoted_string> ] [ in-memory <boolean> ] [ min-update-interval <duration> ]; ..
↪. };
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( primary | master | secondary | slave | response ) ( fail |␣
↪warn | ignore ); // may occur multiple times
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        clients-per-query <integer>;
        cookie-algorithm ( siphash24 );
        cookie-secret <string>; // may occur multiple times
        deny-answer-addresses { <address_match_element>; ... } [ except-from {
↪<string>; ... } ];
        deny-answer-aliases { <string>; ... } [ except-from { <string>; ... } ];
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        directory <quoted_string>;
        disable-algorithms <string> { <string>; ... }; // may occur multiple times
        disable-ds-digests <string> { <string>; ... }; // may occur multiple times
        disable-empty-zone <string>; // may occur multiple times
        dns64 <netprefix> {
                break-dnssec <boolean>;
                clients { <address_match_element>; ... };
                exclude { <address_match_element>; ... };
                mapped { <address_match_element>; ... };
                recursive-only <boolean>;
                suffix <ipv6_address>;
        }; // may occur multiple times
        dns64-contact <string>;
        dns64-server <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnsrps-enable <boolean>;
        dnsrps-library <quoted_string>;
        dnsrps-options { <unspecified-text> };
        dnssec-accept-expired <boolean>;
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-must-be-secure <string> <boolean>; // may occur multiple times,␣
↪deprecated
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        dnssec-validation ( yes | no | auto );
        dnstap { ( all | auth | client | forwarder | resolver | update ) [ (␣
↪query | response ) ]; ... };
        dnstap-identity ( <quoted_string> | none | hostname );
        dnstap-output ( file | unix ) <quoted_string> [ size ( unlimited | <size>␣
```

(continues on next page)

(continued from previous page)

```
↪) ] [ versions ( unlimited | <integer> ) ] [ suffix ( increment | timestamp ) ];
        dnstap-version ( <quoted_string> | none );
        dual-stack-servers [ port <integer> ] { ( <quoted_string> [ port
↪<integer> ] | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port
↪<integer> ] ); ... };
        dump-file <quoted_string>;
        edns-udp-size <integer>;
        empty-contact <string>;
        empty-server <string>;
        empty-zones-enable <boolean>;
        fetch-quota-params <integer> <fixedpoint> <fixedpoint> <fixedpoint>;
        fetches-per-server <integer> [ ( drop | fail ) ];
        fetches-per-zone <integer> [ ( drop | fail ) ];
        flush-zones-on-shutdown <boolean>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        fstrm-set-buffer-hint <integer>;
        fstrm-set-flush-timeout <integer>;
        fstrm-set-input-queue-size <integer>;
        fstrm-set-output-notify-threshold <integer>;
        fstrm-set-output-queue-model ( mpsc | spsc );
        fstrm-set-output-queue-size <integer>;
        fstrm-set-reopen-interval <duration>;
        geoip-directory ( <quoted_string> | none );
        heartbeat-interval <integer>; // deprecated
        hostname ( <quoted_string> | none );
        http-listener-clients <integer>;
        http-port <integer>;
        http-streams-per-connection <integer>;
        https-port <integer>;
        interface-interval <duration>;
        ipv4only-contact <string>;
        ipv4only-enable <boolean>;
        ipv4only-server <string>;
        ixfr-from-differences ( primary | master | secondary | slave | <boolean>␣
↪);
        keep-response-order { <address_match_element>; ... }; // obsolete
        key-directory <quoted_string>;
        lame-ttl <duration>;
        listen-on [ port <integer> ] [ proxy <string> ] [ tls <string> ] [ http
↪<string> ] { <address_match_element>; ... }; // may occur multiple times
        listen-on-v6 [ port <integer> ] [ proxy <string> ] [ tls <string> ] [␣
↪http <string> ] { <address_match_element>; ... }; // may occur multiple times
        lmdb-mapsize <sizeval>;
        managed-keys-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        match-mapped-addresses <boolean>;
        max-cache-size ( default | unlimited | <sizeval> | <percentage> );
        max-cache-ttl <duration>;
        max-clients-per-query <integer>;
```

(continues on next page)

(continued from previous page)

```
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-ncache-ttl <duration>;
        max-query-count <integer>;
        max-query-restarts <integer>;
        max-records <integer>;
        max-records-per-type <integer>;
        max-recursion-depth <integer>;
        max-recursion-queries <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-rsa-exponent-size <integer>;
        max-stale-ttl <duration>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-udp-size <integer>;
        max-validation-failures-per-fetch <integer>; // experimental
        max-validations-per-fetch <integer>; // experimental
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        memstatistics <boolean>;
        memstatistics-file <quoted_string>;
        message-compression <boolean>;
        min-cache-ttl <duration>;
        min-ncache-ttl <duration>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        minimal-any <boolean>;
        minimal-responses ( no-auth | no-auth-recursive | <boolean> );
        multi-master <boolean>;
        new-zones-directory <quoted_string>;
        no-case-compress { <address_match_element>; ... };
        nocookie-udp-size <integer>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-rate <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        nta-lifetime <duration>;
        nta-recheck <duration>;
        nxdomain-redirect <string>;
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        pid-file ( <quoted_string> | none );
        port <integer>;
        preferred-glue <string>;
```

(continues on next page)

(continued from previous page)

```
        prefetch <integer> [ <integer> ];
        provide-ixfr <boolean>;
        qname-minimization ( strict | relaxed | disabled | off );
        query-source [ address ] ( <ipv4_address> | * | none );
        query-source-v6 [ address ] ( <ipv6_address> | * | none );
        querylog <boolean>;
        rate-limit {
                all-per-second <integer>;
                errors-per-second <integer>;
                exempt-clients { <address_match_element>; ... };
                ipv4-prefix-length <integer>;
                ipv6-prefix-length <integer>;
                log-only <boolean>;
                max-table-size <integer>;
                min-table-size <integer>;
                nodata-per-second <integer>;
                nxdomains-per-second <integer>;
                qps-scale <integer>;
                referrals-per-second <integer>;
                responses-per-second <integer>;
                slip <integer>;
                window <integer>;
        };
        recursing-file <quoted_string>;
        recursion <boolean>;
        recursive-clients <integer>;
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-server-cookie <boolean>;
        resolver-query-timeout <integer>;
        resolver-use-dns64 <boolean>;
        response-padding { <address_match_element>; ... } block-size <integer>;
        response-policy { zone <string> [ add-soa <boolean> ] [ log <boolean> ] [␣
↪max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ policy ( cname␣
↪| disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only
↪<quoted_string> ) ] [ recursive-only <boolean> ] [ nsip-enable <boolean> ] [␣
↪nsdname-enable <boolean> ] [ ede <string> ]; ... } [ add-soa <boolean> ] [␣
↪break-dnssec <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval
↪<duration> ] [ min-ns-dots <integer> ] [ nsip-wait-recurse <boolean> ] [␣
↪nsdname-wait-recurse <boolean> ] [ qname-wait-recurse <boolean> ] [␣
↪recursive-only <boolean> ] [ servfail-until-ready <boolean> ] [ nsip-enable
↪<boolean> ] [ nsdname-enable <boolean> ] [ dnsrps-enable <boolean> ] [␣
↪dnsrps-options { <unspecified-text> } ];
        responselog <boolean>;
        reuseport <boolean>;
        root-key-sentinel <boolean>;
        rrset-order { [ class <string> ] [ type <string> ] [ name <quoted_string>␣
↪] <string> <string>; ... };
        secroots-file <quoted_string>;
        send-cookie <boolean>;
        serial-query-rate <integer>;
```

(continues on next page)

(continued from previous page)

```
        serial-update-method ( date | increment | unixtime );
        server-id ( <quoted_string> | none | hostname );
        servfail-ttl <duration>;
        session-keyalg <string>;
        session-keyfile ( <quoted_string> | none );
        session-keyname <string>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        sig0checks-quota <integer>; // experimental
        sig0checks-quota-exempt { <address_match_element>; ... }; // experimental
        sig0key-checks-limit <integer>;
        sig0message-checks-limit <integer>;
        sortlist { <address_match_element>; ... }; // deprecated
        stale-answer-client-timeout ( disabled | off | <integer> );
        stale-answer-enable <boolean>;
        stale-answer-ttl <duration>;
        stale-cache-enable <boolean>;
        stale-refresh-time <duration>;
        startup-notify-rate <integer>;
        statistics-file <quoted_string>;
        synth-from-dnssec <boolean>;
        tcp-advertised-timeout <integer>;
        tcp-clients <integer>;
        tcp-idle-timeout <integer>;
        tcp-initial-timeout <integer>;
        tcp-keepalive-timeout <integer>;
        tcp-listen-queue <integer>;
        tcp-receive-buffer <integer>;
        tcp-reuse-timeout <integer>;
        tcp-send-buffer <integer>;
        tkey-domain <quoted_string>; // obsolete
        tkey-gssapi-credential <quoted_string>; // deprecated
        tkey-gssapi-keytab <quoted_string>;
        tls-port <integer>;
        transfer-format ( many-answers | one-answer );
        transfer-message-size <integer>;
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        transfers-in <integer>;
        transfers-out <integer>;
        transfers-per-ns <integer>;
        trust-anchor-telemetry <boolean>;
        try-tcp-refresh <boolean>;
        udp-receive-buffer <integer>;
        udp-send-buffer <integer>;
        update-check-ksk <boolean>; // obsolete
        update-quota <integer>;
        use-v4-udp-ports { <portrange>; ... }; // deprecated
        use-v6-udp-ports { <portrange>; ... }; // deprecated
        v6-bias <integer>;
```

(continues on next page)

(continued from previous page)

```
        validate-except { <string>; ... };
        version ( <quoted_string> | none );
        zero-no-soa-ttl <boolean>;
        zero-no-soa-ttl-cache <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** topmost

**Tags:** server

Defines global options to be used by BIND 9.

This is the grammar of the `options` statement in the *`named.conf`* file:

### 8.2.13 `options` Block Definition and Usage

The `options` statement sets up global options to be used by BIND. This statement may appear only once in a configuration file. If there is no `options` statement, an options block with each option set to its default is used.

**`attach-cache`**

**Grammar:** `attach-cache <string>;`

**Blocks:** options, view

**Tags:** view

Allows multiple views to share a single cache database.

This option allows multiple views to share a single cache database. Each view has its own cache database by default, but if multiple views have the same operational policy for name resolution and caching, those views can share a single cache to save memory, and possibly improve resolution efficiency, by using this option.

The *`attach-cache`* option may also be specified in *`view`* statements, in which case it overrides the global *`attach-cache`* option.

The `cache_name` specifies the cache to be shared. When the *`named`* server configures views which are supposed to share a cache, it creates a cache with the specified name for the first view of these sharing views. The rest of the views simply refer to the already-created cache.

One common configuration to share a cache is to allow all views to share a single cache. This can be done by specifying *`attach-cache`* as a global option with an arbitrary name.

Another possible operation is to allow a subset of all views to share a cache while the others retain their own caches. For example, if there are three views A, B, and C, and only A and B should share a cache, specify the *`attach-cache`* option as a view of A (or B)'s option, referring to the other view name:

```
view "A" {
  // this view has its own cache
  ...
};
view "B" {
  // this view refers to A's cache
  attach-cache "A";
};
```

(continues on next page)

(continued from previous page)

```
view "C" {
  // this view has its own cache
  ...
};
```

Views that share a cache must have the same policy on configurable parameters that may affect caching. The current implementation requires the following configurable options be consistent among these views: *check-names*, *dnssec-accept-expired*, *dnssec-validation*, *max-cache-ttl*, *max-ncache-ttl*, *max-stale-ttl*, *max-cache-size*, *min-cache-ttl*, *min-ncache-ttl*, and *zero-no-soa-ttl*.

Note that there may be other parameters that may cause confusion if they are inconsistent for different views that share a single cache. For example, if these views define different sets of forwarders that can return different answers for the same question, sharing the answer does not make sense or could even be harmful. It is the administrator's responsibility to ensure that configuration differences in different views do not cause disruption with a shared cache.

**directory**

**Grammar key-store:** `directory <string>;`

**Grammar options:** `directory <quoted_string>;`

**Blocks:** key-store, options

**Tags:** server

Sets the server's working directory.

This sets the working directory of the server. Any non-absolute pathnames in the configuration file are taken as relative to this directory. The default location for most server output files (e.g., `named.run`) is this directory. If a directory is not specified, the working directory defaults to `"."`, the directory from which the server was started. The directory specified should be an absolute path, and *must* be writable by the effective user ID of the *named* process.

The option takes effect only at the time that the configuration option is parsed; if other files are being included before or after specifying the new *directory*, the *directory* option must be listed before any other directive (like `include`) that can work with relative files. The safest way to include files is to use absolute file names.

**dnstap**

**Grammar:** `dnstap { ( all | auth | client | forwarder | resolver | update ) [ ( query | response ) ]; ... };`

**Blocks:** options, view

**Tags:** logging

Enables logging of *dnstap* messages.

*dnstap* is a fast, flexible method for capturing and logging DNS traffic. Developed by Robert Edmonds at Farsight Security, Inc., and supported by multiple DNS implementations, *dnstap* uses `libfstrm` (a lightweight high-speed framing library; see https://github.com/farsightsec/fstrm) to send event payloads which are encoded using Protocol Buffers (`libprotobuf-c`, a mechanism for serializing structured data developed by Google, Inc.; see https://protobuf.dev).

To enable *dnstap* at compile time, the `fstrm` and `protobuf-c` libraries must be available, and BIND must be configured with `--enable-dnstap`.

The *dnstap* option is a bracketed list of message types to be logged. These may be set differently for each view. Supported types are `client`, `auth`, `resolver`, `forwarder`, and `update`. Specifying type `all` causes all *dnstap* messages to be logged, regardless of type.

Each type may take an additional argument to indicate whether to log `query` messages or `response` messages; if not specified, both queries and responses are logged.

Example: To log all authoritative queries and responses, recursive client responses, and upstream queries sent by the resolver, use:

```
dnstap {
  auth;
  client response;
  resolver query;
};
```

**Note**

In the default configuration, the dnstap output for recursive resolver traffic does not include the IP addresses used by server-side sockets. This is caused by the fact that unless the *query source address* is explicitly set, these sockets are bound to wildcard IP addresses and determining the specific IP address used by each of them requires issuing a system call (i.e. incurring a performance penalty).

Logged *dnstap* messages can be parsed using the *dnstap-read* utility (see *dnstap-read - print dnstap data in human-readable form* for details).

For more information on *dnstap*, see https://dnstap.info.

The `fstrm` library has a number of tunables that are exposed in *named.conf*, and can be modified if necessary to improve performance or prevent loss of data. These are:

**fstrm-set-buffer-hint**

**Grammar:** `fstrm-set-buffer-hint <integer>;`

**Blocks:** options

**Tags:** logging

Sets the number of accumulated bytes in the output buffer before forcing a buffer flush.

The indicates the threshold number of bytes to accumulate in the output buffer before forcing a buffer flush. The minimum is 1024, the maximum is 65536, and the default is 8192.

**fstrm-set-flush-timeout**

**Grammar:** `fstrm-set-flush-timeout <integer>;`

**Blocks:** options

**Tags:** logging

Sets the number of seconds that unflushed data remains in the output buffer.

This is the number of seconds to allow unflushed data to remain in the output buffer. The minimum is 1 second, the maximum is 600 seconds (10 minutes), and the default is 1 second.

**fstrm-set-output-notify-threshold**

**Grammar:** `fstrm-set-output-notify-threshold <integer>;`

**Blocks:** options

**Tags:** logging

Sets the number of outstanding queue entries allowed on an input queue before waking the I/O thread.

This indicates the number of outstanding queue entries to allow on an input queue before waking the I/O thread. The minimum is 1 and the default is 32.

**fstrm-set-output-queue-model**

**Grammar:** `fstrm-set-output-queue-model ( mpsc | spsc );`

**Blocks:** options

**Tags:** logging

Sets the queuing semantics to use for queue objects.

This sets the queuing semantics to use for queue objects. The default is `mpsc` (multiple producer, single consumer); the other option is `spsc` (single producer, single consumer).

**fstrm-set-input-queue-size**

**Grammar:** `fstrm-set-input-queue-size <integer>;`

**Blocks:** options

**Tags:** logging

Sets the number of queue entries to allocate for each input queue.

This is the number of queue entries to allocate for each input queue. This value must be a power of 2. The minimum is 2, the maximum is 16384, and the default is 512.

**fstrm-set-output-queue-size**

**Grammar:** `fstrm-set-output-queue-size <integer>;`

**Blocks:** options

**Tags:** logging

Sets the number of queue entries allocated for each output queue.

This specifies the number of queue entries to allocate for each output queue. The minimum is 2, the maximum is system-dependent and based on `IOV_MAX`, and the default is 64.

**fstrm-set-reopen-interval**

**Grammar:** `fstrm-set-reopen-interval <duration>;`

**Blocks:** options

**Tags:** logging

Sets the number of seconds to wait between attempts to reopen a closed output stream.

This sets the number of seconds to wait between attempts to reopen a closed output stream. The minimum is 1 second, the maximum is 600 seconds (10 minutes), and the default is 5 seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value.

Note that all of the above minimum, maximum, and default values are set by the `libfstrm` library, and may be subject to change in future versions of the library. See the `libfstrm` documentation for more information.

**dnstap-output**

**Grammar:** `dnstap-output ( file | unix ) <quoted_string> [ size ( unlimited | <size> ) ] [ versions ( unlimited | <integer> ) ] [ suffix ( increment | timestamp ) ];`

**Blocks:** options

**Tags:** logging

Configures the path to which the *dnstap* frame stream is sent.

This configures the path to which the *dnstap* frame stream is sent if *dnstap* is enabled at compile time and active.

The first argument is either `file` or `unix`, indicating whether the destination is a file or a Unix domain socket. The second argument is the path of the file or socket. (Note: when using a socket, *dnstap* messages are only sent if another process such as `fstrm_capture` (provided with `libfstrm`) is listening on the socket.)

If the first argument is `file`, then up to three additional options can be added: `size` indicates the size to which a *dnstap* log file can grow before being rolled to a new file; `versions` specifies the number of rolled log files to retain; and `suffix` indicates whether to retain rolled log files with an incrementing counter as the suffix (`increment`) or with the current timestamp (`timestamp`). These are similar to the `size`, `versions`, and `suffix` options in a *logging* channel. The default is to allow *dnstap* log files to grow to any size without rolling.

*dnstap-output* can only be set globally in *options*. Currently, it can only be set once while *named* is running; once set, it cannot be changed by *rndc reload* or *rndc reconfig*.

**dnstap-identity**

**Grammar:** `dnstap-identity ( <quoted_string> | none | hostname );`

**Blocks:** options

**Tags:** logging

Specifies an `identity` string to send in *dnstap* messages.

This specifies an `identity` string to send in *dnstap* messages. If set to *hostname*, which is the default, the server's hostname is sent. If set to `none`, no identity string is sent.

**dnstap-version**

**Grammar:** `dnstap-version ( <quoted_string> | none );`

**Blocks:** options

**Tags:** logging

Specifies a *version* string to send in *dnstap* messages.

This specifies a *version* string to send in *dnstap* messages. The default is the version number of the BIND release. If set to `none`, no version string is sent.

**geoip-directory**

**Grammar:** `geoip-directory ( <quoted_string> | none );`

**Blocks:** options

**Tags:** server

Specifies the directory containing GeoIP database files.

When *named* is compiled using the MaxMind GeoIP2 geolocation API, this specifies the directory containing GeoIP database files. By default, the option is set based on the prefix used to build the `libmaxminddb` module; for example, if the library is installed in `/usr/local/lib`, then the default *geoip-directory* is `/usr/local/share/GeoIP`. See *acl* for details about `geoip` ACLs.

**key-directory**

**Grammar:** `key-directory <quoted_string>;`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Indicates the directory where public and private DNSSEC key files are found.

This is the directory where the public and private DNSSEC key files should be found when performing a dynamic update of secure zones, if different than the current working directory. (Note that this option has no effect on the paths for files containing non-DNSSEC keys, such as `rndc.key`, or `session.key`.)

**lmdb-mapsize**

**Grammar:** `lmdb-mapsize <sizeval>;`

**Blocks:** options, view

**Tags:** server

Sets a maximum size for the memory map of the new-zone database in LMDB database format.

When *named* is built with liblmdb, this option sets a maximum size for the memory map of the new-zone database (NZD) in LMDB database format. This database is used to store configuration information for zones added using *rndc addzone*. Note that this is not the NZD database file size, but the largest size that the database may grow to.

Because the database file is memory-mapped, its size is limited by the address space of the *named* process. The default of 32 megabytes was chosen to be usable with 32-bit *named* builds. The largest permitted value is 1 terabyte. Given typical zone configurations without elaborate ACLs, a 32 MB NZD file ought to be able to hold configurations of about 100,000 zones.

**managed-keys-directory**

**Grammar:** `managed-keys-directory <quoted_string>;`

**Blocks:** options

**Tags:** dnssec

Specifies the directory in which to store the files that track managed DNSSEC keys.

This specifies the directory in which to store the files that track managed DNSSEC keys (i.e., those configured using the `initial-key` or `initial-ds` keywords in a *`trust-anchors`* statement). By default, this is the working directory. The directory *must* be writable by the effective user ID of the *`named`* process.

If *`named`* is not configured to use views, managed keys for the server are tracked in a single file called `managed-keys.bind`. Otherwise, managed keys are tracked in separate files, one file per view; each file name is the view name (or, if it contains characters that are incompatible with use as a file name, the SHA256 hash of the view name), followed by the extension `.mkeys`.

(Note: in earlier releases, file names for views always used the SHA256 hash of the view name. To ensure compatibility after upgrading, if a file using the old name format is found to exist, it is used instead of the new format.)

**`max-ixfr-ratio`**

**Grammar:** `max-ixfr-ratio ( unlimited | <percentage> );`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Sets the maximum size for IXFR responses to zone transfer requests.

This sets the size threshold (expressed as a percentage of the size of the full zone) beyond which *`named`* chooses to use an AXFR response rather than IXFR when answering zone transfer requests. See *Incremental Zone Transfers (IXFR)*.

The minimum value is `1%`. The keyword `unlimited` disables ratio checking and allows IXFRs of any size. The default is `100%`.

**`new-zones-directory`**

**Grammar:** `new-zones-directory <quoted_string>;`

**Blocks:** options, view

**Tags:** zone

Specifies the directory where configuration parameters are stored for zones added by *`rndc addzone`*.

This specifies the directory in which to store the configuration parameters for zones added via *`rndc addzone`*. By default, this is the working directory. If set to a relative path, it is relative to the working directory. The directory *must* be writable by the effective user ID of the *`named`* process.

**`qname-minimization`**

**Grammar:** `qname-minimization ( strict | relaxed | disabled | off );`

**Blocks:** options, view

**Tags:** query

Controls QNAME minimization behavior in the BIND 9 resolver.

When this is set to `strict`, BIND follows the QNAME minimization algorithm to the letter, as specified in **RFC 7816**.

Setting this option to `relaxed` causes BIND to fall back to normal (non-minimized) query mode when it receives either NXDOMAIN or other unexpected responses (e.g., SERVFAIL, improper zone cut, REFUSED) to a minimized query.

In `relaxed` mode *`named`* makes NS queries for `<domain>` as it walks down the tree.

`disabled` disables QNAME minimization completely. `off` is a synonym for `disabled`.

The current default is `relaxed`, but it may be changed to `strict` in a future release.

**tkey-gssapi-keytab**

**Grammar:** `tkey-gssapi-keytab <quoted_string>;`

**Blocks:** options

**Tags:** security

Sets the KRB5 keytab file to use for GSS-TSIG updates.

This is the KRB5 keytab file to use for GSS-TSIG updates. If this option is set and `tkey-gssapi-credential` is not set, updates are allowed with any key matching a principal in the specified keytab.

**tkey-gssapi-credential**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `tkey-gssapi-credential <quoted_string>; // deprecated`

**Blocks:** options

**Tags:** security

Sets the security credential for authentication keys requested by the GSS-TSIG protocol.

This is the security credential with which the server should authenticate keys requested by the GSS-TSIG protocol. Currently only Kerberos 5 authentication is available; the credential is a Kerberos principal which the server can acquire through the default system key file, normally `/etc/krb5.keytab`. The location of the keytab file can be overridden using the *`tkey-gssapi-keytab`* option. Normally this principal is of the form `DNS/server.domain`.

**dump-file**

**Grammar:** `dump-file <quoted_string>;`

**Blocks:** options

**Tags:** logging

Indicates the pathname of the file where the server dumps the database after *`rndc dumpdb`*.

This is the pathname of the file the server dumps the database to, when instructed to do so with *`rndc dumpdb`*. If not specified, the default is `named_dump.db`.

**memstatistics-file**

**Grammar:** `memstatistics-file <quoted_string>;`

**Blocks:** options

**Tags:** logging

Sets the pathname of the file where the server writes memory usage statistics on exit.

This is the pathname of the file the server writes memory usage statistics to on exit. If not specified, the default is `named.memstats`.

**pid-file**

**Grammar:** `pid-file ( <quoted_string> | none );`

**Blocks:** options

**Tags:** server

Specifies the pathname of the file where the server writes its process ID.

This is the pathname of the file the server writes its process ID in. If not specified, the default is `/run/named.pid`. The PID file is used by programs that send signals to the running name server. Specifying `pid-file none` disables the use of a PID file; no file is written and any existing one is removed. Note that `none` is a keyword, not a filename, and therefore is not enclosed in double quotes.

**recursing-file**

**Grammar:** `recursing-file <quoted_string>;`

**Blocks:** options

**Tags:** server

Specifies the pathname of the file where the server dumps queries that are currently recursing via *rndc recursing*.

This is the pathname of the file where the server dumps the queries that are currently recursing, when instructed to do so with *rndc recursing*. If not specified, the default is `named.recursing`.

**statistics-file**

**Grammar:** `statistics-file <quoted_string>;`

**Blocks:** options

**Tags:** logging, server

Specifies the pathname of the file where the server appends statistics, when using *rndc stats*.

This is the pathname of the file the server appends statistics to, when instructed to do so using *rndc stats*. If not specified, the default is `named.stats` in the server's current directory. The format of the file is described in *The Statistics File*.

**bindkeys-file**

**Grammar:** `bindkeys-file <quoted_string>; // test only`

**Blocks:** options

**Tags:** dnssec

Specifies the pathname of a file to override the built-in trusted keys provided by *named*.

This is the pathname of a file to override the built-in trusted keys provided by *named*. See the discussion of *dnssec-validation* for details. This is intended for server testing.

**secroots-file**

**Grammar:** `secroots-file <quoted_string>;`

**Blocks:** options

**Tags:** dnssec

Specifies the pathname of the file where the server dumps security roots, when using *rndc secroots*.

This is the pathname of the file the server dumps security roots to, when instructed to do so with *rndc secroots*. If not specified, the default is `named.secroots`.

**session-keyfile**

**Grammar:** `session-keyfile ( <quoted_string> | none );`

**Blocks:** options

**Tags:** security

Specifies the pathname of the file where a TSIG session key is written, when generated by *named* for use by `nsupdate -l`.

This is the pathname of the file into which to write a TSIG session key generated by *named* for use by `nsupdate -l`. If not specified, the default is `/run/session.key`. (See *Dynamic Update Policies*, and in particular the discussion of the *update-policy* statement's `local` option, for more information about this feature.)

**session-keyname**

**Grammar:** `session-keyname <string>;`

**Blocks:** options

**Tags:** security

Specifies the key name for the TSIG session key.

This is the key name to use for the TSIG session key. If not specified, the default is `local-ddns`.

**session-keyalg**

**Grammar:** `session-keyalg <string>;`

**Blocks:** options

**Tags:** security

Specifies the algorithm to use for the TSIG session key.

This is the algorithm to use for the TSIG session key. Valid values are hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, hmac-sha512, and hmac-md5. If not specified, the default is hmac-sha256.

**port**

**Grammar:** `port <integer>;`

**Blocks:** options

**Tags:** server, query

Specifies the UDP/TCP port number the server uses to receive and send DNS protocol traffic.

This is the UDP/TCP port number the server uses to receive and send DNS protocol traffic. The default is 53. This option is mainly intended for server testing; a server using a port other than 53 is not able to communicate with the global DNS.

**tls-port**

**Grammar:** `tls-port <integer>;`

**Blocks:** options

**Tags:** server, query

Specifies the TCP port number the server uses to receive and send DNS-over-TLS protocol traffic.

This is the TCP port number the server uses to receive and send DNS-over-TLS protocol traffic. The default is 853.

**https-port**

**Grammar:** `https-port <integer>;`

**Blocks:** options

**Tags:** server, query

Specifies the TCP port number the server uses to receive and send DNS-over-HTTPS protocol traffic.

This is the TCP port number the server uses to receive and send DNS-over-HTTPS protocol traffic. The default is 443.

**http-port**

**Grammar:** `http-port <integer>;`

**Blocks:** options

**Tags:** server, query

Specifies the TCP port number the server uses to receive and send unencrypted DNS traffic via HTTP.

This is the TCP port number the server uses to receive and send unencrypted DNS traffic via HTTP (a configuration that may be useful when encryption is handled by third-party software or by a reverse proxy).

**http-listener-clients**

**Grammar:** `http-listener-clients <integer>;`

**Blocks:** options

**Tags:** server

Limits the number of active concurrent connections on a per-listener basis.

This sets a hard limit on the number of active concurrent connections on a per-listener basis. The default value is 300; setting it to 0 removes the quota.

**http-streams-per-connection**

**Grammar:** `http-streams-per-connection <integer>;`

**Blocks:** options

**Tags:** server

Limits the number of active concurrent HTTP/2 streams on a per-connection basis.

This sets a hard limit on the number of active concurrent HTTP/2 streams on a per-connection basis. The default value is 100; setting it to 0 removes the limit. Once the limit is exceeded, the server finishes the HTTP session.

**preferred-glue**

**Grammar:** `preferred-glue <string>;`

**Blocks:** options, view

**Tags:** query

Controls the order of glue records in an A or AAAA response.

If specified, the listed type (A or AAAA) is emitted before other glue in the additional section of a query response. The default is to prefer A records when responding to queries that arrived via IPv4, and AAAA when responding to queries that arrived via IPv6.

**disable-algorithms**

**Grammar:** `disable-algorithms <string> { <string>; ... }; // may occur multiple times`

**Blocks:** options, view

**Tags:** dnssec

Disables DNSSEC algorithms from a specified zone.

This disables the specified DNSSEC algorithms at and below the specified zone. Multiple *disable-algorithms* statements are allowed. Only the best-match *disable-algorithms* clause is used to determine the algorithms.

If all supported algorithms are disabled, the zones covered by the *disable-algorithms* setting are treated as insecure.

Configured trust anchors in *trust-anchors* (or *managed-keys* or *trusted-keys*) that match a disabled algorithm are ignored and treated as if they were not configured.

**disable-ds-digests**

**Grammar:** `disable-ds-digests <string> { <string>; ... }; // may occur multiple times`

**Blocks:** options, view

**Tags:** dnssec, zone

Disables DS digest types from a specified zone.

This disables the specified DS digest types at and below the specified name. Multiple *disable-ds-digests* statements are allowed. Only the best-match *disable-ds-digests* clause is used to determine the digest types.

If all supported digest types are disabled, the zones covered by *disable-ds-digests* are treated as insecure.

**dnssec-must-be-secure**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `dnssec-must-be-secure <string> <boolean>; // may occur multiple times, deprecated`

**Blocks:** options, view

**Tags:** deprecated

Defines hierarchies that must or may not be secure (signed and validated).

This option is deprecated and will be removed in a future release.

This specifies hierarchies which must be or may not be secure (signed and validated). If `yes`, then *named* only accepts answers if they are secure. If `no`, then normal DNSSEC validation applies, allowing insecure answers to be accepted. The specified domain must be defined as a trust anchor, for instance in a *trust-anchors* statement, or `dnssec-validation auto` must be active.

**dns64**

**Grammar:**

```
dns64 <netprefix> {
        break-dnssec <boolean>;
        clients { <address_match_element>; ... };
        exclude { <address_match_element>; ... };
        mapped { <address_match_element>; ... };
        recursive-only <boolean>;
        suffix <ipv6_address>;
}; // may occur multiple times
```

**Blocks:** options, view

**Tags:** query

Instructs *named* to return mapped IPv4 addresses to AAAA queries when there are no AAAA records.

This directive instructs *named* to return mapped IPv4 addresses to AAAA queries when there are no AAAA records. It is intended to be used in conjunction with a NAT64. Each *dns64* defines one DNS64 prefix. Multiple DNS64 prefixes can be defined.

Compatible IPv6 prefixes have lengths of 32, 40, 48, 56, 64, and 96, per **RFC 6052**. Bits 64..71 inclusive must be zero, with the most significant bit of the prefix in position 0.

In addition, a reverse IP6.ARPA zone is created for the prefix to provide a mapping from the IP6.ARPA names to the corresponding IN-ADDR.ARPA names using synthesized CNAMEs.

**dns64-server**

**Grammar:** `dns64-server <string>;`

**Blocks:** options, view

**Tags:** server

Specifies the name of the server for *dns64* zones.

**dns64-contact**

**Grammar:** `dns64-contact <string>;`

**Blocks:** options, view

**Tags:** server

Specifies the name of the contact for *dns64* zones.

*dns64-server* and *dns64-contact* can be used to specify the name of the server and contact for the zones. These can be set at the view/options level but not on a per-prefix basis.

*dns64* will also cause IPV4ONLY.ARPA to be created if not explicitly disabled using *ipv4only-enable*.

**clients**

**Grammar:** `clients { <address_match_element>; ... };`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Specifies an access control list (ACL) of clients that are affected by a given *dns64* directive.

Each *dns64* supports an optional *clients* ACL that determines which clients are affected by this directive. If not defined, it defaults to `any;`.

**mapped**

**Grammar:** `mapped { <address_match_element>; ... };`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Specifies an access control list (ACL) of IPv4 addresses that are to be mapped to the corresponding A RRset in *dns64*.

Each *dns64* block supports an optional *mapped* ACL that selects which IPv4 addresses are to be mapped in the corresponding A RRset. If not defined, it defaults to `any;`.

**exclude**

**Grammar:** `exclude { <address_match_element>; ... };`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Allows a list of IPv6 addresses to be ignored if they appear in a domain name's AAAA records in *dns64*.

Normally, DNS64 does not apply to a domain name that owns one or more AAAA records; these records are simply returned. The optional *exclude* ACL allows specification of a list of IPv6 addresses that are ignored if they appear in a domain name's AAAA records; DNS64 is applied to any A records the domain name owns. If not defined, *exclude* defaults to ::ffff:0.0.0.0/96.

**suffix**

**Grammar:** `suffix <ipv6_address>;`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Defines trailing bits for mapped IPv4 address bits in *dns64*.

An optional *suffix* can also be defined to set the bits trailing the mapped IPv4 address bits. By default these bits are set to `::`. The bits matching the prefix and mapped IPv4 address must be zero.

**recursive-only**

**Grammar:** `recursive-only <boolean>;`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Toggles whether *dns64* synthesis occurs only for recursive queries.

If *recursive-only* is set to `yes`, the DNS64 synthesis only happens for recursive queries. The default is `no`.

**break-dnssec**

**Grammar:** `break-dnssec <boolean>;`

**Blocks:** options.dns64, view.dns64

**Tags:** query

Enables *dns64* synthesis even if the validated result would cause a DNSSEC validation failure.

If *break-dnssec* is set to `yes`, the DNS64 synthesis happens even if the result, if validated, would cause a DNSSEC validation failure. If this option is set to `no` (the default), the DO is set on the incoming query, and there are RRSIGs on the applicable records, then synthesis does not happen.

```
acl rfc1918 { 10/8; 192.168/16; 172.16/12; };

dns64 64:FF9B::/96 {
    clients { any; };
    mapped { !rfc1918; any; };
    exclude { 64:FF9B::/96; ::ffff:0000:0000/96; };
    suffix ::;
};
```

**resolver-use-dns64**

**Grammar:** `resolver-use-dns64 <boolean>;`

**Blocks:** options, view

**Tags:** server

Specifies whether to apply DNS64 mappings when sending queries.

If *resolver-use-dns64* is set to `yes`, then the IPv4-to-IPv6 address transformations specified by the *dns64* option are applied to IPv4 server addresses to which recursive queries are sent. This allows a server to perform lookups via a NAT64 connection; queries that would have been sent via IPv4 are instead sent to mapped IPv6 addresses. The default is `no`.

**ipv4only-enable**

**Grammar:** `ipv4only-enable <boolean>;`

**Blocks:** options, view

**Tags:** query

Enables automatic IPv4 zones if a *dns64* block is configured.

This enables or disables automatic zones `ipv4only.arpa`, `170.0.0.192.in-addr.arpa`, and `171.0.0.192.in-addr.arpa`.

By default these zones are loaded if *dns64* is configured.

**ipv4only-server**

**Grammar:** `ipv4only-server <string>;`

**Blocks:** options, view

**Tags:** server, query

Specifies the name of the server for the IPV4ONLY.ARPA zone created by *dns64*.

**ipv4only-contact**

**Grammar:** `ipv4only-contact <string>;`

**Blocks:** options, view

**Tags:** server

Specifies the contact for the IPV4ONLY.ARPA zone created by *dns64*.

*ipv4only-server* and *ipv4only-contact* can be used to specify the name of the server and contact for the IPV4ONLY.ARPA zone created by *dns64*.

**dnssec-loadkeys-interval**

**Grammar:** `dnssec-loadkeys-interval <integer>;`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Sets the frequency of automatic checks of the DNSSEC key repository.

When a zone is configured with `dnssec-policy;`, its key repository must be checked periodically to see whether the next step of a key rollover is due. The *dnssec-loadkeys-interval* option sets the default interval of key repository checks, in minutes, in case the next key event cannot be calculated (e.g. because a DS record needs to be published).

The default is `60` (1 hour), the minimum is `1` (1 minute), and the maximum is `1440` (24 hours); any higher value is silently reduced.

*dnssec-policy*

This specifies which key and signing policy (KASP) should be used for this zone. This is a string referring to a *dnssec-policy* block. The default is `none`.

**dnssec-update-mode**

**Grammar:** `dnssec-update-mode ( maintain | no-resign ); // obsolete`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** obsolete

This option no longer has any effect.

**nta-lifetime**

**Grammar:** `nta-lifetime <duration>;`

**Blocks:** options, view

**Tags:** dnssec

Specifies the lifetime, in seconds, for negative trust anchors added via *rndc nta*.

This specifies the default lifetime, in seconds, for negative trust anchors added via *rndc nta*.

A negative trust anchor selectively disables DNSSEC validation for zones that are known to be failing because of misconfiguration, rather than an attack. When data to be validated is at or below an active NTA (and above any other configured trust anchors), *named* aborts the DNSSEC validation process and treats the data as insecure rather than bogus. This continues until the NTA's lifetime has elapsed. NTAs persist across *named* restarts.

For convenience, TTL-style time-unit suffixes can be used to specify the NTA lifetime in seconds, minutes, or hours. It also accepts ISO 8601 duration formats.

*nta-lifetime* defaults to one hour; it cannot exceed one week.

**nta-recheck**

**Grammar:** `nta-recheck <duration>;`

**Blocks:** options, view

**Tags:** dnssec

Specifies the time interval for checking whether negative trust anchors added via *rndc nta* are still necessary.

This specifies how often to check whether negative trust anchors added via *rndc nta* are still necessary.

A negative trust anchor is normally used when a domain has stopped validating due to operator error; it temporarily disables DNSSEC validation for that domain. In the interest of ensuring that DNSSEC validation is turned back on as soon as possible, *named* periodically sends a query to the domain, ignoring negative trust anchors, to find out whether it can now be validated. If so, the negative trust anchor is allowed to expire early.

Validity checks can be disabled for an individual NTA by using *rndc nta -f*, or for all NTAs by setting *nta-recheck* to zero.

For convenience, TTL-style time-unit suffixes can be used to specify the NTA recheck interval in seconds, minutes, or hours. It also accepts ISO 8601 duration formats.

The default is five minutes. It cannot be longer than *nta-lifetime*, which cannot be longer than a week.

**max-zone-ttl**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar dnssec-policy:** `max-zone-ttl <duration>;`

**Grammar options, view, zone (primary, redirect):** `max-zone-ttl ( unlimited | <duration> ); // deprecated`

**Blocks:** dnssec-policy, options, view, zone (primary, redirect)

**Tags:** deprecated

Specifies a maximum permissible time-to-live (TTL) value, in seconds.

This should now be configured as part of *dnssec-policy*. Use of this option in *options*, *view*, and *zone* blocks is a fatal error if *dnssec-policy* has also been configured for the same zone. In zones without *dnssec-policy*, this option is deprecated, and will be rendered non-operational in a future release.

*max-zone-ttl* specifies a maximum permissible TTL value in seconds. For convenience, TTL-style time-unit suffixes may be used to specify the maximum value. When a zone file is loaded, any record encountered with a TTL higher than *max-zone-ttl* causes the zone to be rejected.

This is needed in DNSSEC-maintained zones because when rolling to a new DNSKEY, the old key needs to remain available until RRSIG records have expired from caches. The *max-zone-ttl* option guarantees that the largest TTL in the zone is no higher than the set value.

When used in *options*, *view* and *zone* blocks, setting *max-zone-ttl* to zero is equivalent to "unlimited".

**stale-answer-ttl**

**Grammar:** `stale-answer-ttl <duration>;`

**Blocks:** options, view

**Tags:** query

Specifies the time to live (TTL) to be returned on stale answers, in seconds.

This specifies the TTL to be returned on stale answers. The default is 30 seconds. The minimum allowed is 1 second; a value of 0 is updated silently to 1 second.

For stale answers to be returned, they must be enabled, either in the configuration file using *stale-answer-enable* or via *rndc serve-stale on*.

**serial-update-method**

**Grammar:** `serial-update-method ( date | increment | unixtime );`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Specifies the update method to be used for the zone serial number in the SOA record.

Zones configured for dynamic DNS may use this option to set the update method to be used for the zone serial number in the SOA record.

With the default setting of `serial-update-method increment;`, the SOA serial number is incremented by one each time the zone is updated.

When set to `serial-update-method unixtime;`, the SOA serial number is set to the number of seconds since the Unix epoch, unless the serial number is already greater than or equal to that value, in which case it is simply incremented by one.

When set to `serial-update-method date;`, the new SOA serial number is the current date in the form "YYYYMMDD", followed by two zeroes, unless the existing serial number is already greater than or equal to that value, in which case it is incremented by one.

**zone-statistics**

**Grammar:** `zone-statistics ( full | terse | none | <boolean> );`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** zone, logging

Controls the level of statistics gathered for all zones.

If `full`, the server collects statistical data on all zones, unless specifically turned off on a per-zone basis by specifying `zone-statistics terse` or `zone-statistics none` in the *zone* statement. The statistical data includes, for example, DNSSEC signing operations and the number of authoritative answers per query type. The default is `terse`, providing minimal statistics on zones (including name and current serial number, but not query type counters), and also information about the currently ongoing incoming zone transfers.

These statistics may be accessed via the `statistics-channel` or using *rndc stats*, which dumps them to the file listed in the *statistics-file*. See also *The Statistics File*.

For backward compatibility with earlier versions of BIND 9, the *zone-statistics* option can also accept `yes` or `no`; `yes` has the same meaning as `full`. As of BIND 9.10, `no` has the same meaning as `none`; previously, it was the same as `terse`.

### Boolean Options

**automatic-interface-scan**

**Grammar:** `automatic-interface-scan <boolean>;`

**Blocks:** options

**Tags:** server

Controls the automatic rescanning of network interfaces when addresses are added or removed.

If `yes` and supported by the operating system, this automatically rescans network interfaces when the interface addresses are added or removed. The default is `yes`. This configuration option does not affect the time-based *interface-interval* option; it is recommended to set the time-based *interface-interval* to 0 when the operator confirms that automatic interface scanning is supported by the operating system.

The *automatic-interface-scan* implementation uses routing sockets for the network interface discovery; therefore, the operating system must support the routing sockets for this feature to work.

**allow-new-zones**

**Grammar:** `allow-new-zones <boolean>;`

**Blocks:** options, view

**Tags:** server, zone

Controls the ability to add zones at runtime via *rndc addzone*.

If `yes`, then zones can be added at runtime via *`rndc addzone`*. The default is `no`.

Newly added zones' configuration parameters are stored so that they can persist after the server is restarted. The configuration information is saved in a file called `viewname.nzf` (or, if *`named`* is compiled with liblmdb, in an LMDB database file called `viewname.nzd`). "viewname" is the name of the view, unless the view name contains characters that are incompatible with use as a file name, in which case a cryptographic hash of the view name is used instead.

Configurations for zones added at runtime are stored either in a new-zone file (NZF) or a new-zone database (NZD), depending on whether *`named`* was linked with liblmdb at compile time. See *rndc - name server control utility* for further details about *`rndc addzone`*.

**`auth-nxdomain`**

**Grammar:** `auth-nxdomain <boolean>;`

**Blocks:** options, view

**Tags:** query

Controls whether BIND, acting as a resolver, provides authoritative NXDOMAIN (domain does not exist) answers.

If `yes`, then the `AA` bit is always set on NXDOMAIN responses, even if the server is not actually authoritative. The default is `no`.

**`memstatistics`**

**Grammar:** `memstatistics <boolean>;`

**Blocks:** options

**Tags:** server, logging

Controls whether memory statistics are written to the file specified by *`memstatistics-file`* at exit.

This writes memory statistics to the file specified by *`memstatistics-file`* at exit. The default is `no` unless *`-m record`* is specified on the command line, in which case it is `yes`.

**`dialup`**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `dialup ( notify | notify-passive | passive | refresh | <boolean> ); // deprecated`

**Blocks:** options, view, zone (primary, secondary, stub)

**Tags:** deprecated

Concentrates zone maintenance so that all transfers take place once every *`heartbeat-interval`*, ideally during a single call.

This option is deprecated and will be removed in a future release.

If `yes`, then the server treats all zones as if they are doing zone transfers across a dial-on-demand dialup link, which can be brought up by traffic originating from this server. Although this setting has different effects according to zone

type, it concentrates the zone maintenance so that everything happens quickly, once every *heartbeat-interval*, ideally during a single call. It also suppresses some normal zone maintenance traffic. The default is no.

If specified in the *view* and *zone* statements, the *dialup* option overrides the global *dialup* option.

If the zone is a primary zone, the server sends out a NOTIFY request to all the secondaries (default). This should trigger the zone serial number check in the secondary (providing it supports NOTIFY), allowing the secondary to verify the zone while the connection is active. The set of servers to which NOTIFY is sent can be controlled by *notify* and *also-notify*.

If the zone is a secondary or stub zone, the server suppresses the regular "zone up to date" (refresh) queries and only performs them when the *heartbeat-interval* expires, in addition to sending NOTIFY requests.

Finer control can be achieved by using *notify*, which only sends NOTIFY messages; notify-passive, which sends NOTIFY messages and suppresses the normal refresh queries; refresh, which suppresses normal refresh processing and sends refresh queries when the *heartbeat-interval* expires; and passive, which disables normal refresh processing.

| dialup mode | normal refresh | heart-beat refresh | heart-beat notify |
|---|---|---|---|
| no (default) | yes | no | no |
| yes | no | yes | yes |
| notify | yes | no | yes |
| refresh | no | yes | no |
| passive | no | no | no |
| notify-passive | no | no | yes |

Note that normal NOTIFY processing is not affected by *dialup*.

**flush-zones-on-shutdown**

**Grammar:** flush-zones-on-shutdown <boolean>;

**Blocks:** options

**Tags:** zone

Controls whether pending zone writes are flushed when the name server exits.

When the name server exits upon receiving SIGTERM, flush or do not flush any pending zone writes. The default is flush-zones-on-shutdown no.

**root-key-sentinel**

**Grammar:** root-key-sentinel <boolean>;

**Blocks:** options, view

**Tags:** server

Controls whether BIND 9 responds to root key sentinel probes.

If yes, the server responds to root key sentinel probes as described in **RFC 8509**:. The default is yes.

**reuseport**

**Grammar:** reuseport <boolean>;

**Blocks:** options

**Tags:** server

Enables kernel load-balancing of sockets.

This option enables kernel load-balancing of sockets on systems which support it, including Linux (SO_REUSEPORT) and FreeBSD (SO_REUSEPORT_LB). This instructs the kernel to distribute incoming socket connections among the networking threads based on a hashing scheme. For more information, see the receive network flow classification options (`rx-flow-hash`) section in the `ethtool` manual page. The default is `yes`.

Enabling *`reuseport`* significantly increases general throughput when incoming traffic is distributed uniformly onto the threads by the operating system. However, in cases where a worker thread is busy with a long-lasting operation, such as processing a Response Policy Zone (RPZ) or Catalog Zone update or an unusually large zone transfer, incoming traffic that hashes onto that thread may be delayed. On servers where these events occur frequently, it may be preferable to disable socket load-balancing so that other threads can pick up the traffic that would have been sent to the busy thread.

Note: this option can only be set when *`named`* first starts. Changes will not take effect during reconfiguration; the server must be restarted.

**`message-compression`**

**Grammar:** `message-compression <boolean>;`

**Blocks:** options, view

**Tags:** query

Controls whether DNS name compression is used in responses to regular queries.

If `yes`, DNS name compression is used in responses to regular queries (not including AXFR or IXFR, which always use compression). Setting this option to `no` reduces CPU usage on servers and may improve throughput. However, it increases response size, which may cause more queries to be processed using TCP; a server with compression disabled is out of compliance with **RFC 1123** Section 6.1.3.2. The default is `yes`.

**`minimal-responses`**

**Grammar:** `minimal-responses ( no-auth | no-auth-recursive | <boolean> );`

**Blocks:** options, view

**Tags:** query

Controls whether the server only adds records to the authority and additional data sections when they are required (e.g. delegations, negative responses). This improves server performance.

This option controls the addition of records to the authority and additional sections of responses. Such records may be included in responses to be helpful to clients; for example, MX records may have associated address records included in the additional section, obviating the need for a separate address lookup. However, adding these records to responses is not mandatory and requires additional database lookups, causing extra latency when marshalling responses.

Responses to DNSKEY, DS, CDNSKEY, and CDS requests will never have optional additional records added. Responses to NS requests will always have additional section processing.

*`minimal-responses`* takes one of four values:

- `no`: the server is as complete as possible when generating responses.
- `yes`: the server only adds records to the authority and additional sections when such records are required by the DNS protocol (for example, when returning delegations or negative responses). This provides the best server performance but may result in more client queries.

- `no-auth`: the server omits records from the authority section except when they are required, but it may still add records to the additional section.
- `no-auth-recursive`: the same as `no-auth` when recursion is requested in the query (RD=1), or the same as `no` if recursion is not requested.

`no-auth` and `no-auth-recursive` are useful when answering stub clients, which usually ignore the authority section. `no-auth-recursive` is meant for use in mixed-mode servers that handle both authoritative and recursive queries.

The default is `no-auth-recursive`.

**`minimal-any`**

**Grammar:** `minimal-any <boolean>;`

**Blocks:** options, view

**Tags:** query

Controls whether the server replies with only one of the RRsets for a query name, when generating a positive response to a query of type ANY over UDP.

If set to `yes`, the server replies with only one of the RRsets for the query name, and its covering RRSIGs if any, when generating a positive response to a query of type ANY over UDP, instead of replying with all known RRsets for the name. Similarly, a query for type RRSIG is answered with the RRSIG records covering only one type. This can reduce the impact of some kinds of attack traffic, without harming legitimate clients. (Note, however, that the RRset returned is the first one found in the database; it is not necessarily the smallest available RRset.) Additionally, *`minimal-responses`* is turned on for these queries, so no unnecessary records are added to the authority or additional sections. The default is `no`.

**`notify`**

**Grammar:** `notify ( explicit | master-only | primary-only | <boolean> );`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Controls whether `NOTIFY` messages are sent on zone changes.

If set to `yes` (the default), DNS NOTIFY messages are sent when a zone the server is authoritative for changes; see *using notify*. The messages are sent to the servers listed in the zone's NS records (except the primary server identified in the SOA MNAME field), and to any servers listed in the *`also-notify`* option.

If set to `primary-only` (or the older keyword `master-only`), notifies are only sent for primary zones. If set to `explicit`, notifies are sent only to servers explicitly listed using *`also-notify`*. If set to `no`, no notifies are sent.

The *`notify`* option may also be specified in the *`zone`* statement, in which case it overrides the `options notify` statement. It would only be necessary to turn off this option if it caused secondary zones to crash.

**`notify-to-soa`**

**Grammar:** `notify-to-soa <boolean>;`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** transfer

Controls whether the name servers in the NS RRset are checked against the SOA MNAME.

If `yes`, do not check the name servers in the NS RRset against the SOA MNAME. Normally a NOTIFY message is not sent to the SOA MNAME (SOA ORIGIN), as it is supposed to contain the name of the ultimate primary server. Sometimes, however, a secondary server is listed as the SOA MNAME in hidden primary configurations; in that case, the ultimate primary should be set to still send NOTIFY messages to all the name servers listed in the NS RRset.

**recursion**

**Grammar:** `recursion <boolean>;`

**Blocks:** options, view

**Tags:** query

Defines whether recursion and caching are allowed.

If `yes`, and a DNS query requests recursion, then the server attempts to do all the work required to answer the query. If recursion is off and the server does not already know the answer, it returns a referral response. The default is `yes`. Note that setting `recursion no` does not prevent clients from getting data from the server's cache; it only prevents new data from being cached as an effect of client queries. Caching may still occur as an effect of the server's internal operation, such as NOTIFY address lookups.

**request-nsid**

**Grammar:** `request-nsid <boolean>;`

**Blocks:** options, server, view, view.server

**Tags:** query

Controls whether an empty EDNS(0) NSID (Name Server Identifier) option is sent with all queries to authoritative name servers during iterative resolution.

If `yes`, then an empty EDNS(0) NSID (Name Server Identifier) option is sent with all queries to authoritative name servers during iterative resolution. If the authoritative server returns an NSID option in its response, then its contents are logged in the `nsid` category at level `info`. The default is `no`.

**require-cookie**

**Grammar:** `require-cookie <boolean>;`

**Blocks:** server, view.server

**Tags:** query

Controls whether responses without a server cookie are accepted.

The `require-cookie` clause can be used to indicate that the remote server is known to support DNS COOKIE. Setting this option to `yes` causes *named* to always retry a request over TCP when it receives a UDP response without a DNS COOKIE from the remote server, even if UDP responses with DNS COOKIE have not been sent by this server before. This prevents spoofed answers from being accepted without a retry over TCP, when *named* has not yet determined whether the remote server supports DNS COOKIE. Setting this option to `no` (the default) causes *named* to rely on autodetection of DNS COOKIE support to determine when to retry a request over TCP.

For DNAME lookups the default is `yes` and it is enforced. Servers serving DNAME must correctly support DNS over TCP.

> **Note**
>
> If a UDP response is signed using TSIG, *named* accepts it even if `require-cookie` is set to `yes` and the response does not contain a DNS COOKIE.

The `send-cookie` clause determines whether the local server adds a COOKIE EDNS option to requests sent to the server. This overrides `send-cookie` set at the view or option level. The *named* server may determine that COOKIE is not supported by the remote server and not add a COOKIE EDNS option to requests.

**require-server-cookie**

**Grammar:** `require-server-cookie <boolean>;`

**Blocks:** options, view

**Tags:** query

Controls whether a valid server cookie is required before sending a full response to a UDP request.

If `yes`, BIND requires a valid server cookie before sending a full response to a UDP request from a cookie-aware client. BADCOOKIE is sent if there is a bad or nonexistent server cookie.

The default is `no`.

Users wishing to test that DNS COOKIE clients correctly handle BADCOOKIE, or who are getting a lot of forged DNS requests with DNS COOKIES present, should set this to `yes`. Setting this to `yes` results in a reduced amplification effect in a reflection attack, as the BADCOOKIE response is smaller than a full response, while also requiring a legitimate client to follow up with a second query with the new, valid, cookie.

**answer-cookie**

**Grammar:** `answer-cookie <boolean>;`

**Blocks:** options

**Tags:** query

Controls whether COOKIE EDNS replies are sent in response to client queries.

When set to the default value of `yes`, COOKIE EDNS options are sent when applicable in replies to client queries. If set to `no`, COOKIE EDNS options are not sent in replies. This can only be set at the global options level, not per-view.

`answer-cookie no` is intended as a temporary measure, for use when *named* shares an IP address with other servers that do not yet support DNS COOKIE. A mismatch between servers on the same address is not expected to cause operational problems, but the option to disable COOKIE responses so that all servers have the same behavior is provided out of an abundance of caution. DNS COOKIE is an important security mechanism, and should not be disabled unless absolutely necessary.

**send-cookie**

**Grammar:** `send-cookie <boolean>;`

**Blocks:** options, server, view, view.server

**Tags:** query

Controls whether a COOKIE EDNS option is sent along with a query.

If `yes`, a COOKIE EDNS option is sent along with the query. If the resolver has previously communicated with the server, the COOKIE returned in the previous transaction is sent. This is used by the server to determine whether the resolver has talked to it before. A resolver sending the correct COOKIE is assumed not to be an off-path attacker sending a spoofed-source query; the query is therefore unlikely to be part of a reflection/amplification attack, so resolvers sending a correct COOKIE option are not subject to response-rate limiting (RRL). Resolvers which do not send a correct COOKIE option may be limited to receiving smaller responses via the *nocookie-udp-size* option.

The *named* server may determine that COOKIE is not supported by the remote server and not add a COOKIE EDNS option to requests.

The default is `yes`.

**stale-answer-enable**

**Grammar:** `stale-answer-enable <boolean>;`

**Blocks:** options, view

**Tags:** server, query

Enables the returning of "stale" cached answers when the name servers for a zone are not answering.

If `yes`, this option enables the returning of "stale" cached answers when the name servers for a zone are not answering and the *stale-cache-enable* option is also enabled. The default is not to return stale answers.

Stale answers can also be enabled or disabled at runtime via *rndc serve-stale on* or *rndc serve-stale off*; these override the configured setting. *rndc serve-stale reset* restores the setting to the one specified in *named.conf*. Note that if stale answers have been disabled by *rndc*, they cannot be re-enabled by reloading or reconfiguring *named*; they must be re-enabled with *rndc serve-stale on*, or the server must be restarted.

Information about stale answers is logged under the `serve-stale` log category.

**stale-answer-client-timeout**

**Grammar:** `stale-answer-client-timeout ( disabled | off | <integer> );`

**Blocks:** options, view

**Tags:** server, query

Defines the amount of time (in milliseconds) that *named* waits before attempting to answer a query with a stale RRset from cache.

This option defines the amount of time (in milliseconds) that *named* waits before attempting to answer the query with a stale RRset from cache. If a stale answer is found, *named* continues the ongoing fetches, attempting to refresh the RRset in cache until the *resolver-query-timeout* interval is reached.

This option is off by default, which is equivalent to setting it to `off` or `disabled`. It also has no effect if *stale-answer-enable* is disabled.

The minimum value, `0`, causes a cached (stale) RRset to be immediately returned if it is available, while still attempting to refresh the data in cache.

When this option is enabled, the only supported value in the current version of BIND 9 is `0`. Non-zero values generate a warning message and are treated as `0`.

**stale-cache-enable**

**Grammar:** `stale-cache-enable <boolean>;`

**Blocks:** options, view

**Tags:** server, query

Enables the retention of "stale" cached answers.

If `yes`, enable the retaining of "stale" cached answers. Default `no`.

**`stale-refresh-time`**

**Grammar:** `stale-refresh-time <duration>;`

**Blocks:** options, view

**Tags:** server, query

Sets the time window for the return of "stale" cached answers before the next attempt to contact, if the name servers for a given zone are not responding.

If the name servers for a given zone are not answering, this sets the time window for which *`named`* will promptly return "stale" cached answers for that RRSet being requested before a new attempt in contacting the servers is made. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default *`stale-refresh-time`* is 30 seconds, as **RFC 8767** recommends that attempts to refresh to be done no more frequently than every 30 seconds. A value of zero disables the feature, meaning that normal resolution will take place first, if that fails only then *`named`* will return "stale" cached answers.

**`nocookie-udp-size`**

**Grammar:** `nocookie-udp-size <integer>;`

**Blocks:** options, view

**Tags:** query

Sets the maximum size of UDP responses that are sent to queries without a valid server COOKIE.

This sets the maximum size of UDP responses that are sent to queries without a valid server COOKIE. A value below 128 is silently raised to 128. The default value is 4096, but the *`max-udp-size`* option may further limit the response size as the default for *`max-udp-size`* is 1232.

**`cookie-algorithm`**

**Grammar:** `cookie-algorithm ( siphash24 );`

**Blocks:** options

**Tags:** server

Sets the algorithm to be used when generating a server cookie.

This sets the algorithm to be used when generating the server cookie. The default is "siphash24", which is the only supported option, as the previously supported "aes" option has been removed.

**`cookie-secret`**

**Grammar:** `cookie-secret <string>; // may occur multiple times`

**Blocks:** options

**Tags:** server

Specifies a shared secret used for generating and verifying EDNS COOKIE options within an anycast cluster.

If set, this is a shared secret used for generating and verifying EDNS COOKIE options within an anycast cluster. If not set, the system generates a random secret at startup. The shared secret is encoded as a hex string and needs to be 128 bits.

If there are multiple secrets specified, the first one listed in *named.conf* is used to generate new server cookies. The others are only used to verify returned cookies.

**response-padding**

**Grammar:** `response-padding { <address_match_element>; ... } block-size <integer>;`

**Blocks:** options, view

**Tags:** query

Adds an EDNS Padding option to encrypted messages, to reduce the chance of guessing the contents based on size.

The EDNS Padding option is intended to improve confidentiality when DNS queries are sent over an encrypted channel, by reducing the variability in packet sizes. If a query:

1. contains an EDNS Padding option,
2. includes a valid server cookie or uses TCP,
3. is not signed using TSIG or SIG(0), and
4. is from a client whose address matches the specified ACL,

then the response is padded with an EDNS Padding option to a multiple of `block-size` bytes. If these conditions are not met, the response is not padded.

If `block-size` is 0 or the ACL is `none;`, this feature is disabled and no padding occurs; this is the default. If `block-size` is greater than 512, a warning is logged and the value is truncated to 512. Block sizes are ordinarily expected to be powers of two (for instance, 128), but this is not mandatory.

**trust-anchor-telemetry**

**Grammar:** `trust-anchor-telemetry <boolean>;`

**Blocks:** options, view

**Tags:** dnssec

Instructs *named* to send specially formed queries once per day to domains for which trust anchors have been configured.

This causes *named* to send specially formed queries once per day to domains for which trust anchors have been configured via, e.g., *trust-anchors* or `dnssec-validation auto`.

The query name used for these queries has the form `_ta-xxxx(-xxxx)(...).<domain>`, where each "xxxx" is a group of four hexadecimal digits representing the key ID of a trusted DNSSEC key. The key IDs for each domain are sorted smallest to largest prior to encoding. The query type is NULL.

By monitoring these queries, zone operators are able to see which resolvers have been updated to trust a new key; this may help them decide when it is safe to remove an old one.

The default is `yes`.

**provide-ixfr**

**Grammar:** `provide-ixfr <boolean>;`

**Blocks:** options, server, view, view.server

**Tags:** transfer

Controls whether a primary responds to an incremental zone request (IXFR) or only responds with a full zone transfer (AXFR).

The *provide-ixfr* clause determines whether the local server, acting as primary, responds with an incremental zone transfer when the given remote server, a secondary, requests it. If set to `yes`, incremental transfer is provided whenever possible. If set to `no`, all transfers to the remote server are non-incremental.

**request-ixfr**

**Grammar:** `request-ixfr <boolean>;`

**Blocks:** options, server, view, zone (mirror, secondary), view.server

**Tags:** transfer

Controls whether a secondary requests an incremental zone transfer (IXFR) or a full zone transfer (AXFR).

The *request-ixfr* statement determines whether the local server, acting as a secondary, requests incremental zone transfers from the given remote server, a primary.

IXFR requests to servers that do not support IXFR automatically fall back to AXFR. Therefore, there is no need to manually list which servers support IXFR and which ones do not; the global default of `yes` should always work. The purpose of the *provide-ixfr* and *request-ixfr* statements is to make it possible to disable the use of IXFR even when both primary and secondary claim to support it: for example, if one of the servers is buggy and crashes or corrupts data when IXFR is used.

It may also be set in the zone block; if set there, it overrides the global or view setting for that zone. It may also be set in the *server* block.

**request-expire**

**Grammar:** `request-expire <boolean>;`

**Blocks:** options, server, view, zone (mirror, secondary), view.server

**Tags:** transfer, query

Specifies whether the local server requests the EDNS EXPIRE value, when acting as a secondary.

The *request-expire* statement determines whether the local server, when acting as a secondary, requests the EDNS EXPIRE value. The EDNS EXPIRE value indicates the remaining time before the zone data expires and needs to be refreshed. This is used when a secondary server transfers a zone from another secondary server; when transferring from the primary, the expiration timer is set from the EXPIRE field of the SOA record instead. The default is `yes`.

**match-mapped-addresses**

**Grammar:** `match-mapped-addresses <boolean>;`

**Blocks:** options

**Tags:** server

Allows IPv4-mapped IPv6 addresses to match address-match list entries for corresponding IPv4 addresses.

If `yes`, then an IPv4-mapped IPv6 address matches any address-match list entries that match the corresponding IPv4 address.

This option was introduced to work around a kernel quirk in some operating systems that causes IPv4 TCP connections, such as zone transfers, to be accepted on an IPv6 socket using mapped addresses. This caused address-match lists designed for IPv4 to fail to match. However, *named* now solves this problem internally. The use of this option is discouraged.

**ixfr-from-differences**

**Grammar zone (mirror, primary, secondary):** `ixfr-from-differences <boolean>;`

**Grammar options, view:** `ixfr-from-differences ( primary | master | secondary | slave | <boolean> );`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Controls how IXFR transfers are calculated.

When `yes` and the server loads a new version of a primary zone from its zone file or receives a new version of a secondary file via zone transfer, it compares the new version to the previous one and calculates a set of differences. The differences are then logged in the zone's journal file so that the changes can be transmitted to downstream secondaries as an incremental zone transfer.

By allowing incremental zone transfers to be used for non-dynamic zones, this option saves bandwidth at the expense of increased CPU and memory consumption at the primary server. In particular, if the new version of a zone is completely different from the previous one, the set of differences is of a size comparable to the combined size of the old and new zone versions, and the server needs to temporarily allocate memory to hold this complete difference set.

*ixfr-from-differences* also accepts `primary` and `secondary` at the view and options levels, which causes *ixfr-from-differences* to be enabled for all primary or secondary zones, respectively. It is off for all zones by default.

Note: if inline signing is enabled for a zone, the user-provided *ixfr-from-differences* setting is ignored for that zone.

**multi-master**

**Grammar:** `multi-master <boolean>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Controls whether serial number mismatch errors are logged.

This should be set when there are multiple primary servers for a zone and the addresses refer to different machines. If `yes`, *named* does not log when the serial number on the primary is less than what *named* currently has. The default is `no`.

**dnssec-validation**

**Grammar:** `dnssec-validation ( yes | no | auto );`

**Blocks:** options, view

**Tags:** dnssec

Enables DNSSEC validation in *named*.

This option enables DNSSEC validation in *named*.

If set to `auto`, DNSSEC validation is enabled and a default trust anchor for the DNS root zone is used. This trust anchor is provided as part of BIND and is kept up-to-date using *Dynamic Trust Anchor Management* key management. Adding an explicit static key using the *trust-anchors* statement, with a `static-key` anchor type (or using the deprecated *trusted-keys* statement) for the root zone, is not supported with the `auto` setting and is treated as a configuration error.

If set to `yes`, DNSSEC validation is enabled, but a trust anchor must be manually configured using a *trust-anchors* statement (or the *managed-keys* or *trusted-keys* statements, both deprecated). If *trust-anchors* is not configured, it is a configuration error. If *trust-anchors* does not include a valid root key, then validation does not take place for names which are not covered by any of the configured trust anchors.

If set to `no`, DNSSEC validation is disabled. (Note: the resolver will still set the DO bit in outgoing queries to indicate that it can accept DNSSEC responses, even if *dnssec-validation* is disabled.)

The default is `auto`, unless BIND is built with `configure --disable-auto-validation`, in which case the default is `yes`.

The default root trust anchor is compiled into *named* and is current as of the release date. If the root key changes, a running BIND server detects this and rolls smoothly to the new key. However, newly installed servers will be unable to start validation, and BIND must be upgraded to a newer version.

**validate-except**

**Grammar:** `validate-except { <string>; ... };`

**Blocks:** options, view

**Tags:** dnssec

Specifies a list of domain names at and beneath which DNSSEC validation should not be performed.

This specifies a list of domain names at and beneath which DNSSEC validation should *not* be performed, regardless of the presence of a trust anchor at or above those names. This may be used, for example, when configuring a top-level domain intended only for local use, so that the lack of a secure delegation for that domain in the root zone does not cause validation failures. (This is similar to setting a negative trust anchor except that it is a permanent configuration, whereas negative trust anchors expire and are removed after a set period of time.)

**dnssec-accept-expired**

**Grammar:** `dnssec-accept-expired <boolean>;`

**Blocks:** options, view

**Tags:** dnssec

Instructs BIND 9 to accept expired DNSSEC signatures when validating.

This accepts expired signatures when verifying DNSSEC signatures. The default is `no`. Setting this option to `yes` leaves *named* vulnerable to replay attacks.

**querylog**

**Grammar:** `querylog <boolean>;`

**Blocks:** options

**Tags:** logging, server

Specifies whether query logging should be active when *named* first starts.

Query logging provides a complete log of all incoming queries and all query errors. This provides more insight into the server's activity, but with a cost to performance which may be significant on heavily loaded servers.

The *querylog* option specifies whether query logging should be active when *named* first starts. If *querylog* is not specified, then query logging is determined by the presence of the logging category `queries`. Please note that *rndc reconfig* and *rndc reload* have no effect on this option, so it cannot be changed once the server is running. However, query logging can be activated at runtime using the command `rndc querylog on`, or deactivated with *rndc querylog off*.

**responselog**

**Grammar:** `responselog <boolean>;`

**Blocks:** options

**Tags:** logging, server

Specifies whether response logging should be active when *named* first starts.

Response logging complements *querylog* by logging the rcode of previous queries along with the queries' name, type and class.

Response logging can also be activated at runtime using the command `rndc responselog on`, or deactivated with *rndc responselog off*.

**check-names**

**Grammar zone (hint, mirror, primary, secondary, stub):** `check-names ( fail | warn | ignore );`

**Grammar options, view:** `check-names ( primary | master | secondary | slave | response ) ( fail | warn | ignore ); // may occur multiple times`

**Blocks:** options, view, zone (hint, mirror, primary, secondary, stub)

**Tags:** query, server

Restricts the character set and syntax of certain domain names in primary files and/or DNS responses received from the network.

This option is used to restrict the character set and syntax of certain domain names in primary files and/or DNS responses received from the network. The default varies according to usage area. For *type primary* zones the default is `fail`. For *type secondary* zones the default is `warn`. For answers received from the network (`response`), the default is `ignore`.

The rules for legal hostnames and mail domains are derived from **RFC 952** and **RFC 821** as modified by **RFC 1123**.

*check-names* applies to the owner names of A, AAAA, and MX records. It also applies to the domain names in the RDATA of NS, SOA, MX, and SRV records. It further applies to the RDATA of PTR records where the owner name indicates that it is a reverse lookup of a hostname (the owner name ends in IN-ADDR.ARPA, IP6.ARPA, or IP6.INT).

Owner names of A and AAAA records starting with the Active Directory Forest prefix labels `gc._msdcs` are excluded if the remainder of the name meets the check-names rules.

Owner names of A records meeting the **RFC 7208** rules for the labels `_spf`, `_spf_verify` and `_spf_rate` are also excluded.

**check-dup-records**

**Grammar:** `check-dup-records ( fail | warn | ignore );`

**Blocks:** options, view, zone (primary)

**Tags:** dnssec, query

Checks primary zones for records that are treated as different by DNSSEC but are semantically equal in plain DNS.

This checks primary zones for records that are treated as different by DNSSEC but are semantically equal in plain DNS. The default is to `warn`. Other possible values are `fail` and `ignore`.

**check-mx**

**Grammar:** `check-mx ( fail | warn | ignore );`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Checks whether an MX record appears to refer to an IP address.

This checks whether the MX record appears to refer to an IP address. The default is to `warn`. Other possible values are `fail` and `ignore`.

**check-wildcard**

**Grammar:** `check-wildcard <boolean>;`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Checks for non-terminal wildcards.

This option is used to check for non-terminal wildcards. The use of non-terminal wildcards is almost always as a result of a lack of understanding of the wildcard-matching algorithm (**RFC 1034**). This option affects primary zones. The default (`yes`) is to check for non-terminal wildcards and issue a warning.

**check-integrity**

**Grammar:** `check-integrity <boolean>;`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Performs post-load zone integrity checks on primary zones.

This performs post-load zone integrity checks on primary zones. It checks that MX and SRV records refer to address (A or AAAA) records and that glue address records exist for delegated zones. For MX and SRV records, only in-zone hostnames are checked (for out-of-zone hostnames, use *named-checkzone*). For NS records, only names below top-of-zone are checked (for out-of-zone names and glue consistency checks, use *named-checkzone*). DS records not at delegations are rejected. The default is `yes`.

The use of the SPF record to publish Sender Policy Framework is deprecated, as the migration from using TXT records to SPF records was abandoned. Enabling this option also checks that a TXT Sender Policy Framework record exists (starts with "v=spf1") if there is an SPF record. Warnings are emitted if the TXT record does not exist; they can be suppressed with *check-spf*.

**check-mx-cname**

**Grammar:** `check-mx-cname ( fail | warn | ignore );`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Sets the response to MX records that refer to CNAMEs.

If *check-integrity* is set, *named* fails, warns, or ignores MX records that refer to CNAMEs. The default is to `warn`.

**check-srv-cname**

**Grammar:** `check-srv-cname ( fail | warn | ignore );`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Sets the response to SRV records that refer to CNAMEs.

If *check-integrity* is set, *named* fails, warns, or ignores SRV records that refer to CNAMEs. The default is to `warn`.

**check-sibling**

**Grammar:** `check-sibling <boolean>;`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Specifies whether to check for sibling glue when performing integrity checks.

This option instructs BIND to also check that sibling glue exists, when performing integrity checks. The default is `yes`.

**check-spf**

**Grammar:** `check-spf ( warn | ignore );`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Specifies whether to check for a TXT Sender Policy Framework record, if an SPF record is present.

If *check-integrity* is set, *named* checks whether there is a TXT Sender Policy Framework record present (starts with "v=spf1"), if there is an SPF record present. The default is `warn`.

**check-svcb**

**Grammar:** `check-svcb <boolean>;`

**Blocks:** options, view, zone (primary)

**Tags:** zone

Specifies whether to perform additional checks on SVCB records.

If yes, *named* checks that SVCB records that start with a _dns label prefixed by an optional _<port> label (e.g. _443._dns.ns1.example) have an alpn parameter, and that the dohpath parameter exists when the alpn indicates that it should be present. The default is yes.

**zero-no-soa-ttl**

**Grammar:** zero-no-soa-ttl <boolean>;

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** zone, query, server

Specifies whether to set the time to live (TTL) of the SOA record to zero, when returning authoritative negative responses to SOA queries.

If yes, when returning authoritative negative responses to SOA queries, *named* sets the TTL of the SOA record returned in the authority section to zero. The default is yes.

**zero-no-soa-ttl-cache**

**Grammar:** zero-no-soa-ttl-cache <boolean>;

**Blocks:** options, view

**Tags:** zone, query, server

Sets the time to live (TTL) to zero when caching a negative response to an SOA query.

If yes, this option instructs BIND to set the TTL to zero when caching a negative response to an SOA query. The default is no.

**update-check-ksk**

**Grammar:** update-check-ksk <boolean>; // obsolete

**Blocks:** options, view, zone (primary, secondary)

**Tags:** obsolete

This option no longer has any effect.

**dnssec-dnskey-kskonly**

**Grammar:** dnssec-dnskey-kskonly <boolean>; // obsolete

**Blocks:** options, view, zone (primary, secondary)

**Tags:** obsolete

This option no longer has any effect.

**try-tcp-refresh**

**Grammar:** try-tcp-refresh <boolean>;

**Blocks:** options, view, zone (mirror, secondary)

**Tags:** transfer

Specifies that BIND 9 should attempt to refresh a zone using TCP if UDP queries fail.

If yes, BIND tries to refresh the zone using TCP if UDP queries fail. The default is yes.

**dnssec-secure-to-insecure**

**Grammar:** `dnssec-secure-to-insecure <boolean>; // obsolete`

**Blocks:** options, view, zone (primary)

**Tags:** obsolete

This option no longer has any effect.

**synth-from-dnssec**

**Grammar:** `synth-from-dnssec <boolean>;`

**Blocks:** options, view

**Tags:** dnssec

Enables support for **RFC 8198**, Aggressive Use of DNSSEC-Validated Cache.

This option enables support for **RFC 8198**, Aggressive Use of DNSSEC-Validated Cache. It allows the resolver to send a smaller number of queries when resolving queries for DNSSEC-signed domains by synthesizing answers from cached NSEC and other RRsets that have been proved to be correct using DNSSEC. The default is `yes`.

> **Note**
>
> DNSSEC validation must be enabled for this option to be effective. This initial implementation only covers synthesis of answers from NSEC records; synthesis from NSEC3 is planned for the future. This will also be controlled by *synth-from-dnssec*.

### Forwarding

The forwarding facility sends queries which cannot be answered using local data to different resolvers. This can be used to create a large site-wide cache on a few servers, reducing traffic over links to external name servers. It can also be used to allow queries by servers that do not have direct access to the Internet, but that wish to look up exterior names anyway. Forwarding occurs only on those queries for which the server is not authoritative and does not have the answer in its cache.

**forward**

**Grammar:** `forward ( first | only );`

**Blocks:** options, view, zone (forward, primary, secondary, static-stub, stub)

**Tags:** query

Allows or disallows fallback to recursion if forwarding has failed; it is always used in conjunction with the *forwarders* statement.

This option is only meaningful if the *forwarders* list is not empty. A value of `first` is the default and causes the server to query the forwarders first; if that does not answer the question, the server then looks for the answer itself. If `only` is specified, the server only queries the forwarders.

**forwarders**

**Grammar:** `forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> | <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };`

**Blocks:** options, view, zone (forward, primary, secondary, static-stub, stub)

**Tags:** query

Defines one or more resolvers to which queries are forwarded.

This specifies a list of IP addresses of DNS resolvers, to which queries which cannot be answered using locally available data are forwarded. The default is the empty list (no forwarding). Each address in the list can be associated with an optional port number and a TLS transport. A default port number and a TLS transport can be set for the entire list.

If a TLS configuration is specified, *named* uses DNS-over-TLS (DoT) connections when connecting to the specified IP address(es), via the TLS configuration referenced by the *tls* statement.

Forwarding can also be configured on a per-domain basis, allowing for the global forwarding options to be overridden in a variety of ways. Particular domains can be set to use different forwarders, or have a different `forward only/first` behavior, or not forward at all; see *zone*.

### Dual-stack Servers

Dual-stack servers are used as servers of last resort, to work around problems in reachability due to the lack of support for either IPv4 or IPv6 on the host machine.

**`dual-stack-servers`**

**Grammar:** `dual-stack-servers [ port <integer> ] { ( <quoted_string> [ port <integer> ] | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ); ... };`

**Blocks:** options, view

**Tags:** server

Specifies host names or addresses of machines with access to both IPv4 and IPv6 transports.

This specifies host names or addresses of machines with access to both IPv4 and IPv6 transports. If a hostname is used, the server must be able to resolve the name using only the transport it has. If the machine is dual-stacked, the *dual-stack-servers* parameter has no effect unless access to a transport has been disabled on the command line (e.g., *named -4*).

### Access Control

Access to the server can be restricted based on the IP address of the requesting system. See *Address Match Lists* for details on how to specify IP address lists.

**`allow-notify`**

**Grammar:** `allow-notify { <address_match_element>; ... };`

**Blocks:** options, view, zone (mirror, secondary)

**Tags:** transfer

Defines an *address_match_list* that is allowed to send `NOTIFY` messages for the zone, in addition to addresses defined in the *primaries* option for the zone.

This ACL specifies which hosts may send NOTIFY messages to inform this server of changes to zones for which it is acting as a secondary server. This is only applicable for secondary zones (i.e., *type secondary* or `slave`).

If this option is set in *view* or *options*, it is globally applied to all secondary zones. If set in the *zone* statement, the global value is overridden.

If not specified, the default is to process NOTIFY messages only from the configured *primaries* for the zone. *allow-notify* can be used to expand the list of permitted hosts, not to reduce it.

**allow-proxy**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `allow-proxy { <address_match_element>; ... }; // experimental`

**Blocks:** options, view

**Tags:** server

Defines an *address_match_list* for the client addresses allowed to send PROXYv2 headers.

The default *address_match_list* is *none*, which means that no client is allowed to do that by default for security reasons, as the PROXYv2 protocol provides an easy way to spoof both source and destination addresses.

This *address_match_list* is primarily meant to have addresses and subnets of the proxies that are allowed to send PROXYv2 headers to BIND. In most cases, we do not recommend setting this *address_match_list* to be very permissive; in particular, we recommend against setting it to *any*, especially in cases when PROXYv2 headers can be accepted on publicly available networking interfaces.

The specified option is the only option that matches against real peer addresses when PROXYv2 headers are used. Most of the options that work with peer addresses use the ones extracted from PROXYv2 headers.

See also: *allow-proxy-on*.

**allow-proxy-on**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `allow-proxy-on { <address_match_element>; ... }; // experimental`

**Blocks:** options, view

**Tags:** server

Defines an *address_match_list* for the interface addresses allowed to accept PROXYv2 headers. The option is mostly intended for multi-homed configurations.

The default *address_match_list* is *any*, which means that accepting PROXYv2 is allowed on any interface.

The option is useful in cases when a user needs to have precise control over which interfaces allow PROXYv2, as it is the only option that matches against real interface addresses when PROXYv2 headers are used. Most options that work with interface addresses use the ones extracted from PROXYv2 headers.

It may be desirable to first set *allow-proxy*.

**allow-query**

**Grammar:** `allow-query { <address_match_element>; ... };`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** query

Specifies which hosts (an IP address list) are allowed to send queries to this resolver.

*allow-query* may also be specified in the *zone* statement, in which case it overrides the `options allow-query` statement. If not specified, the default is to allow queries from all hosts.

> **Note**
>
> *allow-query-cache* is used to specify access to the cache.

**allow-query-on**

**Grammar:** `allow-query-on { <address_match_element>; ... };`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** query

Specifies which local addresses (an IP address list) are allowed to send queries to this resolver. This option is used in multi-homed configurations.

This makes it possible, for instance, to allow queries on internal-facing interfaces but disallow them on external-facing ones, without necessarily knowing the internal network's addresses.

Note that *allow-query-on* is only checked for queries that are permitted by *allow-query*. A query must be allowed by both ACLs, or it is refused.

*allow-query-on* may also be specified in the *zone* statement, in which case it overrides the `options allow-query-on` statement.

If not specified, the default is to allow queries on all addresses.

> **Note**
>
> *allow-query-cache* is used to specify access to the cache.

**allow-query-cache**

**Grammar:** `allow-query-cache { <address_match_element>; ... };`

**Blocks:** options, view

**Tags:** query

Specifies which hosts (an IP address list) can access this server's cache and thus effectively controls recursion.

This option defines an *address_match_list* of IP address(es) which are allowed to issue queries that access the local cache. Without access to the local cache, recursive queries are effectively useless so, in effect, this statement (or its default) controls recursive behavior. This statement's default setting depends on:

1. If *recursion no;* present, it defaults to `allow-query-cache {none;};`. No local cache access permitted.
2. If *recursion yes;* (default), then, if *allow-recursion* is present, it defaults to the value of *allow-recursion*. Local cache access is permitted to the same *address_match_list* as *allow-recursion*.

3. If *recursion yes;* (default), then, if *allow-recursion* is **not** present, it defaults to allow-query-cache {localnets; localhost;};. Local cache access is permitted to *address_match_list* localnets and localhost IP addresses only.

**allow-query-cache-on**

**Grammar:** allow-query-cache-on { <address_match_element>; ... };

**Blocks:** options, view

**Tags:** query

Specifies which hosts (from an IP address list) can access this server's cache. It is used on servers with multiple interfaces.

This specifies which local addresses can send answers from the cache. If *allow-query-cache-on* is not set, then *allow-recursion-on* is used if set. Otherwise, the default is to allow cache responses to be sent from any address. Note: both *allow-query-cache* and *allow-query-cache-on* must be satisfied before a cache response can be sent; a client that is blocked by one cannot be allowed by the other.

**allow-recursion**

**Grammar:** allow-recursion { <address_match_element>; ... };

**Blocks:** options, view

**Tags:** query

Defines an *address_match_list* of clients that are allowed to perform recursive queries.

This specifies which hosts are allowed to make recursive queries through this server. BIND checks to see if the following parameters are set, in order: *allow-query-cache* and *allow-query*. If neither of those parameters is set, the default (localnets; localhost;) is used.

**allow-recursion-on**

**Grammar:** allow-recursion-on { <address_match_element>; ... };

**Blocks:** options, view

**Tags:** query, server

Specifies which local addresses can accept recursive queries.

This specifies which local addresses can accept recursive queries. If *allow-recursion-on* is not set, then *allow-query-cache-on* is used if set; otherwise, the default is to allow recursive queries on all addresses. Any client permitted to send recursive queries can send them to any address on which *named* is listening. Note: both *allow-recursion* and *allow-recursion-on* must be satisfied before recursion is allowed; a client that is blocked by one cannot be allowed by the other.

**allow-update**

**Grammar:** allow-update { <address_match_element>; ... };

**Blocks:** options, view, zone (primary)

**Tags:** transfer

Defines an *address_match_list* of hosts that are allowed to submit dynamic updates for primary zones.

This provides a simple access control list. When set in the *zone* statement for a primary zone, this specifies which hosts are allowed to submit dynamic DNS updates to that zone. The default is to deny updates from all hosts.

Note that allowing updates based on the requestor's IP address is insecure; see *Dynamic Update Security* for details.

In general, this option should only be set at the *zone* level. While a default value can be set at the *options* or *view* level and inherited by zones, this could lead to some zones unintentionally allowing updates.

Updates are written to the zone's filename that is set in *file*.

**allow-update-forwarding**

**Grammar:** `allow-update-forwarding { <address_match_element>; ... };`

**Blocks:** options, view, zone (mirror, secondary)

**Tags:** transfer

Defines an *address_match_list* of hosts that are allowed to submit dynamic updates to a secondary server for transmission to a primary.

When set in the *zone* statement for a secondary zone, this specifies which hosts are allowed to submit dynamic DNS updates and have them be forwarded to the primary. The default is `{ none; }`, which means that no update forwarding is performed.

To enable update forwarding, specify `allow-update-forwarding { any; };` in the *zone* statement. Specifying values other than `{ none; }` or `{ any; }` is usually counterproductive; the responsibility for update access control should rest with the primary server, not the secondary.

Note that enabling the update forwarding feature on a secondary server may expose primary servers to attacks if they rely on insecure IP-address-based access control; see *Dynamic Update Security* for more details.

In general this option should only be set at the *zone* level. While a default value can be set at the *options* or *view* level and inherited by zones, this can lead to some zones unintentionally forwarding updates.

**allow-transfer**

**Grammar:** `allow-transfer [ port <integer> ] [ transport <string> ] { <address_match_element>; ... };`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Defines an *address_match_list* of hosts that are allowed to transfer the zone information from this server.

This specifies which hosts are allowed to receive zone transfers from the server. *allow-transfer* may also be specified in the *zone* statement, in which case it overrides the *allow-transfer* statement set in *options* or *view*.

Transport-level limitations can also be specified. In particular, zone transfers can be restricted to a specific port and/or DNS transport protocol by using the options *port* and `transport`. Either option can be specified; if both are used, both constraints must be satisfied in order for the transfer to be allowed. Zone transfers are currently only possible via the TCP and TLS transports.

For example: `allow-transfer port 853 transport tls { any; };` allows outgoing zone transfers to any host using the TLS transport over port 853.

If *allow-transfer* is not specified, then the default is `none`; outgoing zone transfers are disabled.

> **Warning**
>
> Please note that incoming TLS connections are **not authenticated at the TLS level by default**. Please use *TSIG* to authenticate requestors or consider implementing *Mutual TLS* authentication.

**blackhole**

**Grammar:** `blackhole { <address_match_element>; ... };`

**Blocks:** options

**Tags:** query

Defines an *address_match_list* of hosts to ignore. The server will neither respond to queries from nor send queries to these addresses.

This specifies a list of addresses which the server does not accept queries from or or cannot use to resolve a query. Queries from these addresses are not responded to. The default is `none`.

When configuring this list, note that BIND evaluates Access Control Lists sequentially (first match wins). A common misconception is that the directive `!address;` blocks everything except that address. In reality, it only explicitly exempts `address` from the blackhole; all other IP addresses reach the end of the list without matching, meaning they are also not blackholed.

To successfully blackhole all traffic *except* specific addresses, you must explicitly catch the remaining traffic with `any;` at the end of the list. For example: `!address; any;`

**no-case-compress**

**Grammar:** `no-case-compress { <address_match_element>; ... };`

**Blocks:** options, view

**Tags:** server

Specifies a list of addresses that require case-insensitive compression in responses.

This specifies a list of addresses which require responses to use case-insensitive compression. This ACL can be used when *named* needs to work with clients that do not comply with the requirement in **RFC 1034** to use case-insensitive name comparisons when checking for matching domain names.

If left undefined, the ACL defaults to `none`: case-sensitive compression is used for all clients. If the ACL is defined and matches a client, case is ignored when compressing domain names in DNS responses sent to that client.

This can result in slightly smaller responses; if a response contains the names "example.com" and "example.COM", case-insensitive compression treats the second one as a duplicate. It also ensures that the case of the query name exactly matches the case of the owner names of returned records, rather than matches the case of the records entered in the zone file. This allows responses to exactly match the query, which is required by some clients due to incorrect use of case-sensitive comparisons.

Case-insensitive compression is *always* used in AXFR and IXFR responses, regardless of whether the client matches this ACL.

There are circumstances in which *named* does not preserve the case of owner names of records: if a zone file defines records of different types with the same name, but the capitalization of the name is different (e.g., "www.example.com/A" and "WWW.EXAMPLE.COM/AAAA"), then all responses for that name use the *first* version of the name that was used in the zone file. This limitation may be addressed in a future release. However, domain names specified in the rdata of resource records (i.e., records of type NS, MX, CNAME, etc.) always have their case preserved unless the client matches this ACL.

**resolver-query-timeout**

**Grammar:** `resolver-query-timeout <integer>;`

**Blocks:** options, view

**Tags:** query

Specifies the length of time, in milliseconds, that a resolver attempts to resolve a recursive query before failing.

This is the amount of time, in milliseconds, that the resolver spends attempting to resolve a recursive query before failing. The default is `10000`, the minimum is `301`, and the maximum is `30000`. Setting it to `0` results in the default being used.

This value was originally specified in seconds. Values less than or equal to 300 are treated as seconds and converted to milliseconds before applying the above limits.

### Interfaces

The interfaces, ports, and protocols that the server can use to answer queries may be specified using the *listen-on* and *listen-on-v6* options.

**listen-on**

**Grammar:** `listen-on [ port <integer> ] [ proxy <string> ] [ tls <string> ] [ http <string> ] { <address_match_element>; ... }; // may occur multiple times`

**Blocks:** options

**Tags:** server

Specifies the IPv4 addresses on which a server listens for DNS queries.

**listen-on-v6**

**Grammar:** `listen-on-v6 [ port <integer> ] [ proxy <string> ] [ tls <string> ] [ http <string> ] { <address_match_element>; ... }; // may occur multiple times`

**Blocks:** options

**Tags:** server

Specifies the IPv6 addresses on which a server listens for DNS queries.

The *listen-on* and *listen-on-v6* statements can each take an optional port, PROXYv2 support switch, TLS configuration identifier, and/or HTTP configuration identifier, in addition to an *address_match_list*.

The *address_match_list* in *listen-on* specifies the IPv4 addresses on which the server will listen. (IPv6 addresses are ignored, with a logged warning.) The server listens on all interfaces allowed by the address match list. If no *listen-on* is specified, the default is to listen for standard DNS queries on port 53 of all IPv4 interfaces.

*listen-on-v6* takes an *address_match_list* of IPv6 addresses. The server listens on all interfaces allowed by the address match list. If no *listen-on-v6* is specified, the default is to listen for standard DNS queries on port 53 of all IPv6 interfaces.

When specified, the PROXYv2 support switch `proxy` allows the enabling of PROXYv2 protocol support. The PROXYv2 protocol provides the means for passing connection information, such as a client's source and destination addresses and ports, across multiple layers of NAT or TCP/UDP proxies to back-end servers. The addresses passed by the PROXYv2 protocol are then used, instead of the peer and interface addresses provided by the operating system.

The `proxy` switch can have the following values:

- `plain` - accept plain PROXYv2 headers. This is the only valid option for transports that do not employ encryption. In the case of transports that employ encryption, this value instructs BIND that PROXYv2 headers are sent without encryption before the TLS handshake. In that case, only PROXYv2 headers are not encrypted.
- `encrypted` - accept encrypted PROXYv2 headers. This value instructs BIND that PROXYv2 headers are sent encrypted immediately after the TLS handshake. The option is valid only for transports that employ encryption; encrypted PROXYv2 headers cannot be sent via unencrypted transports.

Please consult the documentation of any proxying front-end software to decide which value should be used. If in doubt, use `plain` for encrypted transports, especially for DNS-over-HTTPS (DoH), but DNS-specific software is likely to need `encrypted`.

It should be noted that when PROXYv2 is enabled on a listener, it loses the ability to accept regular DNS queries without associated PROXYv2 headers.

In some cases, PROXYv2 headers might not contain usable source and destination addresses. In particular, this can happen when the headers use the `LOCAL` command, or headers use address types that are unspecified or unsupported by BIND. If otherwise correct, such headers are accepted by BIND and the real endpoint addresses are used in these cases.

The PROXYv2 protocol is designed to be extensible and can carry additional information in the form of type-length-values (TLVs). Many of the types are defined in the protocol specification, and for some of these, BIND does a reasonable amount of validation in order to detect and reject ill-formed or hand-crafted headers. Apart from that, this additional data, while accepted, is not currently used by BIND for anything else.

By default, no client is allowed to send queries that contain PROXYv2 protocol headers, even when support for the protocol is enabled in a *`listen-on`* statement. Users who are interested in enabling the PROXYv2 protocol support may also want to look at the *`allow-proxy`* and *`allow-proxy-on`* options, to adjust the corresponding ACLs.

If a TLS configuration is specified, *`named`* will listen for DNS-over-TLS (DoT) connections, using the key and certificate specified in the referenced *`tls`* statement. If the name `ephemeral` is used, an ephemeral key and certificate created for the currently running *`named`* process will be used.

If an HTTP configuration is specified, *`named`* listens for DNS-over-HTTPS (DoH) connections using the HTTP endpoint specified in the referenced *`http`* statement. If the name `default` is used, then *`named`* listens for connections at the default endpoint, `/dns-query`.

Use of an *`http`* specification requires *`tls`* to be specified as well. If an unencrypted connection is desired (for example, on load-sharing servers behind a reverse proxy), `tls none` may be used.

If a port number is not specified, the default is 53 for standard DNS, 853 for DNS over TLS, 443 for DNS over HTTPS, and 80 for DNS over HTTP (unencrypted). These defaults may be overridden using the *`port`*, *`tls-port`*, *`https-port`*, and *`http-port`* options.

Multiple *`listen-on`* statements are allowed. For example:

```
listen-on { 5.6.7.8; };
listen-on port 1234 { !1.2.3.4; 1.2/16; };
listen-on port 8853 tls ephemeral { 4.3.2.1; };
listen-on port 8453 tls ephemeral http myserver { 8.7.6.5; };
listen-on port 5300 proxy plain { !1.2.3.4; 1.2/16; };
listen-on port 8953 proxy encrypted tls ephemeral { 4.3.2.1; };
listen-on port 8553 proxy plain tls ephemeral http myserver { 8.7.6.5; };
```

The first two lines instruct the name server to listen for standard DNS queries on port 53 of the IP address 5.6.7.8 and on port 1234 of an address on the machine in net 1.2 that is not 1.2.3.4. The third line instructs the server to

listen for DNS-over-TLS connections on port 8853 of the IP address 4.3.2.1 using the ephemeral key and certifcate. The fourth line enables DNS-over-HTTPS connections on port 8453 of address 8.7.6.5, using the ephemeral key and certificate, and the HTTP endpoint or endpoints configured in an *http* statement with the name `myserver`.

Multiple *listen-on-v6* options can be used. For example:

```
listen-on-v6 { any; };
listen-on-v6 port 1234 { !2001:db8::/32; any; };
listen-on-v6 port 8853 tls example-tls { 2001:db8::100; };
listen-on-v6 port 8453 tls example-tls http default { 2001:db8::100; };
listen-on-v6 port 8000 tls none http myserver { 2001:db8::100; };
listen-on-v6 port 53000 proxy plain { !2001:db8::/32; any; };
listen-on-v6 port 8953 proxy encrypted tls example-tls { 2001:db8::100; };
listen-on-v6 port 8553 proxy plain tls example-tls http default { 2001:db8::100; }
↪;
```

The first two lines instruct the name server to listen for standard DNS queries on port 53 of any IPv6 addresses, and on port 1234 of IPv6 addresses that are not in the prefix 2001:db8::/32. The third line instructs the server to listen for for DNS-over-TLS connections on port 8853 of the address 2001:db8::100, using a TLS key and certificate specified in the a *tls* statement with the name `example-tls`. The fourth instructs the server to listen for DNS-over-HTTPS connections, again using `example-tls`, on the default HTTP endpoint. The fifth line, in which the *tls* parameter is set to `none`, instructs the server to listen for *unencrypted* DNS queries over HTTP at the endpoint specified in `myserver`..

To instruct the server not to listen on any IPv6 addresses, use:

```
listen-on-v6 { none; };
```

### Query Address

**`query-source`**

**Grammar options, view:** `query-source [ address ] ( <ipv4_address> | * | none );`

**Grammar server, view.server:** `query-source [ address ] ( <ipv4_address> | * );`

**Blocks:** options, server, view, view.server

**Tags:** query

Controls the IPv4 address from which queries are issued. If *none*, then no IPv4 address would be used to issue the query and therefore only IPv6 servers are queried.

**`query-source-v6`**

**Grammar options, view:** `query-source-v6 [ address ] ( <ipv6_address> | * | none );`

**Grammar server, view.server:** `query-source-v6 [ address ] ( <ipv6_address> | * );`

**Blocks:** options, server, view, view.server

**Tags:** query

Controls the IPv6 address from which queries are issued. If *none*, then no IPv6 address would be used to issue the query and therefore only IPv4 servers are quried.

If the server does not know the answer to a question, it queries other name servers. *query-source* specifies the address and port used for such queries. For queries sent over IPv6, there is a separate *query-source-v6* option. If `address` is `*` (asterisk) or is omitted, a wildcard IP address (`INADDR_ANY`) is used.

The defaults of the *query-source* and *query-source-v6* options are:

```
query-source address * port *;
query-source-v6 address * port *;
```

**Note**

`port` configuration is deprecated. A warning will be logged when this parameter is used.

**Note**

The address specified in the *query-source* option is used for both UDP and TCP queries, but the port applies only to UDP queries. TCP queries always use a random unprivileged port.

**use-v4-udp-ports**

**Warning**

This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `use-v4-udp-ports { <portrange>; ... }; // deprecated`

**Blocks:** options

**Tags:** deprecated

Specifies a list of ports that are valid sources for UDP/IPv4 messages.

**use-v6-udp-ports**

**Warning**

This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `use-v6-udp-ports { <portrange>; ... }; // deprecated`

**Blocks:** options

**Tags:** deprecated

Specifies a list of ports that are valid sources for UDP/IPv6 messages.

These statements, which are deprecated and will be removed in a future release, specify a list of IPv4 and IPv6 UDP ports that are used as source ports for UDP messages.

If *port* is `*` or is omitted, a random port number from a pre-configured range is selected and used for each query. The port range(s) are specified in the *use-v4-udp-ports* (for IPv4) and *use-v6-udp-ports* (for IPv6) options.

If *use-v4-udp-ports* or *use-v6-udp-ports* is unspecified, *named* checks whether the operating system provides a programming interface to retrieve the system's default range for ephemeral ports. If such an interface is available, *named* uses the corresponding system default range; otherwise, it uses its own defaults:

```
use-v4-udp-ports { range 1024 65535; };
use-v6-udp-ports { range 1024 65535; };
```

**avoid-v4-udp-ports**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `avoid-v4-udp-ports { <portrange>; ... }; // deprecated`

**Blocks:** options

**Tags:** deprecated

Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv4 messages.

**avoid-v6-udp-ports**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `avoid-v6-udp-ports { <portrange>; ... }; // deprecated`

**Blocks:** options

**Tags:** deprecated

Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv6 messages.

These statements, which are deprecated and will be removed in a future release, indicate ranges of port numbers to exclude from those specified in the *avoid-v4-udp-ports* and *avoid-v6-udp-ports* options, respectively.

The defaults of the *avoid-v4-udp-ports* and *avoid-v6-udp-ports* options are:

```
avoid-v4-udp-ports {};
avoid-v6-udp-ports {};
```

For example, with the following configuration:

```
use-v6-udp-ports { range 32768 65535; };
avoid-v6-udp-ports { 40000; range 50000 60000; };
```

UDP ports of IPv6 messages sent from *named* are in one of the following ranges: 32768 to 39999, 40001 to 49999, or 60001 to 65535.

*avoid-v4-udp-ports* and *avoid-v6-udp-ports* can be used to prevent *named* from choosing as its random source port a port that is blocked by a firewall or that is used by other applications; if a query went out with a source port blocked by a firewall, the answer would not pass through the firewall and the name server would have to query again. Note: the desired range can also be represented only with *use-v4-udp-ports* and *use-v6-udp-ports*, and the `avoid-` options are redundant in that sense; they are provided for backward compatibility and to possibly simplify the port specification.

> **Note**
>
> Make sure the ranges are sufficiently large for security. A desirable size depends on several parameters, but we generally recommend it contain at least 16384 ports (14 bits of entropy). Note also that the system's default range when used may be too small for this purpose, and that the range may even be changed while *named* is running; the new range is automatically applied when *named* is reloaded. Explicit configuration of *use-v4-udp-ports* and *use-v6-udp-ports* is encouraged, so that the ranges are sufficiently large and are reasonably independent from the ranges used by other applications.

> **Note**
>
> The operational configuration where *named* runs may prohibit the use of some ports. For example, Unix systems do not allow *named*, if run without root privilege, to use ports less than 1024. If such ports are included in the specified (or detected) set of query ports, the corresponding query attempts will fail, resulting in resolution failures or delay. It is therefore important to configure the set of ports that can be safely used in the expected operational environment.

> **Warning**
>
> Specifying a single port is discouraged, as it removes a layer of protection against spoofing errors.

> **Warning**
>
> The configured *port* must not be the same as the listening port.

> **Note**
>
> See also *transfer-source*, *notify-source* and *parental-source*.

### Zone Transfers

BIND has mechanisms in place to facilitate zone transfers and set limits on the amount of load that transfers place on the system. The following options apply to zone transfers.

**also-notify**

**Grammar:** `also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Defines one or more hosts that are sent `NOTIFY` messages when zone changes occur.

This option defines a global list of IP addresses of name servers that are also sent NOTIFY messages whenever a fresh copy of the zone is loaded, in addition to the servers listed in the zone's NS records. This helps to ensure that copies of the zones quickly converge on stealth servers. Optionally, a port may be specified with each

*also-notify* address to send the notify messages to a port other than the default of 53. An optional TSIG key can also be specified with each address to cause the notify messages to be signed; this can be useful when sending notifies to multiple views. In place of explicit addresses, one or more named *primaries* lists can be used.

If an *also-notify* list is given in a *zone* statement, it overrides the `options also-notify` statement. When a `zone notify` statement is set to `no`, the IP addresses in the global *also-notify* list are not sent NOTIFY messages for that zone. The default is the empty list (no global notification list).

**min-transfer-rate-in**

**Grammar:** `min-transfer-rate-in <integer> <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Specifies the minimum traffic rate below which inbound zone transfers are terminated.

Inbound zone transfers running slower than the given amount of bytes in the given amount of minutes are terminated. This option takes two non-zero integer values. A check is performed periodically every time the configured time interval passes. The default value is `10240 5`, i.e. 10240 bytes in 5 minutes. The maximum time value is 28 days (40320 minutes).

**max-transfer-time-in**

**Grammar:** `max-transfer-time-in <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Specifies the number of minutes after which inbound zone transfers are terminated.

Inbound zone transfers running longer than this many minutes are terminated. The default is 120 minutes (2 hours). The maximum value is 28 days (40320 minutes).

**max-transfer-idle-in**

**Grammar:** `max-transfer-idle-in <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Specifies the number of minutes after which inbound zone transfers making no progress are terminated.

Inbound zone transfers making no progress in this many minutes are terminated. The default is 60 minutes (1 hour). The maximum value is 28 days (40320 minutes).

> **Note**
>
> Inbound zone transfers are also affected by `tcp-idle-timeout`; `max-transfer-idle-in` closes the inbound zone transfer if there is no complete AXFR or no complete IXFR chunk. `tcp-idle-timeout` closes the connection if there is no progress on the TCP level.

**max-transfer-time-out**

**Grammar:** `max-transfer-time-out <integer>;`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Specifies the number of minutes after which outbound zone transfers are terminated.

Outbound zone transfers running longer than this many minutes are terminated. The default is 120 minutes (2 hours). The maximum value is 28 days (40320 minutes).

**max-transfer-idle-out**

**Grammar:** `max-transfer-idle-out <integer>;`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Specifies the number of minutes after which outbound zone transfers making no progress are terminated.

Outbound zone transfers making no progress in this many minutes are terminated. The default is 60 minutes (1 hour). The maximum value is 28 days (40320 minutes).

**notify-rate**

**Grammar:** `notify-rate <integer>;`

**Blocks:** options

**Tags:** transfer, zone

Specifies the rate at which NOTIFY requests are sent during normal zone maintenance operations.

This specifies the rate at which NOTIFY requests are sent during normal zone maintenance operations. (NOTIFY requests due to initial zone loading are subject to a separate rate limit; see below.) The default is 20 per second. The lowest possible rate is one per second; when set to zero, it is silently raised to one.

**primaries**

**Grammar:** `primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };`

**Blocks:** zone (mirror, redirect, secondary, stub)

**Tags:** transfer, zone

Defines one or more servers that zone transfer can be requested from.

This specifies a list of one or more IP addresses of primary servers that the secondary contacts to update its copy of the zone. Primaries list elements can also be names of *remote-servers* blocks.

By default, transfers are made from port 53 on the servers; this can be changed for all servers by specifying a port number before the list of IP addresses, or on a per-server basis after the IP address. Authentication to the primary can also be done with per-server TSIG keys.

**startup-notify-rate**

**Grammar:** `startup-notify-rate <integer>;`

**Blocks:** options

**Tags:** transfer, zone

Specifies the rate at which NOTIFY requests are sent when the name server is first starting, or when new zones have been added.

This is the rate at which NOTIFY requests are sent when the name server is first starting up, or when zones have been newly added to the name server. The default is 20 per second. The lowest possible rate is one per second; when set to zero, it is silently raised to one.

**serial-query-rate**

**Grammar:** `serial-query-rate <integer>;`

**Blocks:** options

**Tags:** transfer

Defines an upper limit on the number of queries per second issued by the server, when querying the SOA RRs used for zone transfers.

Secondary servers periodically query primary servers to find out if zone serial numbers have changed. Each such query uses a minute amount of the secondary server's network bandwidth. To limit the amount of bandwidth used, BIND 9 limits the rate at which queries are sent. The value of the *serial-query-rate* option, an integer, is the maximum number of queries sent per second. The default is 20 per second. The lowest possible rate is one per second; when set to zero, it is silently raised to one.

**transfer-format**

**Grammar:** `transfer-format ( many-answers | one-answer );`

**Blocks:** options, server, view, view.server

**Tags:** transfer

Controls whether multiple records can be packed into a message during zone transfers.

Zone transfers can be sent using two different formats, `one-answer` and `many-answers`. The *transfer-format* option is used on the primary server to determine which format it sends. `one-answer` uses one DNS message per resource record transferred. `many-answers` packs as many resource records as possible into one message. `many-answers` is more efficient; the default is `many-answers`. *transfer-format* may be overridden on a per-server basis by using the *server* block.

**transfer-message-size**

**Grammar:** `transfer-message-size <integer>;`

**Blocks:** options

**Tags:** transfer

Limits the uncompressed size of DNS messages used in zone transfers over TCP.

This is an upper bound on the uncompressed size of DNS messages used in zone transfers over TCP. If a message grows larger than this size, additional messages are used to complete the zone transfer. (Note, however, that this is a hint, not a hard limit; if a message contains a single resource record whose RDATA does not fit within the size limit, a larger message will be permitted so the record can be transferred.)

Valid values are between 512 and 65535 octets; any values outside that range are adjusted to the nearest value within it. The default is `20480`, which was selected to improve message compression; most DNS messages of this size will compress to less than 16536 bytes. Larger messages cannot be compressed as effectively, because 16536 is the largest permissible compression offset pointer in a DNS message.

This option is mainly intended for server testing; there is rarely any benefit in setting a value other than the default.

**transfers-in**

**Grammar:** `transfers-in <integer>;`

**Blocks:** options

**Tags:** transfer

Limits the number of concurrent inbound zone transfers.

This is the maximum number of inbound zone transfers that can run concurrently. The default value is `10`. Increasing *transfers-in* may speed up the convergence of secondary zones, but it also may increase the load on the local system.

**transfers-out**

**Grammar:** `transfers-out <integer>;`

**Blocks:** options

**Tags:** transfer

Limits the number of concurrent outbound zone transfers.

This is the maximum number of outbound zone transfers that can run concurrently. Zone transfer requests in excess of the limit are refused. The default value is `10`.

**transfers-per-ns**

**Grammar:** `transfers-per-ns <integer>;`

**Blocks:** options

**Tags:** transfer

Limits the number of concurrent inbound zone transfers from a remote server.

This is the maximum number of inbound zone transfers that can concurrently transfer from a given remote name server. The default value is `2`. Increasing *transfers-per-ns* may speed up the convergence of secondary zones, but it also may increase the load on the remote name server. *transfers-per-ns* may be overridden on a per-server basis by using the *transfers* phrase of the *server* statement.

**transfer-source**

**Grammar:** `transfer-source ( <ipv4_address> | * );`

**Blocks:** options, server, view, zone (mirror, secondary, stub), view.server

**Tags:** transfer

Defines which local IPv4 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server.

*transfer-source* determines which local address is bound to IPv4 TCP connections used to fetch zones transferred inbound by the server. It also determines the source IPv4 address, and optionally the UDP port, used for the refresh queries and forwarded dynamic updates. If not set, it defaults to a system-controlled value which is usually the address of the interface "closest to" the remote end. This address must appear in the remote end's *allow-transfer* option for the zone being transferred, if one is specified. This statement sets the *transfer-source* for all zones, but can be overridden on a per-view or per-zone basis by including a *transfer-source* statement within the *view* or *zone* block in the configuration file.

> **Note**
>
> `port` configuration is deprecated. A warning will be logged when this parameter is used.

> **Warning**
>
> Specifying a single port is discouraged, as it removes a layer of protection against spoofing errors.

> **Warning**
>
> The configured *port* must not be the same as the listening port.

**`transfer-source-v6`**

**Grammar:** `transfer-source-v6 ( <ipv6_address> | * );`

**Blocks:** options, server, view, zone (mirror, secondary, stub), view.server

**Tags:** transfer

Defines which local IPv6 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server.

This option is the same as *`transfer-source`*, except zone transfers are performed using IPv6.

**`notify-source`**

**Grammar:** `notify-source ( <ipv4_address> | * );`

**Blocks:** options, server, view, zone (mirror, primary, secondary), view.server

**Tags:** transfer

Defines the IPv4 address (and optional port) to be used for outgoing `NOTIFY` messages.

*`notify-source`* determines which local source address, and optionally UDP port, is used to send NOTIFY messages. This address must appear in the secondary server's *`primaries`* zone clause or in an *`allow-notify`* clause. This statement sets the *`notify-source`* for all zones, but can be overridden on a per-zone or per-view basis by including a *`notify-source`* statement within the *`zone`* or *`view`* block in the configuration file.

> **Note**
>
> `port` configuration is deprecated. A warning will be logged when this parameter is used.

> **Warning**
>
> Specifying a single port is discouraged, as it removes a layer of protection against spoofing errors.

> **Warning**
>
> The configured *port* must not be the same as the listening port.

**notify-source-v6**

**Grammar:** `notify-source-v6 ( <ipv6_address> | * );`

**Blocks:** options, server, view, zone (mirror, primary, secondary), view.server

**Tags:** transfer

Defines the IPv6 address (and optional port) to be used for outgoing `NOTIFY` messages.

This option acts like *`notify-source`*, but applies to `NOTIFY` messages sent to IPv6 addresses.

### Server Resource Limits

The following options set limits on the server's resource consumption that are enforced internally by the server rather than by the operating system.

**max-journal-size**

**Grammar:** `max-journal-size ( default | unlimited | <sizeval> );`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer

Controls the size of journal files.

This sets a maximum size for each journal file (see *The Journal File*), expressed in bytes or, if followed by an optional unit suffix ('k', 'm', or 'g'), in kilobytes, megabytes, or gigabytes. When the journal file approaches the specified size, some of the oldest transactions in the journal are automatically removed. The largest permitted value is 2 gigabytes. Very small values are rounded up to 4096 bytes. It is possible to specify `unlimited`, which also means 2 gigabytes. If the limit is set to `default` or left unset, the journal is allowed to grow up to twice as large as the zone. (There is little benefit in storing larger journals.)

This option may also be set on a per-zone basis.

**max-records**

**Grammar:** `max-records <integer>;`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** zone, server

Sets the maximum number of records permitted in a zone.

This sets the maximum number of records permitted in a zone. The default is zero, which means the maximum is unlimited.

**max-records-per-type**

**Grammar:** `max-records-per-type <integer>;`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** server

Sets the maximum number of records that can be stored in an RRset.

This sets the maximum number of resource records that can be stored in an RRset in a database. When configured in *options* or *view*, it controls the cache database; it also sets the default value for zone databases, which can be overridden by setting it at the *zone* level.

If set to a positive value, any attempt to cache, or to add to a zone an RRset with more than the specified number of records, will result in a failure. If set to 0, there is no cap on RRset size. The default is 100.

**max-types-per-name**

**Grammar:** `max-types-per-name <integer>;`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** server

Sets the maximum number of RR types that can be stored for an owner name.

This sets the maximum number of resource record types that can be stored for a single owner name in a database. When configured in *options* or *view*, it controls the cache database and sets the default value for zone databases, which can be overridden by setting it at the *zone* level.

An RR type and its corresponding signature are counted as two types. So, for example, a signed node containing A and AAAA records has four types: A, RRSIG(A), AAAA, and RRSIG(AAAA).

The behavior is slightly different for zone and cache databases:

In a zone, if *max-types-per-name* is set to a positive number, any attempt to add a new resource record set to a name that already has the specified number of types will fail.

In a cache, if *max-types-per-name* is set to a positive number, an attempt to add a new resource record set to a name that already has the specified number of types will temporarily succeed, so that the query can be answered. However, the newly added RRset will immediately be purged.

Certain high-priority types, including SOA, CNAME, DNSKEY, and their corresponding signatures, are always cached. If *max-types-per-name* is set to a very low value, then it may be ignored to allow high-priority types to be cached.

When *max-types-per-name* is set to 0, there is no cap on the number of RR types. The default is 100.

**recursive-clients**

**Grammar:** `recursive-clients <integer>;`

**Blocks:** options

**Tags:** query

Specifies the maximum number of concurrent recursive queries the server can perform.

This sets the maximum number (a "hard quota") of simultaneous recursive lookups the server performs on behalf of clients. The default is `1000`. Because each recursing client uses a fair bit of memory (on the order of 20 kilobytes), the value of the *recursive-clients* option may have to be decreased on hosts with limited memory.

*recursive-clients* defines a "hard quota" limit for pending recursive clients; when more clients than this are pending, new incoming requests are not accepted, and for each incoming request a previous pending request is dropped.

A "soft quota" is also set. When this lower quota is exceeded, incoming requests are accepted, but for each one, a pending request is dropped. If *recursive-clients* is greater than 1000, the soft quota is set to *recursive-clients* minus 100; otherwise it is set to 90% of *recursive-clients*.

**tcp-clients**

**Grammar:** `tcp-clients <integer>;`

**Blocks:** options

**Tags:** server

Specifies the maximum number of simultaneous client TCP connections accepted by the server.

This is the maximum number of simultaneous client TCP connections that the server accepts. The default is `150`.

**clients-per-query**

**Grammar:** `clients-per-query <integer>;`

**Blocks:** options, view

**Tags:** server

Sets the initial minimum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients.

This sets the initial value (minimum) number of simultaneous recursive clients for any given query (<qname,qtype,qclass>) that the server accepts before dropping additional clents. *named* attempts to self-tune this value and changes are logged. The default value is 10.

The chosen value should reflect how many queries come in for a given name in the time it takes to resolve that name.

**max-clients-per-query**

**Grammar:** `max-clients-per-query <integer>;`

**Blocks:** options, view

**Tags:** server

Sets the maximum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients.

This sets the maximum number of simultaneous recursive clients for any given query (<qname,qtype,qclass>) that the server accepts before dropping additional clients.

If the number of queries exceeds *clients-per-query*, *named* assumes that it is dealing with a non-responsive zone and drops additional queries. If it gets a response after dropping queries, it raises the estimate, up to a limit of *max-clients-per-query*. The estimate is then lowered after 20 minutes if it has remained unchanged.

If *max-clients-per-query* is set to zero, there is no upper bound, other than that imposed by *recursive-clients*. If the option is set to a lower value than *clients-per-query*, the value is adjusted to *clients-per-query*.

If *clients-per-query* is set to zero, *max-clients-per-query* no longer applies and there is no upper bound, other than that imposed by *recursive-clients*.

**max-validations-per-fetch**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `max-validations-per-fetch <integer>; // experimental`

**Blocks:** options, view

**Tags:** server

Sets the maximum number of DNSSEC validations that can happen in a single fetch.

This is an **experimental** setting that defines the maximum number of DNSSEC validations that can happen in a single resolver fetch. The default is 16.

**`max-validation-failures-per-fetch`**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `max-validation-failures-per-fetch <integer>; // experimental`

**Blocks:** options, view

**Tags:** server

Sets the maximum number of DNSSEC validation failures that can happen in a single fetch.

This is an **experimental** setting that defines the maximum number of DNSSEC validation failures that can happen in a single resolver fetch. The default is 1.

**`fetches-per-zone`**

**Grammar:** `fetches-per-zone <integer> [ ( drop | fail ) ];`

**Blocks:** options, view

**Tags:** server, query

Sets the maximum number of simultaneous iterative queries allowed to any one domain before the server blocks new queries for data in or beneath that zone.

This sets the maximum number of simultaneous iterative queries to any one domain that the server permits before blocking new queries for data in or beneath that zone. This value should reflect how many fetches would normally be sent to any one zone in the time it would take to resolve them. It should be smaller than *`recursive-clients`*.

When many clients simultaneously query for the same name and type, the clients are all attached to the same fetch, up to the *`max-clients-per-query`* limit, and only one iterative query is sent. However, when clients are simultaneously querying for *different* names or types, multiple queries are sent and *`max-clients-per-query`* is not effective as a limit.

Optionally, this value may be followed by the keyword `drop` or `fail`, indicating whether queries which exceed the fetch quota for a zone are dropped with no response, or answered with SERVFAIL. The default is `drop`.

If *fetches-per-zone* is set to zero, there is no limit on the number of fetches per query and no queries are dropped. The default is zero.

The current list of active fetches can be dumped by running *rndc recursing*. The list includes the number of active fetches for each domain and the number of queries that have been passed (allowed) or dropped (spilled) as a result of the *fetches-per-zone* limit. (Note: these counters are not cumulative over time; whenever the number of active fetches for a domain drops to zero, the counter for that domain is deleted, and the next time a fetch is sent to that domain, it is recreated with the counters set to zero.)

> **Note**
>
> Fetches generated automatically in the result of *prefetch* are exempt from this quota.

**fetches-per-server**

**Grammar:** `fetches-per-server <integer> [ ( drop | fail ) ];`

**Blocks:** options, view

**Tags:** server, query

Sets the maximum number of simultaneous iterative queries allowed to be sent by a server to an upstream name server before the server blocks additional queries.

This sets the maximum number of simultaneous iterative queries that the server allows to be sent to a single upstream name server before blocking additional queries. This value should reflect how many fetches would normally be sent to any one server in the time it would take to resolve them. It should be smaller than *recursive-clients*.

Optionally, this value may be followed by the keyword `drop` or `fail`, indicating whether queries are dropped with no response or answered with SERVFAIL, when all of the servers authoritative for a zone are found to have exceeded the per-server quota. The default is `fail`.

If *fetches-per-server* is set to zero, there is no limit on the number of fetches per query and no queries are dropped. The default is zero.

The *fetches-per-server* quota is dynamically adjusted in response to detected congestion. As queries are sent to a server and either are answered or time out, an exponentially weighted moving average is calculated of the ratio of timeouts to responses. If the current average timeout ratio rises above a "high" threshold, then *fetches-per-server* is reduced for that server. If the timeout ratio drops below a "low" threshold, then *fetches-per-server* is increased. The *fetch-quota-params* options can be used to adjust the parameters for this calculation.

> **Note**
>
> Fetches generated automatically in the result of *prefetch* are exempt from this quota, but they are included in the quota calculations.

**fetch-quota-params**

**Grammar:** `fetch-quota-params <integer> <fixedpoint> <fixedpoint> <fixedpoint>;`

**Blocks:** options, view

**Tags:** server, query

Sets the parameters for dynamic resizing of the *fetches-per-server* quota in response to detected congestion.

This sets the parameters to use for dynamic resizing of the *fetches-per-server* quota in response to detected congestion.

The first argument is an integer value indicating how frequently to recalculate the moving average of the ratio of timeouts to responses for each server. The default is 100, meaning that BIND recalculates the average ratio after every 100 queries have either been answered or timed out.

The remaining three arguments represent the "low" threshold (defaulting to a timeout ratio of 0.1), the "high" threshold (defaulting to a timeout ratio of 0.3), and the discount rate for the moving average (defaulting to 0.7). A higher discount rate causes recent events to weigh more heavily when calculating the moving average; a lower discount rate causes past events to weigh more heavily, smoothing out short-term blips in the timeout ratio. These arguments are all fixed-point numbers with precision of 1/100; at most two places after the decimal point are significant.

**max-cache-size**

**Grammar:** `max-cache-size ( default | unlimited | <sizeval> | <percentage> );`

**Blocks:** options, view

**Tags:** server

Sets the maximum amount of memory to use for an individual cache database and its associated metadata.

This sets the maximum amount of memory to use for an individual cache database and its associated metadata, in bytes or percentage of total physical memory. By default, each view has its own separate cache, which means the total amount of memory required for cache data is the sum of the cache database sizes for all views (unless the *attach-cache* option is used).

When the amount of data in a cache database reaches the configured limit, *named* starts purging non-expired records (following an LRU-based strategy).

The default size limit for each individual cache is:

- 90% of physical memory for views with *recursion* set to `yes` (the default), or
- 2 MB for views with *recursion* set to `no`.

Any positive value smaller than 2 MB is ignored and reset to 2 MB. The keyword `unlimited`, or the value `0`, places no limit on the cache size; records are then purged from the cache only when they expire (according to their TTLs).

> **Note**
>
> For configurations which define multiple views with separate caches and recursion enabled, it is recommended to set *max-cache-size* appropriately for each view, as using the default value of that option (90% of physical memory for each individual cache) may lead to memory exhaustion over time.

> **Note**
>
> *max-cache-size* does not work reliably for a maximum amount of memory of 100 MB or lower.

Upon startup and reconfiguration, caches with a limited size preallocate a small amount of memory (less than 1% of *max-cache-size* for a given view). This preallocation serves as an optimization to eliminate extra latency introduced by resizing internal cache structures.

On systems where detection of the amount of physical memory is not supported, percentage-based values fall back to `unlimited`. Note that the amount of physical memory available is only detected on startup, so *named* does not adjust the cache size limits if the amount of physical memory is changed at runtime.

**tcp-listen-queue**

**Grammar:** `tcp-listen-queue <integer>;`

**Blocks:** options

**Tags:** server

Sets the listen-queue depth.

This sets the listen-queue depth. The default and minimum is 10. If the kernel supports the accept filter "dataready", this also controls how many TCP connections are queued in kernel space waiting for some data before being passed to accept. Non-zero values less than 10 are silently raised. A value of 0 may also be used; on most platforms this sets the listen-queue length to a system-defined default value.

**tcp-initial-timeout**

**Grammar:** `tcp-initial-timeout <integer>;`

**Blocks:** options

**Tags:** server, query

Sets the amount of time (in milliseconds) that the server waits on a new TCP connection for the first message from the client.

This sets the amount of time, in units of 100 milliseconds, that the server waits on a new TCP connection for the first message from the client. The default is 300 (30 seconds), the minimum is 25 (2.5 seconds), and the maximum is 1200 (two minutes). Values above the maximum or below the minimum are adjusted with a logged warning. (Note: this value must be greater than the expected round-trip delay time; otherwise, no client will ever have enough time to submit a message.) This value can be updated at runtime by using *rndc tcp-timeouts*.

**tcp-idle-timeout**

**Grammar:** `tcp-idle-timeout <integer>;`

**Blocks:** options

**Tags:** query

Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is not in use.

This sets the amount of time, in units of 100 milliseconds, that the server waits on an idle TCP connection before closing it, when the client is not using the EDNS TCP keepalive option. The default is 300 (30 seconds), the maximum is 1200 (two minutes), and the minimum is 1 (one-tenth of a second). Values above the maximum or below the minimum are adjusted with a logged warning. See *tcp-keepalive-timeout* for clients using the EDNS TCP keepalive option. This value can be updated at runtime by using *rndc tcp-timeouts*.

**tcp-keepalive-timeout**

**Grammar:** `tcp-keepalive-timeout <integer>;`

**Blocks:** options

**Tags:** query

Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is in use.

This sets the amount of time, in units of 100 milliseconds, that the server waits on an idle TCP connection before closing it, when the client is using the EDNS TCP keepalive option. The default is 300 (30 seconds), the maximum is 65535 (about 1.8 hours), and the minimum is 1 (one-tenth of a second). Values above the maximum or below the minimum are adjusted with a logged warning. This value may be greater than *tcp-idle-timeout* because clients using the EDNS TCP keepalive option are expected to use TCP connections for more than one message. This value can be updated at runtime by using *rndc tcp-timeouts*.

**tcp-reuse-timeout**

**Grammar:** `tcp-reuse-timeout <integer>;`

**Blocks:** options

**Tags:** query

Sets the amount of time (in milliseconds) that an idle outgoing TCP connection is kept open for reuse.

This sets the amount of time, in units of 100 milliseconds, that an idle outgoing TCP or TLS connection opened by *named* (for example, to a forwarder or an authoritative server) is kept open after its last outstanding response has completed, so that it can be reused by a later query instead of being closed and reopened. The default is 50 (5 seconds), and the maximum is 1200 (two minutes). A value of 0 disables keeping idle outgoing TCP or TLS connections open for reuse; it does not affect sharing of a connection while queries are still outstanding. Values above the maximum are adjusted with a logged warning.

**tcp-advertised-timeout**

**Grammar:** `tcp-advertised-timeout <integer>;`

**Blocks:** options

**Tags:** query

Sets the timeout value (in milliseconds) that the server sends in responses containing the EDNS TCP keepalive option.

This sets the timeout value, in units of 100 milliseconds, that the server sends in responses containing the EDNS TCP keepalive option, which informs a client of the amount of time it may keep the session open. The default is 300 (30 seconds), the maximum is 65535 (about 1.8 hours), and the minimum is 0, which signals that the clients must close TCP connections immediately. Ordinarily this should be set to the same value as *tcp-keepalive-timeout*. This value can be updated at runtime by using *rndc tcp-timeouts*.

**update-quota**

**Grammar:** `update-quota <integer>;`

**Blocks:** options

**Tags:** server

Specifies the maximum number of concurrent DNS UPDATE messages that can be processed by the server.

This is the maximum number of simultaneous DNS UPDATE messages that the server will accept, for updating local authoritative zones or forwarding to a primary server. The default is `100`.

**sig0checks-quota**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `sig0checks-quota <integer>; // experimental`

**Blocks:** options

**Tags:** server

Specifies the maximum number of concurrent SIG(0) signature checks that can be processed by the server.

This is the maximum number of simultaneous SIG(0)-signed messages that the server accepts. If the quota is reached, then *named* answers with a status code of REFUSED. The value of `0` disables the quota. The default is `1`.

**sig0checks-quota-exempt**

> **Warning**
>
> This option is experimental and subject to change.

**Grammar:** `sig0checks-quota-exempt { <address_match_element>; ... }; // experimental`

**Blocks:** options

**Tags:** server

Exempts specific clients or client groups from SIG(0) signature checking quota.

DNS clients can be exempted from the SIG(0) signature checking quota with the *sig0checks-quota-exempt* clause, using their IP and/or network addresses. The default value is an empty list.

Example:

```
sig0checks-quota-exempt {
    10.0.0.0/8;
    2001:db8::100;
};
```

**sig0key-checks-limit**

**Grammar:** `sig0key-checks-limit <integer>;`

**Blocks:** options, view

**Tags:** server

Specifies the maximum number of SIG(0) keys to consider when trying to verify a message.

This is the maximum number of keys to consider for a SIG(0)-signed message when trying to verify it. *named* will parse the candidate keys and check whether their key tag and algorithm matches with the expected one before trying to verify the signature. If the limit is reached the message verification fails. The value of `0` disables the limitation. The default is `16`.

**sig0message-checks-limit**

**Grammar:** `sig0message-checks-limit <integer>;`

**Blocks:** options, view

**Tags:** server

Specifies the maximum number of matching SIG(0) keys to try to verify a message.

This is the maximum number of keys which (when correctly parsed and matched against the expected key tag and algorithm) *named* uses to verify a SIG(0)-signed message. If the limit is reached the message verification fails. The value of `0` disables the limitation. The default is `2`.

### Periodic Task Intervals

**heartbeat-interval**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `heartbeat-interval <integer>; // deprecated`

**Blocks:** options

**Tags:** deprecated

Sets the interval at which the server performs zone maintenance tasks for all zones marked as *dialup*.

The server performs zone maintenance tasks for all zones marked as *dialup* whenever this interval expires. The default is 60 minutes. Reasonable values are up to 1 day (1440 minutes). The maximum value is 28 days (40320 minutes). If set to 0, no zone maintenance for these zones occurs.

This option is deprecated and will be removed in a future release.

**interface-interval**

**Grammar:** `interface-interval <duration>;`

**Blocks:** options

**Tags:** server

Sets the interval at which the server scans the network interface list.

The server scans the network interface list on every interval as specified by *interface-interval*.

If set to 0, interface scanning only occurs when the configuration file is loaded, or when *automatic-interface-scan* is enabled and supported by the operating system. After the scan, the server begins listening for queries on any newly discovered interfaces (provided they are allowed by the *listen-on* configuration), and stops listening on interfaces that have gone away. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default is 60 minutes (1 hour); the maximum value is 28 days.

### The *sortlist* Statement

The response to a DNS query may consist of multiple resource records (RRs) forming a resource record set (RRset). The name server normally returns the RRs within the RRset in an indeterminate order (but see the *rrset-order* statement in *RRset Ordering*). The client resolver code should rearrange the RRs as appropriate: that is, using any addresses on the local net in preference to other addresses. However, not all resolvers can do this or are correctly configured. When a client is using a local server, the sorting can be performed in the server, based on the client's address. This only requires configuring the name servers, not all the clients.

**sortlist**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `sortlist { <address_match_element>; ... }; // deprecated`

**Blocks:** options, view

**Tags:** query, deprecated

Controls the ordering of RRs returned to the client, based on the client's IP address.

This option is deprecated and will be removed in a future release.

The *sortlist* statement (see below) takes an *address_match_list* and interprets it in a special way. Each top-level statement in the *sortlist* must itself be an explicit *address_match_list* with one or two elements. The first element (which may be an IP address, an IP prefix, an ACL name, or a nested *address_match_list*) of each top-level list is checked against the source address of the query until a match is found. When the addresses in the first element overlap, the first rule to match is selected.

Once the source address of the query has been matched, if the top-level statement contains only one element, the actual primitive element that matched the source address is used to select the address in the response to move to the beginning of the response. If the statement is a list of two elements, then the second element is interpreted as a topology preference list. Each top-level element is assigned a distance, and the address in the response with the minimum distance is moved to the beginning of the response.

In the following example, any queries received from any of the addresses of the host itself get responses preferring addresses on any of the locally connected networks. Next most preferred are addresses on the 192.168.1/24 network, and after that either the 192.168.2/24 or 192.168.3/24 network, with no preference shown between these two networks. Queries received from a host on the 192.168.1/24 network prefer other addresses on that network to the 192.168.2/24 and 192.168.3/24 networks. Queries received from a host on the 192.168.4/24 or the 192.168.5/24 network only prefer other addresses on their directly connected networks.

```
sortlist {
    // IF the local host
    // THEN first fit on the following nets
    { localhost;
    { localnets;
        192.168.1/24;
        { 192.168.2/24; 192.168.3/24; }; }; };
    // IF on class C 192.168.1 THEN use .1, or .2 or .3
    { 192.168.1/24;
    { 192.168.1/24;
        { 192.168.2/24; 192.168.3/24; }; }; };
```

(continues on next page)

(continued from previous page)

```
    // IF on class C 192.168.2 THEN use .2, or .1 or .3
    { 192.168.2/24;
    { 192.168.2/24;
        { 192.168.1/24; 192.168.3/24; }; }; };
    // IF on class C 192.168.3 THEN use .3, or .1 or .2
    { 192.168.3/24;
    { 192.168.3/24;
        { 192.168.1/24; 192.168.2/24; }; }; };
    // IF .4 or .5 THEN prefer that net
    { { 192.168.4/24; 192.168.5/24; };
    };
};
```

The following example illustrates reasonable behavior for the local host and hosts on directly connected networks. Responses sent to queries from the local host favor any of the directly connected networks. Responses sent to queries from any other hosts on a directly connected network prefer addresses on that same network. Responses to other queries are not sorted.

```
sortlist {
       { localhost; localnets; };
       { localnets; };
};
```

### RRset Ordering

> **Note**
>
> While alternating the order of records in a DNS response between subsequent queries is a known load distribution technique, certain caveats apply (mostly stemming from caching) which usually make it a suboptimal choice for load balancing purposes when used on its own.

**`rrset-order`**

**Grammar:** `rrset-order { [ class <string> ] [ type <string> ] [ name <quoted_string> ] <string> <string>; ... };`

**Blocks:** options, view

**Tags:** query

Defines the order in which equal RRs (RRsets) are returned.

The *`rrset-order`* statement permits configuration of the ordering of the records in a multiple-record response. See also: *`sortlist`*.

Each rule in an *`rrset-order`* statement is defined as follows:

```
[class <class_name>] [type <type_name>] [name "<domain_name>"] order <ordering>
```

The default qualifiers for each rule are:

- If no `class` is specified, the default is `ANY`.
- If no *`type`* is specified, the default is `ANY`.

- If no `name` is specified, the default is `*` (asterisk).

*<domain_name>* only matches the name itself, not any of its subdomains. To make a rule match all subdomains of a given name, a wildcard name (`*.<domain_name>`) must be used. Note that `*.<domain_name>` does *not* match `<domain_name>` itself; to specify RRset ordering for a name and all of its subdomains, two separate rules must be defined: one for `<domain_name>` and one for `*.<domain_name>`.

The legal values for `<ordering>` are:

**`fixed`**
Records are returned in the order they are defined in the zone file.

This value is deprecated and will be removed in a future release.

> **Note**
>
> The `fixed` option is only available if BIND is configured with `--enable-fixed-rrset` at compile time.

**`random`**
Records are returned in a non-deterministic order. The random ordering doesn't guarantee uniform distribution of all permutations.

**`cyclic`**
Records are returned in a cyclic round-robin order, rotating by one record per query.

**`none`**
Records are returned in the order they were retrieved from the database. This order is indeterminate, but remains consistent as long as the database is not modified.

The default RRset order used depends on whether any *`rrset-order`* statements are present in the configuration file used by *`named`*:

- If no *`rrset-order`* statement is present in the configuration file, the implicit default is to return all records in `random` order.
- If any *`rrset-order`* statements are present in the configuration file, but no ordering rule specified in these statements matches a given RRset, the default order for that RRset is `none`.

Note that if multiple *`rrset-order`* statements are present in the configuration file (at both the *`options`* and *`view`* levels), they are *not* combined; instead, the more-specific one (*`view`*) replaces the less-specific one (*`options`*).

If multiple rules within a single *`rrset-order`* statement match a given RRset, the first matching rule is applied.

Example:

```
rrset-order {
    type A name "foo.isc.org" order random;
    type AAAA name "foo.isc.org" order cyclic;
    name "bar.isc.org" order fixed;
    name "*.bar.isc.org" order random;
    name "*.baz.isc.org" order cyclic;
};
```

With the above configuration, the following RRset ordering is used:

| QNAME | QTYPE | RRset Order |
|---|---|---|
| foo.isc.org | A | random |
| foo.isc.org | AAAA | cyclic |
| foo.isc.org | TXT | none |
| sub.foo.isc.org | all | none |
| bar.isc.org | all | fixed |
| sub.bar.isc.org | all | random |
| baz.isc.org | all | none |
| sub.baz.isc.org | all | cyclic |

### Tuning

**lame-ttl**

**Grammar:** `lame-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Sets the resolver's lame cache.

This is always set to 0. More information is available in the security advisory for CVE-2021-25219.

**servfail-ttl**

**Grammar:** `servfail-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Sets the length of time (in seconds) that a SERVFAIL response is cached.

This sets the number of seconds to cache a SERVFAIL response due to DNSSEC validation failure or other general server failure. If set to `0`, SERVFAIL caching is disabled. The SERVFAIL cache is not consulted if a query has the CD (Checking Disabled) bit set; this allows a query that failed due to DNSSEC validation to be retried without waiting for the SERVFAIL TTL to expire.

The maximum value is `30` seconds; any higher value is silently reduced. The default is `1` second.

**min-ncache-ttl**

**Grammar:** `min-ncache-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Specifies the minimum retention time (in seconds) for storage of negative answers in the server's cache.

To reduce network traffic and increase performance, the server stores negative answers. *`min-ncache-ttl`* is used to set a minimum retention time for these answers in the server, in seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default *`min-ncache-ttl`* is `0` seconds. *`min-ncache-ttl`* cannot exceed 90 seconds and is truncated to 90 seconds if set to a greater value.

**min-cache-ttl**

**Grammar:** `min-cache-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Specifies the minimum time (in seconds) that the server caches ordinary (positive) answers.

This sets the minimum time for which the server caches ordinary (positive) answers, in seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default *min-cache-ttl* is `0` seconds. *min-cache-ttl* cannot exceed 90 seconds and is truncated to 90 seconds if set to a greater value.

**max-ncache-ttl**

**Grammar:** `max-ncache-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Specifies the maximum retention time (in seconds) for storage of negative answers in the server's cache.

To reduce network traffic and increase performance, the server stores negative answers. *max-ncache-ttl* is used to set a maximum retention time for these answers in the server, in seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default *max-ncache-ttl* is 10800 seconds (3 hours). *max-ncache-ttl* cannot exceed 7 days and is silently truncated to 7 days if set to a greater value.

**max-cache-ttl**

**Grammar:** `max-cache-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Specifies the maximum time (in seconds) that the server caches ordinary (positive) answers.

This sets the maximum time for which the server caches ordinary (positive) answers, in seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

The default *max-cache-ttl* is 604800 (one week). A value of zero may cause all queries to return SERVFAIL, because of lost caches of intermediate RRsets (such as NS and glue AAAA/A records) in the resolution process.

**max-stale-ttl**

**Grammar:** `max-stale-ttl <duration>;`

**Blocks:** options, view

**Tags:** server

Specifies the maximum time that the server retains records past their normal expiry, to return them as stale records.

If retaining stale RRsets in cache is enabled, and returning of stale cached answers is also enabled, *max-stale-ttl* sets the maximum time for which the server retains records past their normal expiry to return them as stale records,

when the servers for those records are not reachable. The default is 1 day. The minimum allowed is 1 second; a value of 0 is updated silently to 1 second.

For stale answers to be returned, the retaining of them in cache must be enabled via the configuration option *stale-cache-enable*, and returning cached answers must be enabled, either in the configuration file using the *stale-answer-enable* option or by calling *rndc serve-stale on*.

When *stale-cache-enable* is set to no, setting the *max-stale-ttl* has no effect; the value of *max-stale-ttl* is 0 in such a case.

**sig-validity-interval**

**Grammar:** sig-validity-interval <integer> [ <integer> ]; // obsolete

**Blocks:** options, view, zone (primary, secondary)

**Tags:** obsolete

This option no longer has any effect.

**dnskey-sig-validity**

**Grammar:** dnskey-sig-validity <integer>; // obsolete

**Blocks:** options, view, zone (primary, secondary)

**Tags:** obsolete

This option no longer has any effect.

**sig-signing-nodes**

**Grammar:** sig-signing-nodes <integer>;

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Specifies the maximum number of nodes to be examined in each quantum, when signing a zone with a new DNSKEY.

The default is 100.

**sig-signing-signatures**

**Grammar:** sig-signing-signatures <integer>;

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Specifies the threshold for the number of signatures that terminates processing a quantum, when signing a zone with a new DNSKEY.

The default is 10.

**sig-signing-type**

**Grammar:** sig-signing-type <integer>;

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Specifies a private RDATA type to use when generating signing-state records.

The default is `65534`.

This parameter may be removed in a future version, once there is a standard type.

Signing-state records are used internally by *named* to track the current state of a zone-signing process, i.e., whether it is still active or has been completed. The records can be inspected using the command *rndc signing -list zone*. Once *named* has finished signing a zone with a particular key, the signing-state record associated with that key can be removed from the zone by running *rndc signing -clear keyid/algorithm zone*. To clear all of the completed signing-state records for a zone, use *rndc signing -clear all zone*.

**min-refresh-time**

**Grammar:** `min-refresh-time <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Limits the zone refresh interval to no more often than the specified value, in seconds.

This option controls the server's behavior on refreshing a zone (querying for SOA changes). Usually, the SOA refresh values for the zone are used; however, these values are set by the primary, giving secondary server administrators little control over their contents.

This option allows the administrator to set a minimum refresh time in seconds per-zone, per-view, or globally. This option is valid for secondary and stub zones, and clamps the SOA refresh time to the specified value.

The default is 300 seconds.

**max-refresh-time**

**Grammar:** `max-refresh-time <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Limits the zone refresh interval to no less often than the specified value, in seconds.

This option controls the server's behavior on refreshing a zone (querying for SOA changes). Usually, the SOA refresh values for the zone are used; however, these values are set by the primary, giving secondary server administrators little control over their contents.

This option allows the administrator to set a maximum refresh time in seconds per-zone, per-view, or globally. This option is valid for secondary and stub zones, and clamps the SOA refresh time to the specified value.

The default is 2419200 seconds (4 weeks).

**min-retry-time**

**Grammar:** `min-retry-time <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Limits the zone refresh retry interval to no more often than the specified value, in seconds.

This option controls the server's behavior on retrying failed zone transfers. Usually, the SOA retry values for the zone are used; however, these values are set by the primary, giving secondary server administrators little control over their contents.

This option allows the administrator to set a minimum retry time in seconds per-zone, per-view, or globally. This option is valid for secondary and stub zones, and clamps the SOA retry time to the specified value.

The default is 500 seconds.

**max-retry-time**

**Grammar:** `max-retry-time <integer>;`

**Blocks:** options, view, zone (mirror, secondary, stub)

**Tags:** transfer

Limits the zone refresh retry interval to no less often than the specified value, in seconds.

This option controls the server's behavior on retrying failed zone transfers. Usually, the SOA retry values for the zone are used; however, these values are set by the primary, giving secondary server administrators little control over their contents.

This option allows the administrator to set a maximum retry time in seconds per-zone, per-view, or globally. This option is valid for secondary and stub zones, and clamps the SOA retry time to the specified value.

The default is 1209600 seconds (2 weeks).

**edns-udp-size**

**Grammar:** `edns-udp-size <integer>;`

**Blocks:** options, server, view, view.server

**Tags:** query

Sets the maximum advertised EDNS UDP buffer size to control the size of packets received from authoritative servers in response to recursive queries.

This sets the maximum advertised EDNS UDP buffer size, in bytes, to control the size of packets received from authoritative servers in response to recursive queries. Valid values are 512 to 4096; values outside this range are silently adjusted to the nearest value within it. The default value is 1232.

The usual reason for setting *edns-udp-size* to a non-default value is to get UDP answers to pass through broken firewalls that block fragmented packets and/or block UDP DNS packets that are greater than 512 bytes.

When *named* first queries a remote server, it advertises a UDP buffer size of 1232.

Query timeouts observed for any given server affect the buffer size advertised in queries sent to that server. Depending on observed packet dropping patterns, the query is retried over TCP. Per-server EDNS statistics are only retained in memory for the lifetime of a given server's ADB entry.

According to measurements taken by multiple parties, the default value should not be causing the fragmentation. As most of the Internet "core" is able to cope with IP message sizes between 1400-1500 bytes, the 1232 size was chosen as a conservative minimal number that could be changed by the DNS operator to a estimated path MTU, minus the estimated header space. In practice, the smallest MTU witnessed in the operational DNS community is 1500 octets, the Ethernet maximum payload size, so a useful default for the maximum DNS/UDP payload size on **reliable** networks would be 1432.

Any server-specific *edns-udp-size* setting has precedence over all the above rules, i.e. configures a static value for a given *server* block.

**max-udp-size**

**Grammar:** `max-udp-size <integer>;`

**Blocks:** options, server, view, view.server

**Tags:** query

Sets the maximum EDNS UDP message size sent by *named*.

This sets the maximum EDNS UDP message size that *named* sends, in bytes. Valid values are 512 to 4096; values outside this range are silently adjusted to the nearest value within it. The default value is 1232.

This value applies to responses sent by a server; to set the advertised buffer size in queries, see *edns-udp-size*.

The usual reason for setting *max-udp-size* to a non-default value is to allow UDP answers to pass through broken firewalls that block fragmented packets and/or block UDP packets that are greater than 512 bytes. This is independent of the advertised receive buffer (*edns-udp-size*).

Setting this to a low value encourages additional TCP traffic to the name server.

**masterfile-format**

**Grammar:** `masterfile-format ( raw | text );`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, stub)

**Tags:** zone, server

Specifies the file format of zone files.

This specifies the file format of zone files (see *Additional File Formats* for details). The default value is `text`, which is the standard textual representation, except for secondary zones, in which the default value is `raw`. Files in formats other than `text` are typically expected to be generated by the *named-compilezone* tool, or dumped by *named*.

Note that when a zone file in a format other than `text` is loaded, *named* may omit some of the checks which are performed for a file in `text` format. For example, *check-names* only applies when loading zones in `text` format. Zone files in `raw` format should be generated with the same check level as that specified in the *named* configuration file.

When configured in *options*, this statement sets the *masterfile-format* for all zones, but it can be overridden on a per-zone or per-view basis by including a *masterfile-format* statement within the *zone* or *view* block in the configuration file.

**masterfile-style**

**Grammar:** `masterfile-style ( full | relative );`

**Blocks:** options, view, zone (mirror, primary, redirect, secondary, stub)

**Tags:** server

Specifies the format of zone files during a dump, when the *masterfile-format* is `text`.

This specifies the formatting of zone files during dump, when the *masterfile-format* is `text`. This option is ignored with any other *masterfile-format*.

When set to `relative`, records are printed in a multi-line format, with owner names expressed relative to a shared origin. When set to `full`, records are printed in a single-line format with absolute owner names. The `full` format is most suitable when a zone file needs to be processed automatically by a script. The `relative` format is more human-readable, and is thus suitable when a zone is to be edited by hand. The default is `relative`.

**max-query-count**

**Grammar:** `max-query-count <integer>;`

**Blocks:** options, view

**Tags:** server, query

Sets the maximum number of iterative queries while servicing a recursive query.

This sets the maximum number of iterative queries that may be sent by a resolver while looking up a single name. If more queries than this need to be sent before an answer is reached, then recursion is terminated and a SERVFAIL response is returned to the client. The default is `200`.

**max-recursion-depth**

**Grammar:** `max-recursion-depth <integer>;`

**Blocks:** options, view

**Tags:** server

Sets the maximum number of levels of recursion permitted at any one time while servicing a recursive query.

This sets the maximum number of levels of recursion that are permitted at any one time while servicing a recursive query. Resolving a name may require looking up a name server address, which in turn requires resolving another name, etc.; if the number of recursions exceeds this value, the recursive query is terminated and returns SERVFAIL. The default is 7.

**max-recursion-queries**

**Grammar:** `max-recursion-queries <integer>;`

**Blocks:** options, view

**Tags:** server, query

Sets the maximum number of iterative queries while servicing a recursive query.

This sets the maximum number of iterative queries that may be sent by a resolver while looking up a single name. If more queries than this need to be sent before an answer is reached, then recursion is terminated and a SERVFAIL response is returned to the client. (Note: if the answer is a CNAME, then the subsequent lookup for the target of the CNAME is counted separately.) The default is 50.

**max-query-restarts**

**Grammar:** `max-query-restarts <integer>;`

**Blocks:** options, view

**Tags:** server, query

Sets the maximum number of chained CNAMEs to follow

This sets the maximum number of successive CNAME targets to follow when resolving a client query, before terminating the query to avoid a CNAME loop. Valid values are 1 to 255. The default is 11.

**notify-defer**

**Grammar:** `notify-defer <integer>;`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer, zone

Sets the defer time (in seconds) before sending NOTIFY messages for a zone.

This sets the delay, in seconds, to wait before sending a set of NOTIFY messages for a zone. Whenever a NOTIFY message is ready to be sent, sending will be deferred for this duration. This can be useful, for example, when for some operation needs a catalog zone is updated with new member zones before these member zones are actually ready to be tranferred. The delay can be tuned for the catalog zone to an amount of time after which the member zones are usually known to become ready. The default is 0 seconds.

**Warning**

This option is not to be confused with the *notify-delay* option.

**Note**

An implicit *rndc notify* command for a zone overrides the effects of this option.

**Note**

This options is ignored for notifies sent during the *dialup* process.

**notify-delay**

**Grammar:** `notify-delay <integer>;`

**Blocks:** options, view, zone (mirror, primary, secondary)

**Tags:** transfer, zone

Sets the delay (in seconds) between sending sets of NOTIFY messages for a zone.

This sets the delay, in seconds, between sending sets of NOTIFY messages for a zone. Whenever a NOTIFY message is sent for a zone, a timer will be set for this duration. If the zone is updated again before the timer expires, the NOTIFY for that update will be postponed. The default is 5 seconds.

The overall rate at which NOTIFY messages are sent for all zones is controlled by *notify-rate*.

**Warning**

This option is not to be confused with the *notify-defer* option.

**max-rsa-exponent-size**

**Grammar:** `max-rsa-exponent-size <integer>;`

**Blocks:** options

**Tags:** dnssec, query

Sets the maximum RSA exponent size (in bits) when validating.

This sets the maximum RSA exponent size, in bits, that is accepted when validating. Valid values are 35 to 4096 bits. The default, zero, is also accepted and is equivalent to 4096.

**prefetch**

**Grammar:** `prefetch <integer> [ <integer> ];`

**Blocks:** options, view

**Tags:** query

Specifies the "trigger" time-to-live (TTL) value at which prefetch of the current query takes place.

When a query is received for cached data which is to expire shortly, *named* can refresh the data from the authoritative server immediately, ensuring that the cache always has an answer available.

*prefetch* specifies the "trigger" TTL value at which prefetch of the current query takes place; when a cache record with an equal or lower TTL value is encountered during query processing, it is refreshed. Valid trigger TTL values are 1 to 10 seconds. Values larger than 10 seconds are silently reduced to 10. Setting a trigger TTL to zero causes prefetch to be disabled. The default trigger TTL is `2`.

An optional second argument specifies the "eligibility" TTL: the smallest *original* TTL value that is accepted for a record to be eligible for prefetching. The eligibility TTL must be at least six seconds longer than the trigger TTL; if not, *named* silently adjusts it upward. The default eligibility TTL is `9`.

**v6-bias**

**Grammar:** `v6-bias <integer>;`

**Blocks:** options, view

**Tags:** server, query

Indicates the number of milliseconds of preference to give to IPv6 name servers.

When determining the next name server to try, this indicates by how many milliseconds to prefer IPv6 name servers. The default is `50` milliseconds.

**tcp-receive-buffer**

**Grammar:** `tcp-receive-buffer <integer>;`

**Blocks:** options

**Tags:** server

Sets the operating system's receive buffer size for TCP sockets.

**udp-receive-buffer**

**Grammar:** `udp-receive-buffer <integer>;`

**Blocks:** options

**Tags:** server

Sets the operating system's receive buffer size for UDP sockets.

These options control the operating system's receive buffer sizes (`SO_RCVBUF`) for TCP and UDP sockets, respectively. Buffering at the operating system level can prevent packet drops during brief load spikes, but if the buffer size is set too high, a running server could get clogged with outstanding queries that have already timed out. The default is `0`, which means the operating system's default value should be used. The minimum configurable value is `4096`; any nonzero value lower than that is silently raised. The maximum value is determined by the kernel, and values exceeding the maximum are silently reduced.

**tcp-send-buffer**

**Grammar:** `tcp-send-buffer <integer>;`

**Blocks:** options

**Tags:** server

Sets the operating system's send buffer size for TCP sockets.

**udp-send-buffer**

**Grammar:** `udp-send-buffer <integer>;`

**Blocks:** options

**Tags:** server

Sets the operating system's send buffer size for UDP sockets.

These options control the operating system's send buffer sizes (`SO_SNDBUF`) for TCP and UDP sockets, respectively. Buffering at the operating system level can prevent packet drops during brief load spikes, but if the buffer size is set too high, a running server could get clogged with outstanding queries that have already timed out. The default is `0`, which means the operating system's default value should be used. The minimum configurable value is `4096`; any nonzero value lower than that is silently raised. The maximum value is determined by the kernel, and values exceeding the maximum are silently reduced.

### Built-in Server Information Zones

The server provides some helpful diagnostic information through a number of built-in zones under the pseudo-top-level-domain `bind` in the `CHAOS` class. These zones are part of a built-in view (see *view*) of class `CHAOS`, which is separate from the default view of class `IN`. Most global configuration options (*allow-query*, etc.) apply to this view, but some are locally overridden: *notify*, *recursion*, and *allow-new-zones* are always set to `no`, and *rate-limit* is set to allow three responses per second.

To disable these zones, use the options below or hide the built-in `CHAOS` view by defining an explicit view of class `CHAOS` that matches all clients.

**version**

**Grammar:** `version ( <quoted_string> | none );`

**Blocks:** options

**Tags:** server

Specifies the version number of the server to return in response to a `version.bind` query.

This is the version the server should report via a query of the name `version.bind` with type `TXT` and class `CHAOS`. The default is the real version number of this server. Specifying `version none` disables processing of the queries.

Setting *version* to any value (including `none`) also disables queries for `authors.bind TXT CH`.

**hostname**

**Grammar:** `hostname ( <quoted_string> | none );`

**Blocks:** options

**Tags:** server

Specifies the hostname of the server to return in response to a `hostname.bind` query.

This is the hostname the server should report via a query of the name `hostname.bind` with type `TXT` and class `CHAOS`. This defaults to the hostname of the machine hosting the name server, as found by the `gethostname()` function. The primary purpose of such queries is to identify which of a group of anycast servers is actually answering the queries. Specifying `hostname none;` disables processing of the queries.

**server-id**

**Grammar:** `server-id ( <quoted_string> | none | hostname );`

**Blocks:** options

**Tags:** server

Specifies the ID of the server to return in response to a `ID.SERVER` query.

This is the ID the server should report when receiving a Name Server Identifier (NSID) query, or a query of the name `ID.SERVER` with type `TXT` and class `CHAOS`. The primary purpose of such queries is to identify which of a group of anycast servers is actually answering the queries. Specifying `server-id none;` disables processing of the queries. Specifying `server-id hostname;` causes *named* to use the hostname as found by the `gethostname()` function. The default *server-id* is `none`.

### Built-in Empty Zones

The *named* server has some built-in empty zones, for SOA and NS records only. These are for zones that should normally be answered locally and for which queries should not be sent to the Internet's root servers. The official servers that cover these namespaces return NXDOMAIN responses to these queries. In particular, these cover the reverse namespaces for addresses from **RFC 1918**, **RFC 4193**, **RFC 5737**, and **RFC 6598**. They also include the reverse namespace for the IPv6 local address (locally assigned), IPv6 link local addresses, the IPv6 loopback address, and the IPv6 unknown address.

The server attempts to determine whether a built-in zone already exists or is active (covered by a forward-only forwarding declaration), and does not create an empty zone if either is true.

The current list of empty zones is:

- 10.IN-ADDR.ARPA
- 16.172.IN-ADDR.ARPA
- 17.172.IN-ADDR.ARPA
- 18.172.IN-ADDR.ARPA
- 19.172.IN-ADDR.ARPA
- 20.172.IN-ADDR.ARPA
- 21.172.IN-ADDR.ARPA
- 22.172.IN-ADDR.ARPA
- 23.172.IN-ADDR.ARPA
- 24.172.IN-ADDR.ARPA

- 25.172.IN-ADDR.ARPA
- 26.172.IN-ADDR.ARPA
- 27.172.IN-ADDR.ARPA
- 28.172.IN-ADDR.ARPA
- 29.172.IN-ADDR.ARPA
- 30.172.IN-ADDR.ARPA
- 31.172.IN-ADDR.ARPA
- 168.192.IN-ADDR.ARPA
- 64.100.IN-ADDR.ARPA
- 65.100.IN-ADDR.ARPA
- 66.100.IN-ADDR.ARPA
- 67.100.IN-ADDR.ARPA
- 68.100.IN-ADDR.ARPA
- 69.100.IN-ADDR.ARPA
- 70.100.IN-ADDR.ARPA
- 71.100.IN-ADDR.ARPA
- 72.100.IN-ADDR.ARPA
- 73.100.IN-ADDR.ARPA
- 74.100.IN-ADDR.ARPA
- 75.100.IN-ADDR.ARPA
- 76.100.IN-ADDR.ARPA
- 77.100.IN-ADDR.ARPA
- 78.100.IN-ADDR.ARPA
- 79.100.IN-ADDR.ARPA
- 80.100.IN-ADDR.ARPA
- 81.100.IN-ADDR.ARPA
- 82.100.IN-ADDR.ARPA
- 83.100.IN-ADDR.ARPA
- 84.100.IN-ADDR.ARPA
- 85.100.IN-ADDR.ARPA
- 86.100.IN-ADDR.ARPA
- 87.100.IN-ADDR.ARPA
- 88.100.IN-ADDR.ARPA
- 89.100.IN-ADDR.ARPA
- 90.100.IN-ADDR.ARPA
- 91.100.IN-ADDR.ARPA

- 92.100.IN-ADDR.ARPA
- 93.100.IN-ADDR.ARPA
- 94.100.IN-ADDR.ARPA
- 95.100.IN-ADDR.ARPA
- 96.100.IN-ADDR.ARPA
- 97.100.IN-ADDR.ARPA
- 98.100.IN-ADDR.ARPA
- 99.100.IN-ADDR.ARPA
- 100.100.IN-ADDR.ARPA
- 101.100.IN-ADDR.ARPA
- 102.100.IN-ADDR.ARPA
- 103.100.IN-ADDR.ARPA
- 104.100.IN-ADDR.ARPA
- 105.100.IN-ADDR.ARPA
- 106.100.IN-ADDR.ARPA
- 107.100.IN-ADDR.ARPA
- 108.100.IN-ADDR.ARPA
- 109.100.IN-ADDR.ARPA
- 110.100.IN-ADDR.ARPA
- 111.100.IN-ADDR.ARPA
- 112.100.IN-ADDR.ARPA
- 113.100.IN-ADDR.ARPA
- 114.100.IN-ADDR.ARPA
- 115.100.IN-ADDR.ARPA
- 116.100.IN-ADDR.ARPA
- 117.100.IN-ADDR.ARPA
- 118.100.IN-ADDR.ARPA
- 119.100.IN-ADDR.ARPA
- 120.100.IN-ADDR.ARPA
- 121.100.IN-ADDR.ARPA
- 122.100.IN-ADDR.ARPA
- 123.100.IN-ADDR.ARPA
- 124.100.IN-ADDR.ARPA
- 125.100.IN-ADDR.ARPA
- 126.100.IN-ADDR.ARPA
- 127.100.IN-ADDR.ARPA

- 0.IN-ADDR.ARPA
- 127.IN-ADDR.ARPA
- 254.169.IN-ADDR.ARPA
- 2.0.192.IN-ADDR.ARPA
- 100.51.198.IN-ADDR.ARPA
- 113.0.203.IN-ADDR.ARPA
- 255.255.255.255.IN-ADDR.ARPA
- 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
- 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
- 8.B.D.0.1.0.0.2.IP6.ARPA
- D.F.IP6.ARPA
- 8.E.F.IP6.ARPA
- 9.E.F.IP6.ARPA
- A.E.F.IP6.ARPA
- B.E.F.IP6.ARPA
- EMPTY.AS112.ARPA
- HOME.ARPA
- RESOLVER.ARPA

Empty zones can be set at the view level and only apply to views of class IN. Disabled empty zones are only inherited from options if there are no disabled empty zones specified at the view level. To override the options list of disabled zones, disable the root zone at the view level. For example:

```
disable-empty-zone ".";
```

If using the address ranges covered here, reverse zones covering the addresses should already be in place. In practice this appears to not be the case, with many queries being made to the infrastructure servers for names in these spaces. So many, in fact, that sacrificial servers had to be deployed to channel the query load away from the infrastructure servers.

**Note**

The real parent servers for these zones should disable all empty zones under the parent zone they serve. For the real root servers, this is all built-in empty zones. This enables them to return referrals to deeper in the tree.

**empty-server**

**Grammar:** `empty-server <string>;`

**Blocks:** options, view

**Tags:** server, zone

Specifies the server name in the returned SOA record for empty zones.

This specifies the server name that appears in the returned SOA record for empty zones. If none is specified, the zone's name is used.

**empty-contact**

**Grammar:** `empty-contact <string>;`

**Blocks:** options, view

**Tags:** server, zone

Specifies the contact name in the returned SOA record for empty zones.

This specifies the contact name that appears in the returned SOA record for empty zones. If none is specified, "." is used.

**empty-zones-enable**

**Grammar:** `empty-zones-enable <boolean>;`

**Blocks:** options, view

**Tags:** server, zone

Enables or disables all empty zones.

This enables or disables all empty zones. By default, they are enabled.

**disable-empty-zone**

**Grammar:** `disable-empty-zone <string>; // may occur multiple times`

**Blocks:** options, view

**Tags:** server, zone

Disables individual empty zones.

This disables individual empty zones. By default, none are disabled. This option can be specified multiple times.

### Content Filtering

**deny-answer-addresses**

**Grammar:** `deny-answer-addresses { <address_match_element>; ... } [ except-from { <string>; ... } ];`

**Blocks:** options, view

**Tags:** query

Rejects A or AAAA records if the corresponding IPv4 or IPv6 addresses match a given *address_match_list*.

BIND 9 provides the ability to filter out responses from external DNS servers containing certain types of data in the answer section. Specifically, it can reject address (A or AAAA) records if the corresponding IPv4 or IPv6 addresses match the given *address_match_list* of the *deny-answer-addresses* option.

In the *address_match_list* of the *deny-answer-addresses* option, only *ip_address* and *netprefix* are meaningful; any *server_key* is silently ignored.

**deny-answer-aliases**

**Grammar:** `deny-answer-aliases { <string>; ... } [ except-from { <string>; ... } ];`

**Blocks:** options, view

**Tags:** query

Rejects CNAME or DNAME records if the "alias" name matches a given list of *domain_name* elements.

BIND can also reject CNAME or DNAME records if the "alias" name (i.e., the CNAME alias or the substituted query name due to DNAME) matches the given list of *domain_name* elements of the *deny-answer-aliases* option, where "match" means the alias name is a subdomain of one of the listed domain names. If the optional list is specified in the `except-from` argument, records whose query name matches the list are accepted regardless of the filter setting. Likewise, if the alias name is a subdomain of the corresponding zone, the *deny-answer-aliases* filter does not apply; for example, even if "example.com" is specified for *deny-answer-aliases*,

```
www.example.com. CNAME xxx.example.com.
```

returned by an "example.com" server is accepted.

If a response message is rejected due to filtering, the entire message is discarded without being cached and a SERVFAIL error is returned to the client.

This filtering is intended to prevent "DNS rebinding attacks," in which an attacker, in response to a query for a domain name the attacker controls, returns an IP address within the user's own network or an alias name within the user's own domain. A naive web browser or script could then serve as an unintended proxy, allowing the attacker to get access to an internal node of the local network that could not be externally accessed otherwise. See the paper available at https://dl.acm.org/doi/10.1145/1315245.1315298 for more details about these attacks.

For example, with a domain named "example.net" and an internal network using an IPv4 prefix 192.0.2.0/24, an administrator might specify the following rules:

```
deny-answer-addresses { 192.0.2.0/24; } except-from { "example.net"; };
deny-answer-aliases { "example.net"; };
```

If an external attacker let a web browser in the local network look up an IPv4 address of "attacker.example.com", the attacker's DNS server would return a response like this:

```
attacker.example.com. A 192.0.2.1
```

in the answer section. Since the rdata of this record (the IPv4 address) matches the specified prefix 192.0.2.0/24, this response would be ignored.

On the other hand, if the browser looked up a legitimate internal web server "www.example.net" and the following response were returned to the BIND 9 server:

```
www.example.net. A 192.0.2.2
```

it would be accepted, since the owner name "www.example.net" matches the `except-from` element, "example.net".

Note that this is not really an attack on the DNS per se. In fact, there is nothing wrong with having an "external" name mapped to an "internal" IP address or domain name from the DNS point of view; it might actually be provided for a legitimate purpose, such as for debugging. As long as the mapping is provided by the correct owner, it either is not possible or does not make sense to detect whether the intent of the mapping is legitimate within the DNS. The "rebinding" attack must primarily be protected at the application that uses the DNS. For a large site, however, it may be difficult to protect all possible applications at once. This filtering feature is provided only to help such an operational environment; turning it on is generally discouraged unless there is no other choice and the attack is a real threat to applications.

Care should be particularly taken if using this option for addresses within 127.0.0.0/8. These addresses are obviously "internal," but many applications conventionally rely on a DNS mapping from some name to such an address. Filtering out DNS records containing this address spuriously can break such applications.

### Response Policy Zone (RPZ) Rewriting

BIND 9 includes a limited mechanism to modify DNS responses for requests analogous to email anti-spam DNS rejection lists. Responses can be changed to deny the existence of domains (NXDOMAIN), deny the existence of IP addresses for domains (NODATA), or contain other IP addresses or data.

**response-policy**

**Grammar:** `response-policy { zone <string> [ add-soa <boolean> ] [ log <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ policy ( cname | disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only <quoted_string> ) ] [ recursive-only <boolean> ] [ nsip-enable <boolean> ] [ nsdname-enable <boolean> ] [ ede <string> ]; ... } [ add-soa <boolean> ] [ break-dnssec <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ min-ns-dots <integer> ] [ nsip-wait-recurse <boolean> ] [ nsdname-wait-recurse <boolean> ] [ qname-wait-recurse <boolean> ] [ recursive-only <boolean> ] [ servfail-until-ready <boolean> ] [ nsip-enable <boolean> ] [ nsdname-enable <boolean> ] [ dnsrps-enable <boolean> ] [ dnsrps-options { <unspecified-text> } ];`

**Blocks:** options, view

**Tags:** server, query, zone, security

Specifies response policy zones for the view or among global options.

Response policy zones are named in the *response-policy* option for the view, or among the global options if there is no *response-policy* option for the view. Response policy zones are ordinary DNS zones containing RRsets that can be queried normally if allowed. It is usually best to restrict those queries with something like `allow-query { localhost; };`.

A *response-policy* option can support multiple policy zones. To maximize performance, a radix tree is used to quickly identify response policy zones containing triggers that match the current query. This imposes an upper limit of 64 on the number of policy zones in a single *response-policy* option; more than that is a configuration error.

Rules encoded in response policy zones are processed after those defined in *Access Control*. All queries from clients which are not permitted access to the resolver are answered with a status code of REFUSED, regardless of configured RPZ rules.

Five policy triggers can be encoded in RPZ records.

**RPZ-CLIENT-IP**

IP records are triggered by the IP address of the DNS client. Client IP address triggers are encoded in records that have owner names that are subdomains of `rpz-client-ip`, relativized to the policy zone origin name, and that encode an address or address block. IPv4 addresses are represented as `prefixlength.B4.B3.B2.B1.rpz-client-ip`. The IPv4 prefix length must be between 1 and 32. All four bytes - B4, B3, B2, and B1 - must be present. B4 is the decimal value of the least significant byte of the IPv4 address as in IN-ADDR.ARPA.

IPv6 addresses are encoded in a format similar to the standard IPv6 text representation, `prefixlength.W8.W7.W6.W5.W4.W3.W2.W1.rpz-client-ip`. Each of W8,...,W1 is a one- to four-digit hexadecimal number representing 16 bits of the IPv6 address as in the standard text representation of IPv6 addresses, but reversed as in IP6.ARPA. (Note that this representation of IPv6 addresses is different from IP6.ARPA, where each hex digit occupies a label.) All 8 words must be present except when one set of consecutive zero words is replaced with

`.zz.`, analogous to double colons (::) in standard IPv6 text encodings. The IPv6 prefix length must be between 1 and 128.

**QNAME**
QNAME policy records are triggered by query names of requests and targets of CNAME records resolved to generate the response. The owner name of a QNAME policy record is the query name relativized to the policy zone.

**RPZ-IP**
IP triggers are IP addresses in an A or AAAA record in the ANSWER section of a response. They are encoded like client-IP triggers, except as subdomains of `rpz-ip`.

**RPZ-NSDNAME**
NSDNAME triggers match names of authoritative servers for the query name, a parent of the query name, a CNAME for the query name, or a parent of a CNAME. They are encoded as subdomains of `rpz-nsdname`, relativized to the RPZ origin name. NSIP triggers match IP addresses in A and AAAA RRsets for domains that can be checked against NSDNAME policy records. The `nsdname-enable` phrase turns NSDNAME triggers off or on for a single policy zone or for all zones.

If authoritative name servers for the query name are not yet known, *named* recursively looks up the authoritative servers for the query name before applying an RPZ-NSDNAME rule, which can cause a processing delay. To speed up processing at the cost of precision, the `nsdname-wait-recurse` option can be used; when set to `no`, RPZ-NSDNAME rules are only applied when authoritative servers for the query name have already been looked up and cached. If authoritative servers for the query name are not in the cache, the RPZ-NSDNAME rule is ignored, but the authoritative servers for the query name are looked up in the background and the rule is applied to subsequent queries. The default is `yes`, meaning RPZ-NSDNAME rules are always applied, even if authoritative servers for the query name need to be looked up first.

**RPZ-NSIP**
NSIP triggers match the IP addresses of authoritative servers. They are encoded like IP triggers, except as subdomains of `rpz-nsip`. NSDNAME and NSIP triggers are checked only for names with at least `min-ns-dots` dots. The default value of `min-ns-dots` is 1, to exclude top-level domains. The `nsip-enable` phrase turns NSIP triggers off or on for a single policy zone or for all zones.

If a name server's IP address is not yet known, *named* recursively looks up the IP address before applying an RPZ-NSIP rule, which can cause a processing delay. To speed up processing at the cost of precision, the `nsip-wait-recurse` option can be used; when set to `no`, RPZ-NSIP rules are only applied when a name server's IP address has already been looked up and cached. If a server's IP address is not in the cache, the RPZ-NSIP rule is ignored, but the address is looked up in the background and the rule is applied to subsequent queries. The default is `yes`, meaning RPZ-NSIP rules are always applied, even if an address needs to be looked up first.

The query response is checked against all response policy zones, so two or more policy records can be triggered by a response. Because DNS responses are rewritten according to at most one policy record, a single record encoding an action (other than `DISABLED` actions) must be chosen. Triggers, or the records that encode them, are chosen for rewriting in the following order:

1. Choose the triggered record in the zone that appears first in the response-policy option.
2. Prefer CLIENT-IP to QNAME to IP to NSDNAME to NSIP triggers in a single zone.
3. Among NSDNAME triggers, prefer the trigger whose matched name server domain name appears last in the DNSSEC canonical ordering.
4. Among IP or NSIP triggers, prefer the trigger with the longest prefix.
5. Among triggers with the same prefix length, prefer the IP or NSIP trigger that matches the smallest IP address.

When the processing of a response is restarted to resolve DNAME or CNAME records and a policy record set has not been triggered, all response policy zones are again consulted for the DNAME or CNAME names and addresses.

RPZ record sets are any types of DNS record, except DNAME or DNSSEC, that encode actions or responses to individual queries. Any of the policies can be used with any of the triggers. For example, while the `TCP-only` policy is commonly used with `client-IP` triggers, it can be used with any type of trigger to force the use of TCP for responses with owner names in a zone.

**PASSTHRU**
: The auto-acceptance policy is specified by a CNAME whose target is `rpz-passthru`. It causes the response to not be rewritten and is most often used to "poke holes" in policies for CIDR blocks.

**DROP**
: The auto-rejection policy is specified by a CNAME whose target is `rpz-drop`. It causes the response to be discarded. Nothing is sent to the DNS client.

**TCP-Only**
: The "slip" policy is specified by a CNAME whose target is `rpz-tcp-only`. It changes UDP responses to short, truncated DNS responses that require the DNS client to try again with TCP. It is used to mitigate distributed DNS reflection attacks.

**NXDOMAIN**
: The "domain undefined" response is encoded by a CNAME whose target is the root domain (.).

**NODATA**
: The empty set of resource records is specified by a CNAME whose target is the wildcard top-level domain (`*.`). It rewrites the response to NODATA or ANCOUNT=0.

**Local Data**
: A set of ordinary DNS records can be used to answer queries. Queries for record types not in the set are answered with NODATA.

  A special form of local data is a CNAME whose target is a wildcard such as *.example.com. It is used as if an ordinary CNAME after the asterisk (*) has been replaced with the query name. This special form is useful for query logging in the walled garden's authoritative DNS server.

All of the actions specified in all of the individual records in a policy zone can be overridden with a `policy` clause in the *`response-policy`* option. An organization using a policy zone provided by another organization might use this mechanism to redirect domains to its own walled garden.

**GIVEN**
: The placeholder policy says "do not override but perform the action specified in the zone."

**DISABLED**
: The testing override policy causes policy zone records to do nothing but log what they would have done if the policy zone were not disabled. The response to the DNS query is written (or not) according to any triggered policy records that are not disabled. Disabled policy zones should appear first, because they are often not logged if a higher-precedence trigger is found first.

**PASSTHRU; DROP; TCP-Only; NXDOMAIN; NODATA**
: These settings each override the corresponding per-record policy.

**CNAME domain**
: This causes all RPZ policy records to act as if they were "cname domain" records.

By default, the actions encoded in a response policy zone are applied only to queries that ask for recursion (RD=1). That default can be changed for a single policy zone, or for all response policy zones in a view, with a `recursive-only no` clause. This feature is useful for serving the same zone files both inside and outside an **RFC 1918** cloud and using RPZ to delete answers that would otherwise contain **RFC 1918** values on the externally visible name server or view.

Also by default, when *`named`* is started it may start answering to queries before the response policy zones are completely loaded and processed. This can be changed with the `servfail-until-ready yes` option, in which case incoming requests will result in SERVFAIL answer, until all the response policy zones are ready. Note that if one or more response

policy zones fail to load, *named* starts responding to queries according to those zones that did load. Note, that enabling this option has no effect when a DNS Response Policy Service (DNSRPS) interface is used.

Also by default, RPZ actions are applied only to DNS requests that either do not request DNSSEC metadata (DO=0) or when no DNSSEC records are available for the requested name in the original zone (not the response policy zone). This default can be changed for all response policy zones in a view with a `break-dnssec yes` clause. In that case, RPZ actions are applied regardless of DNSSEC. The name of the clause option reflects the fact that results rewritten by RPZ actions cannot verify.

No DNS records are needed for a QNAME or Client-IP trigger; the name or IP address itself is sufficient, so in principle the query name need not be recursively resolved. However, not resolving the requested name can leak the fact that response policy rewriting is in use, and that the name is listed in a policy zone, to operators of servers for listed names. To prevent that information leak, by default any recursion needed for a request is done before any policy triggers are considered. Because listed domains often have slow authoritative servers, this behavior can cost significant time. The `qname-wait-recurse no` option overrides the default and enables that behavior when recursion cannot change a non-error response. The option does not affect QNAME or client-IP triggers in policy zones listed after other zones containing IP, NSIP, and NSDNAME triggers, because those may depend on the A, AAAA, and NS records that would be found during recursive resolution. It also does not affect DNSSEC requests (DO=1) unless `break-dnssec yes` is in use, because the response would depend on whether RRSIG records were found during resolution. Using this option can cause error responses such as SERVFAIL to appear to be rewritten, since no recursion is being done to discover problems at the authoritative server.

**dnsrps-enable**

**Grammar:** `dnsrps-enable <boolean>;`

**Blocks:** options, view

**Tags:** server, security

Turns on the DNS Response Policy Service (DNSRPS) interface.

The `dnsrps-enable yes` option turns on the DNS Response Policy Service (DNSRPS) interface, if it has been compiled in *named* using `configure --enable-dnsrps`.

**dnsrps-library**

**Grammar:** `dnsrps-library <quoted_string>;`

**Blocks:** options

**Tags:** server, security

Specifies the path to the DNS Response Policy Service (DNSRPS) provider library.

This option specifies the path to the DNSRPS provider library. Typically this library is detected when building with `configure --enable-dnsrps` and does not need to be specified in `named.conf`; the option exists to override the default library for testing purposes.

**dnsrps-options**

**Grammar:** `dnsrps-options { <unspecified-text> };`

**Blocks:** options, view

**Tags:** server, security

Provides additional RPZ configuration settings, which are passed to the DNS Response Policy Service (DNSRPS) provider library.

The block provides additional RPZ configuration settings, which are passed through to the DNSRPS provider library. Multiple DNSRPS settings in an *dnsrps-options* string should be separated with semi-colons (;). The DNSRPS provider library is passed a configuration string consisting of the *dnsrps-options* text, concatenated with settings derived from the *response-policy* statement.

Note: the *dnsrps-options* text should only include configuration settings that are specific to the DNSRPS provider. For example, the DNSRPS provider from Farsight Security takes options such as `dnsrpzd-conf`, `dnsrpzd-sock`, and `dnzrpzd-args` (for details of these options, see the `librpz` documentation). Other RPZ configuration settings could be included in *dnsrps-options* as well, but if *named* were switched back to traditional RPZ by setting *dnsrps-enable* to "no", those options would be ignored.

The TTL of a record modified by RPZ policies is set from the TTL of the relevant record in the policy zone. It is then limited to a maximum value. The `max-policy-ttl` clause changes the maximum number of seconds from its default of 5. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

For example, an administrator might use this option statement:

```
response-policy { zone "badlist"; };
```

and this zone statement:

```
zone "badlist" {type primary; file "primary/badlist"; allow-query {none;}; };
```

with this zone file:

```
$TTL 1H
@                       SOA LOCALHOST. named-mgr.example.com (1 1h 15m 30d 2h)
            NS  LOCALHOST.

; QNAME policy records.  There are no periods (.) after the owner names.
nxdomain.domain.com     CNAME   .               ; NXDOMAIN policy
*.nxdomain.domain.com   CNAME   .               ; NXDOMAIN policy
nodata.domain.com       CNAME   *.              ; NODATA policy
*.nodata.domain.com     CNAME   *.              ; NODATA policy
bad.domain.com          A       10.0.0.1        ; redirect to a walled garden
            AAAA    2001:2::1
bzone.domain.com        CNAME   garden.example.com.

; do not rewrite (PASSTHRU) OK.DOMAIN.COM
ok.domain.com           CNAME   rpz-passthru.

; redirect x.bzone.domain.com to x.bzone.domain.com.garden.example.com
*.bzone.domain.com      CNAME   *.garden.example.com.

; IP policy records that rewrite all responses containing A records in 127/8
;       except 127.0.0.1
8.0.0.0.127.rpz-ip      CNAME   .
32.1.0.0.127.rpz-ip     CNAME   rpz-passthru.

; NSDNAME and NSIP policy records
ns.domain.com.rpz-nsdname   CNAME   .
48.zz.2.2001.rpz-nsip       CNAME   .

; auto-reject and auto-accept some DNS clients
```

(continues on next page)

(continued from previous page)

```
112.zz.2001.rpz-client-ip    CNAME   rpz-drop.
8.0.0.0.127.rpz-client-ip    CNAME   rpz-drop.

; force some DNS clients and responses in the example.com zone to TCP
16.0.0.1.10.rpz-client-ip   CNAME   rpz-tcp-only.
example.com                 CNAME   rpz-tcp-only.
*.example.com               CNAME   rpz-tcp-only.
```

Response policy zones can be configured to set an Extended DNS Error (EDE) code on the responses which have been modified by the response policy:

```
response-policy { zone "badlist" ede filtered; };
```

The following settings are supported for the `ede` option:

**`none`**
No Extended DNS Error code is set (default).

**`forged`**
Extended DNS Error code 4 - Forged Answer.

**`blocked`**
Extended DNS Error code 15 - Blocked.

**`censored`**
Extended DNS Error code 16 - Censored.

**`filtered`**
Extended DNS Error code 17 - Filtered.

**`prohibited`**
Extended DNS Error code 18 - Prohibited.

See **RFC 8914** for more information about the Extended DNS Error codes.

RPZ can affect server performance. Each configured response policy zone requires the server to perform one to four additional database lookups before a query can be answered. For example, a DNS server with four policy zones, each with all four kinds of response triggers (QNAME, IP, NSIP, and NSDNAME), requires a total of 17 times as many database lookups as a similar DNS server with no response policy zones. A BIND 9 server with adequate memory and one response policy zone with QNAME and IP triggers might achieve a maximum queries-per-second (QPS) rate about 20% lower. A server with four response policy zones with QNAME and IP triggers might have a maximum QPS rate about 50% lower.

Responses rewritten by RPZ are counted in the `RPZRewrites` statistics.

The `log` clause can be used to optionally turn off rewrite logging for a particular response policy zone. By default, all rewrites are logged.

The `add-soa` option controls whether the RPZ's SOA record is added to the section for traceback of changes from this zone. This can be set at the individual policy zone level or at the response-policy level. The default is `yes`.

Updates to RPZ zones are processed asynchronously; if there is more than one update pending they are bundled together. If an update to a RPZ zone (for example, via IXFR) happens less than `min-update-interval` seconds after the most recent update, the changes are not carried out until this interval has elapsed. The default is `60` seconds. For convenience, TTL-style time-unit suffixes may be used to specify the value. It also accepts ISO 8601 duration formats.

### Response Rate Limiting

**rate-limit**

**Grammar:**

```
rate-limit {
        all-per-second <integer>;
        errors-per-second <integer>;
        exempt-clients { <address_match_element>; ... };
        ipv4-prefix-length <integer>;
        ipv6-prefix-length <integer>;
        log-only <boolean>;
        max-table-size <integer>;
        min-table-size <integer>;
        nodata-per-second <integer>;
        nxdomains-per-second <integer>;
        qps-scale <integer>;
        referrals-per-second <integer>;
        responses-per-second <integer>;
        slip <integer>;
        window <integer>;
};
```

**Blocks:** options, view

**Tags:** query

Controls excessive UDP responses, to prevent BIND 9 from being used to amplify reflection denial-of-service (DoS) attacks.

Excessive, almost-identical UDP *responses* can be controlled by configuring a *rate-limit* clause in an *options* or *view* statement. This mechanism keeps authoritative BIND 9 from being used to amplify reflection denial-of-service (DoS) attacks. Short BADCOOKIE errors or truncated (TC=1) responses can be sent to provide rate-limited responses to legitimate clients within a range of forged, attacked IP addresses. Legitimate clients react to dropped responses by retrying, to BADCOOKIE errors by including a server cookie when retrying, and to truncated responses by switching to TCP.

This mechanism is intended for authoritative DNS servers. It can be used on recursive servers, but can slow applications such as SMTP servers (mail receivers) and HTTP clients (web browsers) that repeatedly request the same domains. When possible, closing "open" recursive servers is better.

Response-rate limiting uses a "credit" or "token bucket" scheme. Each combination of identical response and client has a conceptual "account" that earns a specified number of credits every second. A prospective response debits its account by one. Responses are dropped or truncated while the account is negative.

**window**

**Grammar:** `window <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Specifies the length of time during which responses are tracked.

Responses are tracked within a rolling window of time which defaults to 15 seconds, but which can be configured with the *window* option to any value from 1 to 3600 seconds (1 hour). The account cannot become

more positive than the per-second limit or more negative than *window* times the per-second limit. When the specified number of credits for a class of responses is set to 0, those responses are not rate-limited.

**ipv4-prefix-length**

**Grammar:** `ipv4-prefix-length <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** server

Specifies the prefix lengths of IPv4 address blocks.

**ipv6-prefix-length**

**Grammar:** `ipv6-prefix-length <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** server

Specifies the prefix lengths of IPv6 address blocks.

The notions of "identical response" and "DNS client" for rate limiting are not simplistic. All responses to an address block are counted as if to a single client. The prefix lengths of address blocks are specified with *ipv4-prefix-length* (default 24) and *ipv6-prefix-length* (default 56).

**responses-per-second**

**Grammar:** `responses-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Limits the number of non-empty responses for a valid domain name and record type.

All non-empty responses for a valid domain name (qname) and record type (qtype) are identical and have a limit specified with *responses-per-second* (default 0 or no limit). All valid wildcard domain names are interpreted as the zone's origin name concatenated to the "*" name.

**nodata-per-second**

**Grammar:** `nodata-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Limits the number of empty (NODATA) responses for a valid domain name.

All empty (NODATA) responses for a valid domain, regardless of query type, are identical. Responses in the NODATA class are limited by *nodata-per-second* (default *responses-per-second*).

**nxdomains-per-second**

**Grammar:** `nxdomains-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Limits the number of undefined subdomains for a valid domain name.

Requests for any and all undefined subdomains of a given valid domain result in NXDOMAIN errors, and are identical regardless of query type. They are limited by *nxdomains-per-second* (default *responses-per-second*). This controls some attacks using random names, but can be relaxed or turned off (set to 0) on servers that expect many legitimate NXDOMAIN responses, such as from anti-spam rejection lists.

**referrals-per-second**

**Grammar:** `referrals-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Limits the number of referrals or delegations to a server for a given domain.

Referrals or delegations to the server of a given domain are identical and are limited by *referrals-per-second* (default *responses-per-second*).

Responses generated from local wildcards are counted and limited as if they were for the parent domain name. This controls flooding using random.wild.example.com.

All requests that result in DNS errors other than NXDOMAIN, such as SERVFAIL and FORMERR, are identical regardless of requested name (qname) or record type (qtype). This controls attacks using invalid requests or distant, broken authoritative servers.

**errors-per-second**

**Grammar:** `errors-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** server

Limits the number of errors for a valid domain name and record type.

By default the limit on errors is the same as the *responses-per-second* value, but it can be set separately with *errors-per-second*.

**slip**

**Grammar:** `slip <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Sets the number of "slipped" responses to minimize the use of forged source addresses for an attack.

Many attacks using DNS involve UDP requests with forged source addresses. Rate limiting prevents the use of BIND 9 to flood a network with responses to requests with forged source addresses, but could let a third party block responses to legitimate requests. There is a mechanism that can answer some legitimate requests from a client whose address is being forged in a flood. Setting *slip* to 2 (its default) causes every other UDP request without a valid server cookie to be answered with a small response. The small size and reduced frequency, and resulting lack of amplification, of "slipped" responses make them unattractive for reflection DoS attacks. *slip* must be between 0 and 10. A value of 0 does not "slip"; no small responses are sent due to rate limiting. Rather, all responses are dropped. A value of 1 causes every response to slip; values between 2 and 10 cause every nth response to slip.

If the request included a client cookie, then a "slipped" response is a BADCOOKIE error with a server cookie, which allows a legitimate client to include the server cookie to be exempted from the rate limiting when it retries the request. If the request did not include a cookie, then a "slipped" response is a truncated (TC=1) response, which prompts a legitimate client to switch to TCP and thus be exempted from the rate limiting. Some error responses, including REFUSED and SERVFAIL, cannot be replaced with truncated responses and are instead leaked at the *slip* rate.

(Note: dropped responses from an authoritative server may reduce the difficulty of a third party successfully forging a response to a recursive resolver. The best security against forged responses is for authoritative operators to sign their zones using DNSSEC and for resolver operators to validate the responses. When this is not an option, operators who are more concerned with response integrity than with flood mitigation may consider setting *slip* to 1, causing all rate-limited responses to be truncated rather than dropped. This reduces the effectiveness of rate-limiting against reflection attacks.)

**qps-scale**

**Grammar:** `qps-scale <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Tightens defenses during DNS attacks by scaling back the ratio of the current query-per-second rate.

When the approximate query-per-second rate exceeds the *qps-scale* value, the *responses-per-second*, *errors-per-second*, *nxdomains-per-second*, and *all-per-second* values are reduced by the ratio of the current rate to the *qps-scale* value. This feature can tighten defenses during attacks. For example, with `qps-scale 250; responses-per-second 20;` and a total query rate of 1000 queries/second for all queries from all DNS clients including via TCP, then the effective responses/second limit changes to (250/1000)*20, or 5. Responses to requests that included a valid server cookie, and responses sent via TCP, are not limited but are counted to compute the query-per-second rate.

**exempt-clients**

**Grammar:** `exempt-clients { <address_match_element>; ... };`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Exempts specific clients or client groups from rate limiting.

Communities of DNS clients can be given their own parameters or no rate limiting by putting *rate-limit* statements in *view* statements instead of in the global `option` statement. A *rate-limit* statement in a view replaces, rather than supplements, a *rate-limit* statement among the main options.

DNS clients within a view can be exempted from rate limits with the *exempt-clients* clause.

**all-per-second**

**Grammar:** `all-per-second <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Limits UDP responses of all kinds.

UDP responses of all kinds can be limited with the *all-per-second* phrase. This rate limiting is unlike the rate limiting provided by *responses-per-second*, *errors-per-second*, and

*nxdomains-per-second* on a DNS server, which are often invisible to the victim of a DNS reflection attack. Unless the forged requests of the attack are the same as the legitimate requests of the victim, the victim's requests are not affected. Responses affected by an *all-per-second* limit are always dropped; the *slip* value has no effect. An *all-per-second* limit should be at least 4 times as large as the other limits, because single DNS clients often send bursts of legitimate requests. For example, the receipt of a single mail message can prompt requests from an SMTP server for NS, PTR, A, and AAAA records as the incoming SMTP/TCP/IP connection is considered. The SMTP server can need additional NS, A, AAAA, MX, TXT, and SPF records as it considers the SMTP `Mail From` command. Web browsers often repeatedly resolve the same names that are duplicated in HTML <IMG> tags in a page. *all-per-second* is similar to the rate limiting offered by firewalls but is often inferior. Attacks that justify ignoring the contents of DNS responses are likely to be attacks on the DNS server itself. They usually should be discarded before the DNS server spends resources making TCP connections or parsing DNS requests, but that rate limiting must be done before the DNS server sees the requests.

**max-table-size**

**Grammar:** `max-table-size <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** server

Sets the maximum size of the table used to track requests and rate-limit responses.

**min-table-size**

**Grammar:** `min-table-size <integer>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** query

Sets the minimum size of the table used to track requests and rate-limit responses.

The maximum size of the table used to track requests and rate-limit responses is set with *max-table-size*. Each entry in the table is between 40 and 80 bytes. The table needs approximately as many entries as the number of requests received per second. The default is 20,000. To reduce the cold start of growing the table, *min-table-size* (default 500) can set the minimum table size. Enable *rate-limit* category logging to monitor expansions of the table and inform choices for the initial and maximum table size.

**log-only**

**Grammar:** `log-only <boolean>;`

**Blocks:** options.rate-limit, view.rate-limit

**Tags:** logging, query

Tests rate-limiting parameters without actually dropping any requests.

Use `log-only yes` to test rate-limiting parameters without actually dropping any requests.

Responses dropped by rate limits are included in the `RateDropped` and `QryDropped` statistics. Responses that are truncated by rate limits are included in `RateSlipped` and `RespTruncated`.

### NXDOMAIN Redirection

*named* supports NXDOMAIN redirection via two methods:

- *Redirect zone*
- Redirect namespace

With either method, when *named* gets an NXDOMAIN response it examines a separate namespace to see if the NXDOMAIN response should be replaced with an alternative response.

With a redirect zone (`zone "." { type redirect; };`), the data used to replace the NXDOMAIN is held in a single zone which is not part of the normal namespace. All the redirect information is contained in the zone; there are no delegations.

**nxdomain-redirect**

**Grammar:** `nxdomain-redirect <string>;`

**Blocks:** options, view

**Tags:** query

Appends the specified suffix to the original query name, when replacing an NXDOMAIN with a redirect namespace.

With a redirect namespace (`option { nxdomain-redirect <suffix> };`), the data used to replace the NXDOMAIN is part of the normal namespace and is looked up by appending the specified suffix to the original query name. This roughly doubles the cache required to process NXDOMAIN responses, as both the original NXDOMAIN response and the replacement data (or an NXDOMAIN indicating that there is no replacement) must be stored.

If both a redirect zone and a redirect namespace are configured, the redirect zone is tried first.

## 8.2.14 `server` Block Grammar

**server**

**Grammar:**

```
server <netprefix> {
        bogus <boolean>;
        edns <boolean>;
        edns-udp-size <integer>;
        edns-version <integer>;
        keys <server_key>;
        max-udp-size <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        padding <integer>;
        provide-ixfr <boolean>;
        query-source [ address ] ( <ipv4_address> | * );
        query-source-v6 [ address ] ( <ipv6_address> | * );
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-cookie <boolean>;
        send-cookie <boolean>;
        tcp-keepalive <boolean>;
        tcp-only <boolean>;
```

(continues on next page)

(continued from previous page)

```
        transfer-format ( many-answers | one-answer );
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        transfers <integer>;
}; // may occur multiple times
```

**Blocks:** topmost, view

**Tags:** server

Defines characteristics to be associated with a remote name server.

### 8.2.15 `server` Block Definition and Usage

The *server* statement defines characteristics to be associated with a remote name server. If a prefix length is specified, then a range of servers is covered. Only the most specific server clause applies, regardless of the order in *named.conf*.

The *server* statement can occur at the top level of the configuration file or inside a *view* statement. If a *view* statement contains one or more *server* statements, only those apply to the view and any top-level ones are ignored. If a view contains no *server* statements, any top-level *server* statements are used as defaults.

**bogus**

**Grammar:** `bogus <boolean>;`

**Blocks:** server, view.server

**Tags:** server

Allows a remote server to be ignored.

If a remote server is giving out bad data, marking it as bogus prevents further queries to it. The default value of *bogus* is `no`.

**edns**

**Grammar:** `edns <boolean>;`

**Blocks:** server, view.server

**Tags:** server

Controls the use of the EDNS0 (**RFC 2671**) feature.

The *edns* clause determines whether the local server attempts to use EDNS when communicating with the remote server. The default is `yes`.

**edns-version**

**Grammar:** `edns-version <integer>;`

**Blocks:** server, view.server

**Tags:** server

Sets the maximum EDNS VERSION that is sent to the server(s) by the resolver.

The *edns-version* option sets the maximum EDNS VERSION that is sent to the server(s) by the resolver. The actual EDNS version sent is still subject to normal EDNS version-negotiation rules (see **RFC 6891**), the maximum EDNS version supported by the server, and any other heuristics that indicate that a lower version should be sent. This option is intended to be used when a remote server reacts badly to a given EDNS version or higher; it should be set to the highest version the remote server is known to support. Valid values are 0 to 255; higher values are silently adjusted. This option is not needed until higher EDNS versions than 0 are in use.

**padding**

**Grammar:** `padding <integer>;`

**Blocks:** server, view.server

**Tags:** server

Adds EDNS Padding options to outgoing messages to increase the packet size.

The option adds EDNS Padding options to outgoing messages, increasing the packet size to a multiple of the specified block size. Valid block sizes range from 0 (the default, which disables the use of EDNS Padding) to 512 bytes. Larger values are reduced to 512, with a logged warning. Note: this option is not currently compatible with TSIG or SIG(0), as the EDNS OPT record containing the padding would have to be added to the packet after it had already been signed.

**tcp-only**

**Grammar:** `tcp-only <boolean>;`

**Blocks:** server, view.server

**Tags:** server

Sets the transport protocol to TCP.

The option sets the transport protocol to TCP. The default is to use the UDP transport and to fallback on TCP only when a truncated response is received.

**tcp-keepalive**

**Grammar:** `tcp-keepalive <boolean>;`

**Blocks:** server, view.server

**Tags:** server

Adds EDNS TCP keepalive to messages sent over TCP.

The option adds EDNS TCP keepalive to messages sent over TCP. Note that currently idle timeouts in responses are ignored.

**transfers**

**Grammar:** `transfers <integer>;`

**Blocks:** server, view.server

**Tags:** server

Limits the number of concurrent inbound zone transfers from a server.

*transfers* is used to limit the number of concurrent inbound zone transfers from the specified server. If no *transfers* clause is specified, the limit is set according to the *transfers-per-ns* option.

**keys**

**Blocks:** dnssec-policy, server, view.server

**Tags:** server, security

Specifies one or more *server_key* s to be used with a remote server.

> **Warning**
>
> This option is not to be confused with *keys* in the *dnssec-policy* specification. Although statements with the same name exist in both contexts, they refer to fundamentally incompatible concepts.

In the context of a *server* block, the option identifies a *server_key* defined by the *key* statement, to be used for transaction security (see *TSIG*) when talking to the remote server. When a request is sent to the remote server, a request signature is generated using the key specified here and appended to the message. A request originating from the remote server is not required to be signed by this key.

Only a single key per server is currently supported.

It is possible to override the following values defined in *view* and *options* blocks:

- *edns-udp-size*
- *max-udp-size*
- *notify-source-v6*
- *notify-source*
- *provide-ixfr*
- *query-source-v6*
- *query-source*
- *request-expire*
- *request-ixfr*
- *request-nsid*
- *require-cookie*
- *send-cookie*
- *transfer-format*
- *transfer-source-v6*
- *transfer-source*

### 8.2.16 *statistics-channels* Block Grammar

**statistics-channels**

**Grammar:**

```
statistics-channels {
        inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ] [␣
↪allow { <address_match_element>; ... } ]; // may occur multiple times
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** logging

Specifies the communication channels to be used by system administrators to access statistics information on the name server.

### 8.2.17 *statistics-channels* Block Definition and Usage

The *statistics-channels* statement declares communication channels to be used by system administrators to get access to statistics information on the name server.

This statement is intended to be flexible to support multiple communication protocols in the future, but currently only HTTP access is supported. It requires that BIND 9 be compiled with libxml2 and/or json-c (also known as libjson0); the *statistics-channels* statement is still accepted even if it is built without the library, but any HTTP access fails with an error.

An *inet* control channel is a TCP socket listening at the specified *port* on the specified *ip_address*, which can be an IPv4 or IPv6 address. An *ip_address* of `*` (asterisk) is interpreted as the IPv4 wildcard address; connections are accepted on any of the system's IPv4 addresses. To listen on the IPv6 wildcard address, use an *ip_address* of `::`.

If no port is specified, port 80 is used for HTTP channels. The asterisk (`*`) cannot be used for *port*.

Attempts to open a statistics channel are restricted by the optional `allow` clause. Connections to the statistics channel are permitted based on the *address_match_list*. If no `allow` clause is present, *named* accepts connection attempts from any address. Since the statistics may contain sensitive internal information, the source of connection requests must be restricted appropriately so that only trusted parties can access the statistics channel.

Gathering data exposed by the statistics channel locks various subsystems in *named*, which could slow down query processing if statistics data is requested too often.

An issue in the statistics channel would be considered a security issue only if it could be exploited by unprivileged users circumventing the access control list. In other words, any issue in the statistics channel that could be used to access information unavailable otherwise, or to crash *named*, is not considered a security issue if it can be avoided through the use of a secure configuration.

If no *statistics-channels* statement is present, *named* does not open any communication channels.

The statistics are available in various formats and views, depending on the URI used to access them. For example, if the statistics channel is configured to listen on 127.0.0.1 port 8888, then the statistics are accessible in XML format at http://127.0.0.1:8888/ or http://127.0.0.1:8888/xml. A CSS file is included, which can format the XML statistics into tables when viewed with a stylesheet-capable browser, and into charts and graphs using the Google Charts API when using a JavaScript-capable browser.

Broken-out subsets of the statistics can be viewed at http://127.0.0.1:8888/xml/v3/status (server uptime and last reconfiguration time), http://127.0.0.1:8888/xml/v3/server (server and resolver statistics), http://127.0.0.1:8888/xml/v3/zones (zone statistics), http://127.0.0.1:8888/xml/v3/xfrins (incoming zone transfer statistics), http://127.0.0.1:8888/xml/v3/net (network status and socket statistics), http://127.0.0.1:8888/xml/v3/mem (memory manager statistics), and http://127.0.0.1:8888/xml/v3/traffic (traffic sizes).

The full set of statistics can also be read in JSON format at http://127.0.0.1:8888/json, with the broken-out subsets at http://127.0.0.1:8888/json/v1/status (server uptime and last reconfiguration time), http://127.0.0.1:8888/json/v1/server (server and resolver statistics), http://127.0.0.1:8888/json/v1/zones (zone statistics), http://127.0.0.1:8888/json/v1/xfrins (incoming zone transfer statistics), http://127.0.0.1:8888/json/v1/net (network status and socket statistics), http://127.0.0.1:8888/json/v1/mem (memory manager statistics), and http://127.0.0.1:8888/json/v1/traffic (traffic sizes).

### 8.2.18 *tls* Block Grammar

**tls**

**Grammar:**

```
tls <string> {
        ca-file <quoted_string>;
        cert-file <quoted_string>;
        cipher-suites <string>;
        ciphers <string>;
        dhparam-file <quoted_string>;
        key-file <quoted_string>;
        prefer-server-ciphers <boolean>;
        protocols { <string>; ... };
        remote-hostname <quoted_string>;
        session-tickets <boolean>;
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** security

Configures a TLS connection.

### 8.2.19 *tls* Block Definition and Usage

The *tls* statement is used to configure a TLS connection; this configuration can then be referenced by a *listen-on* or *listen-on-v6* statement to cause *named* to listen for incoming requests via TLS, or in the *primaries* statement for a zone of *type secondary* to cause zone transfer requests to be sent via TLS.

*tls* can only be set at the top level of *named.conf*.

The following options can be specified in a *tls* statement:

**key-file**

**Grammar:** `key-file <quoted_string>;`

**Blocks:** tls

**Tags:** server, security

Specifies the path to a file containing the private TLS key for a connection.

This indicates the path to a file containing the private TLS key to be used for the connection.

**cert-file**

**Grammar:** `cert-file <quoted_string>;`

**Blocks:** tls

**Tags:** server, security

Specifies the path to a file containing the TLS certificate for a connection.

This indicates the path to a file containing the TLS certificate to be used for the connection.

**ca-file**

**Grammar:** `ca-file <quoted_string>;`

**Blocks:** tls

**Tags:** server, security

Specifies the path to a file containing TLS certificates for trusted CA authorities, used to verify remote peer certificates.

This indicates the path to a file containing trusted CA authorities' TLS certificates, used to verify remote peer certificates. Specifying this option enables verification of remote peer certificates. For incoming connections, specifying this option makes BIND require a valid TLS certificate from a client. In the case of outgoing connections, if *remote-hostname* is not specified, the remote server IP address is used instead.

**dhparam-file**

**Grammar:** `dhparam-file <quoted_string>;`

**Blocks:** tls

**Tags:** server, security

Specifies the path to a file containing Diffie-Hellman parameters, for enabling cipher suites.

This indicates the path to a file containing Diffie-Hellman parameters, which is needed to enable the cipher suites depending on the Diffie-Hellman ephemeral key exchange (DHE). Having these parameters specified is essential for enabling perfect forward secrecy capable ciphers in TLSv1.2.

**remote-hostname**

**Grammar:** `remote-hostname <quoted_string>;`

**Blocks:** tls

**Tags:** security

Specifies the expected hostname in the TLS certificate of the remote server.

This specifies the expected hostname in the TLS certificate of the remote server. This option enables a remote server certificate verification. If *ca-file* is not specified, then the platform-specific certificates store is used for verification. This option is used when connecting to a remote peer only and, thus, is ignored when *tls* statements are referenced by *listen-on* or *listen-on-v6* statements.

**protocols**

**Grammar:** `protocols { <string>; ... };`

**Blocks:** tls

**Tags:** security

Specifies the allowed versions of the TLS protocol.

This specifies the allowed versions of the TLS protocol. TLS version 1.2 and higher are supported, depending on the cryptographic library in use. Multiple versions may be specified (e.g. `protocols { TLSv1.2; TLSv1.3; };`).

**cipher-suites**

**Grammar:** `cipher-suites <string>;`

**Blocks:** tls

**Tags:** security

Specifies a list of allowed cipher suites in the order of preference for TLSv1.3 only.

This option defines allowed cipher suites, such as `TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_A` The string must be formed according to the rules specified in the OpenSSL documentation (see https://docs.openssl.org/1.1.1/man1/ciphers/, section "TLS v1.3 cipher suites" for details).

**ciphers**

**Grammar:** `ciphers <string>;`

**Blocks:** tls

**Tags:** security

Specifies a list of allowed ciphers in the order of preference for TLSv1.2 only.

This option defines allowed ciphers, such as `HIGH:!aNULL:!MD5:!SHA1:!SHA256:!SHA384`. The string must be formed according to the rules specified in the OpenSSL documentation (see https://docs.openssl.org/1.1.1/man1/ciphers/ for details).

**prefer-server-ciphers**

**Grammar:** `prefer-server-ciphers <boolean>;`

**Blocks:** tls

**Tags:** server, security

Specifies that server ciphers should be preferred over client ones.

This option specifies that server ciphers should be preferred over client ones.

**session-tickets**

**Grammar:** `session-tickets <boolean>;`

**Blocks:** tls

**Tags:** security

Enables or disables session resumption through TLS session tickets.

This option enables or disables session resumption through TLS session tickets, as defined in **RFC 5077**. Disabling the stateless session tickets might be required in the cases when forward secrecy is needed, or the TLS certificate and key pair is planned to be used across multiple BIND instances.

**Warning**

TLS configuration is subject to change and incompatible changes might be introduced in the future. Users of TLS are encouraged to carefully read release notes when upgrading.

The options described above are used to control different aspects of TLS functioning. Thus, most of them have no well-defined default values, as these depend on the cryptographic library version in use and system-wide cryptographic policy. On the other hand, by specifying the needed options one could have a uniform configuration deployable across a range of platforms.

An example of privacy-oriented, perfect forward secrecy enabled configuration can be found below. It can be used as a starting point.

```
tls local-tls {
    key-file "/path/to/key.pem";
    cert-file "/path/to/fullchain_cert.pem";
    dhparam-file "/path/to/dhparam.pem";
    ciphers
↪"HIGH:!kRSA:!aNULL:!eNULL:!RC4:!3DES:!MD5:!EXP:!PSK:!SRP:!DSS:!SHA1:!SHA256:!SHA384
↪";
    prefer-server-ciphers yes;
    session-tickets no;
};
```

A Diffie-Hellman parameters file can be generated using e.g. OpenSSL, like follows:

```
openssl dhparam -out /path/to/dhparam.pem <3072_or_4096>
```

It is important to ensure that the file is generated on a machine with enough entropy from external sources (e.g. the local computer should be fine, the remote virtual machine or server might not be). These files do not contain any sensitive data and can be shared if required.

There are two built-in TLS connection configurations: `ephemeral`, which uses a temporary key and certificate created for the current *named* session only, and `none`, which can be used when setting up an HTTP listener with no encryption.

The main motivation behind the existence of the `ephemeral` configuration is to aid in testing. Since trusted certificate authorities do not issue the certificates associated with this configuration, these certificates will never be trusted by any clients that verify TLS certificates; they provide encryption of the traffic but no authentication of the transmission channel. That might be enough in the case of deployment in a controlled environment.

It should be noted that on reconfiguration, the `ephemeral` TLS key and the certificate are recreated, and all TLS certificates and keys, as well as associated data, are reloaded from the disk. In that case, listening sockets associated with TLS remain intact.

Note that performing a reconfiguration can cause a short interruption in BIND's ability to process inbound client packets. The length of interruption is environment- and configuration-specific. A good example of when reconfiguration is necessary is when TLS keys and certificates are updated on the disk.

BIND supports the following TLS authentication mechanisms described in **RFC 9103**, Section 9.3: Opportunistic TLS, Strict TLS, and Mutual TLS.

Opportunistic TLS provides encryption for data but does not provide any authentication for the channel. This mode is the default and is used whenever the *remote-hostname* and *ca-file* options are not set in *tls* statements in use. **RFC 9103** allows optional fallback to clear-text DNS in the cases when TLS is not available; however, BIND intentionally does not support that fallback, to protect from unexpected data leaks due to misconfiguration. Both BIND and its complementary tools either successfully establish a secure channel via TLS when instructed to do so, or fail to establish a connection otherwise.

Strict TLS provides server authentication via a pre-configured hostname for outgoing connections. This mechanism offers both channel confidentiality and channel authentication (of the server). In order to achieve Strict TLS, one needs to use *remote-hostname* and, optionally, *ca-file* options in the *tls* statements used for establishing outgoing connections (e.g. the ones used to download zone from primaries via TLS). Providing any of the mentioned options will enable server authentication. If *remote-hostname* is provided but *ca-file* is missing, then the platform-specific certificate authority certificates are used for authentication. The set roughly corresponds to the one used by WEB-browsers to authenticate HTTPS hosts. On the other hand, if *ca-file* is provided but *remote-hostname* is missing, then the remote side's IP address is used instead.

Mutual TLS is an extension to Strict TLS that provides channel confidentiality and mutual channel authentication. It builds up upon the clients offering client certificates when establishing connections and them doing the server authentication as in the case of Strict TLS. The server verifies the provided client certificates and accepts the TLS connection in case of successful verification or rejects it otherwise. In order to instruct the server to require and verify client TLS certificates, one needs to specify the *ca-file* option in *tls* configurations used to configure server listeners. The provided file must contain certificate authority certificates used to issue client certificates. In most cases, one should build one's own TLS certificate authority specifically to issue client certificates and include the certificate authority certificate into the file.

For authenticating zone transfers over TLS, Mutual TLS might be considered a standalone solution, while Strict TLS paired with TSIG-based authentication and, optionally, IP-based access lists, might be considered acceptable for most practical purposes. Mutual TLS has the advantage of not requiring TSIG and thus, not having security issues related to shared cryptographic secrets.

### 8.2.20 *http* Block Grammar

**http**

**Grammar:**

```
http <string> {
        endpoints { <quoted_string>; ... };
        listener-clients <integer>;
        streams-per-connection <integer>;
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** server, query

Configures HTTP endpoints on which to listen for DNS-over-HTTPS (DoH) queries.

### 8.2.21 *http* Block Definition and Usage

The *http* statement is used to configure HTTP endpoints on which to listen for DNS-over-HTTPS (DoH) queries. This configuration can then be referenced by a *listen-on* or *listen-on-v6* statement to cause *named* to listen for incoming requests over HTTPS.

*http* can only be set at the top level of *named.conf*.

The following options can be specified in an *http* statement:

**endpoints**

**Grammar:** `endpoints { <quoted_string>; ... };`

**Blocks:** http

**Tags:** server, query

Specifies a list of HTTP query paths on which to listen.

This specifies a list of HTTP query paths on which to listen. This is the portion of an RFC 3986-compliant URI following the hostname; it must be an absolute path, beginning with "/". The default value is `"/dns-query"`, if omitted.

**`listener-clients`**

**Grammar:** `listener-clients <integer>;`

**Blocks:** http

**Tags:** server, query

Specifies a per-listener quota for active connections.

This option specifies a per-listener quota for active connections.

**`streams-per-connection`**

**Grammar:** `streams-per-connection <integer>;`

**Blocks:** http

**Tags:** server, query

Specifies the maximum number of concurrent HTTP/2 streams over an HTTP/2 connection.

This option specifies the hard limit on the number of concurrent HTTP/2 streams over an HTTP/2 connection.

Any of the options above could be omitted. In such a case, a global value specified in the *`options`* statement is used (see *`http-listener-clients`*, *`http-streams-per-connection`*.

For example, the following configuration enables DNS-over-HTTPS queries on all local addresses:

```
http local {
    endpoints { "/dns-query"; };
};

options {
    ....
    listen-on tls ephemeral http local { any; };
    listen-on-v6 tls ephemeral http local { any; };
};
```

### 8.2.22 *`trust-anchors`* Block Grammar

**`trust-anchors`**

**Grammar:** `trust-anchors { <string> ( static-key | initial-key | static-ds | initial-ds ) <integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times`

**Blocks:** topmost, view

**Tags:** dnssec

Defines *DNSSEC* trust anchors.

### 8.2.23 `trust-anchors` Block Definition and Usage

The `trust-anchors` statement defines DNSSEC trust anchors. DNSSEC is described in *DNSSEC*.

A trust anchor is defined when the public key or public key digest for a non-authoritative zone is known but cannot be securely obtained through DNS, either because it is the DNS root zone or because its parent zone is unsigned. Once a key or digest has been configured as a trust anchor, it is treated as if it has been validated and proven secure.

The resolver attempts DNSSEC validation on all DNS data in subdomains of configured trust anchors. Validation below specified names can be temporarily disabled by using `rndc nta`, or permanently disabled with the `validate-except` option.

All keys listed in `trust-anchors`, and their corresponding zones, are deemed to exist regardless of what parent zones say. Only keys configured as trust anchors are used to validate the DNSKEY RRset for the corresponding name. The parent's DS RRset is not used.

`trust-anchors` may be set at the top level of `named.conf` or within a view. If it is set in both places, the configurations are additive; keys defined at the top level are inherited by all views, but keys defined in a view are only used within that view.

The `trust-anchors` statement can contain multiple trust-anchor entries, each consisting of a domain name, followed by an "anchor type" keyword indicating the trust anchor's format, followed by the key or digest data.

If the anchor type is `static-key` or `initial-key`, it is followed with the key's flags, protocol, and algorithm, plus the Base64 representation of the public key data. This is identical to the text representation of a DNSKEY record. Spaces, tabs, newlines, and carriage returns are ignored in the key data, so the configuration may be split into multiple lines.

If the anchor type is `static-ds` or `initial-ds`, it is followed with the key tag, algorithm, digest type, and the hexadecimal representation of the key digest. This is identical to the text representation of a DS record. Spaces, tabs, newlines, and carriage returns are ignored.

Trust anchors configured with the `static-key` or `static-ds` anchor types are immutable, while keys configured with `initial-key` or `initial-ds` can be kept up-to-date automatically, without intervention from the resolver operator. (`static-key` keys are identical to keys configured using the deprecated `trusted-keys` statement.)

Suppose, for example, that a zone's key-signing key was compromised, and the zone owner had to revoke and replace the key. A resolver which had the original key configured using `static-key` or `static-ds` would be unable to validate this zone any longer; it would reply with a SERVFAIL response code. This would continue until the resolver operator had updated the `trust-anchors` statement with the new key.

If, however, the trust anchor had been configured using `initial-key` or `initial-ds` instead, the zone owner could add a "stand-by" key to the zone in advance. `named` would store the stand-by key, and when the original key was revoked, `named` would be able to transition smoothly to the new key. It would also recognize that the old key had been revoked and cease using that key to validate answers, minimizing the damage that the compromised key could do. This is the process used to keep the ICANN root DNSSEC key up-to-date.

Whereas `static-key` and `static-ds` trust anchors continue to be trusted until they are removed from `named.conf`, an `initial-key` or `initial-ds` is only trusted *once*: for as long as it takes to load the managed key database and start the **RFC 5011** key maintenance process.

It is not possible to mix static with initial trust anchors for the same domain name.

The first time `named` runs with an `initial-key` or `initial-ds` configured in `named.conf`, it fetches the DNSKEY RRset directly from the zone apex, and validates it using the trust anchor specified in `trust-anchors`. If the DNSKEY RRset is validly signed by a key matching the trust anchor, then it is used as the basis for a new managed-keys database.

From that point on, whenever `named` runs, it sees the `initial-key` or `initial-ds` listed in `trust-anchors`, checks to make sure **RFC 5011** key maintenance has already been initialized for the specified domain, and if so, simply moves

on. The key specified in the *trust-anchors* statement is not used to validate answers; it is superseded by the key or keys stored in the managed-keys database.

The next time *named* runs after an initial-key or initial-ds has been *removed* from the *trust-anchors* statement (or changed to a static-key or static-ds), the corresponding zone is removed from the managed-keys database, and **RFC 5011** key maintenance is no longer used for that domain.

In the current implementation, the managed-keys database is stored as a master-format zone file.

On servers which do not use views, this file is named managed-keys.bind. When views are in use, there is a separate managed-keys database for each view; the filename is the view name (or, if a view name contains characters which would make it illegal as a filename, a hash of the view name), followed by the suffix .mkeys.

When the key database is changed, the zone is updated. As with any other dynamic zone, changes are written into a journal file, e.g., managed-keys.bind.jnl or internal.mkeys.jnl. Changes are committed to the primary file as soon as possible afterward, usually within 30 seconds. Whenever *named* is using automatic key maintenance, the zone file and journal file can be expected to exist in the working directory. (For this reason, among others, the working directory should be always be writable by *named*.)

If the *dnssec-validation* option is set to auto, *named* automatically sets up an initial-key for the root zone. This initializing key is built into *named* and is current as of the release date. When the root zone key changes, a running server detects the change and rolls to the new key; however, newly installed servers being run for the first time will need to be on a recent-enough version of BIND to have been built with the current key.

### 8.2.24 *dnssec-policy* Block Grammar

**dnssec-policy**

**Grammar options, view, zone (primary, secondary):** dnssec-policy <string>;

**Grammar topmost:**

```
dnssec-policy <string> {
        cdnskey <boolean>;
        cds-digest-types { <string>; ... };
        dnskey-ttl <duration>;
        inline-signing <boolean>;
        keys { ( csk | ksk | zsk ) [ key-directory | key-store <string> ]␣
↪lifetime <duration_or_unlimited> algorithm <string> [ tag-range <integer>
↪<integer> ] [ <integer> ]; ... };
        manual-mode <boolean>;
        max-zone-ttl <duration>;
        nsec3param [ iterations <integer> ] [ optout <boolean> ] [ salt-length
↪<integer> ];
        offline-ksk <boolean>;
        parent-ds-ttl <duration>;
        parent-propagation-delay <duration>;
        publish-safety <duration>;
        purge-keys <duration>;
        retire-safety <duration>;
        signatures-jitter <duration>;
        signatures-refresh <duration>;
        signatures-validity <duration>;
        signatures-validity-dnskey <duration>;
        zone-propagation-delay <duration>;
}; // may occur multiple times
```

**Blocks:** topmost, options, view, zone (primary, secondary)

**Tags:** dnssec

Defines a key and signing policy (KASP) for zones.

### 8.2.25 *dnssec-policy* Block Definition and Usage

The *dnssec-policy* statement defines a key and signing policy (KASP) for zones.

A KASP determines how one or more zones are signed with DNSSEC. For example, it specifies how often keys should roll, which cryptographic algorithms to use, and how often RRSIG records need to be refreshed. Multiple key and signing policies can be configured with unique policy names.

A policy for a zone is selected using a *dnssec-policy* statement in the *zone* block, specifying the name of the policy that should be used.

**There are three built-in policies:**

- `default`, which uses the *default policy*;
- `insecure`, to be used when the zone should be unsigned gracefully; and
- `none`, which means no DNSSEC policy (the same as not selecting *dnssec-policy* at all; the zone is not signed).

Keys are not shared among zones, which means that one set of keys per zone is generated even if they have the same policy. If multiple views are configured with different versions of the same zone, each separate version uses the same set of signing keys.

If the expected key files that were previously observed have gone missing or are inaccessible, key management is halted. This will prevent rollovers from being started if there is a temporary file access issue. If his problem is permanent it will eventually lead to expired signatures in your zone. Note that if the key files are missing or inaccessible during *named* startup, BIND 9 will try to generate new keys according to the DNSSEC policy, because it has no cached information about existing keys yet.

The *dnssec-policy* statement requires dynamic DNS to be set up, or *inline-signing* to be enabled (which is the default for DNSSEC zones).

If *inline-signing* is enabled, this means that a signed version of the zone is maintained separately and is written out to a different file on disk (the zone's filename plus a `.signed` extension).

If *inline-signing* is disabled, the zone needs to be configured with an *update-policy* or *allow-update*. In such a case, the DNSSEC records are written to the filename set in the original zone's *file*.

Key rollover timing is computed for each key according to the key lifetime defined in the KASP. The lifetime may be modified by zone TTLs and propagation delays, to prevent validation failures. When a key reaches the end of its lifetime, *named* generates and publishes a new key automatically, then deactivates the old key and activates the new one; finally, the old key is retired according to a computed schedule.

Zone-signing key (ZSK) rollovers require no operator input. Key-signing key (KSK) and combined-signing key (CSK) rollovers require action to be taken to submit a DS record to the parent. Rollover timing for KSKs and CSKs is adjusted to take into account delays in processing and propagating DS updates.

The policy `default` causes the zone to be signed with a single combined-signing key (CSK) using the algorithm ECDSAP256SHA256; this key has an unlimited lifetime. This policy can be displayed using the command *named -C*.

**Note**

The default signing policy may change in future releases. This could require changes to a signing policy when upgrading to a new version of BIND. Check the release notes carefully when upgrading to be informed of such changes. To prevent policy changes on upgrade, use an explicitly defined *dnssec-policy*, rather than `default`.

If a *dnssec-policy* statement is modified and the server restarted or reconfigured, *named* attempts to change the policy smoothly from the old one to the new. For example, if the key algorithm is changed, then a new key is generated with the new algorithm, and the old algorithm is retired when the existing key's lifetime ends.

**Note**

Rolling to a new policy while another key rollover is already in progress is not yet supported, and may result in unexpected behavior.

The following options can be specified in a *dnssec-policy* statement:

**cdnskey**

**Grammar:** `cdnskey <boolean>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies whether a CDNSKEY record should be published during KSK rollover.

When set to the default value of `yes`, a CDNSKEY record is published during KSK rollovers when the DS of the successor key may be submitted to the parent.

**cds-digest-types**

**Grammar:** `cds-digest-types { <string>; ... };`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies the digest types to use for CDS resource records.

This indicates the digest types to use when generating CDS resource records. The default is SHA-256 only.

**dnskey-ttl**

**Grammar:** `dnskey-ttl <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies the time-to-live (TTL) for DNSKEY resource records.

This indicates the TTL to use when generating DNSKEY resource records. The default is 1 hour (3600 seconds).

**inline-signing**

**tags**
dnssec

**short**
Specifies whether BIND 9 maintains a separate signed version of a zone.

If `yes`, BIND 9 maintains a separate signed version of the zone. An unsigned zone is transferred in or loaded from disk and the signed version of the zone is served with, possibly, a different serial number. The signed version of the zone is stored in a file that is the zone's filename (set in *`file`*) with a `.signed` extension.

This behavior is enabled by default.

**keys**

**tags**
dnssec

**short**
Specifies the type of keys to be used for DNSSEC signing.

This is a list specifying the algorithms and roles to use when generating keys and signing the zone. Entries in this list do not represent specific DNSSEC keys, which may be changed on a regular basis, but the roles that keys play in the signing policy. For example, configuring a KSK of algorithm RSASHA256 ensures that the DNSKEY RRset always includes a key-signing key for that algorithm.

Here is an example (for illustration purposes only) of some possible entries in a `keys` list:

```
keys {
    ksk key-directory lifetime unlimited algorithm rsasha256 2048;
    zsk lifetime 30d algorithm 8 tag-range 0 32767;
    csk key-store "hsm" lifetime P6MT12H3M15S algorithm ecdsa256;
};
```

This example specifies that three keys should be used in the zone. The first token determines which role the key plays in signing RRsets. If set to `ksk`, then this is a key-signing key; it has the KSK flag set and is only used to sign DNSKEY, CDS, and CDNSKEY RRsets. If set to `zsk`, this is a zone-signing key; the KSK flag is unset, and the key signs all RRsets *except* DNSKEY, CDS, and CDNSKEY. If set to `csk`, the key has the KSK flag set and is used to sign all RRsets.

An optional second token determines where the key is stored. The two available options are `key-store <string>` and `key-directory`.

When using `key-store`, the referenced *`key-store`* describes how the key should be be stored. This can be as a file, or it can be inside a PKCS#11 token.

When using `key-directory`, the key is stored in the zone's configured *`key-directory`*. This is also the default.

When using `tag-range`, valid key tags for managed keys are restricted to this range [`tag-min tag-max`]. The optional `tag-range` is intended to be used in multi-signer scenarios. The default is unlimited ([0..65535]).

The `lifetime` parameter specifies how long a key may be used before rolling over. For convenience, TTL-style time-unit suffixes can be used to specify the key lifetime. It also accepts ISO 8601 duration formats.

In the example above, the first key has an unlimited lifetime, the second key may be used for 30 days, and the third key has a rather peculiar lifetime of 6 months, 12 hours, 3 minutes, and 15 seconds. A lifetime of 0 seconds is the same as `unlimited`.

Note that the lifetime of a key may be extended if retiring it too soon would cause validation failures. The key lifetime must be longer than the time it takes to do a rollover; that is, the lifetime must be more than the publication interval (which is the sum of *dnskey-ttl*, *publish-safety*, and *zone-propagation-delay*). It must also be more than the retire interval (which is the sum of *max-zone-ttl*, *retire-safety*, *zone-propagation-delay*, and signing delay (*signatures-validity* minus *signatures-refresh*) for ZSKs, and the sum of *parent-ds-ttl*, *retire-safety*, and *parent-propagation-delay* for KSKs and CSKs). BIND 9 treats a key lifetime that is too short as an error.

The `algorithm` parameter specifies the key's algorithm, expressed either as a string ("rsasha256", "ecdsa384", etc.) or as a decimal number. An optional second parameter specifies the key's size in bits. If it is omitted, as shown in the example for the second and third keys, an appropriate default size for the algorithm is used. Each KSK/ZSK pair must have the same algorithm. A CSK combines the functionality of a ZSK and a KSK.

**Note**

When changing the `key-directory` or the `key-store`, BIND will be unable to find existing key files. Be sure to copy key files to the new directory before changing the path used in the configuration file. This is also true when changing to a built-in policy, e.g. to `insecure`. In this specific case, the existing key files should be moved to the zone's `key-directory` from the new configuration.

**`manual-mode`**

**Grammar:** `manual-mode <boolean>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Run key management in a manual mode.

If enabled, BIND 9 does not automatically start and progress key rollovers, instead the change is logged. Only after manual confirmation with *rndc dnssec -step* the change is made.

This feature is off by default.

**`offline-ksk`**

**Grammar:** `offline-ksk <boolean>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies whether the DNSKEY, CDS, and CDNSKEY RRsets are being signed offline.

If enabled, BIND 9 does not generate signatures for the DNSKEY, CDS, and CDNSKEY RRsets. Instead, the signed DNSKEY, CDS and CDNSKEY RRsets are looked up from Signed Key Response (SKR) files.

Any existing DNSKEY, CDS, and CDNSKEY RRsets in the unsigned version of the zone are filtered and replaced with RRsets from the SKR file.

This feature is off by default. Configuring `offline-ksk` in conjunction with a CSK is a configuration error.

**purge-keys**

**Grammar:** `purge-keys <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies the amount of time after which DNSSEC keys that have been deleted from the zone can be removed from disk.

This is the amount of time after which DNSSEC keys that have been deleted from the zone can be removed from disk. If a key still determined to have presence (for example in some resolver cache), *named* will not remove the key files.

The default is `P90D` (90 days). Set this option to `0` to never purge deleted keys.

**publish-safety**

**Grammar:** `publish-safety <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Increases the amount of time between when keys are published and when they become active, to allow for unforeseen events.

This is a margin that is added to the pre-publication interval in rollover timing calculations, to give some extra time to cover unforeseen events. This increases the time between when keys are published and when they become active. The default is `PT1H` (1 hour).

**retire-safety**

**Grammar:** `retire-safety <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Increases the amount of time a key remains published after it is no longer active, to allow for unforeseen events.

This is a margin that is added to the post-publication interval in rollover timing calculations, to give some extra time to cover unforeseen events. This increases the time a key remains published after it is no longer active. The default is `PT1H` (1 hour).

**signatures-jitter**

**Grammar:** `signatures-jitter <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies a range for signature expirations.

To prevent all signatures from expiring at the same moment, BIND 9 may vary the validity interval of individual signatures. The validity of a newly generated signature is in the range between *signatures-validity* (maximum) and *signatures-validity*, minus *signatures-jitter* (minimum). The default jitter is 12 hours, and the configured value must be lower than both *signatures-validity* and *signatures-validity-dnskey*.

**signatures-refresh**

**Grammar:** `signatures-refresh <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies how frequently an RRSIG record is refreshed.

This determines how frequently an RRSIG record needs to be refreshed. The signature is renewed when the time until the expiration time is less than the specified interval. The default is `P5D` (5 days), meaning signatures that expire in 5 days or sooner are refreshed. The *signatures-refresh* value must be less than 90% of the minimum value of *signatures-validity* and *signatures-validity-dnskey*.

**signatures-validity**

**Grammar:** `signatures-validity <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Indicates the validity period of an RRSIG record.

This indicates the validity period of an RRSIG record (subject to inception offset and jitter). The default is `P2W` (2 weeks).

The *signatures-validity* should be at least several multiples of the SOA expire interval, to allow for reasonable interaction between the various timer and expiry dates.

**signatures-validity-dnskey**

**Grammar:** `signatures-validity-dnskey <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Indicates the validity period of DNSKEY records.

This is similar to *signatures-validity*, but for DNSKEY records. The default is `P2W` (2 weeks).

**max-zone-ttl**

**tags**
zone, query

**short**
Specifies a maximum permissible time-to-live (TTL) value, in seconds.

This specifies the maximum permissible TTL value for the zone. When a zone file is loaded, any record encountered with a TTL higher than *max-zone-ttl* causes the zone to be rejected.

This ensures that when rolling to a new DNSKEY, the old key will remain available until RRSIG records have expired from caches. The *max-zone-ttl* option guarantees that the largest TTL in the zone is no higher than a known and predictable value.

The default value PT24H (24 hours). A value of zero is treated as if the default value were in use.

**nsec3param**

**Grammar:** `nsec3param [ iterations <integer> ] [ optout <boolean> ] [ salt-length <integer> ];`

**Blocks:** dnssec-policy

**Tags:** dnssec

Specifies the use of NSEC3 instead of NSEC, and sets NSEC3 parameters.

Use NSEC3 instead of NSEC, and optionally set the NSEC3 parameters.

Here is an example of an nsec3 configuration:

```
nsec3param iterations 0 optout no salt-length 0;
```

The default is to use *NSEC*. The iterations, optout, and salt-length parts are optional, but if not set, the values in the example above are the default *NSEC3* parameters. Note that the specific salt string is not specified by the user; *named* creates a salt of the indicated length.

> **Warning**
>
> Do not use extra *iterations*, *salt*, and *opt-out* unless their implications are fully understood. A higher number of iterations causes interoperability problems and opens servers to CPU-exhausting DoS attacks. See **RFC 9276**.

**zone-propagation-delay**

**Grammar:** `zone-propagation-delay <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec, zone

Sets the propagation delay from the time a zone is first updated to when the new version of the zone is served by all secondary servers.

This is the expected propagation delay from the time when a zone is first updated to the time when the new version of the zone is served by all secondary servers. The default is PT5M (5 minutes).

**parent-ds-ttl**

**Grammar:** `parent-ds-ttl <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec

Sets the time to live (TTL) of the DS RRset used by the parent zone.

This is the TTL of the DS RRset that the parent zone uses. The default is `P1D` (1 day).

**`parent-propagation-delay`**

**Grammar:** `parent-propagation-delay <duration>;`

**Blocks:** dnssec-policy

**Tags:** dnssec, zone

Sets the propagation delay from the time the parent zone is updated to when the new version is served by all of the parent zone's name servers.

This is the expected propagation delay from the time when the parent zone is updated to the time when the new version is served by all of the parent zone's name servers. The default is `PT1H` (1 hour).

### Automated KSK Rollovers

BIND has mechanisms in place to facilitate automated KSK rollovers. It publishes CDS and CDNSKEY records that can be used by the parent zone to publish or withdraw the zone's DS records. BIND will query the parental agents to see if the new DS is actually published before withdrawing the old DNSSEC key.

> **Note**
>
> The DS response is not validated so it is recommended to set up a trust relationship with the parental agent. For example, use TSIG to authenticate the parental agent, or point to a validating resolver.

**`parental-agents`**

**Grammar:** `parental-agents [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };`

**Blocks:** zone (primary, secondary)

**Tags:** dnssec

This specifies a list of one or more IP addresses of parental agents that are used to query the zone's DS records during a KSK rollover. The list of parental agents can also contain the names of *`remote-servers`* blocks.

By default, DS queries are sent from port 53 on the servers; this can be changed for all servers by specifying a port number before the list of IP addresses, or on a per-server basis after the IP address. Authentication to the primary can also be done with per-server TSIG keys.

The following options apply to DS queries sent to *`parental-agents`*:

**`checkds`**

**Grammar:** `checkds ( explicit | <boolean> );`

**Blocks:** zone (primary, secondary)

**Tags:** dnssec

Controls whether `DS` queries are sent to parental agents.

If set to `yes`, DS queries are sent when a KSK rollover is in progress. The queries are sent to the servers listed in the parent zone's NS records. This is the default if there are no *`parental-agents`* configured for the zone.

If set to `explicit`, DS queries are sent only to servers explicitly listed using *`parental-agents`*. This is the default if there are parental agents configured.

If set to `no`, no DS queries are sent. Users should manually run *`rndc dnssec -checkds`* with the appropriate parameters, to signal that specific DS records are published and/or withdrawn.

**`parental-source`**

**Grammar:** `parental-source ( <ipv4_address> | * );`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Specifies which local IPv4 source address is used to send parental DS queries.

*`parental-source`* determines which local source address, and optionally UDP port, is used to send parental DS queries. This statement sets the *`parental-source`* for all zones, but can be overridden on a per-zone or per-view basis by including a *`parental-source`* statement within the *`zone`* or *`view`* block in the configuration file.

**Note**

`port` configuration is deprecated. A warning will be logged when this parameter is used.

**Warning**

Specifying a single port is discouraged, as it removes a layer of protection against spoofing errors.

**Warning**

The configured *port* must not be the same as the listening port.

**`parental-source-v6`**

**Grammar:** `parental-source-v6 ( <ipv6_address> | * );`

**Blocks:** options, view, zone (primary, secondary)

**Tags:** dnssec

Specifies which local IPv6 source address is used to send parental DS queries.

This option acts like *`parental-source`*, but applies to parental DS queries sent to IPv6 addresses.

### 8.2.26 *`managed-keys`* Block Grammar

**`managed-keys`**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `managed-keys { <string> ( static-key | initial-key | static-ds | initial-ds ) <integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times, deprecated`

**Blocks:** topmost, view

**Tags:** deprecated

### 8.2.27 *managed-keys* Block Definition and Usage

The *managed-keys* statement has been deprecated in favor of *trust-anchors* with the `initial-key` keyword.

### 8.2.28 *trusted-keys* Block Grammar

**trusted-keys**

> **Warning**
>
> This option is deprecated and will be removed in a future version of BIND.

**Grammar:** `trusted-keys { <string> <integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times, deprecated`

**Blocks:** topmost, view

**Tags:** deprecated

### 8.2.29 *trusted-keys* Block Definition and Usage

The *trusted-keys* statement has been deprecated in favor of *trust-anchors* with the `static-key` keyword.

### 8.2.30 *view* Block Grammar

**view**

**Grammar:**

```
view <string> [ <class> ] {
        allow-new-zones <boolean>;
        allow-notify { <address_match_element>; ... };
        allow-proxy { <address_match_element>; ... }; // experimental
        allow-proxy-on { <address_match_element>; ... }; // experimental
        allow-query { <address_match_element>; ... };
        allow-query-cache { <address_match_element>; ... };
        allow-query-cache-on { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-recursion { <address_match_element>; ... };
        allow-recursion-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
```

(continues on next page)

(continued from previous page)

```
        allow-update { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
        attach-cache <string>;
        auth-nxdomain <boolean>;
        catalog-zones { zone <string> [ default-primaries [ port <integer> ] [␣
↪source ( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { (
↪<server-list> | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port
↪<integer> ] ) [ key <string> ] [ tls <string> ]; ... } ] [ zone-directory
↪<quoted_string> ] [ in-memory <boolean> ] [ min-update-interval <duration> ]; ..
↪. };
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( primary | master | secondary | slave | response ) ( fail |␣
↪warn | ignore ); // may occur multiple times
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        clients-per-query <integer>;
        deny-answer-addresses { <address_match_element>; ... } [ except-from {
↪<string>; ... } ];
        deny-answer-aliases { <string>; ... } [ except-from { <string>; ... } ];
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        disable-algorithms <string> { <string>; ... }; // may occur multiple times
        disable-ds-digests <string> { <string>; ... }; // may occur multiple times
        disable-empty-zone <string>; // may occur multiple times
        dlz <string> {
                database <string>;
                search <boolean>;
        }; // may occur multiple times
        dns64 <netprefix> {
                break-dnssec <boolean>;
                clients { <address_match_element>; ... };
                exclude { <address_match_element>; ... };
                mapped { <address_match_element>; ... };
                recursive-only <boolean>;
                suffix <ipv6_address>;
        }; // may occur multiple times
        dns64-contact <string>;
        dns64-server <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnsrps-enable <boolean>;
        dnsrps-options { <unspecified-text> };
        dnssec-accept-expired <boolean>;
```

(continues on next page)

(continued from previous page)

```
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-must-be-secure <string> <boolean>; // may occur multiple times,␣
↪deprecated
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        dnssec-validation ( yes | no | auto );
        dnstap { ( all | auth | client | forwarder | resolver | update ) [ (␣
↪query | response ) ]; ... };
        dual-stack-servers [ port <integer> ] { ( <quoted_string> [ port
↪<integer> ] | <ipv4_address> [ port <integer> ] | <ipv6_address> [ port
↪<integer> ] ); ... };
        dyndb <string> <quoted_string> { <unspecified-text> }; // may occur␣
↪multiple times
        edns-udp-size <integer>;
        empty-contact <string>;
        empty-server <string>;
        empty-zones-enable <boolean>;
        fetch-quota-params <integer> <fixedpoint> <fixedpoint> <fixedpoint>;
        fetches-per-server <integer> [ ( drop | fail ) ];
        fetches-per-zone <integer> [ ( drop | fail ) ];
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        ipv4only-contact <string>;
        ipv4only-enable <boolean>;
        ipv4only-server <string>;
        ixfr-from-differences ( primary | master | secondary | slave | <boolean>␣
↪);
        key <string> {
                algorithm <string>;
                secret <string>;
        }; // may occur multiple times
        key-directory <quoted_string>;
        lame-ttl <duration>;
        lmdb-mapsize <sizeval>;
        managed-keys { <string> ( static-key | initial-key | static-ds |␣
↪initial-ds ) <integer> <integer> <integer> <quoted_string>; ... }; // may occur␣
↪multiple times, deprecated
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        match-clients { <address_match_element>; ... };
        match-destinations { <address_match_element>; ... };
        match-recursive-only <boolean>;
        max-cache-size ( default | unlimited | <sizeval> | <percentage> );
        max-cache-ttl <duration>;
        max-clients-per-query <integer>;
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-ncache-ttl <duration>;
        max-query-count <integer>;
```

(continues on next page)

(continued from previous page)

```
        max-query-restarts <integer>;
        max-records <integer>;
        max-records-per-type <integer>;
        max-recursion-depth <integer>;
        max-recursion-queries <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-stale-ttl <duration>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-udp-size <integer>;
        max-validation-failures-per-fetch <integer>; // experimental
        max-validations-per-fetch <integer>; // experimental
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        message-compression <boolean>;
        min-cache-ttl <duration>;
        min-ncache-ttl <duration>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        minimal-any <boolean>;
        minimal-responses ( no-auth | no-auth-recursive | <boolean> );
        multi-master <boolean>;
        new-zones-directory <quoted_string>;
        no-case-compress { <address_match_element>; ... };
        nocookie-udp-size <integer>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        nta-lifetime <duration>;
        nta-recheck <duration>;
        nxdomain-redirect <string>;
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        plugin ( query ) <string> [ { <unspecified-text> } ]; // may occur␣
↪multiple times
        preferred-glue <string>;
        prefetch <integer> [ <integer> ];
        provide-ixfr <boolean>;
        qname-minimization ( strict | relaxed | disabled | off );
        query-source [ address ] ( <ipv4_address> | * | none );
        query-source-v6 [ address ] ( <ipv6_address> | * | none );
        rate-limit {
                all-per-second <integer>;
                errors-per-second <integer>;
```

(continues on next page)

(continued from previous page)

```
                exempt-clients { <address_match_element>; ... };
                ipv4-prefix-length <integer>;
                ipv6-prefix-length <integer>;
                log-only <boolean>;
                max-table-size <integer>;
                min-table-size <integer>;
                nodata-per-second <integer>;
                nxdomains-per-second <integer>;
                qps-scale <integer>;
                referrals-per-second <integer>;
                responses-per-second <integer>;
                slip <integer>;
                window <integer>;
        };
        recursion <boolean>;
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-server-cookie <boolean>;
        resolver-query-timeout <integer>;
        resolver-use-dns64 <boolean>;
        response-padding { <address_match_element>; ... } block-size <integer>;
        response-policy { zone <string> [ add-soa <boolean> ] [ log <boolean> ] [␣
↪max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ policy ( cname␣
↪| disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only
↪<quoted_string> ) ] [ recursive-only <boolean> ] [ nsip-enable <boolean> ] [␣
↪nsdname-enable <boolean> ] [ ede <string> ]; ... } [ add-soa <boolean> ] [␣
↪break-dnssec <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval
↪<duration> ] [ min-ns-dots <integer> ] [ nsip-wait-recurse <boolean> ] [␣
↪nsdname-wait-recurse <boolean> ] [ qname-wait-recurse <boolean> ] [␣
↪recursive-only <boolean> ] [ servfail-until-ready <boolean> ] [ nsip-enable
↪<boolean> ] [ nsdname-enable <boolean> ] [ dnsrps-enable <boolean> ] [␣
↪dnsrps-options { <unspecified-text> } ];
        root-key-sentinel <boolean>;
        rrset-order { [ class <string> ] [ type <string> ] [ name <quoted_string>␣
↪] <string> <string>; ... };
        send-cookie <boolean>;
        serial-update-method ( date | increment | unixtime );
        server <netprefix> {
                bogus <boolean>;
                edns <boolean>;
                edns-udp-size <integer>;
                edns-version <integer>;
                keys <server_key>;
                max-udp-size <integer>;
                notify-source ( <ipv4_address> | * );
                notify-source-v6 ( <ipv6_address> | * );
                padding <integer>;
                provide-ixfr <boolean>;
                query-source [ address ] ( <ipv4_address> | * );
                query-source-v6 [ address ] ( <ipv6_address> | * );
                request-expire <boolean>;
```

(continues on next page)

(continued from previous page)

```
                request-ixfr <boolean>;
                request-nsid <boolean>;
                require-cookie <boolean>;
                send-cookie <boolean>;
                tcp-keepalive <boolean>;
                tcp-only <boolean>;
                transfer-format ( many-answers | one-answer );
                transfer-source ( <ipv4_address> | * );
                transfer-source-v6 ( <ipv6_address> | * );
                transfers <integer>;
        }; // may occur multiple times
        servfail-ttl <duration>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        sig0key-checks-limit <integer>;
        sig0message-checks-limit <integer>;
        sortlist { <address_match_element>; ... }; // deprecated
        stale-answer-client-timeout ( disabled | off | <integer> );
        stale-answer-enable <boolean>;
        stale-answer-ttl <duration>;
        stale-cache-enable <boolean>;
        stale-refresh-time <duration>;
        synth-from-dnssec <boolean>;
        transfer-format ( many-answers | one-answer );
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        trust-anchor-telemetry <boolean>;
        trust-anchors { <string> ( static-key | initial-key | static-ds |␣
↪initial-ds ) <integer> <integer> <integer> <quoted_string>; ... }; // may occur␣
↪multiple times
        trusted-keys { <string> <integer> <integer> <integer> <quoted_string>; ...
↪ }; // may occur multiple times, deprecated
        try-tcp-refresh <boolean>;
        update-check-ksk <boolean>; // obsolete
        v6-bias <integer>;
        validate-except { <string>; ... };
        zero-no-soa-ttl <boolean>;
        zero-no-soa-ttl-cache <boolean>;
        zone <string> [ <class> ] {
                in-view <string>;
        };
        zone <string> [ <class> ] {
                type forward;
                forward ( first | only );
                forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address>␣
↪| <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        };
        zone <string> [ <class> ] {
                type hint;
                check-names ( fail | warn | ignore );
```

(continues on next page)

(continued from previous page)

```
                file <quoted_string>;
        };
        zone <string> [ <class> ] {
                type mirror;
                allow-notify { <address_match_element>; ... };
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
                allow-update-forwarding { <address_match_element>; ... };
                also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ]␣
↪[ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                check-names ( fail | warn | ignore );
                database <string>;
                file <quoted_string>;
                ixfr-from-differences <boolean>;
                journal <quoted_string>;
                masterfile-format ( raw | text );
                masterfile-style ( full | relative );
                max-ixfr-ratio ( unlimited | <percentage> );
                max-journal-size ( default | unlimited | <sizeval> );
                max-records <integer>;
                max-records-per-type <integer>;
                max-refresh-time <integer>;
                max-retry-time <integer>;
                max-transfer-idle-in <integer>;
                max-transfer-idle-out <integer>;
                max-transfer-time-in <integer>;
                max-transfer-time-out <integer>;
                max-types-per-name <integer>;
                min-refresh-time <integer>;
                min-retry-time <integer>;
                min-transfer-rate-in <integer> <integer>;
                multi-master <boolean>;
                notify ( explicit | master-only | primary-only | <boolean> );
                notify-defer <integer>;
                notify-delay <integer>;
                notify-source ( <ipv4_address> | * );
                notify-source-v6 ( <ipv6_address> | * );
                primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                request-expire <boolean>;
                request-ixfr <boolean>;
                transfer-source ( <ipv4_address> | * );
                transfer-source-v6 ( <ipv6_address> | * );
                try-tcp-refresh <boolean>;
                zero-no-soa-ttl <boolean>;
                zone-statistics ( full | terse | none | <boolean> );
```

(continues on next page)

(continued from previous page)

```
        };
        zone <string> [ <class> ] {
                type primary;
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
                allow-update { <address_match_element>; ... };
                also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ]␣
↪[ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                check-dup-records ( fail | warn | ignore );
                check-integrity <boolean>;
                check-mx ( fail | warn | ignore );
                check-mx-cname ( fail | warn | ignore );
                check-names ( fail | warn | ignore );
                check-sibling <boolean>;
                check-spf ( warn | ignore );
                check-srv-cname ( fail | warn | ignore );
                check-svcb <boolean>;
                check-wildcard <boolean>;
                checkds ( explicit | <boolean> );
                database <string>;
                dialup ( notify | notify-passive | passive | refresh | <boolean>␣
↪); // deprecated
                dlz <string>;
                dnskey-sig-validity <integer>; // obsolete
                dnssec-dnskey-kskonly <boolean>; // obsolete
                dnssec-loadkeys-interval <integer>;
                dnssec-policy <string>;
                dnssec-secure-to-insecure <boolean>; // obsolete
                dnssec-update-mode ( maintain | no-resign ); // obsolete
                file <quoted_string>;
                forward ( first | only );
                forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address>␣
↪| <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
                inline-signing <boolean>;
                ixfr-from-differences <boolean>;
                journal <quoted_string>;
                key-directory <quoted_string>;
                masterfile-format ( raw | text );
                masterfile-style ( full | relative );
                max-ixfr-ratio ( unlimited | <percentage> );
                max-journal-size ( default | unlimited | <sizeval> );
                max-records <integer>;
                max-records-per-type <integer>;
                max-transfer-idle-out <integer>;
                max-transfer-time-out <integer>;
                max-types-per-name <integer>;
                max-zone-ttl ( unlimited | <duration> ); // deprecated
                notify ( explicit | master-only | primary-only | <boolean> );
```

(continues on next page)

(continued from previous page)

```
                notify-defer <integer>;
                notify-delay <integer>;
                notify-source ( <ipv4_address> | * );
                notify-source-v6 ( <ipv6_address> | * );
                notify-to-soa <boolean>;
                nsec3-test-zone <boolean>; // test only
                parental-agents [ port <integer> ] [ source ( <ipv4_address> | *␣
↪) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [␣
↪port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                parental-source ( <ipv4_address> | * );
                parental-source-v6 ( <ipv6_address> | * );
                serial-update-method ( date | increment | unixtime );
                sig-signing-nodes <integer>;
                sig-signing-signatures <integer>;
                sig-signing-type <integer>;
                sig-validity-interval <integer> [ <integer> ]; // obsolete
                update-check-ksk <boolean>; // obsolete
                update-policy ( local | { ( deny | grant ) <string> ( 6to4-self |␣
↪external | krb5-self | krb5-selfsub | krb5-subdomain | krb5-subdomain-self-rhs␣
↪| ms-self | ms-selfsub | ms-subdomain | ms-subdomain-self-rhs | name | self |␣
↪selfsub | selfwild | subdomain | tcp-self | wildcard | zonesub ) [ <string> ]
↪<rrtypelist>; ... } );
                zero-no-soa-ttl <boolean>;
                zone-statistics ( full | terse | none | <boolean> );
        };
        zone <string> [ <class> ] {
                type redirect;
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                dlz <string>;
                file <quoted_string>;
                masterfile-format ( raw | text );
                masterfile-style ( full | relative );
                max-records <integer>;
                max-records-per-type <integer>;
                max-types-per-name <integer>;
                max-zone-ttl ( unlimited | <duration> ); // deprecated
                primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                zone-statistics ( full | terse | none | <boolean> );
        };
        zone <string> [ <class> ] {
                type secondary;
                allow-notify { <address_match_element>; ... };
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
                allow-update-forwarding { <address_match_element>; ... };
```

(continues on next page)

(continued from previous page)

```
                also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ]␣
↪[ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                check-names ( fail | warn | ignore );
                checkds ( explicit | <boolean> );
                database <string>;
                dialup ( notify | notify-passive | passive | refresh | <boolean>␣
↪); // deprecated
                dlz <string>;
                dnskey-sig-validity <integer>; // obsolete
                dnssec-dnskey-kskonly <boolean>; // obsolete
                dnssec-loadkeys-interval <integer>;
                dnssec-policy <string>;
                dnssec-update-mode ( maintain | no-resign ); // obsolete
                file <quoted_string>;
                forward ( first | only );
                forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address>␣
↪| <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
                inline-signing <boolean>;
                ixfr-from-differences <boolean>;
                journal <quoted_string>;
                key-directory <quoted_string>;
                masterfile-format ( raw | text );
                masterfile-style ( full | relative );
                max-ixfr-ratio ( unlimited | <percentage> );
                max-journal-size ( default | unlimited | <sizeval> );
                max-records <integer>;
                max-records-per-type <integer>;
                max-refresh-time <integer>;
                max-retry-time <integer>;
                max-transfer-idle-in <integer>;
                max-transfer-idle-out <integer>;
                max-transfer-time-in <integer>;
                max-transfer-time-out <integer>;
                max-types-per-name <integer>;
                min-refresh-time <integer>;
                min-retry-time <integer>;
                min-transfer-rate-in <integer> <integer>;
                multi-master <boolean>;
                notify ( explicit | master-only | primary-only | <boolean> );
                notify-defer <integer>;
                notify-delay <integer>;
                notify-source ( <ipv4_address> | * );
                notify-source-v6 ( <ipv6_address> | * );
                notify-to-soa <boolean>;
                nsec3-test-zone <boolean>; // test only
                parental-agents [ port <integer> ] [ source ( <ipv4_address> | *␣
↪) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [␣
↪port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                parental-source ( <ipv4_address> | * );
```

(continues on next page)

(continued from previous page)

```
                parental-source-v6 ( <ipv6_address> | * );
                primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                request-expire <boolean>;
                request-ixfr <boolean>;
                sig-signing-nodes <integer>;
                sig-signing-signatures <integer>;
                sig-signing-type <integer>;
                sig-validity-interval <integer> [ <integer> ]; // obsolete
                transfer-source ( <ipv4_address> | * );
                transfer-source-v6 ( <ipv6_address> | * );
                try-tcp-refresh <boolean>;
                update-check-ksk <boolean>; // obsolete
                zero-no-soa-ttl <boolean>;
                zone-statistics ( full | terse | none | <boolean> );
        };
        zone <string> [ <class> ] {
                type static-stub;
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                forward ( first | only );
                forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address>␣
↪| <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
                max-records <integer>;
                max-records-per-type <integer>;
                max-types-per-name <integer>;
                server-addresses { ( <ipv4_address> | <ipv6_address> ); ... };
                server-names { <string>; ... };
                zone-statistics ( full | terse | none | <boolean> );
        };
        zone <string> [ <class> ] {
                type stub;
                allow-query { <address_match_element>; ... };
                allow-query-on { <address_match_element>; ... };
                check-names ( fail | warn | ignore );
                database <string>;
                dialup ( notify | notify-passive | passive | refresh | <boolean>␣
↪); // deprecated
                file <quoted_string>;
                forward ( first | only );
                forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address>␣
↪| <ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
                masterfile-format ( raw | text );
                masterfile-style ( full | relative );
                max-records <integer>;
                max-records-per-type <integer>;
                max-refresh-time <integer>;
                max-retry-time <integer>;
                max-transfer-idle-in <integer>;
                max-transfer-time-in <integer>;
```

(continues on next page)

(continued from previous page)

```
                max-types-per-name <integer>;
                min-refresh-time <integer>;
                min-retry-time <integer>;
                min-transfer-rate-in <integer> <integer>;
                multi-master <boolean>;
                primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
                transfer-source ( <ipv4_address> | * );
                transfer-source-v6 ( <ipv6_address> | * );
                zone-statistics ( full | terse | none | <boolean> );
        };

        zone-statistics ( full | terse | none | <boolean> );
}; // may occur multiple times
```

**Blocks:** topmost

**Tags:** view

Allows a name server to answer a DNS query differently depending on who is asking.

```
view view_name [ class ] {
    match-clients { address_match_list } ;
    match-destinations { address_match_list } ;
    match-recursive-only <boolean> ;
  [ view_option ; ... ]
  [ zone_statement ; ... ]
} ;
```

### 8.2.31 *view* Block Definition and Usage

The *view* statement is a powerful feature of BIND 9 that lets a name server answer a DNS query differently depending on who is asking. It is particularly useful for implementing split DNS setups without having to run multiple servers.

**match-clients**

**Grammar:** `match-clients { <address_match_element>; ... };`

**Blocks:** view

**Tags:** view

Specifies a view of DNS namespace for a given subset of client IP addresses.

**match-destinations**

**Grammar:** `match-destinations { <address_match_element>; ... };`

**Blocks:** view

**Tags:** view

Specifies a view of DNS namespace for a given subset of destination IP addresses.

Each *view* statement defines a view of the DNS namespace that is seen by a subset of clients. A client matches a view if its source IP address matches the *address_match_list* of the view's *match-clients* clause, and its destination IP address matches the *address_match_list* of the view's *match-destinations* clause. If not specified, both *match-clients* and *match-destinations* default to matching all addresses. In addition to checking IP addresses, *match-clients* and *match-destinations* can also take the name of a TSIG *key*, which provides a mechanism for the client to select the view.

**match-recursive-only**

**Grammar:** `match-recursive-only <boolean>;`

**Blocks:** view

**Tags:** view

Specifies that only recursive requests can match this view of the DNS namespace.

A view can also be specified as *match-recursive-only*, which means that only recursive requests from matching clients match that view. The order of the *view* statements is significant; a client request is resolved in the context of the first *view* that it matches.

Zones defined within a *view* statement are only accessible to clients that match the *view*. By defining a zone of the same name in multiple views, different zone data can be given to different clients: for example, "internal" and "external" clients in a split DNS setup.

Many of the options given in the *options* statement can also be used within a *view* statement, and then apply only when resolving queries with that view. When no view-specific value is given, the value in the *options* statement is used as a default. Also, zone options can have default values specified in the *view* statement; these view-specific defaults take precedence over those in the *options* statement.

Views are class-specific. If no class is given, class IN is assumed. Note that all non-IN views must contain a hint zone, since only the IN class has compiled-in default hints.

If there are no *view* statements in the config file, a default view that matches any client is automatically created in class IN. Any *zone* statements specified on the top level of the configuration file are considered to be part of this default view, and the *options* statement applies to the default view. If any explicit *view* statements are present, all *zone* statements must occur inside *view* statements.

Here is an example of a typical split DNS setup implemented using *view* statements:

```
view "internal" {
      // This should match our internal networks.
      match-clients { 10.0.0.0/8; };

      // Provide recursive service to internal
      // clients only.
      recursion yes;

      // Provide a complete view of the example.com
      // zone including addresses of internal hosts.
      zone "example.com" {
        type primary;
        file "example-internal.db";
      };
};

view "external" {
      // Match all clients not matched by the
```

(continues on next page)

(continued from previous page)

```
        // previous view.
        match-clients { any; };

        // Refuse recursive service to external clients.
        recursion no;

        // Provide a restricted view of the example.com
        // zone containing only publicly accessible hosts.
        zone "example.com" {
         type primary;
         file "example-external.db";
        };
};
```

### 8.2.32 *zone* Block Grammar

**zone**

**Blocks:** topmost, view

**Tags:** zone

Specifies the zone in a BIND 9 configuration.

### 8.2.33 *zone* Block Definition and Usage

#### Zone Types

**type**

**Blocks:** zone (forward, hint, mirror, primary, redirect, secondary, static-stub, stub)

**Tags:** zone

Specifies the kind of zone in a given configuration.

The *type* keyword is required for the *zone* configuration unless it is an *in-view* configuration. Its acceptable values are: *primary* (or `master`), *secondary* (or `slave`), *mirror*, *hint*, *stub*, *static-stub*, *forward*, or *redirect*.

**type primary**

**Grammar:**

```
zone <string> [ <class> ] {
        type primary;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
```

(continues on next page)

(continued from previous page)

```
↪;
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( fail | warn | ignore );
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        checkds ( explicit | <boolean> );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        dlz <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        file <quoted_string>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        inline-signing <boolean>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
        key-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        parental-agents [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
        parental-source ( <ipv4_address> | * );
```

(continues on next page)

(continued from previous page)

```
        parental-source-v6 ( <ipv6_address> | * );
        serial-update-method ( date | increment | unixtime );
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        update-check-ksk <boolean>; // obsolete
        update-policy ( local | { ( deny | grant ) <string> ( 6to4-self |␣
↪external | krb5-self | krb5-selfsub | krb5-subdomain | krb5-subdomain-self-rhs␣
↪| ms-self | ms-selfsub | ms-subdomain | ms-subdomain-self-rhs | name | self |␣
↪selfsub | selfwild | subdomain | tcp-self | wildcard | zonesub ) [ <string> ]
↪<rrtypelist>; ... } );
        zero-no-soa-ttl <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains the main copy of the data for a zone.

A primary zone has a master copy of the data for the zone and is able to provide authoritative answers for it. Type `master` is a synonym for *primary*.

**`type secondary`**

**Grammar:**

```
zone <string> [ <class> ] {
        type secondary;
        allow-notify { <address_match_element>; ... };
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        check-names ( fail | warn | ignore );
        checkds ( explicit | <boolean> );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        dlz <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-policy <string>;
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        file <quoted_string>;
```

(continues on next page)

(continued from previous page)

```
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        inline-signing <boolean>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
        key-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        parental-agents [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls
↪<string> ]; ... };
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        request-expire <boolean>;
        request-ixfr <boolean>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        try-tcp-refresh <boolean>;
        update-check-ksk <boolean>; // obsolete
        zero-no-soa-ttl <boolean>;
```

(continues on next page)

(continued from previous page)

```
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains a duplicate of the data for a zone that has been transferred from a primary server.

A secondary zone is a replica of a primary zone. Type `slave` is a synonym for *`secondary`*. The *`primaries`* list specifies one or more IP addresses of primary servers that the secondary contacts to update its copy of the zone.

A zone may refresh on timer or on receipt of a notify. If a valid notify is received where the notify carries a serial number larger than the one in the SOA currently served, then the secondary will schedule a zone refresh.

A notify is considered valid if the sender is one of the servers in the NS RRset for the zone, has been explicitly allowed using an *`allow-notify`* clause, or is from an address listed in the primary servers clause.

If no notifies have been received, the server will try to refresh the zone. The REFRESH field in the SOA record determines how long after the last zone update it should query the primaries for the SOA record. Again, if the SOA record contains a serial number larger than the one in the SOA currently served, a zone refresh is scheduled. If a notify is received while a refresh is in progress, the serial number of the notify is checked and if it is larger, another refresh for the zone is queued. There will at most be one zone refresh queued.

The primary servers are queried in turn, *`named`* will move on to the next server in the list if either it is unable to get a valid response from the server it is currently querying, or the primary being queried returns the same or smaller SOA than the secondary is currently serving. On the first SOA received that has a serial bigger than the one currently served, *`named`* will initiate a zone transfer with that server. Once the zone transfer has been received and the zone has been updated, then this zone refresh is complete, and no other servers are tried.

When receiving a notify, *`named`* does not first query the sender of the notify. It will continue with the next server in the list that transferred the zone, skipping over unreachable servers. A primary is considered unreachable if the secondary cannot get a response from the server. This state will be cached for 10 minutes, or until a notify is received from that address.

Furthermore, a zone is refreshed when the secondary server is restarted, or when a *`rndc refresh`* command is received.

If a file is specified, then the replica is written to this file whenever the zone is changed, and reloaded from this file on a server restart. Use of a file is recommended, since it often speeds server startup and eliminates a needless waste of bandwidth. Note that for large numbers (in the tens or hundreds of thousands) of zones per server, it is best to use a two-level naming scheme for zone filenames. For example, a secondary server for the zone `example.com` might place the zone contents into a file called `ex/example.com`, where `ex/` is just the first two letters of the zone name. (Most operating systems behave very slowly if there are 100,000 files in a single directory.)

**`type mirror`**

**Grammar:**

```
zone <string> [ <class> ] {
        type mirror;
        allow-notify { <address_match_element>; ... };
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
```

(continues on next page)

(continued from previous page)

```
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        check-names ( fail | warn | ignore );
        database <string>;
        file <quoted_string>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        request-expire <boolean>;
        request-ixfr <boolean>;
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        try-tcp-refresh <boolean>;
        zero-no-soa-ttl <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains a DNSSEC-validated duplicate of the main data for a zone.

A mirror zone is similar to a zone of *type secondary*, except its data is subject to DNSSEC validation before being used in answers. Validation is applied to the entire zone during the zone transfer process, and again when the zone file is loaded from disk upon restarting *named*. If validation of a new version of a mirror zone fails, a

retransfer is scheduled; in the meantime, the most recent correctly validated version of that zone is used until it either expires or a newer version validates correctly. If no usable zone data is available for a mirror zone, due to either transfer failure or expiration, traditional DNS recursion is used to look up the answers instead. Mirror zones cannot be used in a view that does not have recursion enabled.

Answers coming from a mirror zone look almost exactly like answers from a zone of *type secondary*, with the notable exceptions that the AA bit ("authoritative answer") is not set, and the AD bit ("authenticated data") is.

Mirror zones are intended to be used to set up a fast local copy of the root zone (see **RFC 8806**). A default list of primary servers for the IANA root zone is built into *named*, so its mirroring can be enabled using the following configuration:

```
zone "." {
    type mirror;
};
```

Mirror zone validation always happens for the entire zone contents. This ensures that each version of the zone used by the resolver is fully self-consistent with respect to DNSSEC. For incoming mirror zone IXFRs, every revision of the zone contained in the IXFR sequence is validated independently, in the order in which the zone revisions appear on the wire. For this reason, it might be useful to force use of AXFR for mirror zones by setting `request-ixfr no;` for the relevant zone (or view). Other, more efficient zone verification methods may be added in the future.

To make mirror zone contents persist between *named* restarts, use the *file* option.

Mirroring a zone other than root requires an explicit list of primary servers to be provided using the *primaries* option (see *primaries* for details), and a key-signing key (KSK) for the specified zone to be explicitly configured as a trust anchor (see *trust-anchors*).

When configuring NOTIFY for a mirror zone, only `notify no;` and `notify explicit;` can be used at the zone level; any other *notify* setting at the zone level is a configuration error. Using any other *notify* setting at the *options* or *view* level causes that setting to be overridden with `notify explicit;` for the mirror zone. The global default for the *notify* option is `yes`, so mirror zones are by default configured with `notify explicit;`.

Outgoing transfers of mirror zones are disabled by default but may be enabled using *allow-transfer*.

> **Note**
>
> Use of this zone type with any zone other than the root should be considered *experimental* and may cause performance issues, especially for zones that are large and/or frequently updated.

**type hint**

**Grammar:**

```
zone <string> [ <class> ] {
        type hint;
        check-names ( fail | warn | ignore );
        file <quoted_string>;
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains the initial set of root name servers to be used at BIND 9 startup.

The initial set of root name servers is specified using a hint zone. When the server starts, it uses the root hints to find a root name server and get the most recent list of root name servers. If no hint zone is specified for class IN, the server uses a compiled-in default set of root servers hints. Classes other than IN have no built-in default hints.

**type stub**

**Grammar:**

```
zone <string> [ <class> ] {
        type stub;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        check-names ( fail | warn | ignore );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        file <quoted_string>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-time-in <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains a duplicate of the NS records of a zone.

A stub zone specifies a set of name servers to use when contacting the zone for the first time. A stub zone overrides any NS records (delegations) that might exist in the parent zone. Once an authoritative server is reached, the NS records from that server are honored.

A stub zone can work around missing or broken delegations, but comes at expense of maintaining a set of name server addresses in *named.conf*.

> **Warning**
>
> Use of stub zones is not recommended. Proper delegation with NS records in the parent zone should be used.

*named* queries authoritative servers configured as *primaries* to obtain up-to-date NS records. These new NS records are then used to obtain answers from a given zone.

Stub zones can also be used as a way to force the resolution of a given domain to use a particular set of authoritative servers. For example, the caching name servers on a private network using **RFC 1918** addressing may be configured with stub zones for `10.in-addr.arpa` to use a set of internal name servers as the authoritative servers for that domain.

If a BIND 9 primary, serving a parent zone, has child stub zones configured, all the secondary servers for the parent zone also need to have the same child stub zones configured.

Stub zones are not a standard part of the DNS; they are a feature specific to the BIND implementation.

**type static-stub**

**Grammar:**

```
zone <string> [ <class> ] {
        type static-stub;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        max-records <integer>;
        max-records-per-type <integer>;
        max-types-per-name <integer>;
        server-addresses { ( <ipv4_address> | <ipv6_address> ); ... };
        server-names { <string>; ... };
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains statically configured NS records for a zone.

A static-stub zone specifies a set of name servers to use to resolve *all* queries for the given zone. A stub zone overrides any NS records (delegations) that might exist in the parent zone, and also any records received from the otherwise-authoritative server.

Like a *stub* zone, this can work around missing or broken delegations at the parent. Unlike stub, static-stub also overrides any NS records offered by the specified servers.

When recursion is necessary for a query that matches a static-stub zone, the locally configured data (name server names and glue addresses) is always used, even if different authoritative information is cached.

The zone data is configured via the *server-addresses* and *server-names* zone statements. These must point to authoritative servers.

> **Warning**
>
> Use of static-stub zones is not recommended. Proper delegation with NS records in the parent zone should be used.

The zone data is maintained in the form of NS and (if necessary) glue A or AAAA RRs internally, which can be seen by dumping zone databases with *rndc dumpdb -all*. The configured RRs are considered local configuration parameters rather than public data. Non-recursive queries (i.e., those with the RD bit off) to a static-stub zone are therefore prohibited and are responded to with REFUSED.

Since the data is statically configured, no zone maintenance action takes place for a static-stub zone. For example, there is no periodic refresh attempt, and an incoming *NOTIFY* message is rejected with an rcode of NOTAUTH.

Each static-stub zone is configured with internally generated NS and (if necessary) glue A or AAAA RRs.

**type forward**

**Grammar:**

```
zone <string> [ <class> ] {
        type forward;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains forwarding statements that apply to queries within a given domain.

A forward zone is a way to configure forwarding on a per-domain basis. A *zone* statement of type *forward* can contain a *forward* and/or *forwarders* statement, which applies to queries within the domain given by the zone name. If no *forwarders* statement is present, or an empty list for *forwarders* is given, then no forwarding is done for the domain, canceling the effects of any forwarders in the *options* statement. Thus, to use this type of zone to change the behavior of the global *forward* option (that is, "forward first" to, then "forward only", or vice versa), but use the same servers as set globally, re-specify the global forwarders.

**type redirect**

**Grammar:**

```
zone <string> [ <class> ] {
        type redirect;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        dlz <string>;
        file <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-records <integer>;
        max-records-per-type <integer>;
        max-types-per-name <integer>;
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
```

(continues on next page)

(continued from previous page)

```
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer>␣
↪] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... }
↪;
        zone-statistics ( full | terse | none | <boolean> );
};
```

**Blocks:** zone, view.zone

**Tags:** zone

Contains information to answer queries when normal resolution would return NXDOMAIN.

Redirect zones are used to provide answers to queries when normal resolution would result in NXDOMAIN being returned. Only one redirect zone is supported per view. *allow-query* can be used to restrict which clients see these answers.

If the client has requested DNSSEC records (DO=1) and the NXDOMAIN response is signed, no substitution occurs.

To redirect all NXDOMAIN responses to 100.100.100.2 and 2001:fff:ffff::100.100.100.2, configure a type *redirect* zone named ".", with the zone file containing wildcard records that point to the desired addresses: `*. IN A 100.100.100.2` and `*. IN AAAA 2001:ffff:ffff::100.100.100.2`.

As another example, to redirect all Spanish names (under .ES), use similar entries but with the names `*.ES.` instead of `*.`. To redirect all commercial Spanish names (under COM.ES), use wildcard entries called `*.COM.ES.`.

Note that the redirect zone supports all possible types; it is not limited to A and AAAA records.

If a redirect zone is configured with a *primaries* option, then it is transferred in as if it were a secondary zone. Otherwise, it is loaded from a file as if it were a primary zone.

Because redirect zones are not referenced directly by name, they are not kept in the zone lookup table with normal primary and secondary zones. To reload a redirect zone, use *rndc reload -redirect*; to retransfer a redirect zone configured as a secondary, use *rndc retransfer -redirect*. When using *rndc reload* without specifying a zone name, redirect zones are reloaded along with other zones.

**in-view**

**Grammar zone, view.zone:**

```
zone <string> [ <class> ] {
        in-view <string>;
};
```

**Grammar zone (in-view):** `in-view <string>;`

**Blocks:** zone, zone (in-view), view.zone

**Tags:** view, zone

Specifies the view in which a given zone is defined.

When using multiple views, a *type primary* or *type secondary* zone configured in one view can be referenced in a subsequent view. This allows both views to use the same zone without the overhead of loading it more than once. This is configured using a *zone* statement, with an *in-view* option specifying the view in which the zone is defined. A *zone* statement containing *in-view* does not need to specify a type, since that is part of the zone definition in the other view.

See *Multiple Views* for more information.

### Class

The zone's name may optionally be followed by a class. If a class is not specified, class `IN` (for `Internet`) is assumed. This is correct for the vast majority of cases.

The `hesiod` class is named for an information service from MIT's Project Athena. It was used to share information about various systems databases, such as users, groups, printers, and so on. The keyword `HS` is a synonym for hesiod.

Another MIT development is Chaosnet, a LAN protocol created in the mid-1970s. Zone data for it can be specified with the `CHAOS` class.

### Zone Options

***allow-notify***
See the description of *allow-notify* in *Access Control*.

***allow-query***
See the description of *allow-query* in *Access Control*.

***allow-query-on***
See the description of *allow-query-on* in *Access Control*.

***allow-transfer***
See the description of *allow-transfer* in *Access Control*.

***allow-update***
See the description of *allow-update* in *Access Control*.

***update-policy***
This specifies a "Simple Secure Update" policy. See *Dynamic Update Policies*.

***allow-update-forwarding***
See the description of *allow-update-forwarding* in *Access Control*.

***also-notify***
This option is only meaningful if *notify* is active for this zone. The set of machines that receive a `DNS NOTIFY` message for this zone is made up of all the listed name servers (other than the primary) for the zone, plus any IP addresses specified with *also-notify*. A port may be specified with each *also-notify* address to send the notify messages to a port other than the default of 53. A TSIG key may also be specified to cause the `NOTIFY` to be signed by the given key. *also-notify* is not meaningful for stub zones. The default is the empty list.

***check-names***
This option is used to restrict the character set and syntax of certain domain names in primary files and/or DNS responses received from the network. The default varies according to zone type. For *primary* zones the default is `fail`; for *secondary* zones the default is `warn`. It is not implemented for *hint* zones.

***check-mx***
See the description of *check-mx* in *Boolean Options*.

***check-spf***
See the description of *check-spf* in *Boolean Options*.

***check-wildcard***
See the description of *check-wildcard* in *Boolean Options*.

***check-integrity***
See the description of *check-integrity* in *Boolean Options*.

***check-sibling***
See the description of *check-sibling* in *Boolean Options*.

***zero-no-soa-ttl***
See the description of *zero-no-soa-ttl* in *Boolean Options*.

***update-check-ksk***
See the description of *update-check-ksk* in *Boolean Options*.

***dnssec-loadkeys-interval***
See the description of *dnssec-loadkeys-interval* in *options*.

***dnssec-update-mode***
See the description of *dnssec-update-mode* in *options*.

***dnssec-dnskey-kskonly***
See the description of *dnssec-dnskey-kskonly* in *Boolean Options*.

***try-tcp-refresh***
See the description of *try-tcp-refresh* in *Boolean Options*.

**database**

**Grammar:** `database <string>;`

**Blocks:** dlz, zone (mirror, primary, secondary, stub), view.dlz

**Tags:** zone

Specifies the type of database to be used to store zone data.

This specifies the type of database to be used to store the zone data. The string following the *database* keyword is interpreted as a list of whitespace-delimited words. The first word identifies the database type, and any subsequent words are passed as arguments to the database to be interpreted in a way specific to the database type.

The default is `rbt`, BIND 9's native in-memory red-black tree database. This database does not take arguments.

Other values are possible if additional database drivers have been linked into the server. Some sample drivers are included with the distribution but none are linked in by default.

***dialup***
See the description of *dialup* in *Boolean Options*.

**file**

**Grammar logging.channel:** `file <quoted_string> [ versions ( unlimited | <integer> ) ] [ size <size> ] [ suffix ( increment | timestamp ) ];`

**Grammar zone (hint, mirror, primary, redirect, secondary, stub):** `file <quoted_string>;`

**Blocks:** zone (hint, mirror, primary, redirect, secondary, stub), logging.channel

**Tags:** zone

Specifies the zone's filename.

This sets the zone's filename. In *primary*, *hint*, and *redirect* zones which do not have *primaries* defined, zone data is loaded from this file. In *secondary*, *mirror*, *stub*, and *redirect* zones which do have *primaries* defined, zone data is retrieved from another server and saved in this file. This option is not applicable to other zone types.

***forward***
This option is only meaningful if the zone has a forwarders list. The `only` value causes the lookup to fail after trying the forwarders and getting no answer, while `first` allows a normal lookup to be tried.

***forwarders***
This is used to override the list of global forwarders. If it is not specified in a zone of type *forward*, no forwarding is done for the zone and the global options are not used.

**journal**

**Grammar:** `journal <quoted_string>;`

**Blocks:** zone (mirror, primary, secondary)

**Tags:** zone

Allows the default journal's filename to be overridden.

This allows the default journal's filename to be overridden. The default is the zone's filename with "`.jnl`" appended. This is applicable to *primary* and *secondary* zones.

***max-ixfr-ratio***
See the description of *max-ixfr-ratio* in *options*.

***max-journal-size***
See the description of *max-journal-size* in *Server Resource Limits*.

***max-records***
See the description of *max-records* in *Server Resource Limits*.

***min-transfer-rate-in***
See the description of *min-transfer-rate-in* in *Zone Transfers*.

***max-transfer-time-in***
See the description of *max-transfer-time-in* in *Zone Transfers*.

***max-transfer-idle-in***
See the description of *max-transfer-idle-in* in *Zone Transfers*.

***max-transfer-time-out***
See the description of *max-transfer-time-out* in *Zone Transfers*.

***max-transfer-idle-out***
See the description of *max-transfer-idle-out* in *Zone Transfers*.

***notify***
See the description of *notify* in *Boolean Options*.

***notify-delay***
See the description of *notify-delay* in *Tuning*.

***notify-to-soa***
See the description of *notify-to-soa* in *Boolean Options*.

***parental-agents***
This option is only meaningful if the zone is DNSSEC signed. When performing a key rollover, BIND will query the parental agents to see if the new DS is actually published before withdrawing the old DNSSEC key.

***primaries***
For secondary zones, these are the name servers to request zone transfers from.

***zone-statistics***
See the description of *zone-statistics* in *options*.

**server-addresses**

**Grammar:** `server-addresses { ( <ipv4_address> | <ipv6_address> ); ... };`

**Blocks:** zone (static-stub)

**Tags:** query, zone

Specifies a list of IP addresses to which queries should be sent in recursive resolution for a static-stub zone.

This option is only meaningful for *static-stub* zones. This is a list of IP addresses to which queries should be sent in recursive resolution for the zone. A non-empty list for this option internally configures the apex NS RR with associated glue A or AAAA RRs.

For example, if "example.com" is configured as a static-stub zone with 192.0.2.1 and 2001:db8::1234 in a *server-addresses* option, the following RRs are internally configured:

```
example.com. NS example.com.
example.com. A 192.0.2.1
example.com. AAAA 2001:db8::1234
```

These records are used internally to resolve names under the static-stub zone. For instance, if the server receives a query for "www.example.com" with the RD bit on, the server initiates recursive resolution and sends queries to 192.0.2.1 and/or 2001:db8::1234.

**server-names**

**Grammar:** `server-names { <string>; ... };`

**Blocks:** zone (static-stub)

**Tags:** zone

Specifies a list of domain names of name servers that act as authoritative servers of a static-stub zone.

This option is only meaningful for *static-stub* zones. This is a list of domain names of name servers that act as authoritative servers of the static-stub zone. These names are resolved to IP addresses when *named* needs to send queries to these servers. For this supplemental resolution to be successful, these names must not be a subdomain of the origin name of the static-stub zone. That is, when "example.net" is the origin of a static-stub zone, "ns.example" and "master.example.com" can be specified in the *server-names* option, but "ns.example.net" cannot; it is rejected by the configuration parser.

A non-empty list for this option internally configures the apex NS RR with the specified names. For example, if "example.com" is configured as a static-stub zone with "ns1.example.net" and "ns2.example.net" in a *server-names* option, the following RRs are internally configured:

```
example.com. NS ns1.example.net.
example.com. NS ns2.example.net.
```

These records are used internally to resolve names under the static-stub zone. For instance, if the server receives a query for "www.example.com" with the RD bit on, the server initiates recursive resolution, resolves "ns1.example.net" and/or "ns2.example.net" to IP addresses, and then sends queries to one or more of these addresses.

***sig-validity-interval***

See the description of *sig-validity-interval* in *Tuning*.

***sig-signing-nodes***

See the description of *sig-signing-nodes* in *Tuning*.

***sig-signing-signatures***
See the description of *sig-signing-signatures* in *Tuning*.

***sig-signing-type***
See the description of *sig-signing-type* in *Tuning*.

***transfer-source***
See the description of *transfer-source* in *Zone Transfers*.

***transfer-source-v6***
See the description of *transfer-source-v6* in *Zone Transfers*.

***notify-source***
See the description of *notify-source* in *Zone Transfers*.

***notify-source-v6***
See the description of *notify-source-v6* in *Zone Transfers*.

***min-refresh-time*; *max-refresh-time*; *min-retry-time*; *max-retry-time***
See the descriptions in *Tuning*.

***ixfr-from-differences***
See the description of *ixfr-from-differences* in *Boolean Options*. (Note that the *ixfr-from-differences* choices of *primary* and *secondary* are not available at the zone level.)

***key-directory***
See the description of *key-directory* in *options*.

***serial-update-method***
See the description of *serial-update-method* in *options*.

**inline-signing**

**Grammar:** `inline-signing <boolean>;`

**Blocks:** dnssec-policy, zone (primary, secondary)

**Tags:** dnssec, zone

Specifies whether BIND 9 maintains a separate signed version of a zone.

The use of inline signing is determined by the *dnssec-policy* for the zone. If *inline-signing* is explicitly set to `yes` or `no` in *zone*, it overrides any value from *dnssec-policy*.

***multi-master***
See the description of *multi-master* in *Boolean Options*.

***masterfile-format***
See the description of *masterfile-format* in *Tuning*.

***max-zone-ttl***
See the description of *max-zone-ttl* in *options*. The use of this option in *zone* blocks is deprecated and will be rendered non-operational in a future release.

### Dynamic Update Policies

BIND 9 supports two methods of granting clients the right to perform dynamic updates to a zone:

- *allow-update* - a simple access control list
- *update-policy* - fine-grained access control

In both cases, BIND 9 writes the updates to the zone's filename set in *file*.

In the case of a DNSSEC zone where *inline-signing* is disabled, DNSSEC records are also written to the zone's filename.

> **Note**
>
> The zone file can no longer be manually updated while *named* is running; it is now necessary to perform *rndc freeze*, edit, and then perform *rndc thaw*. Comments and formatting in the zone file are lost when dynamic updates occur.

**update-policy**

**Grammar:** `update-policy ( local | { ( deny | grant ) <string> ( 6to4-self | external | krb5-self | krb5-selfsub | krb5-subdomain | krb5-subdomain-self-rhs | ms-self | ms-selfsub | ms-subdomain | ms-subdomain-self-rhs | name | self | selfsub | selfwild | subdomain | tcp-self | wildcard | zonesub ) [ <string> ] <rrtypelist>; ... } );`

**Blocks:** zone (primary)

**Tags:** transfer

Sets fine-grained rules to allow or deny dynamic updates (DDNS), based on requester identity, updated content, etc.

The *update-policy* clause allows more fine-grained control over which updates are allowed. It specifies a set of rules, in which each rule either grants or denies permission for one or more names in the zone to be updated by one or more identities. Identity is determined by the key that signed the update request, using either TSIG or SIG(0). In most cases, *update-policy* rules only apply to key-based identities. There is no way to specify update permissions based on the client source address.

*update-policy* rules are only meaningful for zones of *type primary*, and are not allowed in any other zone type. It is a configuration error to specify both *allow-update* and *update-policy* at the same time.

A pre-defined *update-policy* rule can be switched on with the command `update-policy local;`. *named* automatically generates a TSIG session key when starting and stores it in a file; this key can then be used by local clients to update the zone while *named* is running. By default, the session key is stored in the file `/run/session.key`, the key name is "local-ddns", and the key algorithm is HMAC-SHA256. These values are configurable with the *session-keyfile*, *session-keyname*, and *session-keyalg* options, respectively. A client running on the local system, if run with appropriate permissions, may read the session key from the key file and use it to sign update requests. The zone's update policy is set to allow that key to change any record within the zone. Assuming the key name is "local-ddns", this policy is equivalent to:

```
update-policy { grant local-ddns zonesub any; };
```

with the additional restriction that only clients connecting from the local system are permitted to send updates.

Note that only one session key is generated by *named*; all zones configured to use `update-policy local` accept the same key.

The command `nsupdate -l` implements this feature, sending requests to localhost and signing them using the key retrieved from the session key file.

Other rule definitions look like this:

```
( grant | deny ) identity ruletype  name   types
```

Each rule grants or denies privileges. Rules are checked in the order in which they are specified in the *update-policy* statement. Once a message has successfully matched a rule, the operation is immediately granted or denied, and no further rules are examined. There are 16 types of rules; the rule type is specified by the `ruletype` field, and the interpretation of other fields varies depending on the rule type.

In general, a rule is matched when the key that signed an update request matches the `identity` field, the name of the record to be updated matches the `name` field (in the manner specified by the `ruletype` field), and the type of the record to be updated matches the `types` field. Details for each rule type are described below.

The `identity` field must be set to a fully qualified domain name. In most cases, this represents the name of the TSIG or SIG(0) key that must be used to sign the update request. If the specified name is a wildcard, it is subject to DNS wildcard expansion, and the rule may apply to multiple identities. When a TKEY exchange has been used to create a shared secret, the identity of the key used to authenticate the TKEY exchange is used as the identity of the shared secret. Some rule types use identities matching the client's Kerberos principal (e.g, `"host/machine@REALM"`) or Windows realm (`machine$@REALM`).

The `name` field also specifies a fully qualified domain name. This often represents the name of the record to be updated. Interpretation of this field is dependent on rule type.

If no `types` are explicitly specified, then a rule matches all types except RRSIG, NS, SOA, NSEC, and NSEC3. Types may be specified by name, including `ANY`; ANY matches all types except NSEC and NSEC3, which can never be updated. Note that when an attempt is made to delete all records associated with a name, the rules are checked for each existing record type.

If the type is immediately followed by a number in parentheses, that number is the maximum number of records of that type permitted to exist in the RRset after an update has been applied. For example, `PTR(1)` indicates that only one PTR record is allowed. If an attempt is made to add two PTR records in an update, the second one is silently discarded. If a PTR record already exists, both new records are silently discarded.

If type ANY is specified with a limit, then that limit applies to all types that are not otherwise specified. For example, `A PTR(1) ANY(2)` indicates that an unlimited number of A records can exist, but only one PTR record, and no more than two of any other type.

Typical use with a rule `grant * tcp-self . PTR(1);` in the zone 2.0.192.IN-ADDR.ARPA looks like this:

```
nsupdate -v <<EOF
local 192.0.2.1
del 1.2.0.192.IN-ADDR.ARPA PTR
add 1.2.0.192.IN-ADDR.ARPA 0 PTR EXAMPLE.COM
send
EOF
```

The ruletype field has 18 values: `name`, `subdomain`, `zonesub`, `wildcard`, `self`, `selfsub`, `self-wild`, `ms-self`, `ms-selfsub`, `ms-subdomain`, `ms-subdomain-self-rhs`, `krb5-self`, `krb5-selfsub`, `krb5-subdomain`, `krb5-subdomain-self-rhs`, `tcp-self`, `6to4-self`, and `external`.

**name**
: With exact-match semantics, this rule matches when the name being updated is identical to the contents of the `name` field.

**subdomain**
: This rule matches when the name being updated is a subdomain of, or identical to, the contents of the `name` field.

**zonesub**
: This rule is similar to subdomain, except that it matches when the name being updated is a subdomain of the zone in which the *update-policy* statement appears. This obviates the need to type the zone name twice, and enables the use of a standard *update-policy* statement in multiple zones without modification. When this rule is used, the `name` field is omitted.

**wildcard**
The name field is subject to DNS wildcard expansion, and this rule matches when the name being updated is a valid expansion of the wildcard.

**self**
This rule matches when the name of the record being updated matches the contents of the identity field. The name field is ignored. To avoid confusion, it is recommended that this field be set to the same value as the identity field or to "." The self rule type is most useful when allowing one key per name to update, where the key has the same name as the record to be updated. In this case, the identity field can be specified as * (asterisk).

**selfsub**
This rule is similar to self, except that subdomains of self can also be updated.

**selfwild**
This rule is similar to self, except that only subdomains of self can be updated.

**ms-self**
When a client sends an UPDATE using a Windows machine principal (for example, machine$@REALM), this rule allows records with the absolute name of machine.REALM to be updated.

The realm to be matched is specified in the identity field.

The name field has no effect on this rule; it should be set to "." as a placeholder.

For example, grant EXAMPLE.COM ms-self . A AAAA allows any machine with a valid principal in the realm EXAMPLE.COM to update its own address records.

**ms-selfsub**
This is similar to ms-self, except it also allows updates to any subdomain of the name specified in the Windows machine principal, not just to the name itself.

**ms-subdomain**
When a client sends an UPDATE using a Windows machine principal (for example, machine$@REALM), this rule allows any machine in the specified realm to update any record in the zone or in a specified subdomain of the zone.

The realm to be matched is specified in the identity field.

The name field specifies the subdomain that may be updated. If set to "." or any other name at or above the zone apex, any name in the zone can be updated.

For example, if *update-policy* for the zone "example.com" includes grant EXAMPLE.COM ms-subdomain hosts.example.com. AA AAAA, any machine with a valid principal in the realm EXAMPLE.COM is able to update address records at or below hosts.example.com.

**ms-subdomain-self-rhs**
This rule is similar to ms-subdomain, with an additional restriction that PTR and SRV target names must match the name of the machine identified in the principal.

**krb5-self**
When a client sends an UPDATE using a Kerberos machine principal (for example, host/machine@REALM), this rule allows records with the absolute name of machine to be updated, provided it has been authenticated by REALM. This is similar but not identical to ms-self, due to the machine part of the Kerberos principal being an absolute name instead of an unqualified name.

The realm to be matched is specified in the identity field.

The name field has no effect on this rule; it should be set to "." as a placeholder.

For example, grant EXAMPLE.COM krb5-self . A AAAA allows any machine with a valid principal in the realm EXAMPLE.COM to update its own address records.

**krb5-selfsub**
This is similar to `krb5-self`, except it also allows updates to any subdomain of the name specified in the `machine` part of the Kerberos principal, not just to the name itself.

**krb5-subdomain**
This rule is identical to `ms-subdomain`, except that it works with Kerberos machine principals (i.e., `host/machine@REALM`) rather than Windows machine principals.

**krb5-subdomain-self-rhs**
This rule is similar to `krb5-subdomain`, with an additional restriction that PTR and SRV target names must match the name of the machine identified in the principal.

**tcp-self**
This rule allows updates that have been sent via TCP and for which the standard mapping from the client's IP address into the `in-addr.arpa` and `ip6.arpa` namespaces matches the name to be updated. The `identity` field must match that name. The `name` field should be set to ".". Note that, since identity is based on the client's IP address, it is not necessary for update request messages to be signed.

> **Note**
> It is theoretically possible to spoof these TCP sessions.

**6to4-self**
This allows the name matching a 6to4 IPv6 prefix, as specified in **RFC 3056**, to be updated by any TCP connection from either the 6to4 network or from the corresponding IPv4 address. This is intended to allow NS or DNAME RRsets to be added to the `ip6.arpa` reverse tree.

The `identity` field must match the 6to4 prefix in `ip6.arpa`. The `name` field should be set to ".". Note that, since identity is based on the client's IP address, it is not necessary for update request messages to be signed.

In addition, if specified for an `ip6.arpa` name outside of the `2.0.0.2.ip6.arpa` namespace, the corresponding /48 reverse name can be updated. For example, TCP/IPv6 connections from 2001:DB8:ED0C::/48 can update records at `C.0.D.E.8.B.D.0.1.0.0.2.ip6.arpa`.

> **Note**
> It is theoretically possible to spoof these TCP sessions.

**external**
This rule allows *named* to defer the decision of whether to allow a given update to an external daemon.

The method of communicating with the daemon is specified in the `identity` field, the format of which is "`local`:path", where "path" is the location of a Unix-domain socket. (Currently, "local" is the only supported mechanism.)

Requests to the external daemon are sent over the Unix-domain socket as datagrams with the following format:

```
Protocol version number (4 bytes, network byte order, currently 1)
Request length (4 bytes, network byte order)
Signer (null-terminated string)
Name (null-terminated string)
TCP source address (null-terminated string)
Rdata type (null-terminated string)
Key (null-terminated string)
```

(continues on next page)

(continued from previous page)

```
TKEY token length (4 bytes, network byte order)
TKEY token (remainder of packet)
```

The daemon replies with a four-byte value in network byte order, containing either 0 or 1; 0 indicates that the specified update is not permitted, and 1 indicates that it is.

> **Warning**
>
> The external daemon must not delay communication. This policy is evaluated synchronously; any wait period negatively affects *named* performance.

### Multiple Views

When multiple views are in use, a zone may be referenced by more than one of them. Often, the views contain different zones with the same name, allowing different clients to receive different answers for the same queries. At times, however, it is desirable for multiple views to contain identical zones. The *in-view* zone option provides an efficient way to do this; it allows a view to reference a zone that was defined in a previously configured view. For example:

```
view internal {
    match-clients { 10/8; };

    zone example.com {
    type primary;
    file "example-external.db";
    };
};

view external {
    match-clients { any; };

    zone example.com {
    in-view internal;
    };
};
```

An *in-view* option cannot refer to a view that is configured later in the configuration file.

A *zone* statement which uses the *in-view* option may not use any other options, with the exception of *forward* and *forwarders*. (These options control the behavior of the containing view, rather than change the zone object itself.)

Zone-level ACLs (e.g., allow-query, allow-transfer), and other configuration details of the zone, are all set in the view the referenced zone is defined in. Be careful to ensure that ACLs are wide enough for all views referencing the zone.

An *in-view* zone cannot be used as a response policy zone.

An *in-view* zone is not intended to reference a *forward* zone.

## 8.3 Statements

BIND 9 supports many hundreds of statements; finding the right statement to control a specific behavior or solve a particular problem can be a daunting task. To simplify the task for users, all statements have been assigned one or more tags. Tags are designed to group together statements that have broadly similar functionality; thus, for example, all statements that control the handling of queries or of zone transfers are respectively tagged under **query** and **transfer**.

*DNSSEC Tag Statements* are those that relate to or control DNSSEC.

*Logging Tag Statements* relate to or control logging, and typically only appear in a logging block.

*Query Tag Statements* relate to or control queries.

*Security Tag Statements* relate to or control security features.

*Server Tag Statements* relate to or control server behavior, and typically only appear in a server block.

*Transfer Tag Statements* relate to or control zone transfers.

*View Tag Statements* relate to or control view selection criteria, and typically only appear in a view block.

*Zone Tag Statements* relate to or control zone behavior, and typically only appear in a zone block.

*Deprecated Tag Statements* are those that are now deprecated, but are included here for historical reference.

The following table lists all statements permissible in `named.conf`, with their associated tags; the next section groups the statements by tag. Please note that these sections are a work in progress.

| Statement | Description | Tags |
|---|---|---|
| *acl* | Assigns a symbolic name to an address match list. | server |
| *algorithm* | Defines the algorithm to be used in a key clause. | security |
| *all-per-second* | Limits UDP responses of all kinds. | query |
| *allow-new-zones* | Controls the ability to add zones at runtime via *rndc addzone*. | server, zone |
| *allow-notify* | Defines an *address_match_list* that is allowed to send NOTIFY messages for the zone, in addition to addresses defined in the *primaries* option for the zone. | transfer |
| *allow-proxy* | Defines an *address_match_list* for the client addresses allowed to send PROXYv2 headers. | server |
| *allow-proxy-on* | Defines an *address_match_list* for the interface addresses allowed to accept PROXYv2 headers. The option is mostly intended for multi-homed configurations. | server |
| *allow-query* | Specifies which hosts (an IP address list) are allowed to send queries to this resolver. | query |
| *allow-query-cache* | Specifies which hosts (an IP address list) can access this server's cache and thus effectively controls recursion. | query |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *allow-query-cache-on* | Specifies which hosts (from an IP address list) can access this server's cache. It is used on servers with multiple interfaces. | query |
| *allow-query-on* | Specifies which local addresses (an IP address list) are allowed to send queries to this resolver. This option is used in multi-homed configurations. | query |
| *allow-recursion* | Defines an `address_match_list` of clients that are allowed to perform recursive queries. | query |
| *allow-recursion-on* | Specifies which local addresses can accept recursive queries. | query, server |
| *allow-transfer* | Defines an `address_match_list` of hosts that are allowed to transfer the zone information from this server. | transfer |
| *allow-update* | Defines an `address_match_list` of hosts that are allowed to submit dynamic updates for primary zones. | transfer |
| *allow-update-forwarding* | Defines an `address_match_list` of hosts that are allowed to submit dynamic updates to a secondary server for transmission to a primary. | transfer |
| *also-notify* | Defines one or more hosts that are sent `NOTIFY` messages when zone changes occur. | transfer |
| *answer-cookie* | Controls whether COOKIE EDNS replies are sent in response to client queries. | query |
| *attach-cache* | Allows multiple views to share a single cache database. | view |
| *auth-nxdomain* | Controls whether BIND, acting as a resolver, provides authoritative NXDOMAIN (domain does not exist) answers. | query |
| *automatic-interface-scan* | Controls the automatic rescanning of network interfaces when addresses are added or removed. | server |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *avoid-v4-udp-ports* | Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv4 messages. | deprecated |
| *avoid-v6-udp-ports* | Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv6 messages. | deprecated |
| *bindkeys-file* | Specifies the pathname of a file to override the built-in trusted keys provided by `named`. | dnssec |
| *blackhole* | Defines an `address_match_list` of hosts to ignore. The server will neither respond to queries from nor send queries to these addresses. | query |
| *bogus* | Allows a remote server to be ignored. | server |
| *break-dnssec* | Enables `dns64` synthesis even if the validated result would cause a DNSSEC validation failure. | query |
| *buffered* | Controls flushing of log messages. | logging |
| *ca-file* | Specifies the path to a file containing TLS certificates for trusted CA authorities, used to verify remote peer certificates. | server, security |
| *catalog-zones* | Configures catalog zones in `named.conf`. | zone |
| *category* | Specifies the type of data logged to a particular channel. | logging |
| *cdnskey* | Specifies whether a CDNSKEY record should be published during KSK rollover. | dnssec |
| *cds-digest-types* | Specifies the digest types to use for CDS resource records. | dnssec |
| *cert-file* | Specifies the path to a file containing the TLS certificate for a connection. | server, security |
| *channel* | Defines a stream of data that can be independently logged. | logging |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *check-dup-records* | Checks primary zones for records that are treated as different by DNSSEC but are semantically equal in plain DNS. | dnssec, query |
| *check-integrity* | Performs post-load zone integrity checks on primary zones. | zone |
| *check-mx* | Checks whether an MX record appears to refer to an IP address. | zone |
| *check-mx-cname* | Sets the response to MX records that refer to CNAMEs. | zone |
| *check-names* | Restricts the character set and syntax of certain domain names in primary files and/or DNS responses received from the network. | query, server |
| *check-sibling* | Specifies whether to check for sibling glue when performing integrity checks. | zone |
| *check-spf* | Specifies whether to check for a TXT Sender Policy Framework record, if an SPF record is present. | zone |
| *check-srv-cname* | Sets the response to SRV records that refer to CNAMEs. | zone |
| *check-svcb* | Specifies whether to perform additional checks on SVCB records. | zone |
| *check-wildcard* | Checks for non-terminal wildcards. | zone |
| *checkds* | Controls whether `DS` queries are sent to parental agents. | dnssec |
| *cipher-suites* | Specifies a list of allowed cipher suites in the order of preference for TLSv1.3 only. | security |
| *ciphers* | Specifies a list of allowed ciphers in the order of preference for TLSv1.2 only. | security |
| *clients* | Specifies an access control list (ACL) of clients that are affected by a given *dns64* directive. | query |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *clients-per-query* | Sets the initial minimum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients. | server |
| *controls* | Specifies control channels to be used to manage the name server. | server |
| *cookie-algorithm* | Sets the algorithm to be used when generating a server cookie. | server |
| *cookie-secret* | Specifies a shared secret used for generating and verifying EDNS COOKIE options within an anycast cluster. | server |
| *database* | Specifies the type of database to be used to store zone data. | zone |
| *deny-answer-addresses* | Rejects A or AAAA records if the corresponding IPv4 or IPv6 addresses match a given `address_match_list`. | query |
| *deny-answer-aliases* | Rejects CNAME or DNAME records if the "alias" name matches a given list of `domain_name` elements. | query |
| *dhparam-file* | Specifies the path to a file containing Diffie-Hellman parameters, for enabling cipher suites. | server, security |
| *dialup* | Concentrates zone maintenance so that all transfers take place once every `heartbeat-interval`, ideally during a single call. | deprecated |
| *directory* | Sets the server's working directory. | server |
| *disable-algorithms* | Disables DNSSEC algorithms from a specified zone. | dnssec |
| *disable-ds-digests* | Disables DS digest types from a specified zone. | dnssec, zone |
| *disable-empty-zone* | Disables individual empty zones. | server, zone |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *dlz* | Configures a Dynamically Loadable Zone (DLZ) database in `named.conf`. | zone |
| *dns64* | Instructs `named` to return mapped IPv4 addresses to AAAA queries when there are no AAAA records. | query |
| *dns64-contact* | Specifies the name of the contact for `dns64` zones. | server |
| *dns64-server* | Specifies the name of the server for `dns64` zones. | server |
| *dnskey-sig-validity* | | obsolete |
| *dnskey-ttl* | Specifies the time-to-live (TTL) for DNSKEY resource records. | dnssec |
| *dnsrps-enable* | Turns on the DNS Response Policy Service (DNSRPS) interface. | server, security |
| *dnsrps-library* | Specifies the path to the DNS Response Policy Service (DNSRPS) provider library. | server, security |
| *dnsrps-options* | Provides additional RPZ configuration settings, which are passed to the DNS Response Policy Service (DNSRPS) provider library. | server, security |
| *dnssec-accept-expired* | Instructs BIND 9 to accept expired DNSSEC signatures when validating. | dnssec |
| *dnssec-dnskey-kskonly* | | obsolete |
| *dnssec-loadkeys-interval* | Sets the frequency of automatic checks of the DNSSEC key repository. | dnssec |
| *dnssec-must-be-secure* | Defines hierarchies that must or may not be secure (signed and validated). | deprecated |
| *dnssec-policy* | Defines a key and signing policy (KASP) for zones. | dnssec |
| *dnssec-secure-to-insecure* | | obsolete |
| *dnssec-update-mode* | | obsolete |
| *dnssec-validation* | Enables DNSSEC validation in `named`. | dnssec |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *dnstap* | Enables logging of `dnstap` messages. | logging |
| *dnstap-identity* | Specifies an `identity` string to send in `dnstap` messages. | logging |
| *dnstap-output* | Configures the path to which the `dnstap` frame stream is sent. | logging |
| *dnstap-version* | Specifies a `version` string to send in `dnstap` messages. | logging |
| *dual-stack-servers* | Specifies host names or addresses of machines with access to both IPv4 and IPv6 transports. | server |
| *dump-file* | Indicates the pathname of the file where the server dumps the database after `rndc dumpdb`. | logging |
| *dyndb* | Configures a DynDB database in `named.conf`. | zone |
| *edns* | Controls the use of the EDNS0 (**RFC 2671**) feature. | server |
| *edns-udp-size* | Sets the maximum advertised EDNS UDP buffer size to control the size of packets received from authoritative servers in response to recursive queries. | query |
| *edns-version* | Sets the maximum EDNS VERSION that is sent to the server(s) by the resolver. | server |
| *empty-contact* | Specifies the contact name in the returned SOA record for empty zones. | server, zone |
| *empty-server* | Specifies the server name in the returned SOA record for empty zones. | server, zone |
| *empty-zones-enable* | Enables or disables all empty zones. | server, zone |
| *endpoints* | Specifies a list of HTTP query paths on which to listen. | server, query |
| *errors-per-second* | Limits the number of errors for a valid domain name and record type. | server |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *exclude* | Allows a list of IPv6 addresses to be ignored if they appear in a domain name's AAAA records in `dns64`. | query |
| *exempt-clients* | Exempts specific clients or client groups from rate limiting. | query |
| *fetch-quota-params* | Sets the parameters for dynamic resizing of the `fetches-per-server` quota in response to detected congestion. | server, query |
| *fetches-per-server* | Sets the maximum number of simultaneous iterative queries allowed to be sent by a server to an upstream name server before the server blocks additional queries. | server, query |
| *fetches-per-zone* | Sets the maximum number of simultaneous iterative queries allowed to any one domain before the server blocks new queries for data in or beneath that zone. | server, query |
| *file* | Specifies the zone's filename. | zone |
| *flush-zones-on-shutdown* | Controls whether pending zone writes are flushed when the name server exits. | zone |
| *forward* | Allows or disallows fallback to recursion if forwarding has failed; it is always used in conjunction with the `forwarders` statement. | query |
| *forwarders* | Defines one or more resolvers to which queries are forwarded. | query |
| *fstrm-set-buffer-hint* | Sets the number of accumulated bytes in the output buffer before forcing a buffer flush. | logging |
| *fstrm-set-flush-timeout* | Sets the number of seconds that unflushed data remains in the output buffer. | logging |
| *fstrm-set-input-queue-size* | Sets the number of queue entries to allocate for each input queue. | logging |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *fstrm-set-output-notify-threshold* | Sets the number of outstanding queue entries allowed on an input queue before waking the I/O thread. | logging |
| *fstrm-set-output-queue-model* | Sets the queuing semantics to use for queue objects. | logging |
| *fstrm-set-output-queue-size* | Sets the number of queue entries allocated for each output queue. | logging |
| *fstrm-set-reopen-interval* | Sets the number of seconds to wait between attempts to reopen a closed output stream. | logging |
| *geoip-directory* | Specifies the directory containing GeoIP database files. | server |
| *heartbeat-interval* | Sets the interval at which the server performs zone maintenance tasks for all zones marked as `dialup`. | deprecated |
| *hostname* | Specifies the hostname of the server to return in response to a `hostname.bind` query. | server |
| *http* | Configures HTTP endpoints on which to listen for DNS-over-HTTPS (DoH) queries. | server, query |
| *http-listener-clients* | Limits the number of active concurrent connections on a per-listener basis. | server |
| *http-port* | Specifies the TCP port number the server uses to receive and send unencrypted DNS traffic via HTTP. | server, query |
| *http-streams-per-connection* | Limits the number of active concurrent HTTP/2 streams on a per-connection basis. | server |
| *https-port* | Specifies the TCP port number the server uses to receive and send DNS-over-HTTPS protocol traffic. | server, query |
| *in-view* | Specifies the view in which a given zone is defined. | view, zone |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *inet* | Specifies a TCP socket as a control channel. | server |
| *inline-signing* | Specifies whether BIND 9 maintains a separate signed version of a zone. | dnssec, zone |
| *interface-interval* | Sets the interval at which the server scans the network interface list. | server |
| *ipv4-prefix-length* | Specifies the prefix lengths of IPv4 address blocks. | server |
| *ipv4only-contact* | Specifies the contact for the IPV4ONLY.ARPA zone created by `dns64`. | server |
| *ipv4only-enable* | Enables automatic IPv4 zones if a `dns64` block is configured. | query |
| *ipv4only-server* | Specifies the name of the server for the IPV4ONLY.ARPA zone created by `dns64`. | server, query |
| *ipv6-prefix-length* | Specifies the prefix lengths of IPv6 address blocks. | server |
| *ixfr-from-differences* | Controls how IXFR transfers are calculated. | transfer |
| *journal* | Allows the default journal's filename to be overridden. | zone |
| *key* | Defines a shared secret key for use with *TSIG* or the command channel. | security |
| *key-directory* | Indicates the directory where public and private DNSSEC key files are found. | dnssec |
| *key-file* | Specifies the path to a file containing the private TLS key for a connection. | server, security |
| *key-store* | Configures a DNSSEC key store. | dnssec |
| *keys* | Specifies one or more `server_key`s to be used with a remote server. | server, security |
| *lame-ttl* | Sets the resolver's lame cache. | server |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *listen-on* | Specifies the IPv4 addresses on which a server listens for DNS queries. | server |
| *listen-on-v6* | Specifies the IPv6 addresses on which a server listens for DNS queries. | server |
| *listener-clients* | Specifies a per-listener quota for active connections. | server, query |
| *lmdb-mapsize* | Sets a maximum size for the memory map of the new-zone database in LMDB database format. | server |
| *log-only* | Tests rate-limiting parameters without actually dropping any requests. | logging, query |
| *logging* | Configures logging options for the name server. | logging |
| *managed-keys* | | deprecated |
| *managed-keys-directory* | Specifies the directory in which to store the files that track managed DNSSEC keys. | dnssec |
| *manual-mode* | Run key management in a manual mode. | dnssec |
| *mapped* | Specifies an access control list (ACL) of IPv4 addresses that are to be mapped to the corresponding A RRset in *dns64*. | query |
| *masterfile-format* | Specifies the file format of zone files. | zone, server |
| *masterfile-style* | Specifies the format of zone files during a dump, when the *masterfile-format* is `text`. | server |
| *match-clients* | Specifies a view of DNS namespace for a given subset of client IP addresses. | view |
| *match-destinations* | Specifies a view of DNS namespace for a given subset of destination IP addresses. | view |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *match-mapped-addresses* | Allows IPv4-mapped IPv6 addresses to match address-match list entries for corresponding IPv4 addresses. | server |
| *match-recursive-only* | Specifies that only recursive requests can match this view of the DNS namespace. | view |
| *max-cache-size* | Sets the maximum amount of memory to use for an individual cache database and its associated metadata. | server |
| *max-cache-ttl* | Specifies the maximum time (in seconds) that the server caches ordinary (positive) answers. | server |
| *max-clients-per-query* | Sets the maximum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients. | server |
| *max-ixfr-ratio* | Sets the maximum size for IXFR responses to zone transfer requests. | transfer |
| *max-journal-size* | Controls the size of journal files. | transfer |
| *max-ncache-ttl* | Specifies the maximum retention time (in seconds) for storage of negative answers in the server's cache. | server |
| *max-query-count* | Sets the maximum number of iterative queries while servicing a recursive query. | server, query |
| *max-query-restarts* | Sets the maximum number of chained CNAMEs to follow | server, query |
| *max-records* | Sets the maximum number of records permitted in a zone. | zone, server |
| *max-records-per-type* | Sets the maximum number of records that can be stored in an RRset. | server |
| *max-recursion-depth* | Sets the maximum number of levels of recursion permitted at any one time while servicing a recursive query. | server |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *max-recursion-queries* | Sets the maximum number of iterative queries while servicing a recursive query. | server, query |
| *max-refresh-time* | Limits the zone refresh interval to no less often than the specified value, in seconds. | transfer |
| *max-retry-time* | Limits the zone refresh retry interval to no less often than the specified value, in seconds. | transfer |
| *max-rsa-exponent-size* | Sets the maximum RSA exponent size (in bits) when validating. | dnssec, query |
| *max-stale-ttl* | Specifies the maximum time that the server retains records past their normal expiry, to return them as stale records. | server |
| *max-table-size* | Sets the maximum size of the table used to track requests and rate-limit responses. | server |
| *max-transfer-idle-in* | Specifies the number of minutes after which inbound zone transfers making no progress are terminated. | transfer |
| *max-transfer-idle-out* | Specifies the number of minutes after which outbound zone transfers making no progress are terminated. | transfer |
| *max-transfer-time-in* | Specifies the number of minutes after which inbound zone transfers are terminated. | transfer |
| *max-transfer-time-out* | Specifies the number of minutes after which outbound zone transfers are terminated. | transfer |
| *max-types-per-name* | Sets the maximum number of RR types that can be stored for an owner name. | server |
| *max-udp-size* | Sets the maximum EDNS UDP message size sent by `named`. | query |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *max-validation-failures-per-fetch* | Sets the maximum number of DNSSEC validation failures that can happen in a single fetch. | server |
| *max-validations-per-fetch* | Sets the maximum number of DNSSEC validations that can happen in a single fetch. | server |
| *max-zone-ttl* | Specifies a maximum permissible time-to-live (TTL) value, in seconds. | deprecated |
| *memstatistics* | Controls whether memory statistics are written to the file specified by `memstatistics-file` at exit. | server, logging |
| *memstatistics-file* | Sets the pathname of the file where the server writes memory usage statistics on exit. | logging |
| *message-compression* | Controls whether DNS name compression is used in responses to regular queries. | query |
| *min-cache-ttl* | Specifies the minimum time (in seconds) that the server caches ordinary (positive) answers. | server |
| *min-ncache-ttl* | Specifies the minimum retention time (in seconds) for storage of negative answers in the server's cache. | server |
| *min-refresh-time* | Limits the zone refresh interval to no more often than the specified value, in seconds. | transfer |
| *min-retry-time* | Limits the zone refresh retry interval to no more often than the specified value, in seconds. | transfer |
| *min-table-size* | Sets the minimum size of the table used to track requests and rate-limit responses. | query |
| *min-transfer-rate-in* | Specifies the minimum traffic rate below which inbound zone transfers are terminated. | transfer |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *minimal-any* | Controls whether the server replies with only one of the RRsets for a query name, when generating a positive response to a query of type ANY over UDP. | query |
| *minimal-responses* | Controls whether the server only adds records to the authority and additional data sections when they are required (e.g. delegations, negative responses). This improves server performance. | query |
| *multi-master* | Controls whether serial number mismatch errors are logged. | transfer |
| *new-zones-directory* | Specifies the directory where configuration parameters are stored for zones added by `rndc addzone`. | zone |
| *no-case-compress* | Specifies a list of addresses that require case-insensitive compression in responses. | server |
| *nocookie-udp-size* | Sets the maximum size of UDP responses that are sent to queries without a valid server COOKIE. | query |
| *nodata-per-second* | Limits the number of empty (NODATA) responses for a valid domain name. | query |
| *notify* | Controls whether `NOTIFY` messages are sent on zone changes. | transfer |
| *notify-defer* | Sets the defer time (in seconds) before sending NOTIFY messages for a zone. | transfer, zone |
| *notify-delay* | Sets the delay (in seconds) between sending sets of NOTIFY messages for a zone. | transfer, zone |
| *notify-rate* | Specifies the rate at which NOTIFY requests are sent during normal zone maintenance operations. | transfer, zone |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *notify-source* | Defines the IPv4 address (and optional port) to be used for outgoing `NOTIFY` messages. | transfer |
| *notify-source-v6* | Defines the IPv6 address (and optional port) to be used for outgoing `NOTIFY` messages. | transfer |
| *notify-to-soa* | Controls whether the name servers in the NS RRset are checked against the SOA MNAME. | transfer |
| *nsec3param* | Specifies the use of NSEC3 instead of NSEC, and sets NSEC3 parameters. | dnssec |
| *nta-lifetime* | Specifies the lifetime, in seconds, for negative trust anchors added via *`rndc nta`*. | dnssec |
| *nta-recheck* | Specifies the time interval for checking whether negative trust anchors added via *`rndc nta`* are still necessary. | dnssec |
| *null* | Causes all messages sent to the logging channel to be discarded. | logging |
| *nxdomain-redirect* | Appends the specified suffix to the original query name, when replacing an NXDOMAIN with a redirect namespace. | query |
| *nxdomains-per-second* | Limits the number of undefined subdomains for a valid domain name. | query |
| *offline-ksk* | Specifies whether the DNSKEY, CDS, and CDNSKEY RRsets are being signed offline. | dnssec |
| *options* | Defines global options to be used by BIND 9. | server |
| *padding* | Adds EDNS Padding options to outgoing messages to increase the packet size. | server |
| *parent-ds-ttl* | Sets the time to live (TTL) of the DS RRset used by the parent zone. | dnssec |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *parent-propagation-delay* | Sets the propagation delay from the time the parent zone is updated to when the new version is served by all of the parent zone's name servers. | dnssec, zone |
| *parental-agents* | | dnssec |
| *parental-source* | Specifies which local IPv4 source address is used to send parental DS queries. | dnssec |
| *parental-source-v6* | Specifies which local IPv6 source address is used to send parental DS queries. | dnssec |
| *pid-file* | Specifies the pathname of the file where the server writes its process ID. | server |
| *pkcs11-uri* | | dnssec, pkcs11 |
| *plugin* | Configures plugins in `named.conf`. | server |
| *port* | Specifies the UDP/TCP port number the server uses to receive and send DNS protocol traffic. | server, query |
| *prefer-server-ciphers* | Specifies that server ciphers should be preferred over client ones. | server, security |
| *preferred-glue* | Controls the order of glue records in an A or AAAA response. | query |
| *prefetch* | Specifies the "trigger" time-to-live (TTL) value at which prefetch of the current query takes place. | query |
| *primaries* | Defines one or more servers that zone transfer can be requested from. | transfer, zone |
| *print-category* | Includes the category in log messages. | logging |
| *print-severity* | Includes the severity in log messages. | logging |
| *print-time* | Specifies the time format for log messages. | logging |
| *protocols* | Specifies the allowed versions of the TLS protocol. | security |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *provide-ixfr* | Controls whether a primary responds to an incremental zone request (IXFR) or only responds with a full zone transfer (AXFR). | transfer |
| *publish-safety* | Increases the amount of time between when keys are published and when they become active, to allow for unforeseen events. | dnssec |
| *purge-keys* | Specifies the amount of time after which DNSSEC keys that have been deleted from the zone can be removed from disk. | dnssec |
| *qname-minimization* | Controls QNAME minimization behavior in the BIND 9 resolver. | query |
| *qps-scale* | Tightens defenses during DNS attacks by scaling back the ratio of the current query-per-second rate. | query |
| *query-source* | Controls the IPv4 address from which queries are issued. If *none*, then no IPv4 address would be used to issue the query and therefore only IPv6 servers are queried. | query |
| *query-source-v6* | Controls the IPv6 address from which queries are issued. If *none*, then no IPv6 address would be used to issue the query and therefore only IPv4 servers are quried. | query |
| *querylog* | Specifies whether query logging should be active when `named` first starts. | logging, server |
| *rate-limit* | Controls excessive UDP responses, to prevent BIND 9 from being used to amplify reflection denial-of-service (DoS) attacks. | query |
| *recursing-file* | Specifies the pathname of the file where the server dumps queries that are currently recursing via `rndc recursing`. | server |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *recursion* | Defines whether recursion and caching are allowed. | query |
| *recursive-clients* | Specifies the maximum number of concurrent recursive queries the server can perform. | query |
| *recursive-only* | Toggles whether `dns64` synthesis occurs only for recursive queries. | query |
| *referrals-per-second* | Limits the number of referrals or delegations to a server for a given domain. | query |
| *remote-hostname* | Specifies the expected hostname in the TLS certificate of the remote server. | security |
| *remote-servers* | Defines a list of servers to be used by primary and secondary zones. | server |
| *request-expire* | Specifies whether the local server requests the EDNS EXPIRE value, when acting as a secondary. | transfer, query |
| *request-ixfr* | Controls whether a secondary requests an incremental zone transfer (IXFR) or a full zone transfer (AXFR). | transfer |
| *request-nsid* | Controls whether an empty EDNS(0) NSID (Name Server Identifier) option is sent with all queries to authoritative name servers during iterative resolution. | query |
| *require-cookie* | Controls whether responses without a server cookie are accepted. | query |
| *require-server-cookie* | Controls whether a valid server cookie is required before sending a full response to a UDP request. | query |
| *resolver-query-timeout* | Specifies the length of time, in milliseconds, that a resolver attempts to resolve a recursive query before failing. | query |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *resolver-use-dns64* | Specifies whether to apply DNS64 mappings when sending queries. | server |
| *response-padding* | Adds an EDNS Padding option to encrypted messages, to reduce the chance of guessing the contents based on size. | query |
| *response-policy* | Specifies response policy zones for the view or among global options. | server, query, zone, security |
| *responselog* | Specifies whether response logging should be active when `named` first starts. | logging, server |
| *responses-per-second* | Limits the number of non-empty responses for a valid domain name and record type. | query |
| *retire-safety* | Increases the amount of time a key remains published after it is no longer active, to allow for unforeseen events. | dnssec |
| *reuseport* | Enables kernel load-balancing of sockets. | server |
| *root-key-sentinel* | Controls whether BIND 9 responds to root key sentinel probes. | server |
| *rrset-order* | Defines the order in which equal RRs (RRsets) are returned. | query |
| *search* | Specifies whether a Dynamically Loadable Zone (DLZ) module is queried for an answer to a query name. | query |
| *secret* | Defines a Base64-encoded string to be used as the secret by the algorithm. | security |
| *secroots-file* | Specifies the pathname of the file where the server dumps security roots, when using `rndc secroots`. | dnssec |
| *send-cookie* | Controls whether a COOKIE EDNS option is sent along with a query. | query |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *serial-query-rate* | Defines an upper limit on the number of queries per second issued by the server, when querying the SOA RRs used for zone transfers. | transfer |
| *serial-update-method* | Specifies the update method to be used for the zone serial number in the SOA record. | zone |
| *server* | Defines characteristics to be associated with a remote name server. | server |
| *server-addresses* | Specifies a list of IP addresses to which queries should be sent in recursive resolution for a static-stub zone. | query, zone |
| *server-id* | Specifies the ID of the server to return in response to a `ID.SERVER` query. | server |
| *server-names* | Specifies a list of domain names of name servers that act as authoritative servers of a static-stub zone. | zone |
| *servfail-ttl* | Sets the length of time (in seconds) that a SERVFAIL response is cached. | server |
| *session-keyalg* | Specifies the algorithm to use for the TSIG session key. | security |
| *session-keyfile* | Specifies the pathname of the file where a TSIG session key is written, when generated by *named* for use by `nsupdate -l`. | security |
| *session-keyname* | Specifies the key name for the TSIG session key. | security |
| *session-tickets* | Enables or disables session resumption through TLS session tickets. | security |
| *severity* | Defines the priority level of log messages. | logging |
| *sig-signing-nodes* | Specifies the maximum number of nodes to be examined in each quantum, when signing a zone with a new DNSKEY. | dnssec |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *sig-signing-signatures* | Specifies the threshold for the number of signatures that terminates processing a quantum, when signing a zone with a new DNSKEY. | dnssec |
| *sig-signing-type* | Specifies a private RDATA type to use when generating signing-state records. | dnssec |
| *sig-validity-interval* | | obsolete |
| *sig0checks-quota* | Specifies the maximum number of concurrent SIG(0) signature checks that can be processed by the server. | server |
| *sig0checks-quota-exempt* | Exempts specific clients or client groups from SIG(0) signature checking quota. | server |
| *sig0key-checks-limit* | Specifies the maximum number of SIG(0) keys to consider when trying to verify a message. | server |
| *sig0message-checks-limit* | Specifies the maximum number of matching SIG(0) keys to try to verify a message. | server |
| *signatures-jitter* | Specifies a range for signature expirations. | dnssec |
| *signatures-refresh* | Specifies how frequently an RRSIG record is refreshed. | dnssec |
| *signatures-validity* | Indicates the validity period of an RRSIG record. | dnssec |
| *signatures-validity-dnskey* | Indicates the validity period of DNSKEY records. | dnssec |
| *slip* | Sets the number of "slipped" responses to minimize the use of forged source addresses for an attack. | query |
| *sortlist* | Controls the ordering of RRs returned to the client, based on the client's IP address. | query, deprecated |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *stale-answer-client-timeout* | Defines the amount of time (in milliseconds) that `named` waits before attempting to answer a query with a stale RRset from cache. | server, query |
| *stale-answer-enable* | Enables the returning of "stale" cached answers when the name servers for a zone are not answering. | server, query |
| *stale-answer-ttl* | Specifies the time to live (TTL) to be returned on stale answers, in seconds. | query |
| *stale-cache-enable* | Enables the retention of "stale" cached answers. | server, query |
| *stale-refresh-time* | Sets the time window for the return of "stale" cached answers before the next attempt to contact, if the name servers for a given zone are not responding. | server, query |
| *startup-notify-rate* | Specifies the rate at which NOTIFY requests are sent when the name server is first starting, or when new zones have been added. | transfer, zone |
| *statistics-channels* | Specifies the communication channels to be used by system administrators to access statistics information on the name server. | logging |
| *statistics-file* | Specifies the pathname of the file where the server appends statistics, when using `rndc stats`. | logging, server |
| *stderr* | Directs the logging channel output to the server's standard error stream. | logging |
| *streams-per-connection* | Specifies the maximum number of concurrent HTTP/2 streams over an HTTP/2 connection. | server, query |
| *suffix* | Defines trailing bits for mapped IPv4 address bits in `dns64`. | query |
| *synth-from-dnssec* | Enables support for **RFC 8198**, Aggressive Use of DNSSEC-Validated Cache. | dnssec |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *syslog* | Directs the logging channel to the system log. | logging |
| *tcp-advertised-timeout* | Sets the timeout value (in milliseconds) that the server sends in responses containing the EDNS TCP keepalive option. | query |
| *tcp-clients* | Specifies the maximum number of simultaneous client TCP connections accepted by the server. | server |
| *tcp-idle-timeout* | Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is not in use. | query |
| *tcp-initial-timeout* | Sets the amount of time (in milliseconds) that the server waits on a new TCP connection for the first message from the client. | server, query |
| *tcp-keepalive* | Adds EDNS TCP keepalive to messages sent over TCP. | server |
| *tcp-keepalive-timeout* | Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is in use. | query |
| *tcp-listen-queue* | Sets the listen-queue depth. | server |
| *tcp-only* | Sets the transport protocol to TCP. | server |
| *tcp-receive-buffer* | Sets the operating system's receive buffer size for TCP sockets. | server |
| *tcp-reuse-timeout* | Sets the amount of time (in milliseconds) that an idle outgoing TCP connection is kept open for reuse. | query |
| *tcp-send-buffer* | Sets the operating system's send buffer size for TCP sockets. | server |
| *tkey-gssapi-credential* | Sets the security credential for authentication keys requested by the GSS-TSIG protocol. | security |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *tkey-gssapi-keytab* | Sets the KRB5 keytab file to use for GSS-TSIG updates. | security |
| *tls* | Configures a TLS connection. | security |
| *tls-port* | Specifies the TCP port number the server uses to receive and send DNS-over-TLS protocol traffic. | server, query |
| *transfer-format* | Controls whether multiple records can be packed into a message during zone transfers. | transfer |
| *transfer-message-size* | Limits the uncompressed size of DNS messages used in zone transfers over TCP. | transfer |
| *transfer-source* | Defines which local IPv4 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server. | transfer |
| *transfer-source-v6* | Defines which local IPv6 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server. | transfer |
| *transfers* | Limits the number of concurrent inbound zone transfers from a server. | server |
| *transfers-in* | Limits the number of concurrent inbound zone transfers. | transfer |
| *transfers-out* | Limits the number of concurrent outbound zone transfers. | transfer |
| *transfers-per-ns* | Limits the number of concurrent inbound zone transfers from a remote server. | transfer |
| *trust-anchor-telemetry* | Instructs `named` to send specially formed queries once per day to domains for which trust anchors have been configured. | dnssec |
| *trust-anchors* | Defines *DNSSEC* trust anchors. | dnssec |
| *trusted-keys* | | deprecated |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *try-tcp-refresh* | Specifies that BIND 9 should attempt to refresh a zone using TCP if UDP queries fail. | transfer |
| *type* | Specifies the kind of zone in a given configuration. | zone |
| *type forward* | Contains forwarding statements that apply to queries within a given domain. | zone |
| *type hint* | Contains the initial set of root name servers to be used at BIND 9 startup. | zone |
| *type mirror* | Contains a DNSSEC-validated duplicate of the main data for a zone. | zone |
| *type primary* | Contains the main copy of the data for a zone. | zone |
| *type redirect* | Contains information to answer queries when normal resolution would return NXDOMAIN. | zone |
| *type secondary* | Contains a duplicate of the data for a zone that has been transferred from a primary server. | zone |
| *type static-stub* | Contains statically configured NS records for a zone. | zone |
| *type stub* | Contains a duplicate of the NS records of a zone. | zone |
| *udp-receive-buffer* | Sets the operating system's receive buffer size for UDP sockets. | server |
| *udp-send-buffer* | Sets the operating system's send buffer size for UDP sockets. | server |
| *unix* | Specifies a Unix domain socket as a control channel. | obsolete |
| *update-check-ksk* | | obsolete |
| *update-policy* | Sets fine-grained rules to allow or deny dynamic updates (DDNS), based on requester identity, updated content, etc. | transfer |

continues on next page

Table 1 – continued from previous page

| Statement | Description | Tags |
|---|---|---|
| *update-quota* | Specifies the maximum number of concurrent DNS UPDATE messages that can be processed by the server. | server |
| *use-v4-udp-ports* | Specifies a list of ports that are valid sources for UDP/IPv4 messages. | deprecated |
| *use-v6-udp-ports* | Specifies a list of ports that are valid sources for UDP/IPv6 messages. | deprecated |
| *v6-bias* | Indicates the number of milliseconds of preference to give to IPv6 name servers. | server, query |
| *validate-except* | Specifies a list of domain names at and beneath which DNSSEC validation should not be performed. | dnssec |
| *version* | Specifies the version number of the server to return in response to a `version.bind` query. | server |
| *view* | Allows a name server to answer a DNS query differently depending on who is asking. | view |
| *window* | Specifies the length of time during which responses are tracked. | query |
| *zero-no-soa-ttl* | Specifies whether to set the time to live (TTL) of the SOA record to zero, when returning authoritative negative responses to SOA queries. | zone, query, server |
| *zero-no-soa-ttl-cache* | Sets the time to live (TTL) to zero when caching a negative response to an SOA query. | zone, query, server |
| *zone* | Specifies the zone in a BIND 9 configuration. | zone |
| *zone-propagation-delay* | Sets the propagation delay from the time a zone is first updated to when the new version of the zone is served by all secondary servers. | dnssec, zone |
| *zone-statistics* | Controls the level of statistics gathered for all zones. | zone, logging |

## 8.4 Statements by Tag

These tables group the various statements permissible in `named.conf` by their corresponding tag.

### 8.4.1 DNSSEC Tag Statements

| Statement | Description |
|---|---|
| *bindkeys-file* | Specifies the pathname of a file to override the built-in trusted keys provided by *named*. |
| *cdnskey* | Specifies whether a CDNSKEY record should be published during KSK rollover. |
| *cds-digest-types* | Specifies the digest types to use for CDS resource records. |
| *check-dup-records* | Checks primary zones for records that are treated as different by DNSSEC but are semantically equal in plain DNS. |
| *checkds* | Controls whether `DS` queries are sent to parental agents. |
| *disable-algorithms* | Disables DNSSEC algorithms from a specified zone. |
| *disable-ds-digests* | Disables DS digest types from a specified zone. |
| *dnskey-ttl* | Specifies the time-to-live (TTL) for DNSKEY resource records. |
| *dnssec-accept-expired* | Instructs BIND 9 to accept expired DNSSEC signatures when validating. |
| *dnssec-loadkeys-interval* | Sets the frequency of automatic checks of the DNSSEC key repository. |
| *dnssec-policy* | Defines a key and signing policy (KASP) for zones. |
| *dnssec-validation* | Enables DNSSEC validation in *named*. |
| *inline-signing* | Specifies whether BIND 9 maintains a separate signed version of a zone. |
| *key-directory* | Indicates the directory where public and private DNSSEC key files are found. |
| *key-store* | Configures a DNSSEC key store. |
| *managed-keys-directory* | Specifies the directory in which to store the files that track managed DNSSEC keys. |
| *manual-mode* | Run key management in a manual mode. |

continues on next page

Table 2 – continued from previous page

| Statement | Description |
|---|---|
| *max-rsa-exponent-size* | Sets the maximum RSA exponent size (in bits) when validating. |
| *nsec3param* | Specifies the use of NSEC3 instead of NSEC, and sets NSEC3 parameters. |
| *nta-lifetime* | Specifies the lifetime, in seconds, for negative trust anchors added via `rndc nta`. |
| *nta-recheck* | Specifies the time interval for checking whether negative trust anchors added via `rndc nta` are still necessary. |
| *offline-ksk* | Specifies whether the DNSKEY, CDS, and CDNSKEY RRsets are being signed offline. |
| *parent-ds-ttl* | Sets the time to live (TTL) of the DS RRset used by the parent zone. |
| *parent-propagation-delay* | Sets the propagation delay from the time the parent zone is updated to when the new version is served by all of the parent zone's name servers. |
| *parental-agents* | |
| *parental-source* | Specifies which local IPv4 source address is used to send parental DS queries. |
| *parental-source-v6* | Specifies which local IPv6 source address is used to send parental DS queries. |
| *pkcs11-uri* | |
| *publish-safety* | Increases the amount of time between when keys are published and when they become active, to allow for unforeseen events. |
| *purge-keys* | Specifies the amount of time after which DNSSEC keys that have been deleted from the zone can be removed from disk. |
| *retire-safety* | Increases the amount of time a key remains published after it is no longer active, to allow for unforeseen events. |
| *secroots-file* | Specifies the pathname of the file where the server dumps security roots, when using `rndc secroots`. |
| *sig-signing-nodes* | Specifies the maximum number of nodes to be examined in each quantum, when signing a zone with a new DNSKEY. |

continues on next page

Table 2 – continued from previous page

| Statement | Description |
|---|---|
| *sig-signing-signatures* | Specifies the threshold for the number of signatures that terminates processing a quantum, when signing a zone with a new DNSKEY. |
| *sig-signing-type* | Specifies a private RDATA type to use when generating signing-state records. |
| *signatures-jitter* | Specifies a range for signature expirations. |
| *signatures-refresh* | Specifies how frequently an RRSIG record is refreshed. |
| *signatures-validity* | Indicates the validity period of an RRSIG record. |
| *signatures-validity-dnskey* | Indicates the validity period of DNSKEY records. |
| *synth-from-dnssec* | Enables support for **RFC 8198**, Aggressive Use of DNSSEC-Validated Cache. |
| *trust-anchor-telemetry* | Instructs `named` to send specially formed queries once per day to domains for which trust anchors have been configured. |
| *trust-anchors* | Defines *DNSSEC* trust anchors. |
| *validate-except* | Specifies a list of domain names at and beneath which DNSSEC validation should not be performed. |
| *zone-propagation-delay* | Sets the propagation delay from the time a zone is first updated to when the new version of the zone is served by all secondary servers. |

### 8.4.2 Logging Tag Statements

| Statement | Description |
|---|---|
| *buffered* | Controls flushing of log messages. |
| *category* | Specifies the type of data logged to a particular channel. |
| *channel* | Defines a stream of data that can be independently logged. |
| *dnstap* | Enables logging of `dnstap` messages. |
| *dnstap-identity* | Specifies an `identity` string to send in `dnstap` messages. |
| *dnstap-output* | Configures the path to which the `dnstap` frame stream is sent. |

continues on next page

Table 3 – continued from previous page

| Statement | Description |
|---|---|
| *dnstap-version* | Specifies a `version` string to send in `dnstap` messages. |
| *dump-file* | Indicates the pathname of the file where the server dumps the database after `rndc dumpdb`. |
| *fstrm-set-buffer-hint* | Sets the number of accumulated bytes in the output buffer before forcing a buffer flush. |
| *fstrm-set-flush-timeout* | Sets the number of seconds that unflushed data remains in the output buffer. |
| *fstrm-set-input-queue-size* | Sets the number of queue entries to allocate for each input queue. |
| *fstrm-set-output-notify-threshold* | Sets the number of outstanding queue entries allowed on an input queue before waking the I/O thread. |
| *fstrm-set-output-queue-model* | Sets the queuing semantics to use for queue objects. |
| *fstrm-set-output-queue-size* | Sets the number of queue entries allocated for each output queue. |
| *fstrm-set-reopen-interval* | Sets the number of seconds to wait between attempts to reopen a closed output stream. |
| *log-only* | Tests rate-limiting parameters without actually dropping any requests. |
| *logging* | Configures logging options for the name server. |
| *memstatistics* | Controls whether memory statistics are written to the file specified by `memstatistics-file` at exit. |
| *memstatistics-file* | Sets the pathname of the file where the server writes memory usage statistics on exit. |
| *null* | Causes all messages sent to the logging channel to be discarded. |
| *print-category* | Includes the category in log messages. |
| *print-severity* | Includes the severity in log messages. |
| *print-time* | Specifies the time format for log messages. |
| *querylog* | Specifies whether query logging should be active when `named` first starts. |

continues on next page

Table 3 – continued from previous page

| Statement | Description |
|---|---|
| *responselog* | Specifies whether response logging should be active when `named` first starts. |
| *severity* | Defines the priority level of log messages. |
| *statistics-channels* | Specifies the communication channels to be used by system administrators to access statistics information on the name server. |
| *statistics-file* | Specifies the pathname of the file where the server appends statistics, when using `rndc stats`. |
| *stderr* | Directs the logging channel output to the server's standard error stream. |
| *syslog* | Directs the logging channel to the system log. |
| *zone-statistics* | Controls the level of statistics gathered for all zones. |

### 8.4.3 Query Tag Statements

| Statement | Description |
|---|---|
| *all-per-second* | Limits UDP responses of all kinds. |
| *allow-query* | Specifies which hosts (an IP address list) are allowed to send queries to this resolver. |
| *allow-query-cache* | Specifies which hosts (an IP address list) can access this server's cache and thus effectively controls recursion. |
| *allow-query-cache-on* | Specifies which hosts (from an IP address list) can access this server's cache. It is used on servers with multiple interfaces. |
| *allow-query-on* | Specifies which local addresses (an IP address list) are allowed to send queries to this resolver. This option is used in multi-homed configurations. |
| *allow-recursion* | Defines an `address_match_list` of clients that are allowed to perform recursive queries. |
| *allow-recursion-on* | Specifies which local addresses can accept recursive queries. |
| *answer-cookie* | Controls whether COOKIE EDNS replies are sent in response to client queries. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *auth-nxdomain* | Controls whether BIND, acting as a resolver, provides authoritative NXDOMAIN (domain does not exist) answers. |
| *blackhole* | Defines an `address_match_list` of hosts to ignore. The server will neither respond to queries from nor send queries to these addresses. |
| *break-dnssec* | Enables `dns64` synthesis even if the validated result would cause a DNSSEC validation failure. |
| *check-dup-records* | Checks primary zones for records that are treated as different by DNSSEC but are semantically equal in plain DNS. |
| *check-names* | Restricts the character set and syntax of certain domain names in primary files and/or DNS responses received from the network. |
| *clients* | Specifies an access control list (ACL) of clients that are affected by a given `dns64` directive. |
| *deny-answer-addresses* | Rejects A or AAAA records if the corresponding IPv4 or IPv6 addresses match a given `address_match_list`. |
| *deny-answer-aliases* | Rejects CNAME or DNAME records if the "alias" name matches a given list of `domain_name` elements. |
| *dns64* | Instructs `named` to return mapped IPv4 addresses to AAAA queries when there are no AAAA records. |
| *edns-udp-size* | Sets the maximum advertised EDNS UDP buffer size to control the size of packets received from authoritative servers in response to recursive queries. |
| *endpoints* | Specifies a list of HTTP query paths on which to listen. |
| *exclude* | Allows a list of IPv6 addresses to be ignored if they appear in a domain name's AAAA records in `dns64`. |
| *exempt-clients* | Exempts specific clients or client groups from rate limiting. |
| *fetch-quota-params* | Sets the parameters for dynamic resizing of the `fetches-per-server` quota in response to detected congestion. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *fetches-per-server* | Sets the maximum number of simultaneous iterative queries allowed to be sent by a server to an upstream name server before the server blocks additional queries. |
| *fetches-per-zone* | Sets the maximum number of simultaneous iterative queries allowed to any one domain before the server blocks new queries for data in or beneath that zone. |
| *forward* | Allows or disallows fallback to recursion if forwarding has failed; it is always used in conjunction with the `forwarders` statement. |
| *forwarders* | Defines one or more resolvers to which queries are forwarded. |
| *http* | Configures HTTP endpoints on which to listen for DNS-over-HTTPS (DoH) queries. |
| *http-port* | Specifies the TCP port number the server uses to receive and send unencrypted DNS traffic via HTTP. |
| *https-port* | Specifies the TCP port number the server uses to receive and send DNS-over-HTTPS protocol traffic. |
| *ipv4only-enable* | Enables automatic IPv4 zones if a `dns64` block is configured. |
| *ipv4only-server* | Specifies the name of the server for the IPV4ONLY.ARPA zone created by `dns64`. |
| *listener-clients* | Specifies a per-listener quota for active connections. |
| *log-only* | Tests rate-limiting parameters without actually dropping any requests. |
| *mapped* | Specifies an access control list (ACL) of IPv4 addresses that are to be mapped to the corresponding A RRset in `dns64`. |
| *max-query-count* | Sets the maximum number of iterative queries while servicing a recursive query. |
| *max-query-restarts* | Sets the maximum number of chained CNAMEs to follow |
| *max-recursion-queries* | Sets the maximum number of iterative queries while servicing a recursive query. |
| *max-rsa-exponent-size* | Sets the maximum RSA exponent size (in bits) when validating. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *max-udp-size* | Sets the maximum EDNS UDP message size sent by `named`. |
| *message-compression* | Controls whether DNS name compression is used in responses to regular queries. |
| *min-table-size* | Sets the minimum size of the table used to track requests and rate-limit responses. |
| *minimal-any* | Controls whether the server replies with only one of the RRsets for a query name, when generating a positive response to a query of type ANY over UDP. |
| *minimal-responses* | Controls whether the server only adds records to the authority and additional data sections when they are required (e.g. delegations, negative responses). This improves server performance. |
| *nocookie-udp-size* | Sets the maximum size of UDP responses that are sent to queries without a valid server COOKIE. |
| *nodata-per-second* | Limits the number of empty (NODATA) responses for a valid domain name. |
| *nxdomain-redirect* | Appends the specified suffix to the original query name, when replacing an NXDOMAIN with a redirect namespace. |
| *nxdomains-per-second* | Limits the number of undefined subdomains for a valid domain name. |
| *port* | Specifies the UDP/TCP port number the server uses to receive and send DNS protocol traffic. |
| *preferred-glue* | Controls the order of glue records in an A or AAAA response. |
| *prefetch* | Specifies the "trigger" time-to-live (TTL) value at which prefetch of the current query takes place. |
| *qname-minimization* | Controls QNAME minimization behavior in the BIND 9 resolver. |
| *qps-scale* | Tightens defenses during DNS attacks by scaling back the ratio of the current query-per-second rate. |
| *query-source* | Controls the IPv4 address from which queries are issued. If *none*, then no IPv4 address would be used to issue the query and therefore only IPv6 servers are queried. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *query-source-v6* | Controls the IPv6 address from which queries are issued. If *none*, then no IPv6 address would be used to issue the query and therefore only IPv4 servers are quried. |
| *rate-limit* | Controls excessive UDP responses, to prevent BIND 9 from being used to amplify reflection denial-of-service (DoS) attacks. |
| *recursion* | Defines whether recursion and caching are allowed. |
| *recursive-clients* | Specifies the maximum number of concurrent recursive queries the server can perform. |
| *recursive-only* | Toggles whether `dns64` synthesis occurs only for recursive queries. |
| *referrals-per-second* | Limits the number of referrals or delegations to a server for a given domain. |
| *request-expire* | Specifies whether the local server requests the EDNS EXPIRE value, when acting as a secondary. |
| *request-nsid* | Controls whether an empty EDNS(0) NSID (Name Server Identifier) option is sent with all queries to authoritative name servers during iterative resolution. |
| *require-cookie* | Controls whether responses without a server cookie are accepted. |
| *require-server-cookie* | Controls whether a valid server cookie is required before sending a full response to a UDP request. |
| *resolver-query-timeout* | Specifies the length of time, in milliseconds, that a resolver attempts to resolve a recursive query before failing. |
| *response-padding* | Adds an EDNS Padding option to encrypted messages, to reduce the chance of guessing the contents based on size. |
| *response-policy* | Specifies response policy zones for the view or among global options. |
| *responses-per-second* | Limits the number of non-empty responses for a valid domain name and record type. |
| *rrset-order* | Defines the order in which equal RRs (RRsets) are returned. |
| *search* | Specifies whether a Dynamically Loadable Zone (DLZ) module is queried for an answer to a query name. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *send-cookie* | Controls whether a COOKIE EDNS option is sent along with a query. |
| *server-addresses* | Specifies a list of IP addresses to which queries should be sent in recursive resolution for a static-stub zone. |
| *slip* | Sets the number of "slipped" responses to minimize the use of forged source addresses for an attack. |
| *sortlist* | Controls the ordering of RRs returned to the client, based on the client's IP address. |
| *stale-answer-client-timeout* | Defines the amount of time (in milliseconds) that `named` waits before attempting to answer a query with a stale RRset from cache. |
| *stale-answer-enable* | Enables the returning of "stale" cached answers when the name servers for a zone are not answering. |
| *stale-answer-ttl* | Specifies the time to live (TTL) to be returned on stale answers, in seconds. |
| *stale-cache-enable* | Enables the retention of "stale" cached answers. |
| *stale-refresh-time* | Sets the time window for the return of "stale" cached answers before the next attempt to contact, if the name servers for a given zone are not responding. |
| *streams-per-connection* | Specifies the maximum number of concurrent HTTP/2 streams over an HTTP/2 connection. |
| *suffix* | Defines trailing bits for mapped IPv4 address bits in `dns64`. |
| *tcp-advertised-timeout* | Sets the timeout value (in milliseconds) that the server sends in responses containing the EDNS TCP keepalive option. |
| *tcp-idle-timeout* | Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is not in use. |
| *tcp-initial-timeout* | Sets the amount of time (in milliseconds) that the server waits on a new TCP connection for the first message from the client. |
| *tcp-keepalive-timeout* | Sets the amount of time (in milliseconds) that the server waits on an idle TCP connection before closing it, if the EDNS TCP keepalive option is in use. |

continues on next page

Table 4 – continued from previous page

| Statement | Description |
|---|---|
| *tcp-reuse-timeout* | Sets the amount of time (in milliseconds) that an idle outgoing TCP connection is kept open for reuse. |
| *tls-port* | Specifies the TCP port number the server uses to receive and send DNS-over-TLS protocol traffic. |
| *v6-bias* | Indicates the number of milliseconds of preference to give to IPv6 name servers. |
| *window* | Specifies the length of time during which responses are tracked. |
| *zero-no-soa-ttl* | Specifies whether to set the time to live (TTL) of the SOA record to zero, when returning authoritative negative responses to SOA queries. |
| *zero-no-soa-ttl-cache* | Sets the time to live (TTL) to zero when caching a negative response to an SOA query. |

### 8.4.4 Security Tag Statements

| Statement | Description |
|---|---|
| *algorithm* | Defines the algorithm to be used in a key clause. |
| *ca-file* | Specifies the path to a file containing TLS certificates for trusted CA authorities, used to verify remote peer certificates. |
| *cert-file* | Specifies the path to a file containing the TLS certificate for a connection. |
| *cipher-suites* | Specifies a list of allowed cipher suites in the order of preference for TLSv1.3 only. |
| *ciphers* | Specifies a list of allowed ciphers in the order of preference for TLSv1.2 only. |
| *dhparam-file* | Specifies the path to a file containing Diffie-Hellman parameters, for enabling cipher suites. |
| *dnsrps-enable* | Turns on the DNS Response Policy Service (DNSRPS) interface. |
| *dnsrps-library* | Specifies the path to the DNS Response Policy Service (DNSRPS) provider library. |
| *dnsrps-options* | Provides additional RPZ configuration settings, which are passed to the DNS Response Policy Service (DNSRPS) provider library. |
| *key* | Defines a shared secret key for use with *TSIG* or the command channel. |
| *key-file* | Specifies the path to a file containing the private TLS key for a connection. |
| *keys* | Specifies one or more `server_key` s to be used with a remote server. |
| *prefer-server-ciphers* | Specifies that server ciphers should be preferred over client ones. |
| *protocols* | Specifies the allowed versions of the TLS protocol. |
| *remote-hostname* | Specifies the expected hostname in the TLS certificate of the remote server. |
| *response-policy* | Specifies response policy zones for the view or among global options. |
| *secret* | Defines a Base64-encoded string to be used as the secret by the algorithm. |
| *session-keyalg* | Specifies the algorithm to use for the TSIG session key. |
| *session-keyfile* | Specifies the pathname of the file where a TSIG session key is written, when generated by `named` for use by `nsupdate -l`. |
| *session-keyname* | Specifies the key name for the TSIG session key. |
| *session-tickets* | Enables or disables session resumption through TLS session tickets. |
| *tkey-gssapi-credential* | Sets the security credential for authentication keys requested by the GSS-TSIG protocol. |
| *tkey-gssapi-keytab* | Sets the KRB5 keytab file to use for GSS-TSIG updates. |
| *tls* | Configures a TLS connection. |

### 8.4.5 Server Tag Statements

| Statement | Description |
|---|---|
| *acl* | Assigns a symbolic name to an address match list. |
| *allow-new-zones* | Controls the ability to add zones at runtime via `rndc addzone`. |
| *allow-proxy* | Defines an `address_match_list` for the client addresses allowed to send PROXYv2 headers. |
| *allow-proxy-on* | Defines an `address_match_list` for the interface addresses allowed to accept PROXYv2 headers. The option is mostly intended for multi-homed configurations. |
| *allow-recursion-on* | Specifies which local addresses can accept recursive queries. |
| *automatic-interface-scan* | Controls the automatic rescanning of network interfaces when addresses are added or removed. |
| *bogus* | Allows a remote server to be ignored. |
| *ca-file* | Specifies the path to a file containing TLS certificates for trusted CA authorities, used to verify remote peer certificates. |
| *cert-file* | Specifies the path to a file containing the TLS certificate for a connection. |
| *check-names* | Restricts the character set and syntax of certain domain names in primary files and/or DNS responses received from the network. |
| *clients-per-query* | Sets the initial minimum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients. |
| *controls* | Specifies control channels to be used to manage the name server. |
| *cookie-algorithm* | Sets the algorithm to be used when generating a server cookie. |
| *cookie-secret* | Specifies a shared secret used for generating and verifying EDNS COOKIE options within an anycast cluster. |
| *dhparam-file* | Specifies the path to a file containing Diffie-Hellman parameters, for enabling cipher suites. |
| *directory* | Sets the server's working directory. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *disable-empty-zone* | Disables individual empty zones. |
| *dns64-contact* | Specifies the name of the contact for `dns64` zones. |
| *dns64-server* | Specifies the name of the server for `dns64` zones. |
| *dnsrps-enable* | Turns on the DNS Response Policy Service (DNSRPS) interface. |
| *dnsrps-library* | Specifies the path to the DNS Response Policy Service (DNSRPS) provider library. |
| *dnsrps-options* | Provides additional RPZ configuration settings, which are passed to the DNS Response Policy Service (DNSRPS) provider library. |
| *dual-stack-servers* | Specifies host names or addresses of machines with access to both IPv4 and IPv6 transports. |
| *edns* | Controls the use of the EDNS0 (**RFC 2671**) feature. |
| *edns-version* | Sets the maximum EDNS VERSION that is sent to the server(s) by the resolver. |
| *empty-contact* | Specifies the contact name in the returned SOA record for empty zones. |
| *empty-server* | Specifies the server name in the returned SOA record for empty zones. |
| *empty-zones-enable* | Enables or disables all empty zones. |
| *endpoints* | Specifies a list of HTTP query paths on which to listen. |
| *errors-per-second* | Limits the number of errors for a valid domain name and record type. |
| *fetch-quota-params* | Sets the parameters for dynamic resizing of the `fetches-per-server` quota in response to detected congestion. |
| *fetches-per-server* | Sets the maximum number of simultaneous iterative queries allowed to be sent by a server to an upstream name server before the server blocks additional queries. |
| *fetches-per-zone* | Sets the maximum number of simultaneous iterative queries allowed to any one domain before the server blocks new queries for data in or beneath that zone. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *geoip-directory* | Specifies the directory containing GeoIP database files. |
| *hostname* | Specifies the hostname of the server to return in response to a `hostname.bind` query. |
| *http* | Configures HTTP endpoints on which to listen for DNS-over-HTTPS (DoH) queries. |
| *http-listener-clients* | Limits the number of active concurrent connections on a per-listener basis. |
| *http-port* | Specifies the TCP port number the server uses to receive and send unencrypted DNS traffic via HTTP. |
| *http-streams-per-connection* | Limits the number of active concurrent HTTP/2 streams on a per-connection basis. |
| *https-port* | Specifies the TCP port number the server uses to receive and send DNS-over-HTTPS protocol traffic. |
| *inet* | Specifies a TCP socket as a control channel. |
| *interface-interval* | Sets the interval at which the server scans the network interface list. |
| *ipv4-prefix-length* | Specifies the prefix lengths of IPv4 address blocks. |
| *ipv4only-contact* | Specifies the contact for the IPV4ONLY.ARPA zone created by *dns64*. |
| *ipv4only-server* | Specifies the name of the server for the IPV4ONLY.ARPA zone created by *dns64*. |
| *ipv6-prefix-length* | Specifies the prefix lengths of IPv6 address blocks. |
| *key-file* | Specifies the path to a file containing the private TLS key for a connection. |
| *keys* | Specifies one or more *server_key* s to be used with a remote server. |
| *lame-ttl* | Sets the resolver's lame cache. |
| *listen-on* | Specifies the IPv4 addresses on which a server listens for DNS queries. |
| *listen-on-v6* | Specifies the IPv6 addresses on which a server listens for DNS queries. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *listener-clients* | Specifies a per-listener quota for active connections. |
| *lmdb-mapsize* | Sets a maximum size for the memory map of the new-zone database in LMDB database format. |
| *masterfile-format* | Specifies the file format of zone files. |
| *masterfile-style* | Specifies the format of zone files during a dump, when the `masterfile-format` is `text`. |
| *match-mapped-addresses* | Allows IPv4-mapped IPv6 addresses to match address-match list entries for corresponding IPv4 addresses. |
| *max-cache-size* | Sets the maximum amount of memory to use for an individual cache database and its associated metadata. |
| *max-cache-ttl* | Specifies the maximum time (in seconds) that the server caches ordinary (positive) answers. |
| *max-clients-per-query* | Sets the maximum number of simultaneous recursive clients accepted by the server for any given query before the server drops additional clients. |
| *max-ncache-ttl* | Specifies the maximum retention time (in seconds) for storage of negative answers in the server's cache. |
| *max-query-count* | Sets the maximum number of iterative queries while servicing a recursive query. |
| *max-query-restarts* | Sets the maximum number of chained CNAMEs to follow |
| *max-records* | Sets the maximum number of records permitted in a zone. |
| *max-records-per-type* | Sets the maximum number of records that can be stored in an RRset. |
| *max-recursion-depth* | Sets the maximum number of levels of recursion permitted at any one time while servicing a recursive query. |
| *max-recursion-queries* | Sets the maximum number of iterative queries while servicing a recursive query. |
| *max-stale-ttl* | Specifies the maximum time that the server retains records past their normal expiry, to return them as stale records. |
| *max-table-size* | Sets the maximum size of the table used to track requests and rate-limit responses. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *max-types-per-name* | Sets the maximum number of RR types that can be stored for an owner name. |
| *max-validation-failures-per-fetch* | Sets the maximum number of DNSSEC validation failures that can happen in a single fetch. |
| *max-validations-per-fetch* | Sets the maximum number of DNSSEC validations that can happen in a single fetch. |
| *memstatistics* | Controls whether memory statistics are written to the file specified by `memstatistics-file` at exit. |
| *min-cache-ttl* | Specifies the minimum time (in seconds) that the server caches ordinary (positive) answers. |
| *min-ncache-ttl* | Specifies the minimum retention time (in seconds) for storage of negative answers in the server's cache. |
| *no-case-compress* | Specifies a list of addresses that require case-insensitive compression in responses. |
| *options* | Defines global options to be used by BIND 9. |
| *padding* | Adds EDNS Padding options to outgoing messages to increase the packet size. |
| *pid-file* | Specifies the pathname of the file where the server writes its process ID. |
| *plugin* | Configures plugins in `named.conf`. |
| *port* | Specifies the UDP/TCP port number the server uses to receive and send DNS protocol traffic. |
| *prefer-server-ciphers* | Specifies that server ciphers should be preferred over client ones. |
| *querylog* | Specifies whether query logging should be active when `named` first starts. |
| *recursing-file* | Specifies the pathname of the file where the server dumps queries that are currently recursing via `rndc recursing`. |
| *remote-servers* | Defines a list of servers to be used by primary and secondary zones. |
| *resolver-use-dns64* | Specifies whether to apply DNS64 mappings when sending queries. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *response-policy* | Specifies response policy zones for the view or among global options. |
| *responselog* | Specifies whether response logging should be active when `named` first starts. |
| *reuseport* | Enables kernel load-balancing of sockets. |
| *root-key-sentinel* | Controls whether BIND 9 responds to root key sentinel probes. |
| *server* | Defines characteristics to be associated with a remote name server. |
| *server-id* | Specifies the ID of the server to return in response to a `ID.SERVER` query. |
| *servfail-ttl* | Sets the length of time (in seconds) that a SERVFAIL response is cached. |
| *sig0checks-quota* | Specifies the maximum number of concurrent SIG(0) signature checks that can be processed by the server. |
| *sig0checks-quota-exempt* | Exempts specific clients or client groups from SIG(0) signature checking quota. |
| *sig0key-checks-limit* | Specifies the maximum number of SIG(0) keys to consider when trying to verify a message. |
| *sig0message-checks-limit* | Specifies the maximum number of matching SIG(0) keys to try to verify a message. |
| *stale-answer-client-timeout* | Defines the amount of time (in milliseconds) that `named` waits before attempting to answer a query with a stale RRset from cache. |
| *stale-answer-enable* | Enables the returning of "stale" cached answers when the name servers for a zone are not answering. |
| *stale-cache-enable* | Enables the retention of "stale" cached answers. |
| *stale-refresh-time* | Sets the time window for the return of "stale" cached answers before the next attempt to contact, if the name servers for a given zone are not responding. |
| *statistics-file* | Specifies the pathname of the file where the server appends statistics, when using `rndc stats`. |

continues on next page

Table 5 – continued from previous page

| Statement | Description |
|---|---|
| *streams-per-connection* | Specifies the maximum number of concurrent HTTP/2 streams over an HTTP/2 connection. |
| *tcp-clients* | Specifies the maximum number of simultaneous client TCP connections accepted by the server. |
| *tcp-initial-timeout* | Sets the amount of time (in milliseconds) that the server waits on a new TCP connection for the first message from the client. |
| *tcp-keepalive* | Adds EDNS TCP keepalive to messages sent over TCP. |
| *tcp-listen-queue* | Sets the listen-queue depth. |
| *tcp-only* | Sets the transport protocol to TCP. |
| *tcp-receive-buffer* | Sets the operating system's receive buffer size for TCP sockets. |
| *tcp-send-buffer* | Sets the operating system's send buffer size for TCP sockets. |
| *tls-port* | Specifies the TCP port number the server uses to receive and send DNS-over-TLS protocol traffic. |
| *transfers* | Limits the number of concurrent inbound zone transfers from a server. |
| *udp-receive-buffer* | Sets the operating system's receive buffer size for UDP sockets. |
| *udp-send-buffer* | Sets the operating system's send buffer size for UDP sockets. |
| *update-quota* | Specifies the maximum number of concurrent DNS UPDATE messages that can be processed by the server. |
| *v6-bias* | Indicates the number of milliseconds of preference to give to IPv6 name servers. |
| *version* | Specifies the version number of the server to return in response to a `version.bind` query. |
| *zero-no-soa-ttl* | Specifies whether to set the time to live (TTL) of the SOA record to zero, when returning authoritative negative responses to SOA queries. |
| *zero-no-soa-ttl-cache* | Sets the time to live (TTL) to zero when caching a negative response to an SOA query. |

### 8.4.6 Transfer Tag Statements

| Statement | Description |
|---|---|
| *allow-notify* | Defines an `address_match_list` that is allowed to send `NOTIFY` messages for the zone, in addition to addresses defined in the `primaries` option for the zone. |
| *allow-transfer* | Defines an `address_match_list` of hosts that are allowed to transfer the zone information from this server. |
| *allow-update* | Defines an `address_match_list` of hosts that are allowed to submit dynamic updates for primary zones. |
| *allow-update-forwarding* | Defines an `address_match_list` of hosts that are allowed to submit dynamic updates to a secondary server for transmission to a primary. |
| *also-notify* | Defines one or more hosts that are sent `NOTIFY` messages when zone changes occur. |
| *ixfr-from-differences* | Controls how IXFR transfers are calculated. |
| *max-ixfr-ratio* | Sets the maximum size for IXFR responses to zone transfer requests. |
| *max-journal-size* | Controls the size of journal files. |
| *max-refresh-time* | Limits the zone refresh interval to no less often than the specified value, in seconds. |
| *max-retry-time* | Limits the zone refresh retry interval to no less often than the specified value, in seconds. |
| *max-transfer-idle-in* | Specifies the number of minutes after which inbound zone transfers making no progress are terminated. |
| *max-transfer-idle-out* | Specifies the number of minutes after which outbound zone transfers making no progress are terminated. |
| *max-transfer-time-in* | Specifies the number of minutes after which inbound zone transfers are terminated. |
| *max-transfer-time-out* | Specifies the number of minutes after which outbound zone transfers are terminated. |
| *min-refresh-time* | Limits the zone refresh interval to no more often than the specified value, in seconds. |
| *min-retry-time* | Limits the zone refresh retry interval to no more often than the specified value, in seconds. |

continues on next page

Table 6 – continued from previous page

| Statement | Description |
|---|---|
| *min-transfer-rate-in* | Specifies the minimum traffic rate below which inbound zone transfers are terminated. |
| *multi-master* | Controls whether serial number mismatch errors are logged. |
| *notify* | Controls whether `NOTIFY` messages are sent on zone changes. |
| *notify-defer* | Sets the defer time (in seconds) before sending NOTIFY messages for a zone. |
| *notify-delay* | Sets the delay (in seconds) between sending sets of NOTIFY messages for a zone. |
| *notify-rate* | Specifies the rate at which NOTIFY requests are sent during normal zone maintenance operations. |
| *notify-source* | Defines the IPv4 address (and optional port) to be used for outgoing `NOTIFY` messages. |
| *notify-source-v6* | Defines the IPv6 address (and optional port) to be used for outgoing `NOTIFY` messages. |
| *notify-to-soa* | Controls whether the name servers in the NS RRset are checked against the SOA MNAME. |
| *primaries* | Defines one or more servers that zone transfer can be requested from. |
| *provide-ixfr* | Controls whether a primary responds to an incremental zone request (IXFR) or only responds with a full zone transfer (AXFR). |
| *request-expire* | Specifies whether the local server requests the EDNS EXPIRE value, when acting as a secondary. |
| *request-ixfr* | Controls whether a secondary requests an incremental zone transfer (IXFR) or a full zone transfer (AXFR). |
| *serial-query-rate* | Defines an upper limit on the number of queries per second issued by the server, when querying the SOA RRs used for zone transfers. |
| *startup-notify-rate* | Specifies the rate at which NOTIFY requests are sent when the name server is first starting, or when new zones have been added. |

continues on next page

Table 6 – continued from previous page

| Statement | Description |
|---|---|
| *transfer-format* | Controls whether multiple records can be packed into a message during zone transfers. |
| *transfer-message-size* | Limits the uncompressed size of DNS messages used in zone transfers over TCP. |
| *transfer-source* | Defines which local IPv4 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server. |
| *transfer-source-v6* | Defines which local IPv6 address(es) are bound to TCP connections used to fetch zones transferred inbound by the server. |
| *transfers-in* | Limits the number of concurrent inbound zone transfers. |
| *transfers-out* | Limits the number of concurrent outbound zone transfers. |
| *transfers-per-ns* | Limits the number of concurrent inbound zone transfers from a remote server. |
| *try-tcp-refresh* | Specifies that BIND 9 should attempt to refresh a zone using TCP if UDP queries fail. |
| *update-policy* | Sets fine-grained rules to allow or deny dynamic updates (DDNS), based on requester identity, updated content, etc. |

### 8.4.7 View Tag Statements

| Statement | Description |
|---|---|
| *attach-cache* | Allows multiple views to share a single cache database. |
| *in-view* | Specifies the view in which a given zone is defined. |
| *match-clients* | Specifies a view of DNS namespace for a given subset of client IP addresses. |
| *match-destinations* | Specifies a view of DNS namespace for a given subset of destination IP addresses. |
| *match-recursive-only* | Specifies that only recursive requests can match this view of the DNS namespace. |
| *view* | Allows a name server to answer a DNS query differently depending on who is asking. |

### 8.4.8 Zone Tag Statements

| Statement | Description |
|---|---|
| *allow-new-zones* | Controls the ability to add zones at runtime via `rndc addzone`. |
| *catalog-zones* | Configures catalog zones in `named.conf`. |
| *check-integrity* | Performs post-load zone integrity checks on primary zones. |
| *check-mx* | Checks whether an MX record appears to refer to an IP address. |
| *check-mx-cname* | Sets the response to MX records that refer to CNAMEs. |
| *check-sibling* | Specifies whether to check for sibling glue when performing integrity checks. |
| *check-spf* | Specifies whether to check for a TXT Sender Policy Framework record, if an SPF record is present. |
| *check-srv-cname* | Sets the response to SRV records that refer to CNAMEs. |
| *check-svcb* | Specifies whether to perform additional checks on SVCB records. |
| *check-wildcard* | Checks for non-terminal wildcards. |
| *database* | Specifies the type of database to be used to store zone data. |
| *disable-ds-digests* | Disables DS digest types from a specified zone. |
| *disable-empty-zone* | Disables individual empty zones. |
| *dlz* | Configures a Dynamically Loadable Zone (DLZ) database in `named.conf`. |
| *dyndb* | Configures a DynDB database in `named.conf`. |
| *empty-contact* | Specifies the contact name in the returned SOA record for empty zones. |
| *empty-server* | Specifies the server name in the returned SOA record for empty zones. |
| *empty-zones-enable* | Enables or disables all empty zones. |
| *file* | Specifies the zone's filename. |

continues on next page

Table 7 – continued from previous page

| Statement | Description |
|---|---|
| *flush-zones-on-shutdown* | Controls whether pending zone writes are flushed when the name server exits. |
| *in-view* | Specifies the view in which a given zone is defined. |
| *inline-signing* | Specifies whether BIND 9 maintains a separate signed version of a zone. |
| *journal* | Allows the default journal's filename to be overridden. |
| *masterfile-format* | Specifies the file format of zone files. |
| *max-records* | Sets the maximum number of records permitted in a zone. |
| *new-zones-directory* | Specifies the directory where configuration parameters are stored for zones added by `rndc addzone`. |
| *notify-defer* | Sets the defer time (in seconds) before sending NOTIFY messages for a zone. |
| *notify-delay* | Sets the delay (in seconds) between sending sets of NOTIFY messages for a zone. |
| *notify-rate* | Specifies the rate at which NOTIFY requests are sent during normal zone maintenance operations. |
| *parent-propagation-delay* | Sets the propagation delay from the time the parent zone is updated to when the new version is served by all of the parent zone's name servers. |
| *primaries* | Defines one or more servers that zone transfer can be requested from. |
| *response-policy* | Specifies response policy zones for the view or among global options. |
| *serial-update-method* | Specifies the update method to be used for the zone serial number in the SOA record. |
| *server-addresses* | Specifies a list of IP addresses to which queries should be sent in recursive resolution for a static-stub zone. |
| *server-names* | Specifies a list of domain names of name servers that act as authoritative servers of a static-stub zone. |
| *startup-notify-rate* | Specifies the rate at which NOTIFY requests are sent when the name server is first starting, or when new zones have been added. |

continues on next page

Table 7 – continued from previous page

| Statement | Description |
|---|---|
| *type* | Specifies the kind of zone in a given configuration. |
| *type forward* | Contains forwarding statements that apply to queries within a given domain. |
| *type hint* | Contains the initial set of root name servers to be used at BIND 9 startup. |
| *type mirror* | Contains a DNSSEC-validated duplicate of the main data for a zone. |
| *type primary* | Contains the main copy of the data for a zone. |
| *type redirect* | Contains information to answer queries when normal resolution would return NXDOMAIN. |
| *type secondary* | Contains a duplicate of the data for a zone that has been transferred from a primary server. |
| *type static-stub* | Contains statically configured NS records for a zone. |
| *type stub* | Contains a duplicate of the NS records of a zone. |
| *zero-no-soa-ttl* | Specifies whether to set the time to live (TTL) of the SOA record to zero, when returning authoritative negative responses to SOA queries. |
| *zero-no-soa-ttl-cache* | Sets the time to live (TTL) to zero when caching a negative response to an SOA query. |
| *zone* | Specifies the zone in a BIND 9 configuration. |
| *zone-propagation-delay* | Sets the propagation delay from the time a zone is first updated to when the new version of the zone is served by all secondary servers. |
| *zone-statistics* | Controls the level of statistics gathered for all zones. |

### 8.4.9 Deprecated Tag Statements

| Statement | Description |
|---|---|
| *avoid-v4-udp-ports* | Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv4 messages. |
| *avoid-v6-udp-ports* | Specifies the range(s) of ports to be excluded from use as sources for UDP/IPv6 messages. |
| *dialup* | Concentrates zone maintenance so that all transfers take place once every `heartbeat-interval`, ideally during a single call. |
| *dnssec-must-be-secure* | Defines hierarchies that must or may not be secure (signed and validated). |
| *heartbeat-interval* | Sets the interval at which the server performs zone maintenance tasks for all zones marked as `dialup`. |
| *managed-keys* | |
| *max-zone-ttl* | Specifies a maximum permissible time-to-live (TTL) value, in seconds. |
| *sortlist* | Controls the ordering of RRs returned to the client, based on the client's IP address. |
| *trusted-keys* | |
| *use-v4-udp-ports* | Specifies a list of ports that are valid sources for UDP/IPv4 messages. |
| *use-v6-udp-ports* | Specifies a list of ports that are valid sources for UDP/IPv6 messages. |

## 8.5 BIND 9 Statistics

BIND 9 maintains lots of statistics information and provides several interfaces for users to access those statistics. The available statistics include all statistics counters that are meaningful in BIND 9, and other information that is considered useful.

The statistics information is categorized into the following sections:

**Incoming Requests**
The number of incoming DNS requests for each OPCODE.

**Incoming Queries**
The number of incoming queries for each RR type.

**Outgoing Queries**
The number of outgoing queries for each RR type sent from the internal resolver, maintained per view.

**Incoming Zone Transfers**
Information about in-progress incoming zone transfers.

This section describes the information that can be seen in the HTML table about in-progress incoming zone transfers. It lists the meaning, units, and possible range of values of each column, and the key/attribute/element name (in parentheses) for the JSON and XML output formats.

**`Zone Name` (`name`)**
Text string. This is the name of the zone being transferred, as specified in the `zone` declaration on this server.

**`Zone Type` (`type`)**
Text string. This is the type of zone being transferred, as specified in the `zone` declaration on this server. Possible values are: `secondary`, `stub`, `redirect`, and `mirror`.

**Local Serial (serial)**
32-bit unsigned Integer. This is the current (old) serial number of the zone being transferred. It comes from the SOA record held on the current server.

**Remote Serial (remoteserial)**
32-bit unsigned Integer. This is the new serial number of the zone being transferred. It comes from the SOA record held on the primary server from which the zone is being transferred.

**IXFR (ixfr)**
Boolean. This says whether the transfer is incremental (using IXFR) or full (using AXFR). Possible values are: `Yes` and `No`.

**State (state)**
Text string. This is the current state of the transfer for this zone. Possible values and their meanings are:

**Needs Refresh**
The zone needs a refresh, but the process has not started yet; this can be due to different factors, like the retry interval of the zone.

**Pending**
The zone is flagged for a refresh, but the process is currently in the queue and will start shortly, or is in a waiting state because of rate-limiting; see *serial-query-rate*. The `Duration (s)` timer starts before entering this state.

**Refresh SOA**
BIND is sending a refresh SOA query to get the zone serial number and will then initiate a zone transfer, if necessary. If this step is successful, the `SOA Query` and `Got SOA` states are skipped. Otherwise, the zone transfer procedure can still be initiated, and the SOA request will be attempted using the same transport as the zone transfer. The `Duration (s)` timer restarts before entering this state, and for each attempted connection (note that in UDP mode there can be several retries during one "connection" attempt).

**Deferred**
The zone is going to be refreshed, but the process was deferred due to quota; see *transfers-in* and *transfers-per-ns*. The `Duration (s)` timer restarts before entering this state.

**SOA Query**
BIND is sending an SOA query to get the zone serial number and will then follow with a zone transfer, if necessary. The `Duration (s)` timer restarts before entering this state.

**Got SOA**
An answer for the SOA query from the previous step is received, initiating a transfer.

**Zone Transfer Request**
BIND is waiting for the zone transfer to start. The `Duration (s)` timer restarts before entering this state.

**First Data**
BIND is waiting for the first data record of the transfer.

**Receiving IXFR Data**
BIND is receiving data for an IXFR type incremental zone transfer.

**Finalizing IXFR**
BIND is finalizing an IXFR type incremental zone transfer.

**Receiving AXFR Data**
BIND is receiving data for an AXFR type zone transfer.

**Finalizing AXFR**
BIND is finalizing an AXFR type zone transfer.

> **Note**
>
> State names can change between BIND versions.

**Additional Refresh Queued (refreshqueued)**
Boolean. This shows that the zone is flagged for a refresh. This can be set to `Yes` either when the zone transfer is still in one of the pending states (see the description of the `State` column), or when the transfer is in a running state, but the zone was marked for another refresh again (e.g. because of "notify" request from a primary server). Possible values are: `Yes` and `No`.

**Local Address (localaddr)**
IP address (IPv4 or IPv6, as appropriate) and port number. This shows the source address used to establish the connection for the transfer.

**Remote Address (remoteaddr)**
IP address (IPv4 or IPv6, as appropriate) and port number. This shows the destination address used to establish the connection for the transfer.

**SOA Transport (soatransport)**
Text string. This is the transport protocol in use for the SOA query. Note that this value can potentially change during the process. For example, when the transfer is in the `Refresh SOA` state, the `SOA Transport` of the ongoing query can be shown as `UDP`. If that query fails or times out, it then can be retried using another transport, or the transfer process can be initiated in "SOA before" mode, where the SOA query will be attempted using the same transport as the zone transfer. See the description of the `State` field for more information. Possible values are: `UDP`, `TCP`, `TLS`, and `None`.

**Transport (transport)**
Text string. This is the transport protocol in use for the transfer. Possible values are: `TCP` and `TLS`.

**TSIG Key Name (tsigkeyname)**
Text string. This is the name of the TSIG key specified for use with this zone in the *zone* declaration (if any).

**Duration (s) (duration)**
64-bit unsigned Integer. This is the time, in seconds, that the current major state of the transfer process has been running so far. The timer starts after the refresh SOA request is queued (before the `Pending` state), and then restarts several times during the process to indicate the duration of the current major state. See the descriptions of the different states to find out the states before which this timer restarts.

**Messages Received (nmsg)**
64-bit unsigned Integer. This is the number of DNS messages received. It does not include transport overheads, such as TCP ACK.

**Records Received (nrecs)**
64-bit unsigned Integer. This is the number of individual RRs received so far. If an address record has, for example, five addresses associated with the same name, it counts as five RRs.

**Bytes Received (nbytes)**
64-bit unsigned Integer. This is the number of usable bytes of DNS data. It does not include transport overhead.

**Transfer Rate (B/s) (rate)**
64 bit unsigned Integer. This is the average zone transfer rate in bytes-per-second during the latest full interval that is configured by the *min-transfer-rate-in* configuration option. If no such interval has passed yet, then the overall average rate is reported instead.

> **Note**
>
> Depending on the current state of the transfer, some of the values may be empty or set to – (meaning "not available"). Also, in the case of the JSON output format, the corresponding keys can be missing or values can be set to `NULL`. For example, it is unknown whether a transfer is using AXFR or IXFR until the first data is received (see the description of the `State` column).

**Name Server Statistics**
: Statistics counters for incoming request processing.

**Zone Maintenance Statistics**
: Statistics counters regarding zone maintenance operations, such as zone transfers.

**Resolver Statistics**
: Statistics counters for name resolutions performed in the internal resolver, maintained per view.

**Cache DB RRsets**
: Statistics counters related to cache contents, maintained per view.

  The "NXDOMAIN" counter is the number of names that have been cached as nonexistent. Counters named for RR types indicate the number of active RRsets for each type in the cache database.

  If an RR type name is preceded by an exclamation point (!), it represents the number of records in the cache which indicate that the type does not exist for a particular name; this is also known as "NXRRSET". If an RR type name is preceded by a hash mark (#), it represents the number of RRsets for this type that are present in the cache but whose TTLs have expired; these RRsets may only be used if stale answers are enabled. If an RR type name is preceded by a tilde (~), it represents the number of RRsets for this type that are present in the cache database but are marked for garbage collection; these RRsets cannot be used.

**Socket I/O Statistics**
: Statistics counters for network-related events.

A subset of Name Server Statistics is collected and shown per zone for which the server has the authority, when *`zone-statistics`* is set to `full` (or `yes`), for backward compatibility. See the description of *`zone-statistics`* in *`options`* for further details.

These statistics counters are shown with their zone and view names. The view name is omitted when the server is not configured with explicit views.

There are currently two user interfaces to get access to the statistics. One is in plain-text format, dumped to the file specified by the *`statistics-file`* configuration option; the other is remotely accessible via a statistics channel when the *`statistics-channels`* statement is specified in the configuration file.

### 8.5.1 The Statistics File

The text format statistics dump begins with a line, like:

```
+++ Statistics Dump +++ (973798949)
```

The number in parentheses is a standard Unix-style timestamp, measured in seconds since January 1, 1970. Following that line is a set of statistics information, which is categorized as described above. Each section begins with a line, like:

```
++ Name Server Statistics ++
```

Each section consists of lines, each containing the statistics counter value followed by its textual description; see below for available counters. For brevity, counters that have a value of 0 are not shown in the statistics file.

The statistics dump ends with the line where the number is identical to the number in the beginning line; for example:

```
--- Statistics Dump --- (973798949)
```

### 8.5.2 Statistics Counters

The following lists summarize the statistics counters that BIND 9 provides. For each counter, the abbreviated symbol name is given; these symbols are shown in the statistics information accessed via an HTTP statistics channel. The description of the counter is also shown in the statistics file but, in this document, may be slightly modified for better readability.

#### Name Server Statistics Counters

**Requestv4**
This indicates the number of IPv4 requests received. Note: this also counts non-query requests.

**Requestv6**
This indicates the number of IPv6 requests received. Note: this also counts non-query requests.

**ReqEdns0**
This indicates the number of requests received with EDNS(0).

**ReqBadEDN SVer**
This indicates the number of requests received with an unsupported EDNS version.

**ReqTSIG**
This indicates the number of requests received with TSIG.

**ReqSIG0**
This indicates the number of requests received with SIG(0).

**ReqBadSIG**
This indicates the number of requests received with an invalid (TSIG or SIG(0)) signature.

**ReqTCP**
This indicates the number of TCP requests received.

**AuthQryRej**
This indicates the number of rejected authoritative (non-recursive) queries.

**RecQryRej**
This indicates the number of rejected recursive queries.

**XfrRej**
This indicates the number of rejected zone transfer requests.

**UpdateRej**
This indicates the number of rejected dynamic update requests.

**Response**
This indicates the number of responses sent.

**RespTruncated**
This indicates the number of truncated responses sent.

**RespEDNS0**
This indicates the number of responses sent with EDNS(0).

**RespTSIG**
This indicates the number of responses sent with TSIG.

**RespSIG0**
This indicates the number of responses sent with SIG(0).

**QrySuccess**
This indicates the number of queries that resulted in a successful answer, meaning queries which return a NOERROR response with at least one answer RR. This corresponds to the `success` counter of previous versions of BIND 9.

**QryAuthAns**
This indicates the number of queries that resulted in an authoritative answer.

**QryNoauthAns**
This indicates the number of queries that resulted in a non-authoritative answer.

**QryReferral**
This indicates the number of queries that resulted in a referral answer. This corresponds to the `referral` counter of previous versions of BIND 9.

**QryNxrrset**
This indicates the number of queries that resulted in NOERROR responses with no data. This corresponds to the `nxrrset` counter of previous versions of BIND 9.

**QrySERVFAIL**
This indicates the number of queries that resulted in SERVFAIL.

**QryFORMERR**
This indicates the number of queries that resulted in FORMERR.

**QryNXDOMAIN**
This indicates the number of queries that resulted in NXDOMAIN. This corresponds to the `nxdomain` counter of previous versions of BIND 9.

**QryRecursion**
This indicates the number of queries that caused the server to perform recursion in order to find the final answer. This corresponds to the *`recursion`* counter of previous versions of BIND 9.

**QryDuplicate**
This indicates the number of queries which the server attempted to recurse but for which it discovered an existing query with the same IP address, port, query ID, name, type, and class already being processed. This corresponds to the `duplicate` counter of previous versions of BIND 9.

**QryDropped**
This indicates the number of recursive queries dropped by the server as a result of configured limits. These limits include the settings of the *`fetches-per-zone`*, *`fetches-per-server`*, *`clients-per-query`*, and *`max-clients-per-query`* options, as well as the *`rate-limit`* option. This corresponds to the `dropped` counter of previous versions of BIND 9.

**QryFailure**
This indicates the number of query failures. This corresponds to the `failure` counter of previous versions of BIND 9. Note: this counter is provided mainly for backward compatibility with previous versions; normally, more fine-grained counters such as `AuthQryRej` and `RecQryRej` that would also fall into this counter are provided, so this counter is not of much interest in practice.

**QryNXRedir**
This indicates the number of queries that resulted in NXDOMAIN that were redirected.

**QryNXRedirRLookup**
This indicates the number of queries that resulted in NXDOMAIN that were redirected and resulted in a successful remote lookup.

**XfrReqDone**
This indicates the number of requested and completed zone transfers.

**UpdateReqFwd**
This indicates the number of forwarded update requests.

**UpdateRespFwd**
This indicates the number of forwarded update responses.

**UpdateFwdFail**
This indicates the number of forwarded dynamic updates that failed.

**UpdateDone**
This indicates the number of completed dynamic updates.

**UpdateFail**
This indicates the number of failed dynamic updates.

**UpdateBadPrereq**
This indicates the number of dynamic updates rejected due to a prerequisite failure.

**UpdateQuota**
This indicates the number of times a dynamic update or update forwarding request was rejected because the number of pending requests exceeded *update-quota*.

**RateDropped**
This indicates the number of responses dropped due to rate limits.

**RateSlipped**
This indicates the number of responses truncated by rate limits.

**RPZRewrites**
This indicates the number of response policy zone rewrites.

### Zone Maintenance Statistics Counters

**NotifyOutv4**
This indicates the number of IPv4 notifies sent.

**NotifyOutv6**
This indicates the number of IPv6 notifies sent.

**NotifyInv4**
This indicates the number of IPv4 notifies received.

**NotifyInv6**
This indicates the number of IPv6 notifies received.

**NotifyRej**
This indicates the number of incoming notifies rejected.

**SOAOutv4**
This indicates the number of IPv4 SOA queries sent.

**SOAOutv6**
This indicates the number of IPv6 SOA queries sent.

**AXFRReqv4**
This indicates the number of requested IPv4 AXFRs.

**AXFRReqv6**
This indicates the number of requested IPv6 AXFRs.

**IXFRReqv4**
This indicates the number of requested IPv4 IXFRs.

**IXFRReqv6**
This indicates the number of requested IPv6 IXFRs.

**XfrSuccess**
This indicates the number of successful zone transfer requests.

**XfrFail**
This indicates the number of failed zone transfer requests.

### Resolver Statistics Counters

**Queryv4**
This indicates the number of IPv4 queries sent.

**Queryv6**
This indicates the number of IPv6 queries sent.

**Responsev4**
This indicates the number of IPv4 responses received.

**Responsev6**
This indicates the number of IPv6 responses received.

**NXDOMAIN**
This indicates the number of NXDOMAINs received.

**SERVFAIL**
This indicates the number of SERVFAILs received.

**FORMERR**
This indicates the number of FORMERRs received.

**OtherError**
This indicates the number of other errors received.

**EDNS0Fail**
This indicates the number of EDNS(0) query failures.

**Mismatch**
This indicates the number of mismatched responses received, meaning the DNS ID, response's source address, and/or the response's source port does not match what was expected. (The port must be 53 or as defined by the *port* option.) This may be an indication of a cache poisoning attempt.

**Truncated**
This indicates the number of truncated responses received.

**Lame**
This indicates the number of lame delegations received.

**Retry**
This indicates the number of query retries performed.

**QueryAbort**
This indicates the number of queries aborted due to quota control.

**QuerySockFail**
This indicates the number of failures in opening query sockets. One common reason for such failures is due to a limitation on file descriptors.

**QueryCurUDP**
This indicates the number of UDP queries in progress.

**QueryCurTCP**
This indicates the number of TCP queries in progress.

**QueryTimeout**
This indicates the number of query timeouts.

**GlueFetchv4**
This indicates the number of IPv4 NS address fetches invoked.

**GlueFetchv6**
This indicates the number of IPv6 NS address fetches invoked.

**GlueFetchv4Fail**
This indicates the number of failed IPv4 NS address fetches.

**GlueFetchv6Fail**
This indicates the number of failed IPv6 NS address fetches.

**ValAttempt**
This indicates the number of attempted DNSSEC validations.

**ValOk**
This indicates the number of successful DNSSEC validations.

**ValNegOk**
This indicates the number of successful DNSSEC validations on negative information.

**ValFail**
This indicates the number of failed DNSSEC validations.

**QryRTTnn**
This provides a frequency table on query round-trip times (RTTs). Each `nn` specifies the corresponding frequency. In the sequence of `nn_1`, `nn_2`, ..., `nn_m`, the value of `nn_i` is the number of queries whose RTTs are between `nn_(i-1)` (inclusive) and `nn_i` (exclusive) milliseconds. For the sake of convenience, we define `nn_0` to be 0. The last entry should be represented as `nn_m+`, which means the number of queries whose RTTs are equal to or greater than `nn_m` milliseconds.

**NumFetch**
This indicates the number of active fetches.

**BucketSize**
This indicates the number of the resolver's internal buckets (a static number).

**REFUSED**
This indicates the number of REFUSED responses received.

**ClientCookieOut**
This indicates the number of COOKIE messages sent to an authoritative server with only a client cookie.

**ServerCookieOut**
This indicates the number of COOKIE messages sent to an authoritative server with both a client and a cached server cookie.

**CookieIn**
This indicates the number of COOKIE replies received from an authoritative server.

**CookieClientOk**
This indicates the number of correctly formed COOKIE client responses received.

**BadEDNSVersion**
This indicates the number of bad EDNS version replies received.

**BadCookieRcode**
This indicates the number of BADCOOKIE response codes received from an authoritative server.

**ZoneQuota**
This indicates the number of queries spilled for exceeding the *fetches-per-zone* quota.

**ServerQuota**
This indicates the number of queries spilled for exceeding the *fetches-per-server* quota.

**ClientQuota**
This indicates the number of queries spilled for exceeding the *clients-per-query* quota.

**NextItem**
This indicates the number of times the server waited for the next item after receiving an invalid response.

**Priming**
This indicates the number of priming fetches performed by the resolver.

### Socket I/O Statistics Counters

Socket I/O statistics counters are defined per socket type, which are `UDP4` (UDP/IPv4), `UDP6` (UDP/IPv6), `TCP4` (TCP/IPv4), and `TCP6` (TCP/IPv6). In the following list, `<TYPE>` represents a socket type. Not all counters are available for all socket types; exceptions are noted in the descriptions.

**<TYPE>Open**
This indicates the number of sockets opened successfully.

**<TYPE>OpenFail**
This indicates the number of failures to open sockets.

**<TYPE>Close**
This indicates the number of closed sockets.

**<TYPE>BindFail**
This indicates the number of failures to bind sockets.

**<TYPE>ConnFail**
This indicates the number of failures to connect sockets.

**<TYPE>Conn**
This indicates the number of connections established successfully.

**<TYPE>AcceptFail**
This indicates the number of failures to accept incoming connection requests. This counter does not apply to the `UDP` type.

**<TYPE>Accept**
This indicates the number of incoming connections successfully accepted. This counter does not apply to the `UDP` type.

**<TYPE>SendErr**
This indicates the number of errors in socket send operations.

**<TYPE>RecvErr**
This indicates the number of errors in socket receive operations, including errors of send operations on a connected UDP socket, notified by an ICMP error message.

CHAPTER

NINE

# TROUBLESHOOTING

## 9.1 Common Problems

### 9.1.1 It's Not Working; How Can I Figure Out What's Wrong?

The best solution to installation and configuration issues is to take preventive measures by setting up logging files beforehand. The log files provide hints and information that can be used to identify anything that went wrong and fix the problem.

### 9.1.2 EDNS Compliance Issues

EDNS (Extended DNS) is a standard that was first specified in 1999. It is required for DNSSEC validation, DNS COOKIE options, and other features. There are broken and outdated DNS servers and firewalls still in use which misbehave when queried with EDNS; for example, they may drop EDNS queries rather than replying with FORMERR. BIND and other recursive name servers have traditionally employed workarounds in this situation, retrying queries in different ways and eventually falling back to plain DNS queries without EDNS.

Such workarounds cause unnecessary resolution delays, increase code complexity, and prevent deployment of new DNS features. In February 2019, all major DNS software vendors removed these workarounds; see https://www.dnsflagday.net/2019/ for further details. This change was implemented in BIND as of release 9.14.0.

As a result, some domains may be non-resolvable without manual intervention. In these cases, resolution can be restored by adding *server* clauses for the offending servers, or by specifying `edns no` or `send-cookie no`, depending on the specific noncompliance.

To determine which *server* clause to use, run the following commands to send queries to the authoritative servers for the broken domain:

```
dig soa <zone> @<server> +dnssec
dig soa <zone> @<server> +dnssec +nocookie
dig soa <zone> @<server> +noedns
```

If the first command fails but the second succeeds, the server most likely needs `send-cookie no`. If the first two fail but the third succeeds, then the server needs EDNS to be fully disabled with `edns no`.

Please contact the administrators of noncompliant domains and encourage them to upgrade their broken DNS servers.

### 9.1.3 Inspecting Encrypted DNS Traffic

> **Note**
>
> This feature requires support from the cryptographic library that BIND 9 is built against. For OpenSSL, version 1.1.1 or newer is required (use *named -V* to check).

By definition, TLS-encrypted traffic (e.g. DNS over TLS, DNS over HTTPS) is opaque to packet sniffers, which makes debugging problems with encrypted DNS close to impossible. However, Wireshark offers a solution to this problem by being able to read key log files. In order to make *named* prepare such a file, set the `SSLKEYLOGFILE` environment variable to either:

- the string `config` (`SSLKEYLOGFILE=config`); this requires defining a *logging channel* which will handle messages belonging to the `sslkeylog` category,
- the path to the key file to write (`SSLKEYLOGFILE=/path/to/file`); this is equivalent to the following *logging* configuration:

```
channel default_sslkeylogfile {
    file "${SSLKEYLOGFILE}" versions 10 size 100m suffix timestamp;
};

category sslkeylog {
    default_sslkeylogfile;
};
```

> **Note**
>
> When using `SSLKEYLOGFILE=config`, augmenting the log channel output using options like *print-time* or *print-severity* is strongly discouraged as it will likely make the key log file unusable.

When the `SSLKEYLOGFILE` environment variable is set, each TLS connection established by *named* (both incoming and outgoing) causes about 1 kilobyte of data to be written to the key log file.

> **Warning**
>
> Due to the limitations of the current logging code in BIND 9, enabling TLS pre-master secret logging adversely affects *named* performance.

## 9.2 Incrementing and Changing the Serial Number

Zone serial numbers are just numbers — they are not date-related. However, many people set them to a number that represents a date, usually of the form YYYYMMDDRR. Occasionally they make a mistake and set the serial number to a date in the future, then try to correct it by setting it to the current date. This causes problems because serial numbers are used to indicate that a zone has been updated. If the serial number on the secondary server is lower than the serial number on the primary, the secondary server attempts to update its copy of the zone.

Setting the serial number to a lower number on the primary server than the one on the secondary server means that the secondary will not perform updates to its copy of the zone.

The solution to this is to add 2147483647 (2^31-1) to the number, reload the zone and make sure all secondaries have updated to the new zone serial number, then reset it to the desired number and reload the zone again.

## 9.3 Where Can I Get Help?

The BIND-users mailing list, at https://lists.isc.org/mailman/listinfo/bind-users, is an excellent resource for peer user support. In addition, ISC maintains a Knowledgebase of helpful articles at https://kb.isc.org.

Internet Systems Consortium (ISC) offers annual support agreements for BIND 9, ISC DHCP, and Kea DHCP. All paid support contracts include advance security notifications; some levels include service level agreements (SLAs), premium software features, and increased priority on bug fixes and feature requests.

Please contact info@isc.org or visit https://www.isc.org/contact/ for more information.

CHAPTER

# TEN

# BUILDING BIND 9

To build on a Unix or Linux system, use:

```
$ autoreconf -fi ### (only if building from the git repository)
$ ./configure
$ make
```

Several environment variables affect compilation, and they can be set before running `configure`. The most significant ones are:

| Variable | Description |
|---|---|
| `CC` | The C compiler to use. `configure` tries to figure out the right one for supported systems. |
| `CFLAGS` | The C compiler flags. Defaults to include -g and/or -O2 as supported by the compiler. Please include `-g` if `CFLAGS` needs to be set. |
| `LDFLAGS` | The linker flags. Defaults to an empty string. |

Additional environment variables affecting the build are listed at the end of the `configure` help text, which can be obtained by running the command:

```
$ ./configure --help
```

If using Emacs, the `make tags` command may be helpful.

## 10.1 Required Libraries

To build BIND 9, the following packages must be installed:

- a C11-compliant compiler
- `libcrypto`, `libssl`
- `liburcu`
- `libuv`
- `perl`
- `pkg-config` / `pkgconfig` / `pkgconf`

BIND 9.20 requires `libuv` 1.34.0 or higher; using `libuv` >= 1.40.0 is recommended. Compiling or running with `libuv` 1.35.0 or 1.36.0 is not supported, as this could lead to an assertion failure in the UDP receive code. On older systems

an updated `libuv` package needs to be installed from sources, such as EPEL, PPA, or other native sources. The other option is to build and install `libuv` from source.

OpenSSL 1.0.2e or newer is required. If the OpenSSL library is installed in a nonstandard location, specify the prefix using `--with-openssl=<PREFIX>` on the `configure` command line. To use a PKCS#11 hardware service module for cryptographic operations, `engine_pkcs11` from the OpenSC project must be compiled and used.

The Userspace RCU library `liburcu` (https://liburcu.org/) is used for lock-free data structures and concurrent safe memory reclamation.

On Linux, process capabilities are managed in user space using the `libcap` library (https://git.kernel.org/pub/scm/libs/libcap/libcap.git/), which can be installed on most Linux systems via the `libcap-dev` or `libcap-devel` package.

To build BIND from the git repository, the following tools must also be installed:

- `autoconf` (includes `autoreconf`)
- `automake`
- `libtool`

## 10.2 Optional Features

To see a full list of configuration options, run `configure --help`.

To improve performance, use of the `jemalloc` library (https://jemalloc.net/) is strongly recommended. Version 4.0.0 or newer is required when in use.

To support **DNS over HTTPS (DoH)**, the server must be linked with `libnghttp2` (https://nghttp2.org/). If the library is unavailable, `--disable-doh` can be used to disable DoH support.

To support the HTTP statistics channel, the server must be linked with at least one of the following libraries: `libxml2` (https://gitlab.gnome.org/GNOME/libxml2/-/wikis/home) or `json-c` (https://github.com/json-c/json-c). If these are installed at a nonstandard location, then:

- for `libxml2`, specify the prefix using `--with-libxml2=/prefix`,
- for `json-c`, adjust `PKG_CONFIG_PATH`.

To support compression on the HTTP statistics channel, the server must be linked against `zlib` (https://zlib.net/). If this is installed in a nonstandard location, specify the prefix using `--with-zlib=/prefix`.

To support storing configuration data for runtime-added zones in an LMDB database, the server must be linked with `liblmdb` (https://github.com/LMDB/lmdb). If this is installed in a nonstandard location, specify the prefix using `--with-lmdb=/prefix`.

To support MaxMind GeoIP2 location-based ACLs, the server must be linked with `libmaxminddb` (https://maxmind.github.io/libmaxminddb/). This is turned on by default if the library is found; if the library is installed in a nonstandard location, specify the prefix using `--with-maxminddb=/prefix`. GeoIP2 support can be switched off with `--disable-geoip`.

For DNSTAP packet logging, `libfstrm` (https://github.com/farsightsec/fstrm) and `libprotobuf-c` (https://protobuf.dev) must be installed, and BIND must be configured with `--enable-dnstap`.

To support internationalized domain names in *`dig`*, `libidn2` (https://www.gnu.org/software/libidn/#libidn2) must be installed. If the library is installed in a nonstandard location, specify the prefix using `--with-libidn2=/prefix` or adjust `PKG_CONFIG_PATH`.

For line editing in *`nsupdate`* and *`nslookup`*, either the `readline` (https://tiswww.case.edu/php/chet/readline/rltop.html) or the `libedit` library (https://www.thrysoee.dk/editline/) must be installed. If these are installed at a nonstandard location, adjust `PKG_CONFIG_PATH`. `readline` is used by default, and `libedit` can be explicitly requested using `--with-readline=libedit`.

On some platforms it is necessary to explicitly request large file support to handle files bigger than 2GB. This can be done by using `--enable-largefile` on the `configure` command line.

Support for the "fixed" RRset-order option can be enabled or disabled by specifying `--enable-fixed-rrset` or `--disable-fixed-rrset` on the `configure` command line. By default, fixed RRset-order is disabled to reduce memory footprint.

The `--enable-querytrace` option causes *named* to log every step while processing every query. The `--enable-singletrace` option turns on the same verbose tracing, but allows an individual query to be separately traced by setting its query ID to 0. These options should only be enabled when debugging, because they have a significant negative impact on query performance.

`make install` installs *named* and the various BIND 9 libraries. By default, installation is into /usr/local, but this can be changed with the `--prefix` option when running `configure`.

The option `--sysconfdir` can be specified to set the directory where configuration files such as *named.conf* go by default; `--localstatedir` can be used to set the default parent directory of `run/named.pid`. `--sysconfdir` defaults to `$prefix/etc` and `--localstatedir` defaults to `$prefix/var`.

## 10.3 macOS

Building on macOS assumes that the "Command Tools for Xcode" are installed. These can be downloaded from https://developer.apple.com/xcode/resources/ or, if Xcode is already installed, simply run `xcode-select --install`. (Note that an Apple ID may be required to access the download page.)

CHAPTER

ELEVEN

# DNSSEC GUIDE

## 11.1 Preface

### 11.1.1 Organization

This document provides introductory information on how DNSSEC works, how to configure BIND 9 to support some common DNSSEC features, and some basic troubleshooting tips. The chapters are organized as follows:

*Introduction* covers the intended audience for this document, assumed background knowledge, and a basic introduction to the topic of DNSSEC.

*Getting Started* covers various requirements before implementing DNSSEC, such as software versions, hardware capacity, network requirements, and security changes.

*Validation* walks through setting up a validating resolver, and gives both more information on the validation process and some examples of tools to verify that the resolver is properly validating answers.

*Signing* explains how to set up a basic signed authoritative zone, details the relationship between a child and a parent zone, and discusses ongoing maintenance tasks.

*Basic DNSSEC Troubleshooting* provides some tips on how to analyze and diagnose DNSSEC-related problems.

*Advanced Discussions* covers several topics, including key generation, key storage, key management, NSEC and NSEC3, and some disadvantages of DNSSEC.

*Recipes* provides several working examples of common DNSSEC solutions, with step-by-step details.

*Commonly Asked Questions* lists some commonly asked questions and answers about DNSSEC.

### 11.1.2 Acknowledgements

This document was originally authored by Josh Kuo of DeepDive Networking. He can be reached at josh.kuo@gmail.com.

Thanks to the following individuals (in no particular order) who have helped in completing this document: Jeremy C. Reed, Heidi Schempf, Stephen Morris, Jeff Osborn, Vicky Risk, Jim Martin, Evan Hunt, Mark Andrews, Michael McNally, Kelli Blucher, Chuck Aurora, Francis Dupont, Rob Nagy, Ray Bellis, Matthijs Mekking, and Suzanne Goldlust.

Special thanks goes to Cricket Liu and Matt Larson for their selflessness in knowledge sharing.

Thanks to all the reviewers and contributors, including John Allen, Jim Young, Tony Finch, Timothe Litt, and Dr. Jeffry A. Spain.

The sections on key rollover and key timing metadata borrowed heavily from the Internet Engineering Task Force draft titled "DNSSEC Key Timing Considerations" by S. Morris, J. Ihren, J. Dickinson, and W. Mekking, subsequently published as **RFC 7583**.

Icons made by Freepik and SimpleIcon from Flaticon, licensed under Creative Commons BY 3.0.

## 11.2 Introduction

### 11.2.1 Who Should Read this Guide?

This guide is intended as an introduction to DNSSEC for the DNS administrator who is already comfortable working with the existing BIND and DNS infrastructure. He or she might be curious about DNSSEC, but may not have had the time to investigate DNSSEC, to learn whether DNSSEC should be a part of his or her environment, and understand what it means to deploy it in the field.

This guide provides basic information on how to configure DNSSEC using BIND 9.16.0 or later. Most of the information and examples in this guide also apply to versions of BIND later than 9.9.0, but some of the key features described here were only introduced in version 9.16.0. Readers are assumed to have basic working knowledge of the Domain Name System (DNS) and related network infrastructure, such as concepts of TCP/IP. In-depth knowledge of DNS and TCP/IP is not required. The guide assumes no prior knowledge of DNSSEC or related technology such as public key cryptography.

### 11.2.2 Who May Not Want to Read this Guide?

If you are already operating a DNSSEC-signed zone, you may not learn much from the first half of this document, and you may want to start with *Advanced Discussions*. If you want to learn about details of the protocol extension, such as data fields and flags, or the new record types, this document can help you get started but it does not include all the technical details.

If you are experienced in DNSSEC, you may find some of the concepts in this document to be overly simplified for your taste, and some details are intentionally omitted at times for ease of illustration.

If you administer a large or complex BIND environment, this guide may not provide enough information for you, as it is intended to provide only basic, generic working examples.

If you are a top-level domain (TLD) operator, or administer zones under signed TLDs, this guide can help you get started, but it does not provide enough details to serve all of your needs.

If your DNS environment uses DNS products other than (or in addition to) BIND, this document may provide some background or overlapping information, but you should check each product's vendor documentation for specifics.

Finally, deploying DNSSEC on internal or private networks is not covered in this document, with the exception of a brief discussion in *DNSSEC on Private Networks*.

### 11.2.3 What is DNSSEC?

The Domain Name System (DNS) was designed in a day and age when the Internet was a friendly and trusting place. The protocol itself provides little protection against malicious or forged answers. DNS Security Extensions (DNSSEC) addresses this need, by adding digital signatures into DNS data so that each DNS response can be verified for integrity (the answer did not change during transit) and authenticity (the data came from the true source, not an impostor). In the ideal world, when DNSSEC is fully deployed, every single DNS answer can be validated and trusted.

DNSSEC does not provide a secure tunnel; it does not encrypt or hide DNS data. It operates independently of an existing Public Key Infrastructure (PKI). It does not need SSL certificates or shared secrets. It was designed with backwards compatibility in mind, and can be deployed without impacting "old" unsecured domain names.

DNSSEC is deployed on the three major components of the DNS infrastructure:

- *Recursive Servers*: People use recursive servers to lookup external domain names such as `www.example.com`. Operators of recursive servers need to enable DNSSEC validation. With validation enabled, recursive servers carry out additional tasks on each DNS response they receive to ensure its authenticity.
- *Authoritative Servers*: People who publish DNS data on their name servers need to sign that data. This entails creating additional resource records, and publishing them to parent domains where necessary. With DNSSEC enabled, authoritative servers respond to queries with additional DNS data, such as digital signatures and keys, in addition to the standard answers.

- *Applications*: This component lives on every client machine, from web servers to smart phones. This includes resolver libraries on different operating systems, and applications such as web browsers.

In this guide, we focus on the first two components, Recursive Servers and Authoritative Servers, and only lightly touch on the third component. We look at how DNSSEC works, how to configure a validating resolver, how to sign DNS zone data, and other operational tasks and considerations.

### 11.2.4 What Does DNSSEC Add to DNS?

**Note**

Public Key Cryptography works on the concept of a pair of keys: one made available to the world publicly, and one kept in secrecy privately. Not surprisingly, they are known as a public key and a private key. If you are not familiar with the concept, think of it as a cleverly designed lock, where one key locks and one key unlocks. In DNSSEC, we give out the unlocking public key to the rest of the world, while keeping the locking key private. To learn how this is used to secure DNS messages, see *How Are Answers Verified?*.

DNSSEC introduces eight new resource record types:

- RRSIG (digital resource record signature)
- DNSKEY (public key)
- DS (parent-child)
- NSEC (proof of nonexistence)
- NSEC3 (proof of nonexistence)
- NSEC3PARAM (proof of nonexistence)
- CDS (child-parent signaling)
- CDNSKEY (child-parent signaling)

This guide does not go deep into the anatomy of each resource record type; the details are left for the reader to research and explore. Below is a short introduction on each of the new record types:

- *RRSIG*: With DNSSEC enabled, just about every DNS answer (A, PTR, MX, SOA, DNSKEY, etc.) comes with at least one resource record signature, or RRSIG. These signatures are used by recursive name servers, also known as validating resolvers, to verify the answers received. To learn how digital signatures are generated and used, see *How Are Answers Verified?*.
- *DNSKEY*: DNSSEC relies on public-key cryptography for data authenticity and integrity. There are several keys used in DNSSEC, some private, some public. The public keys are published to the world as part of the zone data, and they are stored in the DNSKEY record type.

  In general, keys in DNSSEC are used for one or both of the following roles: as a Zone Signing Key (ZSK), used to protect all zone data; or as a Key Signing Key (KSK), used to protect the zone's keys. A key that is used for both roles is referred to as a Combined Signing Key (CSK). We talk about keys in more detail in *DNSSEC Keys*.
- *DS*: One of the critical components of DNSSEC is that the parent zone can "vouch" for its child zone. The DS record is verifiable information (generated from one of the child's public keys) that a parent zone publishes about its child as part of the chain of trust. To learn more about the Chain of Trust, see *Chain of Trust*.
- *NSEC, NSEC3, NSEC3PARAM*: These resource records all deal with a very interesting problem: proving that something does not exist. We look at these record types in more detail in *Proof of Non-Existence (NSEC and NSEC3)*.

- *CDS, CDNSKEY*: The CDS and CDNSKEY resource records apply to operational matters and are a way to signal to the parent zone that the DS records it holds for the child zone should be updated. This is covered in more detail in *The CDS and CDNSKEY Resource Records*.

### 11.2.5 How Does DNSSEC Change DNS Lookup?

Traditional (insecure) DNS lookup is simple: a recursive name server receives a query from a client to lookup a name like `www.isc.org`. The recursive name server tracks down the authoritative name server(s) responsible, sends the query to one of the authoritative name servers, and waits for it to respond with the answer.

With DNSSEC validation enabled, a validating recursive name server (a.k.a. a *validating resolver*) asks for additional resource records in its query, hoping the remote authoritative name servers respond with more than just the answer to the query, but some proof to go along with the answer as well. If DNSSEC responses are received, the validating resolver performs cryptographic computation to verify the authenticity (the origin of the data) and integrity (that the data was not altered during transit) of the answers, and even asks the parent zone as part of the verification. It repeats this process of get-key, validate, ask-parent, and its parent, and its parent, all the way until the validating resolver reaches a key that it trusts. In the ideal, fully deployed world of DNSSEC, all validating resolvers only need to trust one key: the root key.

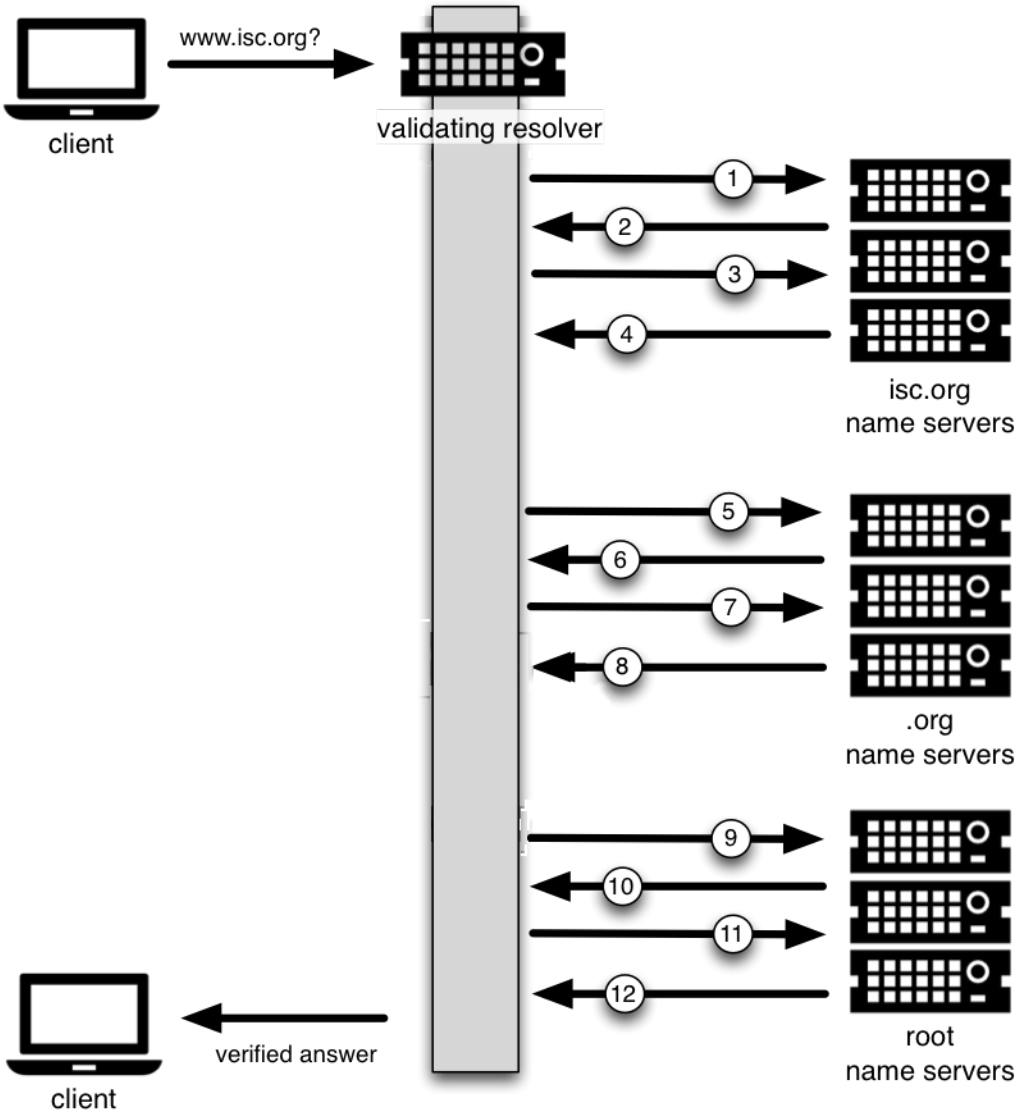
### 11.2.6 The 12-Step DNSSEC Validation Process (Simplified)

The following example shows the 12 steps of the DNSSEC validating process at a very high level, looking up the name `www.isc.org`:

1. Upon receiving a DNS query from a client to resolve `www.isc.org`, the validating resolver follows standard DNS protocol to track down the name server for `isc.org`, and sends it a DNS query to ask for the A record of `www.isc.org`. But since this is a DNSSEC-enabled resolver, the outgoing query has a bit set indicating it wants DNSSEC answers, hoping the name server that receives it is DNSSEC-enabled and can honor this secure request.
2. The `isc.org` name server is DNSSEC-enabled, so it responds with both the answer (in this case, an A record) and a digital signature for verification purposes.
3. The validating resolver requires cryptographic keys to be able to verify the digital signature, so it asks the `isc.org` name server for those keys.
4. The `isc.org` name server responds with the cryptographic keys (and digital signatures of the keys) used to generate the digital signature that was sent in #2. At this point, the validating resolver can use this information to verify the answers received in #2.

   Let's take a quick break here and look at what we've got so far... how can our server trust this answer? If a clever attacker had taken over the `isc.org` name server(s), of course she would send matching keys and signatures. We need to ask someone else to have confidence that we are really talking to the real `isc.org` name server. This is a critical part of DNSSEC: at some point, the DNS administrators at `isc.org` uploaded some cryptographic information to its parent, `.org`, maybe through a secure web form, maybe through an email exchange, or perhaps in person. In any event, at some point some verifiable information about the child (`isc.org`) was sent to the parent (`.org`) for safekeeping.
5. The validating resolver asks the parent (`.org`) for the verifiable information it keeps on its child, `isc.org`.
6. Verifiable information is sent from the `.org` server. At this point, the validating resolver compares this to the answer it received in #4; if the two of them match, it proves the authenticity of `isc.org`.

   Let's examine this process. You might be thinking to yourself, what if the clever attacker that took over `isc.org` also compromised the `.org` servers? Of course all this information would match! That's why we turn our attention now to the `.org` server, interrogate it for its cryptographic keys, and move one level up to `.org`'s parent, root.
7. The validating resolver asks the `.org` authoritative name server for its cryptographic keys, to verify the answers received in #6.
8. The `.org` name server responds with the answer (in this case, keys and signatures). At this point, the validating resolver can verify the answers received in #6.


www.isc.org?
client
validating resolver
1
2
3
4
isc.org
name servers
5
6
7
8
.org
name servers
9
10
11
12
root
name servers
verified answer
client

9. The validating resolver asks root (`.org`'s parent) for the verifiable information it keeps on its child, `.org`.
10. The root name server sends back the verifiable information it keeps on `.org`. The validating resolver uses this information to verify the answers received in #8.

    So at this point, both `isc.org` and `.org` check out. But what about root? What if this attacker is really clever and somehow tricked us into thinking she's the root name server? Of course she would send us all matching information! So we repeat the interrogation process and ask for the keys from the root name server.
11. The validating resolver asks the root name server for its cryptographic keys to verify the answer(s) received in #10.
12. The root name server sends its keys; at this point, the validating resolver can verify the answer(s) received in #10.

### 11.2.7 Chain of Trust

But what about the root server itself? Who do we go to verify root's keys? There's no parent zone for root. In security, you have to trust someone, and in the perfectly protected world of DNSSEC (we talk later about the current imperfect state and ways to work around it), each validating resolver would only have to trust one entity, that is, the root name server. The validating resolver already has the root key on file (we discuss later how we got the root key file). So after the answer in #12 is received, the validating resolver compares it to the key it already has on file. Providing one of the keys in the answer matches the one on file, we can trust the answer from root. Thus we can trust `.org`, and thus we can trust `isc.org`. This is known as the "chain of trust" in DNSSEC.

We revisit this 12-step process again later in *How Does DNSSEC Change DNS Lookup (Revisited)?* with more technical details.

### 11.2.8 Why is DNSSEC Important? (Why Should I Care?)

You might be thinking to yourself: all this DNSSEC stuff sounds wonderful, but why should I care? Below are some reasons why you may want to consider deploying DNSSEC:

1. *Being a good netizen*: By enabling DNSSEC validation (as described in *Validation*) on your DNS servers, you're protecting your users and yourself a little more by checking answers returned to you; by signing your zones (as described in *Signing*), you are making it possible for other people to verify your zone data. As more people adopt DNSSEC, the Internet as a whole becomes more secure for everyone.
2. *Compliance*: You may not even get a say in implementing DNSSEC, if your organization is subject to compliance standards that mandate it. For example, the US government set a deadline in 2008 to have all `.gov` subdomains signed by December 2009.[1] So if you operate a subdomain in `.gov`, you must implement DNSSEC to be compliant. ICANN also requires that all new top-level domains support DNSSEC.
3. *Enhanced Security*: Okay, so the big lofty goal of "let's be good" doesn't appeal to you, and you don't have any compliance standards to worry about. Here is a more practical reason why you should consider DNSSEC: in the event of a DNS-based security breach, such as cache poisoning or domain hijacking, after all the financial and brand damage done to your domain name, you might be placed under scrutiny for any preventive measure that could have been put in place. Think of this like having your website only available via HTTP but not HTTPS.
4. *New Features*: DNSSEC brings not only enhanced security, but also a whole new suite of features. Once DNS can be trusted completely, it becomes possible to publish SSL certificates in DNS, or PGP keys for fully automatic cross-platform email encryption, or SSH fingerprints.... New features are still being developed, but they all rely on a trustworthy DNS infrastructure. To take a peek at these next-generation DNS features, check out *Introduction to DANE*.

[1] The Office of Management and Budget (OMB) for the US government published a memo in 2008, requesting all `.gov` subdomains to be DNSSEC-signed by December 2009. This explains why `.gov` is the most-deployed DNSSEC domain currently, with around 90% of subdomains signed.

### 11.2.9 How Does DNSSEC Change My Job as a DNS Administrator?

With this protocol extension, some of the things you were used to in DNS have changed. As the DNS administrator, you have new maintenance tasks to perform on a regular basis (as described in *Maintenance Tasks*); when there is a DNS resolution problem, you have new troubleshooting techniques and tools to use (as described in *Basic DNSSEC Troubleshooting*). BIND 9 tries its best to make these things as transparent and seamless as possible. In this guide, we try to use configuration examples that result in the least amount of work for BIND 9 DNS administrators.

## 11.3 Getting Started

### 11.3.1 Software Requirements

This guide assumes BIND 9.18.0 or newer, although the more elaborate manual procedures do work with all versions of BIND later than 9.9.

We recommend running the latest stable version to get the most complete DNSSEC configuration, as well as the latest security fixes.

### 11.3.2 Hardware Requirements

#### Recursive Server Hardware

Enabling DNSSEC validation on a recursive server makes it a *validating resolver*. The job of a validating resolver is to fetch additional information that can be used to computationally verify the answer set. Contrary to popular belief, the increase in resource consumption is very modest:

1. *CPU*: a validating resolver executes cryptographic functions on cache-miss answers, which leads to increased CPU usage. Thanks to standard DNS caching and contemporary CPUs, the increase in CPU-time consumption in a steady state is negligible - typically on the order of 5%. For a brief period (a few minutes) after the resolver starts, the increase might be as much as 20%, but it quickly decreases as the DNS cache fills in.
2. *System memory*: DNSSEC leads to larger answer sets and occupies more memory space. With typical ISP traffic and the state of the Internet as of mid-2022, memory consumption for the cache increases by roughly 20%.
3. *Network interfaces*: although DNSSEC does increase the amount of DNS traffic overall, in practice this increase is often within measurement error.

#### Authoritative Server Hardware

On the authoritative server side, DNSSEC is enabled on a zone-by-zone basis. When a zone is DNSSEC-enabled, it is also known as "signed." Below are the expected changes to resource consumption caused by serving DNSSEC-signed zones:

1. *CPU*: a DNSSEC-signed zone requires periodic re-signing, which is a cryptographic function that is CPU-intensive. If your DNS zone is dynamic or changes frequently, that also adds to higher CPU loads.
2. *System storage*: A signed zone is definitely larger than an unsigned zone. How much larger? See *Your Zone, Before and After DNSSEC* for a comparison example. The final size depends on the structure of the zone, the signing algorithm, the number of keys, the choice of NSEC or NSEC3, the ratio of signed delegations, the zone file format, etc. Usually, the size of a signed zone ranges from a negligible increase to as much as three times the size of the unsigned zone.
3. *System memory*: Larger DNS zone files take up not only more storage space on the file system, but also more space when they are loaded into system memory. The final memory consumption also depends on all the variables listed above: in the typical case the increase is around half of the unsigned zone memory consumption, but it can be as high as three times for some corner cases.

4. *Network interfaces*: While your authoritative name servers will begin sending back larger responses, it is unlikely that you need to upgrade your network interface card (NIC) on the name server unless you have some truly outdated hardware.

One factor to consider, but over which you really have no control, is the number of users who query your domain name who themselves have DNSSEC enabled. As of mid-2022, measurements by APNIC show 41% of Internet users send DNSSEC-aware queries. This means that more DNS queries for your domain will take advantage of the additional security features, which will result in increased system load and possibly network traffic.

### 11.3.3 Network Requirements

From a network perspective, DNS and DNSSEC packets are very similar; DNSSEC packets are just bigger, which means DNS is more likely to use TCP. You should test for the following two items to make sure your network is ready for DNSSEC:

1. *DNS over TCP*: Verify network connectivity over TCP port 53, which may mean updating firewall policies or Access Control Lists (ACL) on routers. See *Wait... DNS Uses TCP?* for more details.
2. *Large UDP packets*: Some network equipment, such as firewalls, may make assumptions about the size of DNS UDP packets and incorrectly reject DNS traffic that appears "too big." Verify that the responses your name server generates are being seen by the rest of the world: see *What's EDNS All About (And Why Should I Care)?* for more details.

### 11.3.4 Operational Requirements

#### Parent Zone

Before starting your DNSSEC deployment, check with your parent zone administrators to make sure they support DNSSEC. This may or may not be the same entity as your registrar. As you will see later in *Working With the Parent Zone*, a crucial step in DNSSEC deployment is establishing the parent-child trust relationship. If your parent zone does not yet support DNSSEC, contact that administrator to voice your concerns.

#### Security Requirements

Some organizations may be subject to stricter security requirements than others. Check to see if your organization requires stronger cryptographic keys be generated and stored, and how often keys need to be rotated. The examples presented in this document are not intended for high-value zones. We cover some of these security considerations in *Advanced Discussions*.

## 11.4 Validation

### 11.4.1 Easy-Start Guide for Recursive Servers

This section provides the basic information needed to set up a working DNSSEC-aware recursive server, also known as a validating resolver. A validating resolver performs validation for each remote response received, following the chain of trust to verify that the answers it receives are legitimate, through the use of public key cryptography and hashing functions.

#### Enabling DNSSEC Validation

So how do we turn on DNSSEC validation? It turns out that you may not need to reconfigure your name server at all, since the most recent versions of BIND 9 - including packages and distributions - have shipped with DNSSEC validation enabled by default. Before making any configuration changes, check whether you already have DNSSEC validation enabled by following the steps described in *So You Think You Are Validating (How To Test A Recursive Server)*.

In earlier versions of BIND, including 9.11-ESV, DNSSEC validation must be explicitly enabled. To do this, you only need to add one line to the `options` section of your configuration file:

```
options {
     ...
     dnssec-validation auto;
     ...
 };
```

Restart *named* or run *rndc reconfig*, and your recursive server is now happily validating each DNS response. If this does not work for you, you may have some other network-related configurations that need to be adjusted. Take a look at *Network Requirements* to make sure your network is ready for DNSSEC.

#### Effects of Enabling DNSSEC Validation

Once DNSSEC validation is enabled, any DNS response that does not pass the validation checks results in a failure to resolve the domain name (often a SERVFAIL status seen by the client). If everything has been configured properly, this is the correct result; it means that an end user has been protected against a malicious attack.

However, if there is a DNSSEC configuration issue (sometimes outside of the administrator's control), a specific name or sometimes entire domains may "disappear" from the DNS, and become unreachable through that resolver. For the end user, the issue may manifest itself as name resolution being slow or failing altogether; some parts of a URL not loading; or the web browser returning an error message indicating that the page cannot be displayed. For example, if root name servers were misconfigured with the wrong information about `.org`, it could cause all validation for `.org` domains to fail. To end users, it would appear that all `.org` web sites were out of service.[2] Should you encounter DNSSEC-related problems, don't be tempted to disable validation; there is almost certainly a solution that leaves validation enabled. A basic troubleshooting guide can be found in *Basic DNSSEC Troubleshooting*.

### 11.4.2 So You Think You Are Validating (How To Test A Recursive Server)

Now that you have reconfigured your recursive server and restarted it, how do you know that your recursive name server is actually verifying each DNS query? There are several ways to check, and we've listed a few of them below.

#### Using Web-Based Tools to Verify

For most people, the simplest way to check if a recursive name server is indeed validating DNS queries is to use one of the many web-based tools available.

Configure your client computer to use the newly reconfigured recursive server for DNS resolution; then use one of these web-based tests to confirm that it is in fact validating DNS responses.

- Internet.nl
- DNSSEC or Not (VeriSign)

#### Using *dig* to Verify

Web-based DNSSEC-verification tools often employ JavaScript. If you don't trust the JavaScript magic that the web-based tools rely on, you can take matters into your own hands and use a command-line DNS tool to check your validating resolver yourself.

While *nslookup* is popular, partly because it comes pre-installed on most systems, it is not DNSSEC-aware. *dig*, on the other hand, fully supports the DNSSEC standard and comes as a part of BIND. If you do not have *dig* already installed on your system, install it by downloading it from ISC's website.

*dig* is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name servers that were queried. Most seasoned DNS administrators use *dig* to troubleshoot DNS problems because of its flexibility, ease of use, and clarity of output.

[2] Of course, something like this could happen for reasons other than DNSSEC: for example, the root publishing the wrong addresses for the `.org` nameservers.

The example below shows how to use *dig* to query the name server 10.53.0.1 for the A record for `ftp.isc.org` when DNSSEC validation is enabled (i.e. the default). The address 10.53.0.1 is only used as an example; replace it with the actual address or host name of your recursive name server.

```
$ dig @10.53.0.1 ftp.isc.org. A +dnssec +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.1 ftp.isc.org a +dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48742
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: 29a9705c2160b08c010000005e67a4a102b9ae079c1b24c8 (good)
;; QUESTION SECTION:
;ftp.isc.org.       IN A

;; ANSWER SECTION:
ftp.isc.org.        300 IN A 149.20.1.49
ftp.isc.org.        300 IN RRSIG A 13 3 300 (
                20200401191851 20200302184340 27566 isc.org.
                e9Vkb6/6aHMQk/t23Im71ioiDUhB06sncsduoW9+Asl4
                L3TZtpLvZ5+zudTJC2coI4D/D9AXte1cD6FV6iS6PQ== )

;; Query time: 452 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 14:30:57 GMT 2020
;; MSG SIZE  rcvd: 187
```

The important detail in this output is the presence of the `ad` flag in the header. This signifies that BIND has retrieved all related DNSSEC information related to the target of the query (`ftp.isc.org`) and that the answer received has passed the validation process described in *How Are Answers Verified?*. We can have confidence in the authenticity and integrity of the answer, that `ftp.isc.org` really points to the IP address 149.20.1.49, and that it was not a spoofed answer from a clever attacker.

Unlike earlier versions of BIND, the current versions of BIND always request DNSSEC records (by setting the `do` bit in the query they make to upstream servers), regardless of DNSSEC settings. However, with validation disabled, the returned signature is not checked. This can be seen by explicitly disabling DNSSEC validation. To do this, add the line `dnssec-validation no;` to the "options" section of the configuration file, i.e.:

```
options {
    ...
    dnssec-validation no;
    ...
};
```

If the server is restarted (to ensure a clean cache) and the same *dig* command executed, the result is very similar:

```
$ dig @10.53.0.1 ftp.isc.org. A +dnssec +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.1 ftp.isc.org a +dnssec +multiline
; (1 server found)
```

(continues on next page)

(continued from previous page)

```
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39050
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: a8dc9d1b9ec45e75010000005e67a8a69399741fdbe126f2 (good)
;; QUESTION SECTION:
;ftp.isc.org.       IN A

;; ANSWER SECTION:
ftp.isc.org.        300 IN A 149.20.1.49
ftp.isc.org.        300 IN RRSIG A 13 3 300 (
                20200401191851 20200302184340 27566 isc.org.
                e9Vkb6/6aHMQk/t23Im71ioiDUhB06sncsduoW9+Asl4
                L3TZtpLvZ5+zudTJC2coI4D/D9AXte1cD6FV6iS6PQ== )

;; Query time: 261 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 14:48:06 GMT 2020
;; MSG SIZE  rcvd: 187
```

However, this time there is no `ad` flag in the header. Although *dig* is still returning the DNSSEC-related resource records, it is not checking them, and thus cannot vouch for the authenticity of the answer. If you do carry out this test, remember to re-enable DNSSEC validation (by removing the `dnssec-validation no;` line from the configuration file) before continuing.

### 11.4.3 Verifying Protection From Bad Domain Names

It is also important to make sure that DNSSEC is protecting your network from domain names that fail to validate; such failures could be caused by attacks on your system, attempting to get it to accept false DNS information. Validation could fail for a number of reasons: maybe the answer doesn't verify because it's a spoofed response; maybe the signature was a replayed network attack that has expired; or maybe the child zone has been compromised along with its keys, and the parent zone's information tells us that things don't add up. There is a domain name specifically set up to fail DNSSEC validation, `www.dnssec-failed.org`.

With DNSSEC validation enabled (the default), an attempt to look up that name fails:

```
$ dig @10.53.0.1 www.dnssec-failed.org. A

; <<>> DiG 9.16.0 <<>> @10.53.0.1 www.dnssec-failed.org. A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 22667
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 69c3083144854587010000005e67bb57f5f90ff2688e455d (good)
;; QUESTION SECTION:
;www.dnssec-failed.org.     IN  A
```

(continues on next page)

(continued from previous page)

```
;; Query time: 2763 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 16:07:51 GMT 2020
;; MSG SIZE  rcvd: 78
```

On the other hand, if DNSSEC validation is disabled (by adding the statement `dnssec-validation no;` to the *options* clause in the configuration file), the lookup succeeds:

```
$ dig @10.53.0.1 www.dnssec-failed.org. A

; <<>> DiG 9.16.0 <<>> @10.53.0.1 www.dnssec-failed.org. A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54704
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 251eee58208917f90100000005e67bb6829f6dabc5ae6b7b9 (good)
;; QUESTION SECTION:
;www.dnssec-failed.org.     IN  A

;; ANSWER SECTION:
www.dnssec-failed.org.  7200    IN  A   68.87.109.242
www.dnssec-failed.org.  7200    IN  A   69.252.193.191

;; Query time: 439 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 16:08:08 GMT 2020
;; MSG SIZE  rcvd: 110
```

Do not be tempted to disable DNSSEC validation just because some names are failing to resolve. Remember, DNSSEC protects your DNS lookup from hacking. The next section describes how to quickly check whether the failure to successfully look up a name is due to a validation failure.

### How Do I Know I Have a Validation Problem?

Since all DNSSEC validation failures result in a general `SERVFAIL` message, how do we know if it was really a validation error? Fortunately, there is a flag in *dig*, ("CD" for "checking disabled") which tells the server to disable DNSSEC validation. If you receive a `SERVFAIL` message, re-run the query a second time and set the *dig +cd* flag. If the query succeeds with *dig +cd*, but ends in `SERVFAIL` without it, you know you are dealing with a validation problem. So using the previous example of `www.dnssec-failed.org` and with DNSSEC validation enabled in the resolver:

```
$ dig @10.53.0.1 www.dnssec-failed.org A +cd

; <<>> DiG 9.16.0 <<>> @10.53.0.1 www.dnssec-failed.org. A +cd
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62313
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

(continues on next page)

(continued from previous page)

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 73ca1be3a74dd2cf010000005e67c8c8e6df64b519cd87fd (good)
;; QUESTION SECTION:
;www.dnssec-failed.org.      IN  A

;; ANSWER SECTION:
www.dnssec-failed.org.  7197    IN  A   68.87.109.242
www.dnssec-failed.org.  7197    IN  A   69.252.193.191

;; Query time: 0 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 17:05:12 GMT 2020
;; MSG SIZE  rcvd: 110
```

For more information on troubleshooting, please see *Basic DNSSEC Troubleshooting*.

### 11.4.4 Validation Easy Start Explained

In *Easy-Start Guide for Recursive Servers*, we used one line of configuration to turn on DNSSEC validation: the act of chasing down signatures and keys, making sure they are authentic. Now we are going to take a closer look at what DNSSEC validation actually does, and some other options.

#### *dnssec-validation*

```
options {
    dnssec-validation auto;
};
```

This "auto" line enables automatic DNSSEC trust anchor configuration using the `managed-keys` feature. In this case, no manual key configuration is needed. There are three possible choices for the `dnssec-validation` option:

- *yes*: DNSSEC validation is enabled, but a trust anchor must be manually configured. No validation actually takes place until at least one trusted key has been manually configured.
- *no*: DNSSEC validation is disabled, and the recursive server behaves in the "old-fashioned" way of performing insecure DNS lookups.
- *auto*: DNSSEC validation is enabled, and a default trust anchor (included as part of BIND 9) for the DNS root zone is used. This is the default; BIND automatically does this if there is no `dnssec-validation` line in the configuration file.

Let's discuss the difference between *yes* and *auto*. If set to *yes*, the trust anchor must be manually defined and maintained using the `trust-anchors` statement (with either the `static-key` or `static-ds` modifier) in the configuration file; if set to *auto* (the default, and as shown in the example), then no further action should be required as BIND includes a copy[3] of the root key. When set to *auto*, BIND automatically keeps the keys (also known as trust anchors, discussed in *Trust Anchors*) up-to-date without intervention from the DNS administrator.

When using *yes*, please note that if `trust-anchors` does not include a valid root key, then validation does not take place for names which are not covered by any of the configured trust anchors.

We recommend using the default *auto* unless there is a good reason to require a manual trust anchor. To learn more about trust anchors, please refer to *Trusted Keys and Managed Keys*.

[3] The root zone was signed in July 2010 and, as at the time of this writing (mid-2020), the key has been changed once, in October 2018. The intention going forward is to roll the key once every five years.

### How Does DNSSEC Change DNS Lookup (Revisited)?

Now you've enabled validation on your recursive name server and verified that it works. What exactly changed? In *How Does DNSSEC Change DNS Lookup?* we looked at a very high-level, simplified version of the 12 steps of the DNSSEC validation process. Let's revisit that process now and see what your validating resolver is doing in more detail. Again, as an example we are looking up the A record for the domain name `www.isc.org` (see *The 12-Step DNSSEC Validation Process (Simplified)*):

1. The validating resolver queries the `isc.org` name servers for the A record of `www.isc.org`. This query has the `DNSSEC OK` (`do`) bit set to 1, notifying the remote authoritative server that DNSSEC answers are desired.
2. Since the zone `isc.org` is signed, and its name servers are DNSSEC-aware, it responds with the answer to the A record query plus the RRSIG for the A record.
3. The validating resolver queries for the DNSKEY for `isc.org`.
4. The `isc.org` name server responds with the DNSKEY and RRSIG records. The DNSKEY is used to verify the answers received in #2.
5. The validating resolver queries the parent (`.org`) for the DS record for `isc.org`.
6. The `.org` name server is also DNSSEC-aware, so it responds with the DS and RRSIG records. The DS record is used to verify the answers received in #4.
7. The validating resolver queries for the DNSKEY for `.org`.
8. The `.org` name server responds with its DNSKEY and RRSIG. The DNSKEY is used to verify the answers received in #6.
9. The validating resolver queries the parent (root) for the DS record for `.org`.
10. The root name server, being DNSSEC-aware, responds with DS and RRSIG records. The DS record is used to verify the answers received in #8.
11. The validating resolver queries for the DNSKEY for root.
12. The root name server responds with its DNSKEY and RRSIG. The DNSKEY is used to verify the answers received in #10.

After step #12, the validating resolver takes the DNSKEY received and compares it to the key or keys it has configured, to decide whether the received key can be trusted. We talk about these locally configured keys, or trust anchors, in *Trust Anchors*.

With DNSSEC, every response includes not just the answer, but a digital signature (RRSIG) as well, so the validating resolver can verify the answer received. That is what we look at in the next section, *How Are Answers Verified?*.

### How Are Answers Verified?

> **Note**
>
> Keep in mind, as you read this section, that although words like "encryption" and "decryption" are used here from time to time, DNSSEC does not provide privacy. Public key cryptography is used to verify data *authenticity* (who sent it) and data *integrity* (it did not change during transit), but any eavesdropper can still see DNS requests and responses in clear text, even when DNSSEC is enabled.

So how exactly are DNSSEC answers verified? Let's first see how verifiable information is generated. On the authoritative server, each DNS record (or message) is run through a hash function, and this hashed value is then encrypted by a private key. This encrypted hash value is the digital signature.

When the validating resolver queries for the resource record, it receives both the plain-text message and the digital signature(s). The validating resolver knows the hash function used (it is listed in the digital signature record itself), so

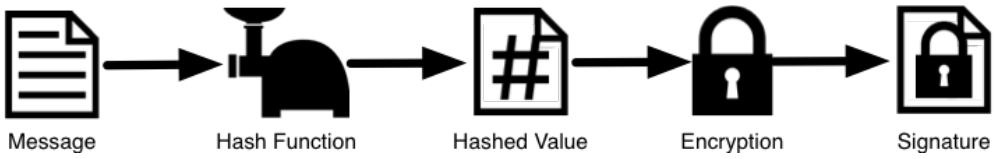



Fig. 1: Signature Generation

it can take the plain-text message and run it through the same hash function to produce a hashed value, which we'll call hash value X. The validating resolver can also obtain the public key (published as DNSKEY records), decrypt the digital signature, and get back the original hashed value produced by the authoritative server, which we'll call hash value Y. If hash values X and Y are identical, and the time is correct (more on what this means below), the answer is verified, meaning this answer came from the authoritative server (authenticity), and the content remained intact during transit (integrity).

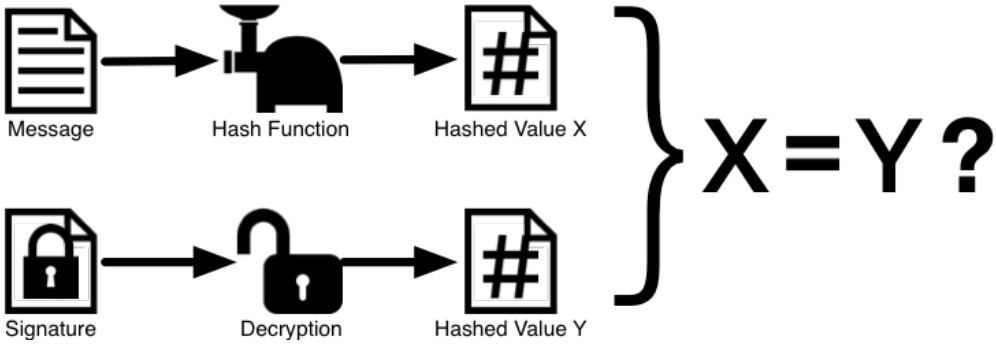



Fig. 2: Signature Verification

Take the A record `ftp.isc.org`, for example. The plain text is:

```
ftp.isc.org.      4 IN A  149.20.1.49
```

The digital signature portion is:

```
ftp.isc.org.       300 IN RRSIG A 13 3 300 (
                20200401191851 20200302184340 27566 isc.org.
                e9Vkb6/6aHMQk/t23Im71ioiDUhB06sncsduoW9+Asl4
                L3TZtpLvZ5+zudTJC2coI4D/D9AXte1cD6FV6iS6PQ== )
```

When a validating resolver queries for the A record `ftp.isc.org`, it receives both the A record and the RRSIG record. It runs the A record through a hash function (in this example, SHA256 as indicated by the number 13, signifying ECDSAP256SHA256) and produces hash value X. The resolver also fetches the appropriate DNSKEY record to decrypt the signature, and the result of the decryption is hash value Y.

But wait, there's more! Just because X equals Y doesn't mean everything is good. We still have to look at the time. Remember we mentioned a little earlier that we need to check if the time is correct? Look at the two timestamps in our example above:

- Signature Expiration: 20200401191851
- Signature Inception: 20200302184340

This tells us that this signature was generated UTC March 2nd, 2020, at 6:43:40 PM (20200302184340), and it is good until UTC April 1st, 2020, 7:18:51 PM (20200401191851). The validating resolver's current system time needs to fall

between these two timestamps. If it does not, the validation fails, because it could be an attacker replaying an old captured answer set from the past, or feeding us a crafted one with incorrect future timestamps.

If the answer passes both the hash value check and the timestamp check, it is validated and the authenticated data (`ad`) bit is set, and the response is sent to the client; if it does not verify, a SERVFAIL is returned to the client.

### 11.4.5 Trust Anchors

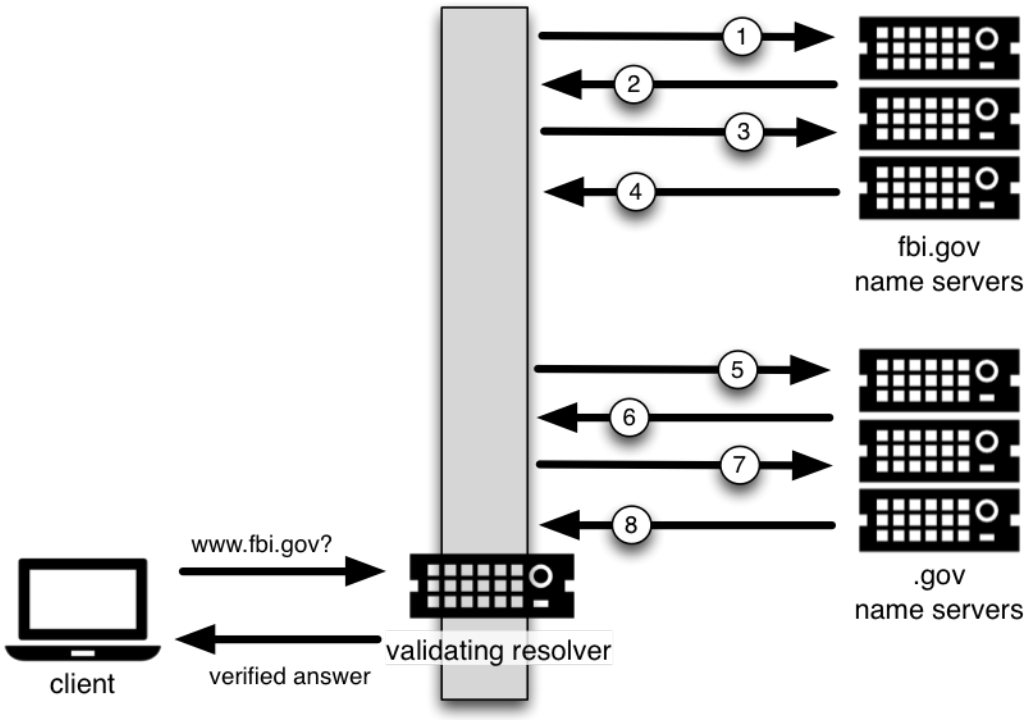
A trust anchor is a key that is placed into a validating resolver, so that the validator can verify the results of a given request with a known or trusted public key (the trust anchor). A validating resolver must have at least one trust anchor installed to perform DNSSEC validation.

### 11.4.6 How Trust Anchors are Used

In the section *How Does DNSSEC Change DNS Lookup (Revisited)?*, we walked through the 12 steps of the DNSSEC lookup process. At the end of the 12 steps, a critical comparison happens: the key received from the remote server and the key we have on file are compared to see if we trust it. The key we have on file is called a trust anchor, sometimes also known as a trust key, trust point, or secure entry point.

The 12-step lookup process describes the DNSSEC lookup in the ideal world, where every single domain name is signed and properly delegated, and where each validating resolver only needs to have one trust anchor - that is, the root's public key. But there is no restriction that the validating resolver must only have one trust anchor. In fact, in the early stages of DNSSEC adoption, it was not unusual for a validating resolver to have more than one trust anchor.

For instance, before the root zone was signed (in July 2010), some validating resolvers that wished to validate domain names in the `.gov` zone needed to obtain and install the key for `.gov`. A sample lookup process for `www.fbi.gov` at that time would have been eight steps rather than 12:




1. The validating resolver queried `fbi.gov` name server for the A record of `www.fbi.gov`.
2. The FBI's name server responded with the answer and its RRSIG.
3. The validating resolver queried the FBI's name server for its DNSKEY.

4. The FBI's name server responded with the DNSKEY and its RRSIG.
5. The validating resolver queried a `.gov` name server for the DS record of `fbi.gov`.
6. The `.gov` name server responded with the DS record and the associated RRSIG for `fbi.gov`.
7. The validating resolver queried the `.gov` name server for its DNSKEY.
8. The `.gov` name server responded with its DNSKEY and the associated RRSIG.

This all looks very similar, except it's shorter than the 12 steps that we saw earlier. Once the validating resolver receives the DNSKEY file in #8, it recognizes that this is the manually configured trusted key (trust anchor), and never goes to the root name servers to ask for the DS record for `.gov`, or ask the root name servers for their DNSKEY.

In fact, whenever the validating resolver receives a DNSKEY, it checks to see if this is a configured trusted key to decide whether it needs to continue chasing down the validation chain.

### Trusted Keys and Managed Keys

Since the resolver is validating, we must have at least one key (trust anchor) configured. How did it get here, and how do we maintain it?

If you followed the recommendation in *Easy-Start Guide for Recursive Servers*, by setting *dnssec-validation* to *auto*, there is nothing left to do. BIND already includes a copy of the root key, and automatically updates it when the root key changes.[Page 341, 3] It looks something like this:

```
trust-anchors {
        # This key (20326) was published in the root zone in 2017.
        . initial-key 257 3 8
↪"AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3
                +/4RgWOq7HrxRixHlFlExOLAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kv
                ArMtNROxVQuCaSnIDdD5LKyWbRd2n9WGe2R8PzgCmr3EgVLrjyBxWezF
                0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eNbuv7pr+e
                oZG+SrDK6nWeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfd
                RUfhHdY6+cn8HFRm+2hM8AnXGXws9555KrUB5qihylGa8subX2Nn6UwN
                R1AkUTV74bU=";
};
```

You can, of course, decide to manage this key manually yourself. First, you need to make sure that *dnssec-validation* is set to *yes* rather than *auto*:

```
options {
    dnssec-validation yes;
};
```

Then, download the root key manually from a trustworthy source, and put it into a *trust-anchors* statement as shown below:

```
trust-anchors {
        # This key (20326) was published in the root zone in 2017.
        . static-key 257 3 8 "AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3
                +/4RgWOq7HrxRixHlFlExOLAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kv
                ArMtNROxVQuCaSnIDdD5LKyWbRd2n9WGe2R8PzgCmr3EgVLrjyBxWezF
                0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/b58Da+sqqls3eNbuv7pr+e
                oZG+SrDK6nWeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/ilBmSVIzuDWfd
                RUfhHdY6+cn8HFRm+2hM8AnXGXws9555KrUB5qihylGa8subX2Nn6UwN
                R1AkUTV74bU=";
};
```

While this *trust-anchors* statement looks similar to the built-in version above, the built-in key has the `initial-key` modifier, whereas in the statement in the configuration file, that is replaced by `static-key`. There is an important difference between the two: a key defined with `static-key` is always trusted until it is deleted from the configuration file. With the `initial-key` modifier, keys are only trusted once: for as long as it takes to load the managed key database and start the key maintenance process. Thereafter, BIND uses the managed keys database (`managed-keys.bind.jnl`) as the source of key information.

**Warning**

Remember, if you choose to manage the keys on your own, whenever the key changes (which, for most zones, happens on a periodic basis), the configuration needs to be updated manually. Failure to do so will result in breaking nearly all DNS queries for the subdomain of the key. So if you are manually managing `.gov`, all domain names in the `.gov` space may become unresolvable; if you are manually managing the root key, you could break all DNS requests made to your recursive name server.

Explicit management of keys was common in the early days of DNSSEC, when neither the root zone nor many top-level domains were signed. Since then, over 90% of the top-level domains have been signed, including all the largest ones. Unless you have a particular need to manage keys yourself, it is best to use the BIND defaults and let the software manage the root key.

### 11.4.7 What's EDNS All About (And Why Should I Care)?

#### EDNS Overview

Traditional DNS responses are typically small in size (less than 512 bytes) and fit nicely into a small UDP packet. The Extension mechanism for DNS (EDNS, or EDNS(0)) offers a mechanism to send DNS data in larger packets over UDP. To support EDNS, both the DNS server and the network need to be properly prepared to support the larger packet sizes and multiple fragments.

This is important for DNSSEC, since the *dig +do* bit that signals DNSSEC-awareness is carried within EDNS, and DNSSEC responses are larger than traditional DNS ones. If DNS servers and the network environment cannot support large UDP packets, it will cause retransmission over TCP, or the larger UDP responses will be discarded. Users will likely experience slow DNS resolution or be unable to resolve certain names at all.

Note that EDNS applies regardless of whether you are validating DNSSEC, because BIND has DNSSEC enabled by default.

Please see *Network Requirements* for more information on what DNSSEC expects from the network environment.

#### EDNS on DNS Servers

For many years, BIND has had EDNS enabled by default, and the UDP packet size is set to a maximum of 4096 bytes. The DNS administrator should not need to perform any reconfiguration. You can use *dig* to verify that your server supports EDNS and see the UDP packet size it allows with this *dig* command:

```
$ dig @10.53.0.1 www.isc.org. A +dnssec +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.1 ftp.isc.org a +dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48742
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

(continues on next page)

(continued from previous page)

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: 29a9705c2160b08c010000005e67a4a102b9ae079c1b24c8 (good)
;; QUESTION SECTION:
;ftp.isc.org.       IN A

;; ANSWER SECTION:
ftp.isc.org.        300 IN A 149.20.1.49
ftp.isc.org.        300 IN RRSIG A 13 3 300 (
                20200401191851 20200302184340 27566 isc.org.
                e9Vkb6/6aHMQk/t23Im71ioiDUhB06sncsduoW9+Asl4
                L3TZtpLvZ5+zudTJC2coI4D/D9AXte1cD6FV6iS6PQ== )

;; Query time: 452 msec
;; SERVER: 10.53.0.1#53(10.53.0.1)
;; WHEN: Tue Mar 10 14:30:57 GMT 2020
;; MSG SIZE  rcvd: 187
```

There is a helpful testing tool available (provided by DNS-OARC) that you can use to verify resolver behavior regarding EDNS support: https://www.dns-oarc.net/oarc/services/replysizetest/ .

Once you've verified that your name servers have EDNS enabled, that should be the end of the story, right? Unfortunately, EDNS is a hop-by-hop extension to DNS. This means the use of EDNS is negotiated between each pair of hosts in a DNS resolution process, which in turn means if one of your upstream name servers (for instance, your ISP's recursive name server that your name server forwards to) does not support EDNS, you may experience DNS lookup failures or be unable to perform DNSSEC validation.

### Support for Large Packets on Network Equipment

If both your recursive name server and your ISP's name servers support EDNS, we are all good here, right? Not so fast. Since these large packets have to traverse the network, the network infrastructure itself must allow them to pass.

When data is physically transmitted over a network, it has to be broken down into chunks. The size of the data chunk is known as the Maximum Transmission Unit (MTU), and it can differ from network to network. IP fragmentation occurs when a large data packet needs to be broken down into chunks smaller than the MTU; these smaller chunks then need to be reassembled back into the large data packet at their destination. IP fragmentation is not necessarily a bad thing, and it most likely occurs on your network today.

Some network equipment, such as a firewall, may make assumptions about DNS traffic. One of these assumptions may be how large each DNS packet is. When a firewall sees a larger DNS packet than it expects, it may either reject the large packet or drop its fragments because the firewall thinks it's an attack. This configuration probably didn't cause problems in the past, since traditional DNS packets are usually pretty small in size. However, with DNSSEC, these configurations need to be updated, since DNSSEC traffic regularly exceeds 1500 bytes (a common MTU value). If the configuration is not updated to support a larger DNS packet size, it often results in the larger packets being rejected, and to the end user it looks like the queries go unanswered. Or in the case of fragmentation, only a part of the answer makes it to the validating resolver, and your validating resolver may need to re-ask the question again and again, creating the appearance for end users that the DNS/network is slow.

While you are updating the configuration on your network equipment, make sure TCP port 53 is also allowed for DNS traffic.

#### Wait... DNS Uses TCP?

Yes. DNS uses TCP port 53 as a fallback mechanism, when it cannot use UDP to transmit data. This has always been the case, even long before the arrival of DNSSEC. Traditional DNS relies on TCP port 53 for operations such as zone transfer. The use of DNSSEC, or DNS with IPv6 records such as AAAA, increases the chance that DNS data will be transmitted via TCP.

Due to the increased packet size, DNSSEC may fall back to TCP more often than traditional (insecure) DNS. If your network blocks or filters TCP port 53 today, you may already experience instability with DNS resolution, before even deploying DNSSEC.

## 11.5 Signing

### 11.5.1 Easy-Start Guide for Signing Authoritative Zones

This section provides the basic information needed to set up a DNSSEC-enabled authoritative name server. A DNSSEC-enabled (or "signed") zone contains additional resource records that are used to verify the authenticity of its zone information.

To convert a traditional (insecure) DNS zone to a secure one, we need to create some additional records (DNSKEY, RRSIG, and NSEC or NSEC3), and upload verifiable information (such as a DS record) to the parent zone to complete the chain of trust. For more information about DNSSEC resource records, please see *What Does DNSSEC Add to DNS?*.

> **Note**
>
> In this chapter, we assume all configuration files, key files, and zone files are stored in `/etc/bind`, and most examples show commands run as the root user. This may not be ideal, but the point is not to distract from what is important here: learning how to sign a zone. There are many best practices for deploying a more secure BIND installation, with techniques such as jailed process and restricted user privileges, but those are not covered in this document. We trust you, a responsible DNS administrator, to take the necessary precautions to secure your system.

For the examples below, we work with the assumption that there is an existing insecure zone `example.com` that we are converting to a secure zone.

#### Enabling Automated DNSSEC Zone Maintenance and Key Generation

To sign a zone, add the following statement to its *zone* clause in the BIND 9 configuration file:

```
options {
    directory "/etc/bind";
    recursion no;
    ...
};

zone "example.com" in {
    ...
    dnssec-policy default;
    ...
};
```

The *dnssec-policy* statement causes the zone to be signed and turns on automatic maintenance for the zone. This includes re-signing the zone as signatures expire and replacing keys on a periodic basis. The value `default` selects the default policy, which contains values suitable for most situations. We cover the creation of a custom policy in *Creating a Custom DNSSEC Policy*, but for the moment we are accepting the default values.

Using *dnssec-policy* requires dynamic DNS or *inline-signing* to be enabled.

When the configuration file is updated, tell *named* to reload the configuration file by running *rndc reconfig*:

```
# rndc reconfig
```

And that's it - BIND signs your zone.

At this point, before you go away and merrily add *dnssec-policy* statements to all your zones, we should mention that, like a number of other BIND configuration options, its scope depends on where it is placed. In the example above, we placed it in a *zone* clause, so it applied only to the zone in question. If we had placed it in a *view* clause, it would have applied to all zones in the view; and if we had placed it in the *options* clause, it would have applied to all zones served by this instance of BIND.

### Verification

The BIND 9 reconfiguration starts the process of signing the zone. First, it generates a key for the zone and includes it in the published zone. The log file shows messages such as these:

```
07-Apr-2020 16:02:55.045 zone example.com/IN (signed): reconfiguring zone keys
07-Apr-2020 16:02:55.045 reloading configuration succeeded
07-Apr-2020 16:02:55.046 keymgr: DNSKEY example.com/ECDSAP256SHA256/10376 (CSK)␣
↪created for policy default
07-Apr-2020 16:02:55.046 Fetching example.com/ECDSAP256SHA256/10376 (CSK) from key␣
↪repository.
07-Apr-2020 16:02:55.046 DNSKEY example.com/ECDSAP256SHA256/10376 (CSK) is now␣
↪published
07-Apr-2020 16:02:55.046 DNSKEY example.com/ECDSAP256SHA256/10376 (CSK) is now active
07-Apr-2020 16:02:55.048 zone example.com/IN (signed): next key event: 07-Apr-2020␣
↪18:07:55.045
```

It then starts signing the zone. How long this process takes depends on the size of the zone, the speed of the server, and how much activity is taking place. We can check what is happening by using *rndc*, entering the command:

```
# rndc signing -list example.com
```

While the signing is in progress, the output is something like:

```
Signing with key 10376/ECDSAP256SHA256
```

and when it is finished:

```
Done signing with key 10376/ECDSAP256SHA256
```

When the second message appears, the zone is signed.

Before moving on to the next step of coordinating with the parent zone, let's make sure everything looks good using *delv*. We want to simulate what a validating resolver will check, by telling *delv* to use a specific trust anchor.

First, we need to make a copy of the key created by BIND. This is in the directory you set with the *directory* statement in your configuration file's *options* clause, and is named something like `Kexample.com.+013.10376.key`:

```
# cp /etc/bind/Kexample.com.+013+10376.key /tmp/example.key
```

The original key file looks like this (with the actual key shortened for ease of display, and comments omitted):

```
# cat /etc/bind/Kexample.com.+013+10376.key

...
example.com. 3600 IN DNSKEY 257 3 13 6saiq99qDB...dqp+o0dw==
```

We want to edit the copy to be in the *trust-anchors* format, so that it looks like this:

```
# cat /tmp/example.key
trust-anchors {
    example.com. static-key 257 3 13 "6saiq99qDB...dqp+o0dw==";
};
```

Now we can run the *delv* command and instruct it to use this trusted-key file to validate the answer it receives from the authoritative name server 192.168.1.13:

```
$ delv @192.168.1.13 -a /tmp/example.key +root=example.com example.com. SOA +multiline
; fully validated
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020040703 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        600 IN RRSIG SOA 13 2 600 (
                20200421150255 20200407140255 10376 example.com.
                jBsz92zwAcGMNV/yu167aKQZvFyC7BiQe1WEnlogdLTF
                oq4yBQumOhO5WX61LjA17l1DuLWcd/ASwlUZWFGCYQ== )
```

### Uploading Information to the Parent Zone

Once everything is complete on our name server, we need to generate some information to be uploaded to the parent zone to complete the chain of trust. The format and the upload methods are actually dictated by your parent zone's administrator, so contact your registrar or parent zone administrator to find out what the actual format should be and how to deliver or upload the information to the parent zone.

What about your zone between the time you signed it and the time your parent zone accepts the upload? To the rest of the world, your zone still appears to be insecure, because if a validating resolver attempts to validate your domain name via your parent zone, your parent zone will indicate that you are not yet signed (as far as it knows). The validating resolver will then give up attempting to validate your domain name, and will fall back to the insecure DNS. Until you complete this final step with your parent zone, your zone remains insecure.

**Note**

Before uploading to your parent zone, verify that your newly signed zone has propagated to all of your name servers (usually via zone transfers). If some of your name servers still have unsigned zone data while the parent tells the world it should be signed, validating resolvers around the world cannot resolve your domain name.

Here are some examples of what you may upload to your parent zone, with the DNSKEY/DS data shortened for display. Note that no matter what format may be required, the end result is the parent zone publishing DS record(s) based on the information you upload. Again, contact your parent zone administrator(s) to find out the correct format for their system.

1. DS record format:

```
example.com. 3600 IN DS 10376 13 2 B92E22CAE0...33B8312EF0
```

2. DNSKEY format:

```
example.com. 3600 IN DNSKEY 257 3 13 6saiq99qDB...dqp+o0dw==
```

The DS record format may be generated from the DNSKEY using the *dnssec-dsfromkey* tool, which is covered in *DS Record Format*. For more details and examples on how to work with your parent zone, please see *Working With the Parent Zone*.

#### So... What Now?

Congratulations! Your zone is signed, your secondary servers have received the new zone data, and the parent zone has accepted your upload and published your DS record. Your zone is now officially DNSSEC-enabled. What happens next? That is basically it - BIND takes care of everything else. As for updating your zone file, you can continue to update it the same way as prior to signing your zone; the normal work flow of editing a zone file and using the *rndc* command to reload the zone still works as usual, and although you are editing the unsigned version of the zone, BIND generates the signed version automatically.

Curious as to what all these commands did to your zone file? Read on to *Your Zone, Before and After DNSSEC* and find out. If you are interested in how to roll this out to your existing primary and secondary name servers, check out *DNSSEC Signing* in the *Recipes* chapter.

### 11.5.2 Your Zone, Before and After DNSSEC

When we assigned the default DNSSEC policy to the zone, we provided the minimal amount of information to convert a traditional DNS zone into a DNSSEC-enabled zone. This is what the zone looked like before we started:

```
$ dig @192.168.1.13 example.com. AXFR +multiline +onesoa

; <<>> DiG 9.16.0 <<>> @192.168.1.13 example.com AXFR +multiline +onesoa
; (1 server found)
;; global options: +cmd
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020040700 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        600 IN NS ns1.example.com.
ftp.example.com.    600 IN A 192.168.1.200
ns1.example.com.    600 IN A 192.168.1.1
web.example.com.    600 IN CNAME www.example.com.
www.example.com.    600 IN A 192.168.1.100
```

Below shows the test zone `example.com` after reloading the server configuration. Clearly, the zone grew in size, and the number of records multiplied:

```
# dig @192.168.1.13 example.com. AXFR +multiline +onesoa

; <<>> DiG 9.16.0 <<>> @192.168.1.13 example.com AXFR +multiline +onesoa
; (1 server found)
;; global options: +cmd
```

(continues on next page)

(continued from previous page)

```
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020040703 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        300 IN RRSIG NSEC 13 2 300 (
                20200413050536 20200407140255 10376 example.com.
                drtV1rJbo5OMi65OJtu7Jmg/thgpdTWrzr6O3Pzt12+B
                oCxMAv3orWWYjfP2n9w5wj0rx2Mt2ev7MOOG8IOUCA== )
example.com.        300 IN NSEC ftp.example.com. NS SOA RRSIG NSEC DNSKEY TYPE65534
example.com.        600 IN RRSIG NS 13 2 600 (
                20200413130638 20200407140255 10376 example.com.
                2ipmzm1Ei6vfE9OLowPMsxLBCbjrCpWPgWJ0ekwZBbux
                MLffZOXn8clt0Ql2U9iCPdyoQryuJCiojHSE2d6nrw== )
example.com.        600 IN RRSIG SOA 13 2 600 (
                20200421150255 20200407140255 10376 example.com.
                jBsz92zwAcGMNV/yu167aKQZvFyC7BiQe1WEnlogdLTF
                oq4yBQumOhO5WX61LjA17l1DuLWcd/ASwlUZWFGCYQ== )
example.com.        0 IN RRSIG TYPE65534 13 2 0 (
                20200413050536 20200407140255 10376 example.com.
                Xjkom24N6qeCJjg9BMUfuWf+euLeZB169DHvLYZPZNlm
                GgM2czUDPio6VpQbUw6JE5DSNjuGjgpgXC5SipC42g== )
example.com.        3600 IN RRSIG DNSKEY 13 2 3600 (
                20200421150255 20200407140255 10376 example.com.
                maK75+28oUyDtci3V7wjTsuhgkLUZW+Q++q46Lea6bKn
                Xj77kXcLNogNdUOr5am/6O6cnPeJKJWsnmTLISm62g== )
example.com.        0 IN TYPE65534 \# 5 ( 0D28880001 )
example.com.        3600 IN DNSKEY 257 3 13 (
                6saiq99qDBb5b4G4cx13cPjFTrIvUs3NW44SvbbHorHb
                kXwOzeGAWyPORN+pwEV/LP9+FHAF/JzAJYdqp+o0dw==
                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 10376
example.com.        600 IN NS ns1.example.com.
ftp.example.com.    600 IN RRSIG A 13 3 600 (
                20200413130638 20200407140255 10376 example.com.
                UYo1njeUA49VhKnPSS3JO4G+/Xd2PD4m3Vaacnd191yz
                BIoouEBAGPcrEM2BNrgR0op1EWSus9tG86SM1ZHGuQ== )
ftp.example.com.    300 IN RRSIG NSEC 13 3 300 (
                20200413130638 20200407140255 10376 example.com.
                rPADrAMAPIPSF3S45OSY8kXBTYMS3nrZg4Awj7qRL+/b
                sOKy6044MbIbjg+YWL69dBjKoTSeEGSCSt73uIxrYA== )
ftp.example.com.    300 IN NSEC ns1.example.com. A RRSIG NSEC
ftp.example.com.    600 IN A 192.168.1.200
ns1.example.com.    600 IN RRSIG A 13 3 600 (
                20200413130638 20200407140255 10376 example.com.
                Yeojg7qrJmxL6uLTnALwKU5byNldZ9Ggj5XjcbpPvujQ
                ocG/ovGBg6pdugXC9UxE39bCDl8dua1frjDcRCCZAA== )
ns1.example.com.    300 IN RRSIG NSEC 13 3 300 (
                20200413130638 20200407140255 10376 example.com.
                vukgQme6k7JwCf/mJOOzHXbE3fKtSro+Kc10T6dHMdsc
                oM1/oXioZvgBZ9cKrQhIAUt7r1KUnrUwM6Je36wWFA== )
```

(continues on next page)

(continued from previous page)

```
ns1.example.com.    300 IN NSEC web.example.com. A RRSIG NSEC
ns1.example.com.    600 IN A 192.168.1.1
web.example.com.    600 IN RRSIG CNAME 13 3 600 (
                20200413130638 20200407140255 10376 example.com.
                JXi4WYypofD5geUowVqlqJyHzvcRnsvU/ONhTBaUCw5Y
                XtifKAXRHWrUL1HIwt37JYPLf5uYu90RfkWLj0GqTQ== )
web.example.com.    300 IN RRSIG NSEC 13 3 300 (
                20200413130638 20200407140255 10376 example.com.
                XF4Hsd58dalL+s6Qu99bG80PQyMf7ZrHEzDiEflRuykP
                DfBRuf34z27vj70LO1lp2ZiX4BB1ahcEK2ae9ASAmA== )
web.example.com.    300 IN NSEC www.example.com. CNAME RRSIG NSEC
web.example.com.    600 IN CNAME www.example.com.
www.example.com.    600 IN RRSIG A 13 3 600 (
                20200413050536 20200407140255 10376 example.com.
                mACKXrDOF5JMWqncSiQ3pYWA6abyGDJ4wgGCumjLXhPy
                0cMzJmKv2s7G6+tW3TsA6BK3UoMfv30oblY2Mnl4/A== )
www.example.com.    300 IN RRSIG NSEC 13 3 300 (
                20200413050536 20200407140255 10376 example.com.
                1YQ22odVt0TeP5gbNJwkvS684ipDmx6sEOsF0eCizhCv
                x8osuOATdlPjIEztt+rveaErZ2nsoLor5k1nQAHsbQ== )
www.example.com.    300 IN NSEC example.com. A RRSIG NSEC
www.example.com.    600 IN A 192.168.1.100
```

But this is a really messy way to tell if the zone is set up properly with DNSSEC. Fortunately, there are tools to help us with that. Read on to *How To Test Authoritative Zones* to learn more.

### 11.5.3 How To Test Authoritative Zones

So we've activated DNSSEC and uploaded some data to our parent zone. How do we know our zone is signed correctly? Here are a few ways to check.

#### Look for Key Data in Your Zone

One way to see if your zone is signed is to check for the presence of DNSKEY record types. In our example, we created a single key, and we expect to see it returned when we query for it.

```
$ dig @192.168.1.13 example.com. DNSKEY +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.6 example.com DNSKEY +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18637
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: efe186423313fb66010000005e8c997e99864f7d69ed7c11 (good)
;; QUESTION SECTION:
;example.com.       IN DNSKEY
```

(continues on next page)

(continued from previous page)

```
;; ANSWER SECTION:
example.com.        3600 IN DNSKEY 257 3 13 (
                6saiq99qDBb5b4G4cx13cPjFTrIvUs3NW44SvbbHorHb
                kXwOzeGAWyPORN+pwEV/LP9+FHAF/JzAJYdqp+o0dw==
                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 10376
```

### Look for Signatures in Your Zone

Another way to see if your zone data is signed is to check for the presence of a signature. With DNSSEC, every record[4] now comes with at least one corresponding signature, known as an RRSIG.

```
$ dig @192.168.1.13 example.com. SOA +dnssec +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.6 example.com SOA +dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45219
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: 75adff4f4ce916b2010000005e8c99c0de47eabb7951b2f5 (good)
;; QUESTION SECTION:
;example.com.       IN SOA

;; ANSWER SECTION:
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020040703 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        600 IN RRSIG SOA 13 2 600 (
                20200421150255 20200407140255 10376 example.com.
                jBsz92zwAcGMNV/yu167aKQZvFyC7BiQe1WEnlogdLTF
                oq4yBQumOhO5WX61LjA17l1DuLWcd/ASwlUZWFGCYQ== )
```

The serial number was automatically incremented from the old, unsigned version. *named* keeps track of the serial number of the signed version of the zone independently of the unsigned version. If the unsigned zone is updated with a new serial number that is higher than the one in the signed copy, then the signed copy is increased to match it; otherwise, the two are kept separate.

### Examine the Zone File

Our original zone file `example.com.db` remains untouched, and *named* has generated three additional files automatically for us (shown below). The signed DNS data is stored in `example.com.db.signed` and in the associated journal file.

[4] Well, almost every record: NS records and glue records for delegations do not have RRSIG records. If there are no delegations, then every record in your zone is signed and comes with its own RRSIG.

```
# cd /etc/bind
# ls
example.com.db  example.com.db.jbk  example.com.db.signed  example.com.db.signed.jnl
```

A quick description of each of the files:

- `.jbk`: a transient file used by *named*
- `.signed`: the signed version of the zone in raw format
- `.signed.jnl`: a journal file for the signed version of the zone

These files are stored in raw (binary) format for faster loading. To reveal the human-readable version, use *named-compilezone* as shown below. In the example below, we run the command on the raw format zone `example.com.db.signed` to produce a text version of the zone `example.com.text`:

```
# named-compilezone -f raw -F text -o example.com.text example.com␣
↪example.com.db.signed
zone example.com/IN: loaded serial 2014112008 (DNSSEC signed)
dump zone to example.com.text...done
OK
```

### Check the Parent

Although this is not strictly related to whether the zone is signed, a critical part of DNSSEC is the trust relationship between the parent and the child. Just because we, the child, have all the correctly signed records in our zone does not mean it can be fully validated by a validating resolver, unless our parent's data agrees with ours. To check if our upload to the parent was successful, ask the parent name server for the DS record of our child zone; we should get back the DS record(s) containing the information we uploaded in *Uploading Information to the Parent Zone*:

```
$ dig example.com. DS

; <<>> DiG 9.16.0 <<>> example.com DS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16954
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: db280d5b52576780010000005e8c9bf5b0d8de103d934e5d (good)
;; QUESTION SECTION:
;example.com.           IN  DS

;; ANSWER SECTION:
example.com.  61179 IN  DS  10376 13 2␣
↪B92E22CAE0B41430EC38D3F7EDF1183C3A94F4D4748569250C15EE33B8312EF0
```

### External Testing Tools

We recommend two tools, below: Verisign DNSSEC Debugger and DNSViz. Others can be found via a simple online search. These excellent online tools are an easy way to verify that your domain name is fully secured.

#### Verisign DNSSEC Debugger

URL: https://dnssec-debugger.verisignlabs.com/

This tool shows a nice summary of checks performed on your domain name. You can expand it to view more details for each of the items checked, to get a detailed report.

#### DNSViz

URL: https://dnsviz.net/

DNSViz provides a visual analysis of the DNSSEC authentication chain for a domain name and its resolution path in the DNS namespace.

### 11.5.4 Signing Easy Start Explained

#### Enable Automatic DNSSEC Maintenance Explained

Signing a zone requires a number of separate steps:

- Generation of the keys to sign the zone.
- Inclusion of the keys into the zone.
- Signing of the records in the file (including the generation of the NSEC or NSEC3 records).

Maintaining a signed zone comprises a set of ongoing tasks:

- Re-signing the zone as signatures approach expiration.
- Generation of new keys as the time approaches for a key roll.
- Inclusion of new keys into the zone when the rollover starts.
- Transition from signing the zone with the old set of keys to signing the zone with the new set of keys.
- Waiting the appropriate interval before removing the old keys from the zone.
- Deleting the old keys.

That is quite complex, and it is all handled in BIND 9 with the single `dnssec-policy default` statement. We will see later on (in the *Creating a Custom DNSSEC Policy* section) how these actions can be tuned, by setting up our own DNSSEC policy with customized parameters. However, in many cases the defaults are adequate.

*dnssec-policy* is the preferred way to run DNSSEC in a zone, but sometimes a more "hands-on" approach to signing and key maintenance is needed. For this reason, we cover manual signing techniques in *Manual Signing*.

### 11.5.5 Working With the Parent Zone

As mentioned in *Uploading Information to the Parent Zone*, the format of the information uploaded to your parent zone is dictated by your parent zone administrator. The two main formats are:

1. DS record format
2. DNSKEY format

Check with your parent zone to see which format they require.

But how can you get each of the formats from your existing data?

When *named* turned on automatic DNSSEC maintenance, essentially the first thing it did was to create the DNSSEC keys and put them in the directory you specified in the configuration file. If you look in that directory, you will see three files with names like `Kexample.com.+013+10376.key`, `Kexample.com.+013+10376.private`, and `Kexample.com.+013+10376.state`. The one we are interested in is the one with the `.key` suffix, which contains the zone's public

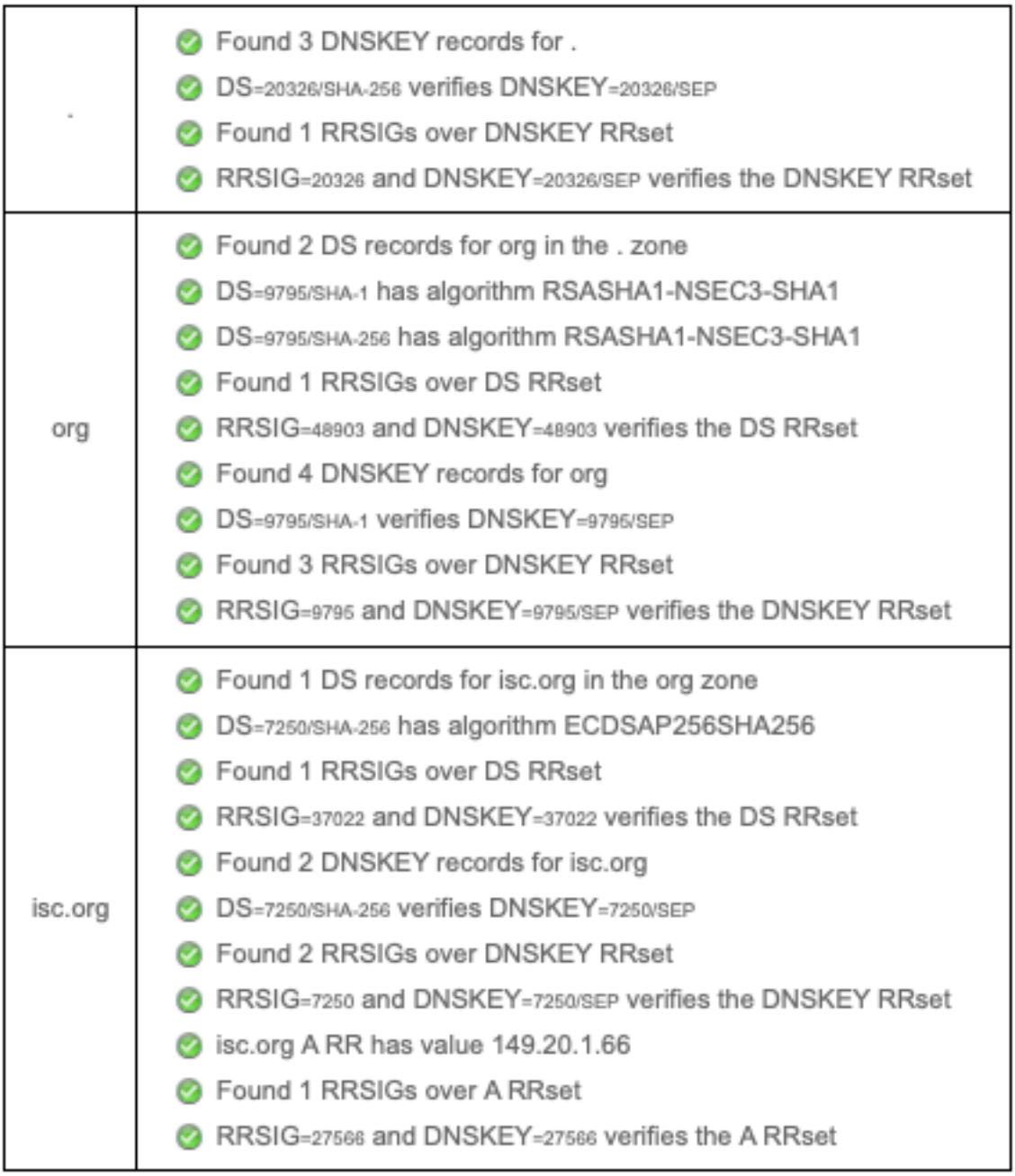



Fig. 3: Verisign DNSSEC Debugger

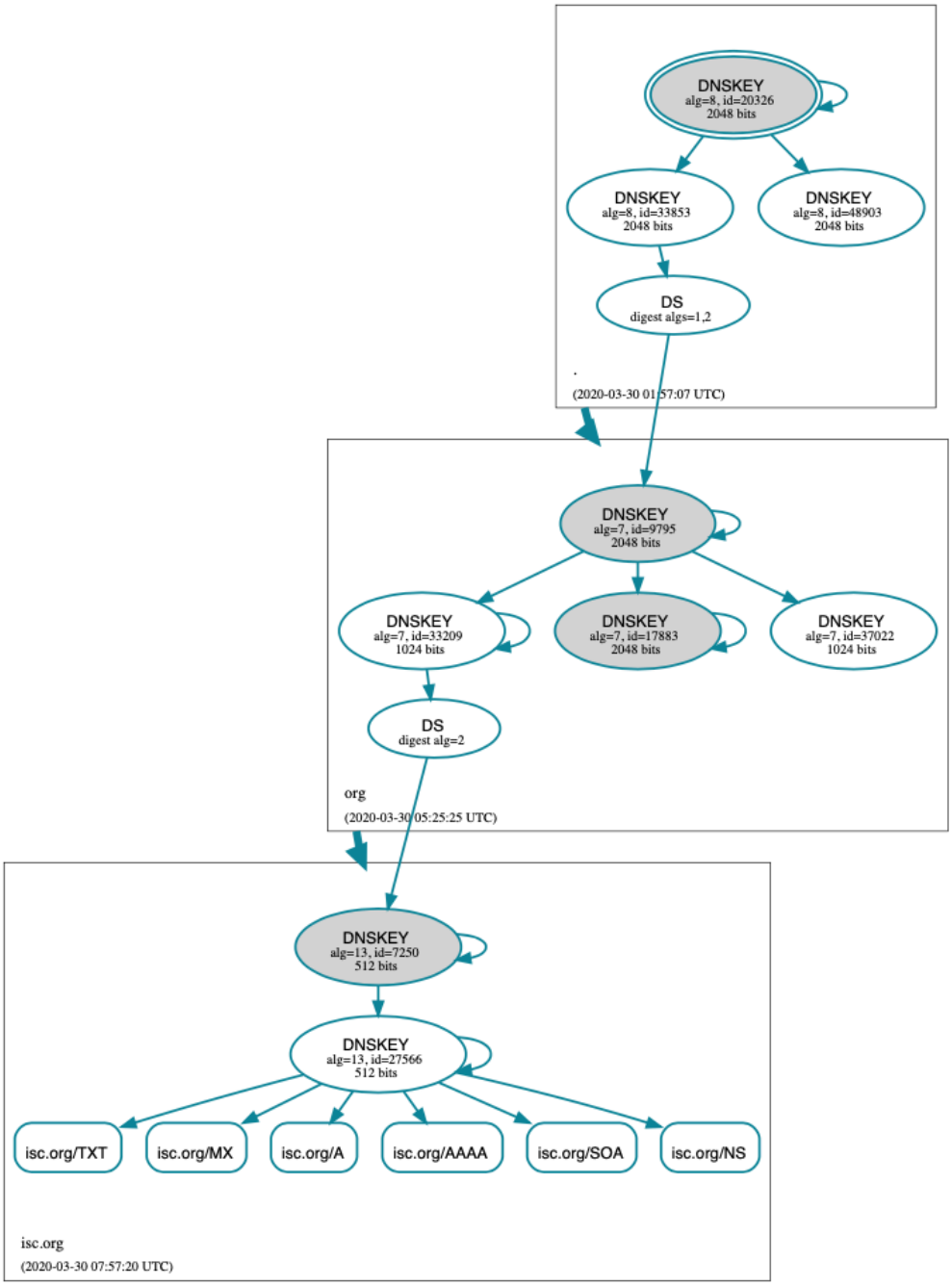

DNSKEY
alg=8, id=20326
2048 bits
DNSKEY
alg=8, id=33853
2048 bits
DNSKEY
alg=8, id=48903
2048 bits
DS
digest algs=1,2
.
(2020-03-30 01:57:07 UTC)
DNSKEY
alg=7, id=9795
2048 bits
DNSKEY
alg=7, id=33209
1024 bits
DNSKEY
alg=7, id=17883
2048 bits
DNSKEY
alg=7, id=37022
1024 bits
DS
digest alg=2
org
(2020-03-30 05:25:25 UTC)
DNSKEY
alg=13, id=7250
512 bits
DNSKEY
alg=13, id=27566
512 bits
isc.org/TXT
isc.org/MX
isc.org/A
isc.org/AAAA
isc.org/SOA
isc.org/NS
isc.org
(2020-03-30 07:57:20 UTC)

Fig. 4: DNSViz

key. (The other files contain the zone's private key and the DNSSEC state associated with the key.) This public key is used to generate the information we need to pass to the parent.

#### DS Record Format

Below is an example of a DS record format generated from the KSK we created earlier (`Kexample.com.+013+10376.key`):

```
# cd /etc/bind
 dnssec-dsfromkey Kexample.com.+013+10376.key
example.com. IN DS 10376 13 2␣
↪B92E22CAE0B41430EC38D3F7EDF1183C3A94F4D4748569250C15EE33B8312EF0
```

Some registrars ask their customers to manually specify the types of algorithm and digest used. In this example, 13 represents the algorithm used, and 2 represents the digest type (SHA-256). The key tag or key ID is 10376.

#### DNSKEY Format

Below is an example of the same key ID (10376) using DNSKEY format (with the actual key shortened for ease of display):

```
example.com. 3600 IN DNSKEY 257 3 13 (6saiq99qDB...dqp+o0dw==) ; key id = 10376
```

The key itself is easy to find (it's difficult to miss that long base64 string) in the file.

```
# cd /etc/bind
# cat Kexample.com.+013+10376.key
; This is a key-signing key, keyid 10376, for example.com.
; Created: 20200407150255 (Tue Apr  7 16:02:55 2020)
; Publish: 20200407150255 (Tue Apr  7 16:02:55 2020)
; Activate: 20200407150255 (Tue Apr  7 16:02:55 2020)
example.com. 3600 IN DNSKEY 257 3 13 6saiq99qDB...dqp+o0dw==
```

### 11.5.6 Creating a Custom DNSSEC Policy

The remainder of this section describes the contents of a custom DNSSEC policy. *Advanced Discussions* describes the concepts involved here and the pros and cons of choosing particular values. If you are not already familiar with DNSSEC, it may be worth reading that chapter first.

Setting up your own DNSSEC policy means that you must include a *dnssec-policy* clause in the zone file. This sets values for the various parameters that affect the signing of zones and the rolling of keys. The following is an example of such a clause:

```
dnssec-policy standard {
    dnskey-ttl 600;
    keys {
        ksk lifetime 365d algorithm ecdsap256sha256;
        zsk lifetime 60d algorithm ecdsap256sha256;
    };
    max-zone-ttl 600;
    parent-ds-ttl 600;
    parent-propagation-delay 2h;
    publish-safety 7d;
    retire-safety 7d;
```

(continues on next page)

(continued from previous page)

```
    signatures-refresh 5d;
    signatures-validity 15d;
    signatures-validity-dnskey 15d;
    zone-propagation-delay 2h;
};
```

The policy has multiple parts:

- The name must be specified. As each zone can use a different policy, *named* needs to be able to distinguish between policies. This is done by giving each policy a name, such as `standard` in the above example.
- The *keys* clause lists all keys that should be in the zone, along with their associated parameters. In this example, we are using the conventional KSK/ZSK split, with the KSK changed every year and the ZSK changed every two months (the `default` DNSSEC policy sets a CSK that is never changed). Keys are created using the ECDSAPS256SHA256 algorithm; each KSK/ZSK pair must have the same algorithm. A CSK combines the functionality of a ZSK and a KSK.
- The parameters ending in `-ttl` are, as expected, the TTLs of the associated records. Remember that during a key rollover, we have to wait for records to expire from caches? The values here tell BIND 9 the maximum amount of time it has to wait for this to happen. Values can be set for the DNSKEY records in your zone, the non-DNSKEY records in your zone, and the DS records in the parent zone.
- Another set of time-related parameters are those ending in `-propagation-delay`. These tell BIND how long it takes for a change in zone contents to become available on all secondary servers. (This may be non-negligible: for example, if a large zone is transferred over a slow link.)
- The policy also sets values for the various signature parameters: how long the signatures on the DNSKEY and non-DNSKEY records are valid, and how often BIND should re-sign the zone.
- The parameters ending in `-safety` are there to give you a bit of leeway in case a key roll doesn't go to plan. When introduced into the zone, the *publish-safety* time is the amount of additional time, over and above that calculated from the other parameters, during which the new key is in the zone but before BIND starts to sign records with it. Similarly, the *retire-safety* is the amount of additional time, over and above that calculated from the other parameters, during which the old key is retained in the zone before being removed.
- Finally, the *purge-keys* option allows you to clean up key files automatically after a period of time. If a key has been removed from the zone, this option will determine how long its key files will be retained on disk.

(You do not have to specify all the items listed above in your policy definition. Any that are not set simply take the default value.)

Usually, the exact timing of a key roll, or how long a signature remains valid, is not critical. For this reason, err on the side of caution when setting values for the parameters. It is better to have an operation like a key roll take a few days longer than absolutely required, than it is to have a quick key roll but have users get validation failures during the process.

Having defined a new policy called "standard", we now need to tell *named* to use it. We do this by adding a `dnssec-policy standard;` statement to the configuration file. Like many other configuration statements, it can be placed in the *options* statement (thus applying to all zones on the server), a *view* statement (applying to all zones in the view), or a *zone* statement (applying only to that zone). In this example, we'll add it to the *zone* statement:

```
zone "example.net" in {
    ...
    dnssec-policy standard;
    ...
};
```

Finally, tell *named* to use the new policy:

```
# rndc reconfig
```

... and that's it. *named* now applies the "standard" policy to your zone.

### 11.5.7 Maintenance Tasks

Zone data is signed and the parent zone has published your DS records: at this point your zone is officially secure. When other validating resolvers look up information in your zone, they are able to follow the 12-step process as described in *How Does DNSSEC Change DNS Lookup (Revisited)?* and verify the authenticity and integrity of the answers.

There is not that much left for you, as the DNS administrator, to do on an ongoing basis. Whenever you update your zone, BIND automatically re-signs your zone with new RRSIG and NSEC/NSEC3 records, and even increments the serial number for you. If you choose to split your keys into a KSK and ZSK, the rolling of the ZSK is completely automatic. Rolling of a KSK or CSK may require some manual intervention, though, so let's examine two more DNSSEC-related resource records, CDS and CDNSKEY.

#### The CDS and CDNSKEY Resource Records

Passing the DS record to the organization running the parent zone has always been recognized as a bottleneck in the key rollover process. To automate the process, the CDS and CDNSKEY resource records were introduced.

The CDS and CDNSKEY records are identical to the DS and DNSKEY records, except in the type code and the name. When such a record appears in the child zone, it is a signal to the parent that it should update the DS it has for that zone. In essence, when the parent notices the presence of the CDS and/or CDNSKEY record(s) in the child zone, it checks these records to verify that they are signed by a valid key for the zone. If the record(s) successfully validate, the parent zone's DS RRset for the child zone is changed to correspond to the CDS (or CDNSKEY) records. (For more information on how the signaling works and the issues surrounding it, please refer to **RFC 7344** and **RFC 8078**.)

#### Working with the Parent Zone (2)

Once the zone is signed, the only required manual tasks are to monitor KSK or CSK key rolls and pass the new DS record to the parent zone. However, if the parent can process CDS or CDNSKEY records, you may not even have to do that.[5]

When the time approaches for the roll of a KSK or CSK, BIND adds a CDS and a CDNSKEY record for the key in question to the apex of the zone. If your parent zone supports polling for CDS/CDNSKEY records, they are uploaded and the DS record published in the parent - at least ideally.

If BIND is configured with *parental-agents*, it will check for the DS presence. Let's look at the following configuration excerpt:

```
remote-servers "net" {
    10.53.0.11; 10.53.0.12;
};

zone "example.net" in {
    ...
    dnssec-policy standard;
    parental-agents { "net"; };
    checkds explicit;
    ...
};
```

[5] For security reasons, a parent zone that supports CDS/CDNSKEY may require the DS record to be manually uploaded when we first sign the zone. Until our zone is signed, the parent cannot be sure that a CDS or CDNSKEY record it finds by querying our zone really comes from our zone; thus, it needs to use some other form of secure transfer to obtain the information.

BIND will check for the presence of the DS record in the parent zone by querying its parental agents (defined in **RFC 7344** to be the entities that the child zone has a relationship with to change its delegation information). In the example above, The zone *example.net* is configured with two parental agents, at the addresses 10.53.0.11 and 10.53.0.12. These addresses are used as an example only. Both addresses will have to respond with a DS RRset that includes the DS record identifying the key that is being rolled. If one or both don't have the DS included yet the rollover is paused, and the check for DS presence is retried after an hour. The same applies for DS withdrawal.

The example also has `checkds` set to *explicit*. This means that only the addresses defined in `parental-agents` are being queried. If set to *yes*, the parental agents are being looked up by querying for the parent NS records.

Alternatively, you can use the `rndc` tool to tell `named` that the DS record has been published or withdrawn. For example:

```
# rndc dnssec -checkds published example.net
```

This command should also be used when `checkds` is set to *no*.

If your parent zone doesn't support CDS/CDNSKEY, you will have to supply the DNSKEY or DS record to the parent zone manually when a new KSK appears in your zone, presumably using the same mechanism you used to upload the records for the first time. Again, you need to use the `rndc` tool to tell `named` that the DS record has been published.

### 11.5.8 Manual Signing

Manual signing of a zone was the first method of signing introduced into BIND and offers, as the name suggests, no automation. The user must handle everything: create the keys, sign the zone file with them, load the signed zone, periodically re-sign the zone, and manage key rolls, including interaction with the parent. A user certainly can do all this, but why not use one of the automated methods?

Although use of the automatic `dnssec-policy` is the preferred way to sign zones in BIND, there are occasions where a manual approach may be needed. `dnssec-policy` does not currently support the use of external hardware, so if your security policy requires it, you need to use manual signing.

BIND 9 ships with several tools that are used in this process, which are explained in more detail below. In all cases, the `-h` option prints a full list of parameters. Note that the DNSSEC tools require the keyset files to be in the working directory or the directory specified by the `-d` option.

To convert a traditional (insecure) DNS zone to a secure one, we need to create various additional records (DNSKEY, RRSIG, NSEC/NSEC3) and, as with fully automatic signing, to upload verifiable information (such as a DS record) to the parent zone to complete the chain of trust.

The first step is to create the keys as described in *Generate Keys*, then using the BIND-provided tools `dnssec-keygen` to create the keys and `dnssec-signzone` to sign the zone. The signed zone is stored in another file and is the one you tell BIND to load. To update the zone (for example, to add a resource record), you update the unsigned zone, re-sign it, and tell `named` to load the updated signed copy. The same goes for refreshing signatures or rolling keys; the user is responsible for providing the signed zone served by `named`. (In the case of rolling keys, you are also responsible for ensuring that the keys are added and removed at the correct times.)

Why would you want to sign your zone this way? You probably wouldn't in the normal course of events, but as there may be circumstances in which it is required, the scripts have been left in the BIND distribution.

**Note**

Again, we assume all configuration files, key files, and zone files are stored in `/etc/bind`, and most examples show commands run as the root user. This may not be ideal, but the point is not to distract from what is important here: learning how to sign a zone. There are many best practices for deploying a more secure BIND installation, with techniques such as jailed process and restricted user privileges, but those are not covered in this document. We trust you, a responsible DNS administrator, to take the necessary precautions to secure your system.

For our examples below, we work with the assumption that there is an existing insecure zone `example.com` that we are converting to a secure version. The secure version uses both a KSK and a ZSK.

### Generate Keys

Everything in DNSSEC centers around keys, so we begin by generating our own keys.

```
# cd /etc/bind/keys
# dnssec-keygen -a ECDSAP256SHA256 example.com
Generating key pair.........................+++++ ......................+++++
Kexample.com.+013+34371
# dnssec-keygen -a ECDSAP256SHA256 -f KSK example.com
Generating key pair.......................+++ ..............................+++
Kexample.com.+013+00472
```

This command generates four key files in `/etc/bind/keys`:

- Kexample.com.+013+34371.key
- Kexample.com.+013+34371.private
- Kexample.com.+013+00472.key
- Kexample.com.+013+00472.private

The two files ending in `.key` are the public keys. These contain the DNSKEY resource records that appear in the zone. The two files ending in `.private` are the private keys, and contain the information that *named* actually uses to sign the zone.

Of the two pairs, one is the zone-signing key (ZSK), and one is the key-signing key (KSK).[6] We can tell which is which by looking at the file contents (the actual keys are shortened here for ease of display):

```
# cat Kexample.com.+013+34371.key
; This is a zone-signing key, keyid 34371, for example.com.
; Created: 20200616104249 (Tue Jun 16 11:42:49 2020)
; Publish: 20200616104249 (Tue Jun 16 11:42:49 2020)
; Activate: 20200616104249 (Tue Jun 16 11:42:49 2020)
example.com. IN DNSKEY 256 3 13 AwEAAfel66...LqkA7cvn8=
# cat Kexample.com.+013+00472.key
; This is a key-signing key, keyid 472, for example.com.
; Created: 20200616104254 (Tue Jun 16 11:42:54 2020)
; Publish: 20200616104254 (Tue Jun 16 11:42:54 2020)
; Activate: 20200616104254 (Tue Jun 16 11:42:54 2020)
example.com. IN DNSKEY 257 3 13 AwEAAbCR6U...l8xPjokVU=
```

The first line of each file tells us what type of key it is. Also, by looking at the actual DNSKEY record, we can tell them apart: 256 is ZSK, and 257 is KSK.

The name of the file also tells us something about the contents. See chapter *Zone keys* for more details.

Make sure that these files are readable by *named* and that the `.private` files are not readable by anyone else.

Alternatively, the *dnssec-keyfromlabel* program is used to get a key pair from a crypto hardware device and build the key files. Its usage is similar to *dnssec-keygen*.

[6] Only one key file - for either a KSK or ZSK - is needed to signal the presence of the zone. *dnssec-keygen* creates files of both types as needed.

### Setting Key Timing Information

Key files contain time information related to rolling keys. This is placed there by *dnssec-keygen* when the file is created, and it can be modified using *dnssec-settime*. By default, only a limited amount of timing information is included in the file, as illustrated in the examples in the previous section.

Note that *dnssec-policy* does set key timing information, but it uses its own state machine to determine what actions to perform.

But when performing manual signing the key parameters and the timing information in the key files, you can implement any DNSSEC policy you want for your zones.

All the dates are the same, and are the date and time that *dnssec-keygen* created the key. We can use *dnssec-settime* to modify the dates. The dates can also be modified using an editor, but that is likely to be more error-prone than using *dnssec-settime*. For example, to publish this key in the zone on 1 July 2020, use it to sign records for a year starting on 15 July 2020, and remove it from the zone at the end of July 2021, we can use the following command:

```
# dnssec-settime -P 20200701 -A 20200715 -I 20210715 -D 20210731␣
↪Kexample.com.+013+34371.key
./Kexample.com.+013+34371.key
./Kexample.com.+013+34371.private
```

which would set the contents of the key file to:

```
; This is a zone-signing key, keyid 34371, for example.com.
; Created: 20200616104249 (Tue Jun 16 11:42:49 2020)
; Publish: 20200701000000 (Wed Jul  1 01:00:00 2020)
; Activate: 20200715000000 (Wed Jul 15 01:00:00 2020)
; Inactive: 20210715000000 (Thu Jul 15 01:00:00 2021)
; Delete: 20210731000000 (Sat Jul 31 01:00:00 2021)
example.com. IN DNSKEY 256 3 13 AwEAAfel66...LqkA7cvn8=
```

(The actual key is truncated here to improve readability.)

Below is a complete list of each of the metadata fields, and how each one affects the signing of your zone:

1. *Created*: This records the date on which the key was created. It is not used in calculations; it is useful simply for documentation purposes.
2. *Publish*: This sets the date on which a key is to be published to the zone. After that date, the key is included in the zone but is not used to sign it. This allows validating resolvers to get a copy of the new key in their cache before there are any resource records signed with it. By default, if not specified at creation time, this is set to the current time, meaning the key is published as soon as *named* picks it up.
3. *Activate*: This sets the date on which the key is to be activated. After that date, resource records are signed with the key. By default, if not specified during creation time, this is set to the current time, meaning the key is used to sign data as soon as *named* picks it up.
4. *Revoke:* This sets the date on which the key is to be revoked. After that date, the key is flagged as revoked, although it is still included in the zone and used to sign it. This is used to notify validating resolvers that this key is about to be removed or retired from the zone. (This state is not used in normal day-to-day operations. See **RFC 5011** to understand the circumstances where it may be used.)
5. *Inactive*: This sets the date on which the key is to become inactive. After that date, the key is still included in the zone, but it is no longer used to sign it. This sets the "expiration" or "retire" date for a key.
6. *Delete*: This sets the date on which the key is to be deleted. After that date, the key is no longer included in the zone, but it continues to exist on the file system or key repository.

This can be summarized as follows:

Table 1: Key Metadata Comparison

| Metadata | Included in Zone File? | Used to Sign Data? | Purpose |
|---|---|---|---|
| Created | No | No | Recording of key creation |
| Publish | Yes | No | Introduction of a key soon to be active |
| Activate | Yes | Yes | Activation date for new key |
| Revoke | Yes | Yes | Notification of a key soon to be retired |
| Inactive | Yes | No | Inactivation or retirement of a key |
| Delete | No | No | Deletion or removal of a key from a zone |

The publication date is the date the key should be introduced into the zone. The activation date can be used to determine when to sign resource records. With "Inactive" you signal when the signer should stop generating new signatures with the given key, and the "Delete" metadata specifies when the key should be removed from the zone.

Finally, we should note that the *dnssec-keygen* command supports the same set of switches so we could have set the dates when we created the key.

### Signing the Zone

Now, edit the zone file to make sure the proper DNSKEY entries are included. The public keys should be inserted into the zone file by including the `.key` files using `$INCLUDE` statements.

Use the command *dnssec-signzone*. Any `keyset` files corresponding to secure sub-zones should be present. The zone signer generates `NSEC`, `NSEC3`, and `RRSIG` records for the zone, as well as `DS` for the child zones if *-g* is specified. If *-g* is not specified, then DS RRsets for the secure child zones need to be added manually.

The following command signs the zone, assuming it is in a file called `zone.child.example`, using manually specified keys:

```
# cd /etc/bind/keys/example.com/
# dnssec-signzone -t -N INCREMENT -o example.com -f /
↪etc/bind/db/example.com.signed.db \
    /etc/bind/db/example.com.db Kexample.com.+013+17694.key␣
↪Kexample.com.+013+06817.key
Verifying the zone using the following algorithms: ECDSAP256SHA256.
Zone fully signed:
Algorithm: ECDSAP256SHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                            ZSKs: 1 active, 0 stand-by, 0 revoked
/etc/bind/db/example.com.signed.db
Signatures generated:                       17
Signatures retained:                         0
Signatures dropped:                          0
Signatures successfully verified:            0
Signatures unsuccessfully verified:          0
Signing time in seconds:                 0.046
Signatures per second:                 364.634
Runtime in seconds:                      0.055
```

The *-o* switch explicitly defines the domain name (`example.com` in this case), while the *-f* switch specifies the output file name. The second line has three parameters: the unsigned zone name (`/etc/bind/db/example.com.db`), the ZSK file name, and the KSK file name. This also generates a plain-text file `/etc/bind/db/example.com.signed.db`, which can be manually verified for correctness.

*dnssec-signzone* also produces keyset and dsset files. These are used to provide the parent zone administrators with the DNSKEY records (or their corresponding DS records) that are the secure entry point to the zone.

By default, all zone keys which have an available private key are used to generate signatures. You can use the *-S* to only include keys that have the "Activate" timing metadata in the past and the "Inactive" timing metadata in the future (or not present).

#### Reloading the Zone

Now it is time to inform BIND that a new signed zonefile is available. We can do this with the *rndc reload example.com* command.

#### Verifying That the Zone Is Signed Correctly

You should now check that the zone is signed. Follow the steps in *Verification*.

#### Uploading the DS Record to the Parent

As described in *Uploading Information to the Parent Zone*, we must now upload the new information to the parent zone. The format of the information and how to generate it is described in *Working With the Parent Zone*, although it is important to remember that you must use the contents of the KSK file that you generated above as part of the process.

The file dsset-example.com (created by *dnssec-signzone* when it signed the example.com zone) contains the DS record for the zone's KSK.

If not yet done so, you will need to pass that to the administrator of the parent zone, to be placed in the zone. When the DS record is published in the parent zone, your zone is fully signed.

#### Checking That Your Zone Can Be Validated

Finally, follow the steps in *How To Test Authoritative Zones* to confirm that a query recognizes the zone as properly signed and vouched for by the parent zone.

#### Re-signing the Zone

Since this is a manual process, you will need to re-sign periodically, as well as every time the zone data changes. You will also need to manually roll the keys by adding and removing DNSKEY records (and interacting with the parent) at the appropriate times.

#### So... What Now?

Once the zone is signed, it must be monitored as described in *Maintenance Tasks*. However, as the time approaches for a key roll, you must create the new key. Of course, it is possible to create keys for the next fifty years all at once and set the key times appropriately. Whether the increased risk in having the private key files for future keys available on disk offsets the overhead of having to remember to create a new key before a rollover depends on your organization's security policy.

### 11.5.9 Offline KSK

For operational reasons, it is possible to keep the KSK offline. Doing so minimizes the risk of the key being compromised through theft or loss.

This effectively means that the private keys of the KSKs and the ZSKs are located in two physically separate places. The KSK is kept completely offline, and the ZSK is used in the primary DNS server to sign the zone data. The DNSKEY, CDS, and CDNSKEY RRsets are signed separately by the KSK.

Because of this, CSKs are incompatible with Offline KSK.

To enable Offline KSK in BIND 9, add the following to the *dnssec-policy* configuration:

```
dnssec-policy "offline-ksk" {
  ...
  offline-ksk yes;
};
```

With this configuration, BIND 9 will no longer generate signatures for the DNSKEY, CDS, and CDNSKEY RRsets, nor will it generate keys for rollovers.

Before enabling Offline KSK, the keys and signed RRsets must be pregenerated. This can be done with the *dnssec-ksr* program, which is used to create Signed Key Response (SKR) files that can be imported into BIND 9.

Creating SKR files is a four-step process. First, the ZSKs must be pregenerated; then, a Key Signing Request (KSR) is created. This file is presented to the KSK operators to be signed. The result is a SKR file that is returned to the ZSK operators, to be imported into the DNS server.

### Pregenerating ZSKs

First we need to pregenerate ZSKs for the future. Let's say we want to generate enough keys for the next two years; this will create several key files, depending on the *dnssec-policy* used. If the ZSK lifetime is six months, this will create about four keys (other timing metadata may cause an extra key to be generated).

This can be done with the *dnssec-ksr* program:

```
# dnssec-ksr -i now -e +2y -k offline-ksk -l named.conf keygen example.net
Kexample.net.+013+63278
Kexample.net.+013+13211
Kexample.net.+013+50958
Kexample.net.+013+12403
```

The timing metadata is set accordingly in the key files. Keys that already exist in the *key-directory* are taken into consideration when pregenerating keys; if the above command is run multiple times quickly in succession, no additional keys are generated.

### Key Signing Request

Now that we have keys that can be published in the zone, we need to get signatures for the DNSKEY RRset to be used in the future. For that, we generate a Key Signing Request (KSR). In this example, we are using the same DNSSEC policy and interval.

```
# dnssec-ksr -i now -e +2y -k offline-ksk -l named.conf request example.net
;; KeySigningRequest 1.0 20240813133035 (Tue Aug 13 15:30:35 2024)
example.net. 3600 IN DNSKEY 256 3 13␣
↪Z8WRuXJr9v7cSUZpJuQKN/1pZuLPEgoWx4eQOhVI8Edz49F7xpbxnGar␣
↪aLelIIIlWuRyjdvUtsnitAfWvyGjqQ==
;; KeySigningRequest 1.0 20250215111826 (Sat Feb 15 12:18:26 2025)
example.net. 3600 IN DNSKEY 256 3 13␣
↪ph7zZ/QgvwHuq2U1aYoMT3MqPUZYEq6y4qNwOb8uzurVISxL0XyhYH+Q␣
↪ngEOV2ECgndMjn8e1ujH/d0H3cPX8A==
example.net. 3600 IN DNSKEY 256 3 13␣
↪Z8WRuXJr9v7cSUZpJuQKN/1pZuLPEgoWx4eQOhVI8Edz49F7xpbxnGar␣
↪aLelIIIlWuRyjdvUtsnitAfWvyGjqQ==
;; KeySigningRequest 1.0 20250225142826 (Tue Feb 25 15:28:26 2025)
example.net. 3600 IN DNSKEY 256 3 13␣
↪ph7zZ/QgvwHuq2U1aYoMT3MqPUZYEq6y4qNwOb8uzurVISxL0XyhYH+Q␣
```

(continues on next page)

(continued from previous page)

```
↪ngEOV2ECgndMjn8e1ujH/d0H3cPX8A==
...
```

The output shows that the ZSK rollovers are pre-planned, which will result in a number of key bundles. Each bundle contains a start time and the ZSKs that need to be published from that time.

The data needs to be stored in a file and can be handed over to the KSK operators, and can be secured by encryption and/or digital signature.

### Signed Key Response

The KSK operators receive a KSR file that contain ZSK sets for a given interval. By signing the KSR, a Signed Key Response (SKR) is created that consists of numerous response bundles; for each bundle, the DNSKEY RRset needs to be constructed by combining the records of the KSK and ZSKs. Then, a signature is generated for the constructed RRset. In addition, the signed CDS and CDNSKEY RRsets are added.

Again the same interval and DNSSEC policy should be used. Below is the command for signing a KSR file "example.net.ksr".

```
# dnssec-ksr -i now -e +2y -k offline-ksk -l named.conf -K ksk -f example.net.ksr␣
↪sign example.net
;; SignedKeyResponse 1.0 20240813134020 (Tue Aug 13 15:40:20 2024)
example.net. 3600 IN DNSKEY 257 3 13␣
↪vV2+6W+cFd3nn8eLrswUnhrPIxdgmslFWwF45MlCPIhjXIp6PpvaHC8k␣
↪Y2RH46UrbWINDEo7k5wqvUncakKhJw==
example.net. 3600 IN DNSKEY 256 3 13␣
↪Z8WRuXJr9v7cSUZpJuQKN/1pZuLPEgoWx4eQOhVI8Edz49F7xpbxnGar␣
↪aLelIIIlWuRyjdvUtsnitAfWvyGjqQ==
example.net. 3600 IN RRSIG DNSKEY 13 2 3600 20240827134020 20240813124020 6221␣
↪example.net. gkiw6M72Gi8XDu8XEAnPVR+AF4K7j1fApt2puLWgChayvaWrMPIbG2jP␣
↪gvd/RJiJSsdGBx4P3GYdNqfFskNKIA==
example.net. 3600 IN CDNSKEY 257 3 13␣
↪vV2+6W+cFd3nn8eLrswUnhrPIxdgmslFWwF45MlCPIhjXIp6PpvaHC8k␣
↪Y2RH46UrbWINDEo7k5wqvUncakKhJw==
example.net. 3600 IN RRSIG CDNSKEY 13 2 3600 20240827134020 20240813124020 6221␣
↪example.net. 1hAwRv2Nbkwfv8KWXdM9eBedgFZapECZJN4iTKj/yb50mjrPjK9JiQ92␣
↪m/xSFUC6gRxMkoPnaULYs+3Qc/XqDA==
example.net. 3600 IN CDS 6221 13 2␣
↪A9EEDE51FA154B90259A1B8788D26C53C20AFE759D3B5FEA0349675A EEC0479D
example.net. 3600 IN RRSIG CDS 13 2 3600 20240827134020 20240813124020 6221␣
↪example.net. TtbCbxTP4WEm5W8ZOdD3DgVlDSz0sdimm5YO28Bi+kP2ZVEM72A0B9QP␣
↪pCiXKrRjCLN2aguqNlRzupWiwb22cA==
;; SignedKeyResponse 1.0 20240822134020 (Thu Aug 22 15:40:20 2024)
example.net. 3600 IN DNSKEY 257 3 13␣
↪vV2+6W+cFd3nn8eLrswUnhrPIxdgmslFWwF45MlCPIhjXIp6PpvaHC8k␣
↪Y2RH46UrbWINDEo7k5wqvUncakKhJw==
example.net. 3600 IN DNSKEY 256 3 13␣
↪Z8WRuXJr9v7cSUZpJuQKN/1pZuLPEgoWx4eQOhVI8Edz49F7xpbxnGar␣
↪aLelIIIlWuRyjdvUtsnitAfWvyGjqQ==
...
```

The output is stored in a file and can be given back to the ZSK operators.

#### Importing the SKR

Now that we have an SKR file, it needs to be imported into the DNS server, via the *rndc skr* command. Let's say the SKR is stored in a file "example.net.skr":

```
# rndc skr -import example.net.skr example.net
```

From now on, when it is time for a new signature for the DNSKEY, CDS, or CDNSKEY RRset, instead of it being generated, it will be looked up in the SKR data.

When the SKR data is nearing the end of its lifetime, simply repeat the four-step process for the next period.

## 11.6 Basic DNSSEC Troubleshooting

In this chapter, we cover some basic troubleshooting techniques, some common DNSSEC symptoms, and their causes and solutions. This is not a comprehensive "how to troubleshoot any DNS or DNSSEC problem" guide, because that could easily be an entire book by itself.

### 11.6.1 Query Path

The first step in troubleshooting DNS or DNSSEC should be to determine the query path. Whenever you are working with a DNS-related issue, it is always a good idea to determine the exact query path to identify the origin of the problem.

End clients, such as laptop computers or mobile phones, are configured to talk to a recursive name server, and the recursive name server may in turn forward requests on to other recursive name servers before arriving at the authoritative name server. The giveaway is the presence of the Authoritative Answer (`aa`) flag in a query response: when present, we know we are talking to the authoritative server; when missing, we are talking to a recursive server. The example below shows an answer to a query for `www.example.com` without the Authoritative Answer flag:

```
$ dig @10.53.0.3 www.example.com A

; <<>> DiG 9.16.0 <<>> @10.53.0.3 www.example.com a
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62714
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c823fe302625db5b010000005e722b504d81bb01c2227259 (good)
;; QUESTION SECTION:
;www.example.com.       IN  A

;; ANSWER SECTION:
www.example.com.    60  IN  A   10.1.0.1

;; Query time: 3 msec
;; SERVER: 10.53.0.3#53(10.53.0.3)
;; WHEN: Wed Mar 18 14:08:16 GMT 2020
;; MSG SIZE  rcvd: 88
```

Not only do we not see the `aa` flag, we see an `ra` flag, which indicates Recursion Available. This indicates that the server we are talking to (10.53.0.3 in this example) is a recursive name server: although we were able to get an answer for `www.example.com`, we know that the answer came from somewhere else.

If we query the authoritative server directly, we get:

```
$ dig @10.53.0.2 www.example.com A

; <<>> DiG 9.16.0 <<>> @10.53.0.2 www.example.com a
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39542
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available
...
```

The `aa` flag tells us that we are now talking to the authoritative name server for `www.example.com`, and that this is not a cached answer it obtained from some other name server; it served this answer to us right from its own database. In fact, the Recursion Available (`ra`) flag is not present, which means this name server is not configured to perform recursion (at least not for this client), so it could not have queried another name server to get cached results.

### 11.6.2 Visible DNSSEC Validation Symptoms

After determining the query path, it is necessary to determine whether the problem is actually related to DNSSEC validation. You can use the *`dig +cd`* flag to disable validation, as described in *How Do I Know I Have a Validation Problem?*.

When there is indeed a DNSSEC validation problem, the visible symptoms, unfortunately, are very limited. With DNSSEC validation enabled, if a DNS response is not fully validated, it results in a generic SERVFAIL message, as shown below when querying against a recursive name server at 192.168.1.7:

```
$ dig @10.53.0.3 www.example.org. A

; <<>> DiG 9.16.0 <<>> @10.53.0.3 www.example.org A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 28947
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d1301968aca086ad010000005e723a7113603c01916d136b (good)
;; QUESTION SECTION:
;www.example.org.       IN  A

;; Query time: 3 msec
;; SERVER: 10.53.0.3#53(10.53.0.3)
;; WHEN: Wed Mar 18 15:12:49 GMT 2020
;; MSG SIZE  rcvd: 72
```

With *`delv`*, a "resolution failed" message is output instead:

```
$ delv @10.53.0.3 www.example.org. A +rtrace
;; fetch: www.example.org/A
;; resolution failed: SERVFAIL
```

BIND 9 logging features may be useful when trying to identify DNSSEC errors.

### 11.6.3 Basic Logging

DNSSEC validation error messages show up in *syslog* as a query error by default. Here is an example of what it may look like:

```
validating www.example.org/A: no valid signature found
RRSIG failed to verify resolving 'www.example.org/A/IN': 10.53.0.2#53
```

Usually, this level of error logging is sufficient. Debug logging, described in *BIND DNSSEC Debug Logging*, gives information on how to get more details about why DNSSEC validation may have failed.

### 11.6.4 BIND DNSSEC Debug Logging

A word of caution: before you enable debug logging, be aware that this may dramatically increase the load on your name servers. Enabling debug logging is thus not recommended for production servers.

With that said, sometimes it may become necessary to temporarily enable BIND debug logging to see more details of how and whether DNSSEC is validating. DNSSEC-related messages are not recorded in *syslog* by default, even if query log is enabled; only DNSSEC errors show up in *syslog*.

The example below shows how to enable debug level 3 (to see full DNSSEC validation messages) in BIND 9 and have it sent to *syslog*:

```
logging {
   channel dnssec_log {
        syslog daemon;
        severity debug 3;
        print-category yes;
    };
    category dnssec { dnssec_log; };
};
```

The example below shows how to log DNSSEC messages to their own file (here, `/var/log/dnssec.log`):

```
logging {
    channel dnssec_log {
        file "/var/log/dnssec.log";
        severity debug 3;
    };
    category dnssec { dnssec_log; };
};
```

After turning on debug logging and restarting BIND, a large number of log messages appear in *syslog*. The example below shows the log messages as a result of successfully looking up and validating the domain name `ftp.isc.org`.

```
validating ./NS: starting
validating ./NS: attempting positive response validation
  validating ./DNSKEY: starting
  validating ./DNSKEY: attempting positive response validation
  validating ./DNSKEY: verify rdataset (keyid=20326): success
  validating ./DNSKEY: marking as secure (DS)
validating ./NS: in validator_callback_dnskey
validating ./NS: keyset with trust secure
validating ./NS: resuming validate
validating ./NS: verify rdataset (keyid=33853): success
```

(continues on next page)

(continued from previous page)

```
validating ./NS: marking as secure, noqname proof not needed
validating ftp.isc.org/A: starting
validating ftp.isc.org/A: attempting positive response validation
validating isc.org/DNSKEY: starting
validating isc.org/DNSKEY: attempting positive response validation
  validating isc.org/DS: starting
  validating isc.org/DS: attempting positive response validation
validating org/DNSKEY: starting
validating org/DNSKEY: attempting positive response validation
  validating org/DS: starting
  validating org/DS: attempting positive response validation
  validating org/DS: keyset with trust secure
  validating org/DS: verify rdataset (keyid=33853): success
  validating org/DS: marking as secure, noqname proof not needed
validating org/DNSKEY: in validator_callback_ds
validating org/DNSKEY: dsset with trust secure
validating org/DNSKEY: verify rdataset (keyid=9795): success
validating org/DNSKEY: marking as secure (DS)
  validating isc.org/DS: in fetch_callback_dnskey
  validating isc.org/DS: keyset with trust secure
  validating isc.org/DS: resuming validate
  validating isc.org/DS: verify rdataset (keyid=33209): success
  validating isc.org/DS: marking as secure, noqname proof not needed
validating isc.org/DNSKEY: in validator_callback_ds
validating isc.org/DNSKEY: dsset with trust secure
validating isc.org/DNSKEY: verify rdataset (keyid=7250): success
validating isc.org/DNSKEY: marking as secure (DS)
validating ftp.isc.org/A: in fetch_callback_dnskey
validating ftp.isc.org/A: keyset with trust secure
validating ftp.isc.org/A: resuming validate
validating ftp.isc.org/A: verify rdataset (keyid=27566): success
validating ftp.isc.org/A: marking as secure, noqname proof not needed
```

Note that these log messages indicate that the chain of trust has been established and `ftp.isc.org` has been successfully validated.

If validation had failed, you would see log messages indicating errors. We cover some of the most validation problems in the next section.

### 11.6.5 Common Problems

#### Security Lameness

Similar to lame delegation in traditional DNS, security lameness refers to the condition when the parent zone holds a set of DS records that point to something that does not exist in the child zone. As a result, the entire child zone may "disappear," having been marked as bogus by validating resolvers.

Below is an example attempting to resolve the A record for a test domain name `www.example.net`. From the user's perspective, as described in *How Do I Know I Have a Validation Problem?*, only a SERVFAIL message is returned. On the validating resolver, we see the following messages in *syslog*:

```
named[126063]: validating example.net/DNSKEY: no valid signature found (DS)
named[126063]: no valid RRSIG resolving 'example.net/DNSKEY/IN': 10.53.0.2#53
named[126063]: broken trust chain resolving 'www.example.net/A/IN': 10.53.0.2#53
```

This gives us a hint that it is a broken trust chain issue. Let's take a look at the DS records that are published for the zone (with the keys shortened for ease of display):

```
$ dig @10.53.0.3 example.net. DS

; <<>> DiG 9.16.0 <<>> @10.53.0.3 example.net DS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59602
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7026d8f7c6e77e2a010000005e735d7c9d038d061b2d24da (good)
;; QUESTION SECTION:
;example.net.           IN  DS

;; ANSWER SECTION:
example.net.        256 IN  DS  14956 8 2 9F3CACD...D3E3A396

;; Query time: 0 msec
;; SERVER: 10.53.0.3#53(10.53.0.3)
;; WHEN: Thu Mar 19 11:54:36 GMT 2020
;; MSG SIZE  rcvd: 116
```

Next, we query for the DNSKEY and RRSIG of `example.net` to see if there's anything wrong. Since we are having trouble validating, we can use the *dig +cd* option to temporarily disable checking and return results, even though they do not pass the validation tests. The *dig +multiline* option causes *dig* to print the type, algorithm type, and key id for DNSKEY records. Again, some long strings are shortened for ease of display:

```
$ dig @10.53.0.3 example.net. DNSKEY +dnssec +cd +multiline

; <<>> DiG 9.16.0 <<>> @10.53.0.3 example.net DNSKEY +cd +multiline +dnssec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42980
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: 4b5e7c88b3680c350100000005e73722057551f9f8be1990e (good)
;; QUESTION SECTION:
;example.net.       IN DNSKEY

;; ANSWER SECTION:
example.net.        287 IN DNSKEY 256 3 8 (
                AwEAAbu3NX...ADU/D7xjFFDu+8WRIn
                ) ; ZSK; alg = RSASHA256 ; key id = 35328
example.net.        287 IN DNSKEY 257 3 8 (
                AwEAAbKtU1...PPP4aQZTybk75ZW+uL
                6OJMAF63NO0s1nAZM2EWAVasbnn/X+J4N2rLuhk=
                ) ; KSK; alg = RSASHA256 ; key id = 27247
```

(continues on next page)

(continued from previous page)

```
example.net.        287 IN RRSIG DNSKEY 8 2 300 (
                20811123173143 20180101000000 27247 example.net.
                Fz1sjClIoF...YEjzpAWuAj9peQ== )
example.net.        287 IN RRSIG DNSKEY 8 2 300 (
                20811123173143 20180101000000 35328 example.net.
                seKtUeJ4/l...YtDc1rcXTVlWIOw= )

;; Query time: 0 msec
;; SERVER: 10.53.0.3#53(10.53.0.3)
;; WHEN: Thu Mar 19 13:22:40 GMT 2020
;; MSG SIZE  rcvd: 962
```

Here is the problem: the parent zone is telling the world that `example.net` is using the key 14956, but the authoritative server indicates that it is using keys 27247 and 35328. There are several potential causes for this mismatch: one possibility is that a malicious attacker has compromised one side and changed the data. A more likely scenario is that the DNS administrator for the child zone did not upload the correct key information to the parent zone.

### Incorrect Time

In DNSSEC, every record comes with at least one RRSIG, and each RRSIG contains two timestamps: one indicating when it becomes valid, and one when it expires. If the validating resolver's current system time does not fall within the two RRSIG timestamps, error messages appear in the BIND debug log.

The example below shows a log message when the RRSIG appears to have expired. This could mean the validating resolver system time is incorrectly set too far in the future, or the zone administrator has not kept up with RRSIG maintenance.

```
validating example.com/DNSKEY: verify failed due to bad signature (keyid=19036):␣
↪RRSIG has expired
```

The log below shows that the RRSIG validity period has not yet begun. This could mean the validation resolver's system time is incorrectly set too far in the past, or the zone administrator has incorrectly generated signatures for this domain name.

```
validating example.com/DNSKEY: verify failed due to bad signature (keyid=4521): RRSIG␣
↪validity period has not begun
```

### Unable to Load Keys

This is a simple yet common issue. If the key files are present but unreadable by *named* for some reason, the *syslog* returns clear error messages, as shown below:

```
named[32447]: zone example.com/IN (signed): reconfiguring zone keys
named[32447]: dns_dnssec_findmatchingkeys: error reading key file Kexample.com.
↪+008+06817.private: permission denied
named[32447]: dns_dnssec_findmatchingkeys: error reading key file Kexample.com.
↪+008+17694.private: permission denied
named[32447]: zone example.com/IN (signed): next key event: 27-Nov-2014 20:04:36.521
```

However, if no keys are found, the error is not as obvious. Below shows the *syslog* messages after executing `rndc reload` with the key files missing from the key directory:

```
named[32516]: received control channel command 'reload'
named[32516]: loading configuration from '/etc/bind/named.conf'
```

(continues on next page)

(continued from previous page)

```
named[32516]: using default UDP/IPv4 port range: [1024, 65535]
named[32516]: using default UDP/IPv6 port range: [1024, 65535]
named[32516]: sizing zone task pool based on 6 zones
named[32516]: the working directory is not writable
named[32516]: reloading configuration succeeded
named[32516]: reloading zones succeeded
named[32516]: all zones loaded
named[32516]: running
named[32516]: zone example.com/IN (signed): reconfiguring zone keys
named[32516]: zone example.com/IN (signed): next key event: 27-Nov-2014 20:07:09.292
```

This happens to look exactly the same as if the keys were present and readable, and appears to indicate that *named* loaded the keys and signed the zone. It even generates the internal (raw) files:

```
# cd /etc/bind/db
# ls
example.com.db  example.com.db.jbk  example.com.db.signed
```

If *named* really loaded the keys and signed the zone, you should see the following files:

```
# cd /etc/bind/db
# ls
example.com.db  example.com.db.jbk  example.com.db.signed  example.com.db.signed.jnl
```

So, unless you see the `*.signed.jnl` file, your zone has not been signed.

### Invalid Trust Anchors

In most cases, you never need to explicitly configure trust anchors. *named* supplies the current root trust anchor and, with the default setting of *dnssec-validation*, updates it on the infrequent occasions when it is changed.

However, in some circumstances you may need to explicitly configure your own trust anchor. As we saw in the *Trust Anchors* section, whenever a DNSKEY is received by the validating resolver, it is compared to the list of keys the resolver explicitly trusts to see if further action is needed. If the two keys match, the validating resolver stops performing further verification and returns the answer(s) as validated.

But what if the key file on the validating resolver is misconfigured or missing? Below we show some examples of log messages when things are not working properly.

First of all, if the key you copied is malformed, BIND does not even start and you will likely find this error message in syslog:

```
named[18235]: /etc/bind/named.conf.options:29: bad base64 encoding
named[18235]: loading configuration: failure
```

If the key is a valid base64 string but the key algorithm is incorrect, or if the wrong key is installed, the first thing you will notice is that virtually all of your DNS lookups result in SERVFAIL, even when you are looking up domain names that have not been DNSSEC-enabled. Below shows an example of querying a recursive server 10.53.0.3:

```
$ dig @10.53.0.3 www.example.com. A

; <<>> DiG 9.16.0 <<>> @10.53.0.3 www.example.org A +dnssec
; (1 server found)
;; global options: +cmd
```

(continues on next page)

(continued from previous page)

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 29586
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: ee078fc321fa1367010000005e73a58bf5f205ca47e04bed (good)
;; QUESTION SECTION:
;www.example.org.       IN  A
```

*delv* shows a similar result:

```
$ delv @192.168.1.7 www.example.com. +rtrace
;; fetch: www.example.com/A
;; resolution failed: SERVFAIL
```

The next symptom you see is in the DNSSEC log messages:

```
managed-keys-zone: DNSKEY set for zone '.' could not be verified with current keys
validating ./DNSKEY: starting
validating ./DNSKEY: attempting positive response validation
validating ./DNSKEY: no DNSKEY matching DS
validating ./DNSKEY: no DNSKEY matching DS
validating ./DNSKEY: no valid signature found (DS)
```

These errors are indications that there are problems with the trust anchor.

### 11.6.6 Negative Trust Anchors

BIND 9.11 introduced Negative Trust Anchors (NTAs) as a means to *temporarily* disable DNSSEC validation for a zone when you know that the zone's DNSSEC is misconfigured.

NTAs are added using the *rndc* command, e.g.:

```
$ rndc nta example.com
 Negative trust anchor added: example.com/_default, expires 19-Mar-2020 19:57:42.000
```

The list of currently configured NTAs can also be examined using *rndc*, e.g.:

```
$ rndc nta -dump
 example.com/_default: expiry 19-Mar-2020 19:57:42.000
```

The default lifetime of an NTA is one hour, although by default, BIND polls the zone every five minutes to see if the zone correctly validates, at which point the NTA automatically expires. Both the default lifetime and the polling interval may be configured via *named.conf*, and the lifetime can be overridden on a per-zone basis using the `-lifetime duration` parameter to `rndc nta`. Both timer values have a permitted maximum value of one week.

### 11.6.7 NSEC3 Troubleshooting

BIND includes a tool called *nsec3hash* that runs through the same steps as a validating resolver, to generate the correct hashed name based on NSEC3PARAM parameters. The command takes the following parameters in order: salt, algorithm, iterations, and domain. For example, if the salt is 1234567890ABCDEF, hash algorithm is 1, and iteration is 10, to get the NSEC3-hashed name for `www.example.com` we would execute a command like this:

```
$ nsec3hash 1234567890ABCEDF 1 10 www.example.com
RN7I9ME6E1I6BDKIP91B9TCE4FHJ7LKF (salt=1234567890ABCEDF, hash=1, iterations=10)
```

Zero-length salt can be specified as -.

While it is unlikely you would construct a rainbow table of your own zone data, this tool may be useful when troubleshooting NSEC3 problems.

## 11.7 Advanced Discussions

### 11.7.1 Signature Validity Periods and Zone Re-Signing Intervals

In *How Are Answers Verified?*, we saw that record signatures have a validity period outside of which they are not valid. This means that at some point, a signature will no longer be valid and a query for the associated record will fail DNSSEC validation. But how long should a signature be valid for?

The maximum value for the validity period should be determined by the impact of a replay attack: if this is low, the period can be long; if high, the period should be shorter. There is no "right" value, but periods of between a few days to a month are common.

Deciding a minimum value is probably an easier task. Should something fail (e.g., a hidden primary distributing to secondary servers that actually answer queries), how long will it take before the failure is noticed, and how long before it is fixed? If you are a large 24x7 operation with operators always on-site, the answer might be less than an hour. In smaller companies, if the failure occurs just after everyone has gone home for a long weekend, the answer might be several days.

Again, there are no "right" values - they depend on your circumstances. The signature validity period you decide to use should be a value between the two bounds. At the time of this writing (mid-2020), the default policy used by BIND sets a value of 14 days.

To keep the zone valid, the signatures must be periodically refreshed since they expire - i.e., the zone must be periodically re-signed. The frequency of the re-signing depends on your network's individual needs. For example, signing puts a load on your server, so if the server is very highly loaded, a lower re-signing frequency is better. Another consideration is the signature lifetime: obviously the intervals between signings must not be longer than the signature validity period. But if you have set a signature lifetime close to the minimum (see above), the signing interval must be much shorter. What would happen if the system failed just before the zone was re-signed?

Again, there is no single "right" answer; it depends on your circumstances. The BIND 9 default policy sets the signature refresh interval to 5 days.

### 11.7.2 Proof of Non-Existence (NSEC and NSEC3)

How do you prove that something does not exist? This zen-like question is an interesting one, and in this section we provide an overview of how DNSSEC solves the problem.

Why is it even important to have authenticated denial of existence in DNS? Couldn't we just send back "hey, what you asked for does not exist," and somehow generate a digital signature to go with it, proving it really is from the correct authoritative source? Aside from the technical challenge of signing something that doesn't exist, this solution has flaws, one of which is it gives an attacker a way to create the appearance of denial of service by replaying this message on the network.

Let's use a little story, told three different ways, to illustrate how proof of nonexistence works. In our story, we run a small company with three employees: Alice, Edward, and Susan. For reasons that are far too complicated to go into, they don't have email accounts; instead, email for them is sent to a single account and a nameless intern passes the message to them. The intern has access to our private DNSSEC key to create signatures for their responses.

If we followed the approach of giving back the same answer no matter what was asked, when people emailed and asked for the message to be passed to "Bob," our intern would simply answer "Sorry, that person doesn't work here" and sign

this message. This answer could be validated because our intern signed the response with our private DNSSEC key. However, since the signature doesn't change, an attacker could record this message. If the attacker were able to intercept our email, when the next person emailed asking for the message to be passed to Susan, the attacker could return the exact same message: "Sorry, that person doesn't work here," with the same signature. Now the attacker has successfully fooled the sender into thinking that Susan doesn't work at our company, and might even be able to convince all senders that no one works at this company.

To solve this problem, two different solutions were created. We will look at the first one, NSEC, next.

### NSEC

The NSEC record is used to prove that something does not exist, by providing the name before it and the name after it. Using our tiny company example, this would be analogous to someone sending an email for Bob and our nameless intern responding with with: "I'm sorry, that person doesn't work here. The name before the location where 'Bob' would be is Alice, and the name after that is Edward." Let's say another email was received for a non-existent person, this time Oliver; our intern would respond "I'm sorry, that person doesn't work here. The name before the location where 'Oliver' would be is Edward, and the name after that is Susan." If another sender asked for Todd, the answer would be: "I'm sorry, that person doesn't work here. The name before the location where 'Todd' would be is Susan, and there are no other names after that."

So we end up with four NSEC records:

```
example.com.         300 IN  NSEC    alice.example.com.  A RRSIG NSEC
alice.example.com.   300 IN  NSEC    edward.example.com. A RRSIG NSEC
edward.example.com.  300 IN  NSEC    susan.example.com.  A RRSIG NSEC
susan.example.com.   300 IN  NSEC    example.com.        A RRSIG NSEC
```

What if the attacker tried to use the same replay method described earlier? If someone sent an email for Edward, none of the four answers would fit. If attacker replied with message #2, "I'm sorry, that person doesn't work here. The name before it is Alice, and the name after it is Edward," it is obviously false, since "Edward" is in the response; and the same goes for #3, Edward and Susan. As for #1 and #4, Edward does not fall in the alphabetical range before Alice or after Susan, so the sender can logically deduce that it was an incorrect answer.

When BIND signs your zone, the zone data is automatically sorted on the fly before generating NSEC records, much like how a phone directory is sorted.

The NSEC record allows for a proof of non-existence for record types. If you ask a signed zone for a name that exists but for a record type that doesn't (for that name), the signed NSEC record returned lists all of the record types that *do* exist for the requested domain name.

NSEC records can also be used to show whether a record was generated as the result of a wildcard expansion. The details of this are not within the scope of this document, but are described well in **RFC 7129**.

Unfortunately, the NSEC solution has a few drawbacks, one of which is trivial "zone walking." In our story, a curious person can keep sending emails, and our nameless, gullible intern keeps divulging information about our employees. Imagine if the sender first asked: "Is Bob there?" and received back the names Alice and Edward. Our sender could then email again: "Is Edwarda there?", and will get back Edward and Susan. (No, "Edwarda" is not a real name. However, it is the first name alphabetically after "Edward" and that is enough to get the intern to reply with a message telling us the next valid name after Edward.) Repeat the process enough times and the person sending the emails eventually learns every name in our company phone directory. For many of you, this may not be a problem, since the very idea of DNS is similar to a public phone book: if you don't want a name to be known publicly, don't put it in DNS! Consider using DNS views (split DNS) and only display your sensitive names to a select audience.

The second potential drawback of NSEC is a bigger zone file and memory consumption; there is no opt-out mechanism for insecure child zones, so each name in the zone will get an additional NSEC record and a RRSIG record to go with it. In practice this is a problem only for parent-zone operators dealing with mostly insecure child zones, such as `com.`. To learn more about opt-out, please see *NSEC3 Opt-Out*.

### NSEC3

NSEC3 adds two additional features that NSEC does not have:

1. It offers no easy zone enumeration.
2. It provides a mechanism for the parent zone to exclude insecure delegations (i.e., delegations to zones that are not signed) from the proof of non-existence.

Recall that in *NSEC* we provided a range of names to prove that something does not exist. But as it turns out, even disclosing these ranges of names becomes a problem: this made it very easy for the curious-minded to look at our entire zone. Not only that, unlike a zone transfer, this "zone walking" is more resource-intensive. So how do we disclose something without actually disclosing it?

The answer is actually quite simple: hashing functions, or one-way hashes. Without going into many details, think of it like a magical meat grinder. A juicy piece of ribeye steak goes in one end, and out comes a predictable shape and size of ground meat (hash) with a somewhat unique pattern. No matter how hard you try, you cannot turn the ground meat back into the ribeye steak: that's what we call a one-way hash.

NSEC3 basically runs the names through a one-way hash before giving them out, so the recipients can verify the non-existence without any knowledge of the other names in the zone.

So let's tell our little story for the third time, this time with NSEC3. In this version, our intern is not given a list of actual names; he is given a list of "hashed" names. So instead of Alice, Edward, and Susan, the list he is given reads like this (hashes shortened for easier reading):

```
FSK5.... (produced from Edward)
JKMA.... (produced from Susan)
NTQ0.... (produced from Alice)
```

Then, an email is received for Bob again. Our intern takes the name Bob through a hash function, and the result is L8J2..., so he replies: "I'm sorry, that person doesn't work here. The name before that is JKMA..., and the name after that is NTQ0...". There, we proved Bob doesn't exist, without giving away any names! To put that into proper NSEC3 resource records, they would look like this (again, hashes shortened for ease of display):

```
FSK5....example.com. 300 IN NSEC3 1 0 0 -  JKMA... A RRSIG
JKMA....example.com. 300 IN NSEC3 1 0 0 -  NTQ0... A RRSIG
NTQ0....example.com. 300 IN NSEC3 1 0 0 -  FSK5... A RRSIG
```

**Note**

Just because we employed one-way hash functions does not mean there is no way for a determined individual to figure out our zone data.

Most names published in the DNS are rarely secret or unpredictable. They are published to be memorable, used and consumed by humans. They are often recorded in many other network logs such as email logs, certificate transparency logs, web page links, intrusion detection systems, malware scanners, email archives, etc. Many times a simple dictionary of commonly used domain-name prefixes (www, mail, imap, login, database, etc.) can be used to quickly reveal a large number of labels within a zone. Additionally, if an adversary really wants to expend significant CPU resources to mount an offline dictionary attack on a zone's NSEC3 chain, they will likely be able to find most of the "guessable" names despite any level of hashing.

Also, it is still possible to gather all of our NSEC3 records and hashed names and perform an offline brute-force attack by trying all possible combinations to figure out what the original name is. In our meat-grinder analogy, this would be like someone buying all available cuts of meat and grinding them up at home using the same model of meat grinder, and comparing the output with the meat you gave him. It is expensive and time-consuming (especially with real meat), but like everything else in cryptography, if someone has enough resources and time, nothing is truly private forever. If

you are concerned about someone performing this type of attack on your zone data, use some of the special techniques described in **RFC 4470**.

### NSEC3PARAM

> **Warning**
>
> Before we dive into the details of NSEC3 parametrization, please note: the defaults should not be changed without a strong justification and a full understanding of the potential impact. See **RFC 9276**.

The above NSEC3 examples used four parameters: 1, 0, 0, and zero-length salt. 1 represents the algorithm, 0 represents the opt-out flag, 0 represents the number of additional iterations, and - is the salt. Let's look at how each one can be configured:

**Algorithm**
**NSEC3 Hashing Algorithm**
: The only currently defined value is 1 for SHA-1, so there is no configuration field for it.

**Opt-out**
: Setting this bit to 1 enables NSEC3 opt-out, which is discussed in *NSEC3 Opt-Out*.

**Iterations**
: Iterations defines the number of _additional_ times to apply the algorithm when generating an NSEC3 hash. More iterations consume more resources for both authoritative servers and validating resolvers. The considerations here are similar to those seen in *Key Sizes*, of security versus resources.

> **Warning**
>
> Do not use values higher than zero. A value of zero provides one round of SHA-1 hashing and protects from non-determined attackers.
>
> A greater number of additional iterations causes interoperability problems and opens servers to CPU-exhausting DoS attacks, while providing only doubtful security benefits.

**Salt**
: A salt value, which can be combined with an FQDN to influence the resulting hash. Salt is discussed in more detail in *NSEC3 Salt*.

### NSEC3 Opt-Out

First things first: For most DNS administrators who do not manage a huge number of insecure delegations, the NSEC3 opt-out featuere is not relevant. See **RFC 9276**.

Opt-out allows for blocks of unsigned delegations to be covered by a single NSEC3 record. In other words, use of the opt-out allows large registries to only sign as many NSEC3 records as there are signed DS or other RRsets in the zone; with opt-out, unsigned delegations do not require additional NSEC3 records. This sacrifices the tamper-resistance proof of non-existence offered by NSEC3 in order to reduce memory and CPU overheads, and decreases the effectiveness of the cache (**RFC 8198**).

Why would that ever be desirable? If a significant number of delegations are not yet securely delegated, meaning they lack DS records and are still insecure or unsigned, generating DNSSEC records for all their NS records might consume lots of memory and is not strictly required by the child zones.

This resource-saving typically makes a difference only for *huge* zones like `com.`. Imagine that you are the operator of busy top-level domains such as `com.`, with millions of insecure delegated domain names. As of mid-2022, around 3% of

all `com.` zones are signed. Basically, without opt-out, with 1,000,000 delegations, only 30,000 of which are secure, you still have to generate NSEC RRsets for the other 970,000 delegations; with NSEC3 opt-out, you will have saved yourself 970,000 sets of records.

In contrast, for a small zone the difference is operationally negligible and the drawbacks outweigh the benefits.

If NSEC3 opt-out is truly essential for a zone, the following configuration can be added to *dnssec-policy*; for example, to create an NSEC3 chain using the SHA-1 hash algorithm, with the opt-out flag, no additional iterations, and no extra salt, use:

```
dnssec-policy "nsec3" {
    ...
    nsec3param iterations 0 optout yes salt-length 0;
};
```

To learn more about how to configure NSEC3 opt-out, please see *NSEC3 Opt-Out*.

#### NSEC3 Salt

> **Warning**
>
> Contrary to popular belief, adding salt provides little value. Each DNS zone is always uniquely salted using the zone name. **Operators should use a zero-length salt value.**

The properties of this extra salt are complicated and beyond scope of this document. For detailed description why the salt in the context of DNSSEC provides little value please see **RFC 9276**.

#### NSEC or NSEC3?

So which is better: NSEC or NSEC3? There is no single right answer here that fits everyone; it comes down to a given network's needs or requirements.

In most cases, NSEC is a good choice for zone administrators. It relieves the authoritative servers and resolver of the additional cryptographic operations that NSEC3 requires, and NSEC is comparatively easier to troubleshoot than NSEC3.

NSEC3 comes with many drawbacks and should be implemented only if zone enumeration prevention is really needed, or when opt-out provides a significant reduction in memory and CPU overheads (in other words, with a huge zone with mostly insecure delegations).

### 11.7.3 DNSSEC Keys

#### Types of Keys

Although DNSSEC documentation talks about three types of keys, they are all the same thing - but they have different roles. The roles are:

**Zone-Signing Key (ZSK)**
: This is the key used to sign the zone. It signs all records in the zone apart from the DNSSEC key-related RRsets: DNSKEY, CDS, and CDNSKEY.

**Key-Signing Key (KSK)**
: This is the key used to sign the DNSSEC key-related RRsets and is the key used to link the parent and child zones. The parent zone stores a digest of the KSK. When a resolver verifies the chain of trust it checks to see that the DS record in the parent (which holds the digest of a key) matches a key in the DNSKEY RRset, and that it is able to use that key to verify the DNSKEY RRset. If it can do that, the resolver knows that it can trust the DNSKEY resource records, and so can use one of them to validate the other records in the zone.

**Combined Signing Key (CSK)**
: A CSK combines the functionality of a ZSK and a KSK. Instead of having one key for signing the zone and one for linking the parent and child zones, a CSK is a single key that serves both roles.

It is important to realize the terms ZSK, KSK, and CSK describe how the keys are used - all these keys are represented by DNSKEY records. The following examples are the DNSKEY records from a zone signed with a KSK and ZSK:

```
$ dig @192.168.1.12 example.com DNSKEY

; <<>> DiG 9.16.0 <<>> @192.168.1.12 example.com dnskey +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54989
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5258d7ed09db0d760100000005ea1cc8c672d8db27a464e37 (good)
;; QUESTION SECTION:
;example.com.       IN DNSKEY

;; ANSWER SECTION:
example.com.        60 IN DNSKEY 256 3 13 (
                tAeXLtIQ3aVDqqS/1UVRt9AE6/nzfoAuaT1Vy4dYl2CK
                pLNcUJxME1Z//pnGXY+HqDU7Gr5HkJY8V0W3r5fzlw==
                ) ; ZSK; alg = ECDSAP256SHA256 ; key id = 63722
example.com.        60 IN DNSKEY 257 3 13 (
                cxkNegsgubBPXSra5ug2P8rWy63B8jTnS4n0IYSsD9eW
                VhiyQDmdgevKUhfG3SE1wbLChjJc2FAbvSZ1qk03Nw==
                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 42933
```

... and a zone signed with just a CSK:

```
$ dig @192.168.1.13 example.com DNSKEY

; <<>> DiG 9.16.0 <<>> @192.168.1.13 example.com dnskey +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22628
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: bf19ee914b5df46e010000005ea1cd02b66c06885d274647 (good)
;; QUESTION SECTION:
;example.com.       IN DNSKEY

;; ANSWER SECTION:
example.com.        60 IN DNSKEY 257 3 13 (
                p0XM6AJ68qid2vtOdyGaeH1jnrdk2GhZeVvGzXfP/PNa
```

(continues on next page)

(continued from previous page)

```
                71wGtzR6jdUrTbXo5Z1W5QeeJF4dls4lh4z7DByF5Q==
                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 1231
```

The only visible difference between the records (apart from the key data itself) is the value of the flags fields; this is 256 for a ZSK and 257 for a KSK or CSK. Even then, the flags field is only a hint to the software using it as to the role of the key: zones can be signed by any key. The fact that a CSK and KSK both have the same flags emphasizes this. A KSK usually only signs the DNSSEC key-related RRsets in a zone, whereas a CSK is used to sign all records in the zone.

The original idea of separating the function of the key into a KSK and ZSK was operational. With a single key, changing it for any reason is "expensive," as it requires interaction with the parent zone (e.g., uploading the key to the parent may require manual interaction with the organization running that zone). By splitting it, interaction with the parent is required only if the KSK is changed; the ZSK can be changed as often as required without involving the parent.

The split also allows the keys to be of different lengths. So the ZSK, which is used to sign the record in the zone, can be of a (relatively) short length, lowering the load on the server. The KSK, which is used only infrequently, can be of a much longer length. The relatively infrequent use also allows the private part of the key to be stored in a way that is more secure but that may require more overhead to access, e.g., on an HSM (see *Hardware Security Modules (HSMs)*).

In the early days of DNSSEC, the idea of splitting the key went more or less unchallenged. However, with the advent of more powerful computers and the introduction of signaling methods between the parent and child zones (see *The CDS and CDNSKEY Resource Records*), the advantages of a ZSK/KSK split are less clear and, for many zones, a single key is all that is required.

As with many questions related to the choice of DNSSEC policy, the decision on which is "best" is not clear and depends on your circumstances.

### Which Algorithm?

There are three algorithm choices for DNSSEC as of this writing (mid-2020):

- RSA
- Elliptic Curve DSA (ECDSA)
- Edwards Curve Digital Security Algorithm (EdDSA)

All are supported in BIND 9, but only RSA and ECDSA (specifically RSASHA256 and ECDSAP256SHA256) are mandatory to implement in DNSSEC. However, RSA is a little long in the tooth, and ECDSA/EdDSA are emerging as the next new cryptographic standards. In fact, the US federal government recommended discontinuing RSA use altogether by September 2015 and migrating to using ECDSA or similar algorithms.

For now, use ECDSAP256SHA256 but keep abreast of developments in this area. For details about rolling over DNSKEYs to a new algorithm, see *Algorithm Rollovers*.

### Key Sizes

If using RSA keys, the choice of key sizes is a classic issue of finding the balance between performance and security. The larger the key size, the longer it takes for an attacker to crack the key; but larger keys also mean more resources are needed both when generating signatures (authoritative servers) and verifying signatures (recursive servers).

Of the two sets of keys, ZSK is used much more frequently. ZSK is used whenever zone data changes or when signatures expire, so performance certainly is of a bigger concern. As for KSK, it is used less frequently, so performance is less of a factor, but its impact is bigger because of its role in signing other keys.

In earlier versions of this guide, the following key lengths were chosen for each set, with the recommendation that they be rotated more frequently for better security:

- *ZSK*: RSA 1024 bits, rollover every year
- *KSK*: RSA 2048 bits, rollover every five years

These should be considered minimum RSA key sizes. At the time of this writing (mid-2020), the root zone and many TLDs are already using 2048 bit ZSKs. If you choose to implement larger key sizes, keep in mind that larger key sizes result in larger DNS responses, which this may mean more load on network resources. Depending on your network configuration, end users may even experience resolution failures due to the increased response sizes, as discussed in *What's EDNS All About (And Why Should I Care)?*.

ECDSA key sizes can be much smaller for the same level of security, e.g., an ECDSA key length of 224 bits provides the same level of security as a 2048-bit RSA key. Currently BIND 9 sets a key size of 256 for all ECDSA keys.

### Key Storage

#### Public Key Storage

The beauty of a public key cryptography system is that the public key portion can and should be distributed to as many people as possible. As the administrator, you may want to keep the public keys on an easily accessible file system for operational ease, but there is no need to securely store them, since both ZSK and KSK public keys are published in the zone data as DNSKEY resource records.

Additionally, a hash of the KSK public key is also uploaded to the parent zone (see *Working With the Parent Zone* for more details), and is published by the parent zone as DS records.

#### Private Key Storage

Ideally, private keys should be stored offline, in secure devices such as a smart card. Operationally, however, this creates certain challenges, since the private key is needed to create RRSIG resource records, and it is a hassle to bring the private key out of storage every time the zone file changes or signatures expire.

A common approach to strike the balance between security and practicality is to have two sets of keys: a ZSK set and a KSK set. A ZSK private key is used to sign zone data, and can be kept online for ease of use, while a KSK private key is used to sign just the DNSKEY (the ZSK); it is used less frequently, and can be stored in a much more secure and restricted fashion.

For example, a KSK private key stored on a USB flash drive that is kept in a fireproof safe, only brought online once a year to sign a new pair of ZSKs, combined with a ZSK private key stored on the network file system and available for routine use, may be a good balance between operational flexibility and security.

For more information on changing keys, please see *Key Rollovers*.

### Hardware Security Modules (HSMs)

A Hardware Security Module (HSM) may come in different shapes and sizes, but as the name indicates, it is a physical device or devices, usually with some or all of the following features:

- Tamper-resistant key storage
- Strong random-number generation
- Hardware for faster cryptographic operations

Most organizations do not incorporate HSMs into their security practices due to cost and the added operational complexity.

BIND supports Public Key Cryptography Standard #11 (PKCS #11) for communication with HSMs and other cryptographic support devices. For more information on how to configure BIND to work with an HSM, please refer to the BIND 9 Administrator Reference Manual.

### 11.7.4 Rollovers

#### Key Rollovers

A key rollover is where one key in a zone is replaced by a new one. There are arguments for and against regularly rolling keys. In essence these are:

Pros:

1. Regularly changing the key hinders attempts at determination of the private part of the key by cryptanalysis of signatures.
2. It gives administrators practice at changing a key; should a key ever need to be changed in an emergency, they would not be doing it for the first time.

Cons:

1. A lot of effort is required to hack a key, and there are probably easier ways of obtaining it, e.g., by breaking into the systems on which it is stored.
2. Rolling the key adds complexity to the system and introduces the possibility of error. We are more likely to have an interruption to our service than if we had not rolled it.

Whether and when to roll the key is up to you. How serious would the damage be if a key were compromised without you knowing about it? How serious would a key roll failure be?

Before going any further, it is worth noting that if you sign your zone with *dnssec-policy*, you don't really need to concern yourself with the details of a key rollover: BIND 9 takes care of it all for you. If you are doing a manual key roll, you do need to familiarize yourself with the various steps involved and the timing details.

Rolling a key is not as simple as replacing the DNSKEY statement in the zone. That is an essential part of it, but timing is everything. For example, suppose that we run the `example.com` zone and that a friend queries for the AAAA record of `www.example.com`. As part of the resolution process (described in *How Does DNSSEC Change DNS Lookup?*), their recursive server looks up the keys for the `example.com` zone and uses them to verify the signature associated with the AAAA record. We'll assume that the records validated successfully, so they can use the address to visit `example.com`'s website.

Let's also assume that immediately after the lookup, we want to roll the ZSK for `example.com`. Our first attempt at this is to remove the old DNSKEY record and signatures, add a new DNSKEY record, and re-sign the zone with it. So one minute our server is serving the old DNSKEY and records signed with the old key, and the next minute it is serving the new key and records signed with it. We've achieved our goal - we are serving a zone signed with the new keys; to check this is really the case, we booted up our laptop and looked up the AAAA record `ftp.example.com`. The lookup succeeded so all must be well. Or is it? Just to be sure, we called our friend and asked them to check. They tried to lookup `ftp.example.com` but got a SERVFAIL response from their recursive server. What's going on?

The answer, in a word, is "caching." When our friend looked up `www.example.com`, their recursive server retrieved and cached not only the AAAA record, but also a lot of other records. It cached the NS records for `com` and `example.com`, as well as the AAAA (and A) records for those name servers (and this action may, in turn, have caused the lookup and caching of other NS and AAAA/A records). Most importantly for this example, it also looked up and cached the DNSKEY records for the root, `com`, and `example.com` zones. When a query was made for `ftp.example.com`, the recursive server believed it already had most of the information we needed. It knew what nameservers served `example.com` and their addresses, so it went directly to one of those to get the AAAA record for `ftp.example.com` and its associated signature. But when it tried to validate the signature, it used the cached copy of the DNSKEY, and that is when our friend had the problem. Their recursive server had a copy of the old DNSKEY in its cache, but the AAAA record for `ftp.example.com` was signed with the new key. So, not surprisingly, the signature could not validate.

How should we roll the keys for `example.com`? A clue to the answer is to note that the problem came about because the DNSKEY records were cached by the recursive server. What would have happened had our friend flushed the DNSKEY records from the recursive server's cache before making the query? That would have worked; those records would have been retrieved from `example.com`'s nameservers at the same time that we retrieved the AAAA record for

`ftp.example.com`. Our friend's server would have obtained the new key along with the AAAA record and associated signature created with the new key, and all would have been well.

As it is obviously impossible for us to notify all recursive server operators to flush our DNSKEY records every time we roll a key, we must use another solution. That solution is to wait for the recursive servers to remove old records from caches when they reach their TTL. How exactly we do this depends on whether we are trying to roll a ZSK, a KSK, or a CSK.

### ZSK Rollover Methods

The ZSK can be rolled in one of the following two ways:

1. *Pre-Publication*: Publish the new ZSK into zone data before it is actually used. Wait at least one TTL interval, so the world's recursive servers know about both keys, then stop using the old key and generate a new RRSIG using the new key. Wait at least another TTL, so the cached old key data is expunged from the world's recursive servers, and then remove the old key.

   The benefit of the pre-publication approach is it does not dramatically increase the zone size; however, the duration of the rollover is longer. If insufficient time has passed after the new ZSK is published, some resolvers may only have the old ZSK cached when the new RRSIG records are published, and validation may fail. This is the method described in *ZSK Rollover*.

2. *Double-Signature*: Publish the new ZSK and new RRSIG, essentially doubling the size of the zone. Wait at least one TTL interval, and then remove the old ZSK and old RRSIG.

   The benefit of the double-signature approach is that it is easier to understand and execute, but it causes a significantly increased zone size during a rollover event.

### KSK Rollover Methods

Rolling the KSK requires interaction with the parent zone, so operationally this may be more complex than rolling ZSKs. There are three methods of rolling the KSK:

1. *Double-KSK*: Add the new KSK to the DNSKEY RRset, which is then signed with both the old and new keys. After waiting for the old RRset to expire from caches, change the DS record in the parent zone. After waiting a further TTL interval for this change to be reflected in caches, remove the old key from the RRset.

   Basically, the new KSK is added first at the child zone and used to sign the DNSKEY; then the DS record is changed, followed by the removal of the old KSK. Double-KSK keeps the interaction with the parent zone to a minimum, but for the duration of the rollover, the size of the DNSKEY RRset is increased.

2. *Double-DS*: Publish the new DS record. After waiting for this change to propagate into caches, change the KSK. After a further TTL interval during which the old DNSKEY RRset expires from caches, remove the old DS record.

   Double-DS is the reverse of Double-KSK: the new DS is published at the parent first, then the KSK at the child is updated, then the old DS at the parent is removed. The benefit is that the size of the DNSKEY RRset is kept to a minimum, but interactions with the parent zone are increased to two events. This is the method described in *KSK Rollover*.

3. *Double-RRset*: Add the new KSK to the DNSKEY RRset, which is then signed with both the old and new key, and add the new DS record to the parent zone. After waiting a suitable interval for the old DS and DNSKEY RRsets to expire from caches, remove the old DNSKEY and old DS record.

   Double-RRset is the fastest way to roll the KSK (i.e., it has the shortest rollover time), but has the drawbacks of both of the other methods: a larger DNSKEY RRset and two interactions with the parent.

#### CSK Rollover Methods

Rolling the CSK is more complex than rolling either the ZSK or KSK, as the timing constraints relating to both the parent zone and the caching of records by downstream recursive servers must be taken into account. There are numerous possible methods that are a combination of ZSK rollover and KSK rollover methods. BIND 9 automatic signing uses a combination of ZSK Pre-Publication and Double-KSK rollover.

#### Emergency Key Rollovers

Keys are generally rolled on a regular schedule - if you choose to roll them at all. But sometimes, you may have to rollover keys out-of-schedule due to a security incident. The aim of an emergency rollover is to re-sign the zone with a new key as soon as possible, because when a key is suspected of being compromised, a malicious attacker (or anyone who has access to the key) could impersonate your server and trick other validating resolvers into believing that they are receiving authentic, validated answers.

During an emergency rollover, follow the same operational procedures described in *Rollovers*, with the added task of reducing the TTL of the current active (potentially compromised) DNSKEY RRset, in an attempt to phase out the compromised key faster before the new key takes effect. The time frame should be significantly reduced from the 30-days-apart example, since you probably do not want to wait up to 60 days for the compromised key to be removed from your zone.

Another method is to carry a spare key with you at all times. If you have a second key pre-published and that one is not compromised at the same time as the first key, you could save yourself some time by immediately activating the spare key if the active key is compromised. With pre-publication, all validating resolvers should already have this spare key cached, thus saving you some time.

With a KSK emergency rollover, you also need to consider factors related to your parent zone, such as how quickly they can remove the old DS records and publish the new ones.

As with many other facets of DNSSEC, there are multiple aspects to take into account when it comes to emergency key rollovers. For more in-depth considerations, please check out **RFC 7583**.

#### Algorithm Rollovers

From time to time, new digital signature algorithms with improved security are introduced, and it may be desirable for administrators to roll over DNSKEYs to a new algorithm, e.g., from RSASHA1 (algorithm 5 or 7) to RSASHA256 (algorithm 8). The algorithm rollover steps must be followed with care to avoid breaking DNSSEC validation.

If you are managing DNSSEC by using the *dnssec-policy* configuration, *named* handles these steps for you. Simply change the algorithm for the relevant keys, and *named* uses the new algorithm when the key is next rolled. It performs a smooth transition to the new algorithm, ensuring that the zone remains valid throughout rollover.

### 11.7.5 Other Topics

#### DNSSEC and Dynamic Updates

Dynamic DNS (DDNS) is actually independent of DNSSEC. DDNS provides a mechanism, separate from editing the zone file or zone database, to edit DNS data. Most DNS clients and servers are able to handle dynamic updates, and DDNS can also be integrated as part of your DHCP environment.

When you have both DNSSEC and dynamic updates in your environment, updating zone data works the same way as with traditional (insecure) DNS: you can use *rndc freeze* before editing the zone file, and *rndc thaw* when you have finished editing, or you can use the command *nsupdate* to add, edit, or remove records like this:

```
$ nsupdate
> server 192.168.1.13
> update add xyz.example.com. 300 IN A 1.1.1.1
```

(continues on next page)

(continued from previous page)

```
> send
> quit
```

The examples provided in this guide make *named* automatically re-sign the zone whenever its content has changed. If you decide to sign your own zone file manually, you need to remember to execute the *dnssec-signzone* command whenever your zone file has been updated.

As far as system resources and performance are concerned, be mindful that with a DNSSEC zone that changes frequently, every time the zone changes your system is executing a series of cryptographic operations to (re)generate signatures and NSEC or NSEC3 records.

#### DNSSEC on Private Networks

Let's clarify what we mean: in this section, "private networks" really refers to a private or internal DNS view. Most DNS products offer the ability to have different versions of DNS answers, depending on the origin of the query. This feature is often called "DNS views" or "split DNS," and is most commonly implemented as an "internal" versus an "external" setup.

For instance, your organization may have a version of `example.com` that is offered to the world, and its names most likely resolve to publicly reachable IP addresses. You may also have an internal version of `example.com` that is only accessible when you are on the company's private networks or via a VPN connection. These private networks typically fall under 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16 for IPv4.

So what if you want to offer DNSSEC for your internal version of `example.com`? This can be done: the golden rule is to use the same key for both the internal and external versions of the zones. This avoids problems that can occur when machines (e.g., laptops) move between accessing the internal and external zones, when it is possible that they may have cached records from the wrong zone.

#### Introduction to DANE

With your DNS infrastructure secured with DNSSEC, information can now be stored in DNS and its integrity and authenticity can be proved. One of the new features that takes advantage of this is the DNS-Based Authentication of Named Entities, or DANE. This improves security in a number of ways, including:

- The ability to store self-signed X.509 certificates and bypass having to pay a third party (such as a Certificate Authority) to sign the certificates (**RFC 6698**).
- Improved security for clients connecting to mail servers (**RFC 7672**).
- A secure way of getting public PGP keys (**RFC 7929**).

### 11.7.6 Disadvantages of DNSSEC

DNSSEC, like many things in this world, is not without its problems. Below are a few challenges and disadvantages that DNSSEC faces.

1. *Increased, well, everything*: With DNSSEC, signed zones are larger, thus taking up more disk space; for DNSSEC-aware servers, the additional cryptographic computation usually results in increased system load; and the network packets are bigger, possibly putting more strains on the network infrastructure.
2. *Different security considerations*: DNSSEC addresses many security concerns, most notably cache poisoning. But at the same time, it may introduce a set of different security considerations, such as amplification attack and zone enumeration through NSEC. These concerns are still being identified and addressed by the Internet community.
3. *More complexity*: If you have read this far, you have probably already concluded this yourself. With additional resource records, keys, signatures, and rotations, DNSSEC adds many more moving pieces on top of the existing

DNS machine. The job of the DNS administrator changes, as DNS becomes the new secure repository of everything from spam avoidance to encryption keys, and the amount of work involved to troubleshoot a DNS-related issue becomes more challenging.

4. *Increased fragility*: The increased complexity means more opportunities for things to go wrong. Before DNSSEC, DNS was essentially "add something to the zone and forget it." With DNSSEC, each new component - re-signing, key rollover, interaction with parent zone, key management - adds more opportunity for error. It is entirely possible that a failure to validate a name may come down to errors on the part of one or more zone operators rather than the result of a deliberate attack on the DNS.

5. *New maintenance tasks*: Even if your new secure DNS infrastructure runs without any hiccups or security breaches, it still requires regular attention, from re-signing to key rollovers. While most of these can be automated, some of the tasks, such as KSK rollover, remain manual for the time being.

6. *Not enough people are using it today*: While it's estimated (as of mid-2020) that roughly 30% of the global Internet DNS traffic is validating,[7] that doesn't mean that many of the DNS zones are actually signed. What this means is, even if your company's zone is signed today, fewer than 30% of the Internet's servers are taking advantage of this extra security. It gets worse: with less than 1.5% of the `com.` domains signed, even if your DNSSEC validation is enabled today, it's not likely to buy you or your users a whole lot more protection until these popular domain names decide to sign their zones.

The last point may have more impact than you realize. Consider this: HTTP and HTTPS make up the majority of traffic on the Internet. While you may have secured your DNS infrastructure through DNSSEC, if your web hosting is outsourced to a third party that does not yet support DNSSEC in its own domain, or if your web page loads contents and components from insecure domains, end users may experience validation problems when trying to access your web page. For example, although you may have signed the zone `company.com`, the web address `www.company.com` may actually be a CNAME to `foo.random-cloud-provider.com`. As long as `random-cloud-provider.com` remains an insecure DNS zone, users cannot fully validate everything when they visit your web page and could be redirected elsewhere by a cache poisoning attack.

## 11.8 Recipes

This chapter provides step-by-step "recipes" for some common DNSSEC configurations.

### 11.8.1 DNSSEC Signing

There are two recipes here: the first shows an example using DNSSEC signing on the primary server, which has been covered in this guide; the second shows how to setup a "bump in the wire" between a hidden primary and the secondary servers to seamlessly sign the zone "on the fly."

#### Primary Server DNSSEC Signing

In this recipe, our servers are illustrated as shown in *DNSSEC Signing Recipe #1*: we have a primary server (192.168.1.1) and three secondary servers (192.168.1.2, 192.168.1.3, and 192.168.1.4) that receive zone transfers. To get the zone signed, we need to reconfigure the primary server. Once reconfigured, a signed version of the zone is generated on the fly; zone transfers take care of synchronizing the signed zone data to all secondary name servers, without configuration or software changes on them.

Using the method described in *Easy-Start Guide for Signing Authoritative Zones*, we just need to add a *`dnssec-policy`* statement to the relevant zone clause. This is what the *`named.conf`* zone statement looks like on the primary server, 192.168.1.1:

```
zone "example.com" IN {
    type primary;
```

(continues on next page)

[7] Based on APNIC statistics at https://stats.labs.apnic.net/dnssec/XA

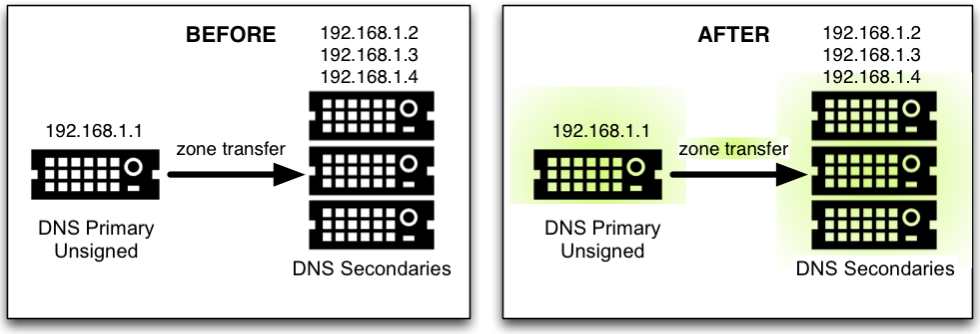



Fig. 5: DNSSEC Signing Recipe #1

(continued from previous page)

```
    file "db/example.com.db";
    key-directory "keys/example.com";
    dnssec-policy default;
    allow-transfer { 192.168.1.2; 192.168.1.3; 192.168.1.4; };
};
```

We have chosen to use the default policy, storing the keys generated for the zone in the directory `keys/example.com`. To use a custom policy, define the policy in the configuration file and select it in the zone statement (as described in *Creating a Custom DNSSEC Policy*).

On the secondary servers, *named.conf* does not need to be updated, and it looks like this:

```
zone "example.com" IN {
    type secondary;
    file "db/example.com.db";
    primaries { 192.168.1.1; };
};
```

In fact, the secondary servers do not even need to be running BIND; they can run any DNS product that supports DNSSEC.

### "Bump in the Wire" Signing

In this recipe, we take advantage of the power of automated signing by placing an additional name server (192.168.1.5) between the hidden primary (192.168.1.1) and the DNS secondaries (192.168.1.2, 192.168.1.3, and 192.168.1.4). The additional name server, 192.168.1.5, acts as a "bump in the wire," taking an unsigned zone from the hidden primary, and sending out signed data on the other end to the secondary name servers. The steps described in this recipe may be used as part of a DNSSEC deployment strategy, since it requires only minimal changes made to the existing hidden DNS primary and DNS secondaries.

It is important to remember that 192.168.1.1 in this case is a hidden primary not exposed to the world, and it must not be listed in the NS RRset. Otherwise the world will get conflicting answers: unsigned answers from the hidden primary and signed answers from the other name servers.

The only configuration change needed on the hidden primary, 192.168.1.1, is to make sure it allows our middle box to perform a zone transfer:

```
zone "example.com" IN {
    ...
    allow-transfer { 192.168.1.5; };
```

(continues on next page)

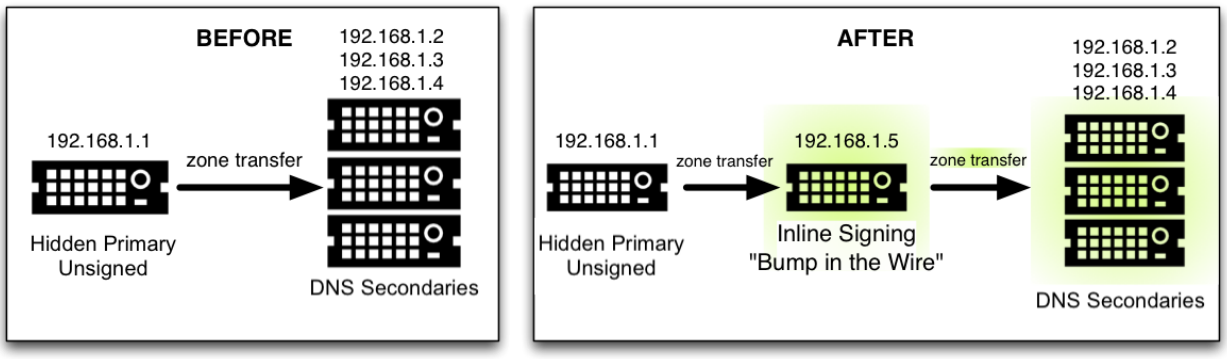



Fig. 6: DNSSEC Signing Recipe #2

(continued from previous page)

```
    ...
};
```

On the middle box, 192.168.1.5, all the tasks described in *Easy-Start Guide for Signing Authoritative Zones* still need to be performed, such as generating key pairs and uploading information to the parent zone. This server is configured as secondary to the hidden primary 192.168.1.1 to receive the unsigned data; then, using keys accessible to this middle box, to sign data on the fly; and finally, to send out the signed data via zone transfer to the other three DNS secondaries. Its `named.conf` zone statement looks like this:

```
zone example.com {
    type secondary;
    primaries { 192.168.1.1; };
    file "db/example.com.db";
    key-directory "keys/example.com";
    dnssec-policy default;
    allow-transfer { 192.168.1.2; 192.168.1.3; 192.168.1.4; };
};
```

(As before, the default policy has been selected here. See *Creating a Custom DNSSEC Policy* for instructions on how to define and use a custom policy.)

Finally, on the three secondary servers, the configuration should be updated to receive a zone transfer from 192.168.1.5 (the middle box) instead of from 192.168.1.1 (the hidden primary). If using BIND, the `named.conf` file looks like this:

```
zone "example.com" IN {
    type secondary;
    file "db/example.com.db";
    primaries { 192.168.1.5; };    # this was 192.168.1.1 before!
};
```

### 11.8.2 Rollovers

If you are signing your zone using a `dnssec-policy` statement, this section is not really relevant to you. In the policy statement, you set how long you want your keys to be valid for, the time taken for information to propagate through your zone, the time it takes for your parent zone to register a new DS record, etc., and that's more or less it. `named` implements everything for you automatically, apart from uploading the new DS records to your parent zone - which is covered in *Uploading Information to the Parent Zone*. (Some screenshots from a session where a KSK is uploaded to the parent zone are presented here for convenience.) However, these recipes may be useful in describing what happens through the

rollover process and what you should be monitoring.

### ZSK Rollover

This recipe covers how to perform a ZSK rollover using what is known as the Pre-Publication method. For other ZSK rolling methods, please see *ZSK Rollover Methods* in *Advanced Discussions*.

Below is a sample timeline for a ZSK rollover to occur on January 1, 2021:

1. December 1, 2020 (one month before rollover)
   - Generate new ZSK
   - Add DNSKEY for new ZSK to zone
2. January 1, 2021 (day of rollover)
   - New ZSK used to replace RRSIGs for the bulk of the zone
3. February 1, 2021 (one month after rollover)
   - Remove old ZSK DNSKEY RRset from zone
   - DNSKEY signatures made with KSK are changed

The current active ZSK has the ID 17694 in the example below. For more information on key management and rollovers, please see *Rollovers*.

### One Month Before ZSK Rollover

On December 1, 2020, a month before the example rollover, you (as administrator) should change the parameters on the current key (17694). Set it to become inactive on January 1, 2021 and be deleted from the zone on February 1, 2021; also, generate a successor key (51623):

```
# cd /etc/bind/keys/example.com/
# dnssec-settime -I 20210101 -D 20210201 Kexample.com.+008+17694
./Kexample.com.+008+17694.key/GoDaddy

./Kexample.com.+008+17694.private
# dnssec-keygen -S Kexample.com.+008+17694
Generating key pair..++++++ ..........++++++
Kexample.com.+008+51623
```

The first command gets us into the key directory `/etc/bind/keys/example.com/`, where keys for `example.com` are stored.

The second, *dnssec-settime*, sets an inactive (*-I*) date of January 1, 2021, and a deletion (*-D*) date of February 1, 2021, for the current ZSK (`Kexample.com.+008+17694`).

The third command, *dnssec-keygen*, creates a successor key, using the exact same parameters (algorithms, key sizes, etc.) as the current ZSK. The new ZSK created in our example is `Kexample.com.+008+51623`.

Make sure the successor keys are readable by *named*.

*named*'s logging messages indicate when the next key checking event is scheduled to occur, the frequency of which can be controlled by *dnssec-loadkeys-interval*. The log message looks like this:

```
zone example.com/IN (signed): next key event: 01-Dec-2020 00:13:05.385
```

And you can check the publish date of the key by looking at the key file:

```
# cd /etc/bind/keys/example.com
# cat Kexample.com.+008+51623.key
; This is a zone-signing key, keyid 51623, for example.com.
; Created: 20201130160024 (Mon Dec  1 00:00:24 2020)
; Publish: 20201202000000 (Fri Dec  2 08:00:00 2020)
; Activate: 20210101000000 (Sun Jan  1 08:00:00 2021)
...
```

Since the publish date is set to the morning of December 2, and our example scenario takes place on December 1, the next morning you will notice that your zone has gained a new DNSKEY record, but the new ZSK is not yet being used to generate signatures. Below is the abbreviated output - with shortened DNSKEY and RRSIG - when querying the authoritative name server, 192.168.1.13:

```
$ dig @192.168.1.13 example.com. DNSKEY +dnssec +multiline

...
;; ANSWER SECTION:
example.com.        600 IN DNSKEY 257 3 8 (
                AwEAAcWDps...lM3NRn/G/R
                ) ; KSK; alg = RSASHA256; key id = 6817
example.com.        600 IN DNSKEY 256 3 8 (
                AwEAAbi6Vo...qBW5+iAqNz
                ) ; ZSK; alg = RSASHA256; key id = 51623
example.com.        600 IN DNSKEY 256 3 8 (
                AwEAAcjGaU...0rzuu55If5
                ) ; ZSK; alg = RSASHA256; key id = 17694
example.com.        600 IN RRSIG DNSKEY 8 2 600 (
                20210101000000 20201201230000 6817 example.com.
                LAiaJM26T7...FU9syh/TQ= )
example.com.        600 IN RRSIG DNSKEY 8 2 600 (
                20210101000000 20201201230000 17694 example.com.
                HK4EBbbOpj...n5V6nvAkI= )
...
```

For good measure, let's take a look at the SOA record and its signature for this zone. Notice the RRSIG is signed by the current ZSK, 17694. This will come in handy later when you want to verify whether the new ZSK is in effect:

```
$ dig @192.168.1.13 example.com. SOA +dnssec +multiline

...
;; ANSWER SECTION:
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020120102 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        600 IN RRSIG SOA 8 2 600 (
                20201230160109 20201130150109 17694 example.com.
                YUTC8rFULaWbW+nAHzbfGwNqzARHevpryzRIJMvZBYPo
                NAeejNk9saNAoCYKWxGJ0YBc2k+r5fYq1Mg4ll2JkBF5
                buAsAYLw8vEOIxVpXwlArY+oSp9T1w2wfTZ0vhVIxaYX
```

(continues on next page)

(continued from previous page)

```
                6dkcz4I3wbDx2xmG0yngtA6A8lAchERx2EGy0RM= )
```

These are all the manual tasks you need to perform for a ZSK rollover. If you have followed the configuration examples in this guide of using *inline-signing* and *dnssec-policy*, everything else is automated for you by BIND.

### Day of ZSK Rollover

On the actual day of the rollover, although there is technically nothing for you to do, you should still keep an eye on the zone to make sure new signatures are being generated by the new ZSK (51623 in this example). The easiest way is to query the authoritative name server 192.168.1.13 for the SOA record as you did a month ago:

```
$ dig @192.168.1.13 example.com. SOA +dnssec +multiline

...
;; ANSWER SECTION:
example.com.        600 IN SOA ns1.example.com. admin.example.com. (
                2020112011 ; serial
                1800       ; refresh (30 minutes)
                900        ; retry (15 minutes)
                2419200    ; expire (4 weeks)
                300        ; minimum (5 minutes)
                )
example.com.        600 IN RRSIG SOA 8 2 600 (
                20210131000000 20201231230000 51623 example.com.
                J4RMNpJPOmMidElyBugJp0RLqXoNqfvo/2AT6yAAvx9X
                zZRL1cuhkRcyCSLZ9Z+zZ2y4u2lvQGrNiondaKdQCor7
                uTqH5WCPoqalOCBjqU7c7vlAM27O9RD11nzPNpVQ7xPs
                y5nkGqf83OXTK26IfnjU1jqiUKSzg6QR7+XpLk0= )
...
```

As you can see, the signature generated by the old ZSK (17694) has disappeared, replaced by a new signature generated from the new ZSK (51623).

**Note**

Not all signatures will disappear magically on the same day; it depends on when each one was generated. In the worst-case scenario, a new signature could have been signed by the old ZSK (17694) moments before it was deactivated, meaning that the signature could live for almost 30 more days, until just before February 1.

This is why it is important to keep the old ZSK in the zone and not delete it right away.

### One Month After ZSK Rollover

Again, technically there is nothing you need to do on this day, but it doesn't hurt to verify that the old ZSK (17694) is now completely gone from your zone. *named* will not touch `Kexample.com.+008+17694.private` and `Kexample.com.+008+17694.key` on your file system. Running the same *dig* command for DNSKEY should suffice:

```
$ dig @192.168.1.13 example.com. DNSKEY +multiline +dnssec

...
;; ANSWER SECTION:
example.com.        600 IN DNSKEY 257 3 8 (
```

(continues on next page)

(continued from previous page)

```
                AwEAAcWDps...lM3NRn/G/R
                ) ; KSK; alg = RSASHA256; key id = 6817
example.com.        600 IN DNSKEY 256 3 8 (
                AwEAAdeCGr...1DnEfX+Xzn
                ) ; ZSK; alg = RSASHA256; key id = 51623
example.com.        600 IN RRSIG DNSKEY 8 2 600 (
                20170203000000 20170102230000 6817 example.com.
                KHY8P0zE21...Y3szrmjAM= )
example.com.        600 IN RRSIG DNSKEY 8 2 600 (
                20170203000000 20170102230000 51623 example.com.
                G2g3crN17h...Oe4gw6gH8= )
...
```

Congratulations, the ZSK rollover is complete! As for the actual key files (the files ending in `.key` and `.private`), they may be deleted at this point, but they do not have to be.

### KSK Rollover

This recipe describes how to perform KSK rollover using the Double-DS method. For other KSK rolling methods, please see *KSK Rollover Methods* in *Advanced Discussions*. The registrar used in this recipe is GoDaddy. Also for this recipe, we are keeping the number of DS records down to just one per active set using just SHA-1, for the sake of better clarity, although in practice most zone operators choose to upload two DS records as shown in *Working With the Parent Zone*. For more information on key management and rollovers, please see *Rollovers*.

Below is a sample timeline for a KSK rollover to occur on January 1, 2021:

1. December 1, 2020 (one month before rollover)
    - Change timer on the current KSK
    - Generate new KSK and DS records
    - Add DNSKEY for the new KSK to zone
    - Upload new DS records to parent zone
2. January 1, 2021 (day of rollover)
    - Use the new KSK to sign all DNSKEY RRsets, which generates new RRSIGs
    - Add new RRSIGs to the zone
    - Remove RRSIG for the old ZSK from zone
    - Start using the new KSK to sign DNSKEY
3. February 1, 2021 (one month after rollover)
    - Remove the old KSK DNSKEY from zone
    - Remove old DS records from parent zone

The current active KSK has the ID 24828, and this is the DS record that has already been published by the parent zone:

```
# dnssec-dsfromkey -a SHA-1 Kexample.com.+007+24828.key
example.com. IN DS 24828 7 1 D4A33E8DD550A9567B4C4971A34AD6C4B80A6AD3
```

### One Month Before KSK Rollover

On December 1, 2020, a month before the planned rollover, you (as administrator) should change the parameters on the current key. Set it to become inactive on January 1, 2021, and be deleted from the zone on February 1st, 2021; also generate a successor key (23550). Finally, generate a new DS record based on the new key, 23550:

```
# cd /etc/bind/keys/example.com/
# dnssec-settime -I 20210101 -D 20210201 Kexample.com.+007+24828
./Kexample.com.+007+24828.key
./Kexample.com.+007+24828.private
# dnssec-keygen -S Kexample.com.+007+24828
Generating key pair.....................................................................
↪.....................++ ....................................++
Kexample.com.+007+23550
# dnssec-dsfromkey -a SHA-1 Kexample.com.+007+23550.key
example.com. IN DS 23550 7 1 54FCF030AA1C79C0088FDEC1BD1C37DAA2E70DFB
```

The first command gets us into the key directory `/etc/bind/keys/example.com/`, where keys for `example.com` are stored.

The second, *dnssec-settime*, sets an inactive (*-I*) date of January 1, 2021, and a deletion (*-D*) date of February 1, 2021 for the current KSK (`Kexample.com.+007+24828`).

The third command, *dnssec-keygen*, creates a successor key, using the exact same parameters (algorithms, key sizes, etc.) as the current KSK. The new key pair created in our example is `Kexample.com.+007+23550`.

The fourth and final command, *dnssec-dsfromkey*, creates a DS record from the new KSK (23550), using SHA-1 as the digest type. Again, in practice most people generate two DS records for both supported digest types (SHA-1 and SHA-256), but for our example here we are only using one to keep the output small and hopefully clearer.

Make sure the successor keys are readable by *named*.

The *syslog* message indicates when the next key checking event is. The log message looks like this:

```
zone example.com/IN (signed): next key event: 01-Dec-2020 00:13:05.385
```

You can check the publish date of the key by looking at the key file:

```
# cd /etc/bind/keys/example.com
# cat Kexample.com.+007+23550.key
; This is a key-signing key, keyid 23550, for example.com.
; Created: 20201130160024 (Thu Dec  1 00:00:24 2020)
; Publish: 20201202000000 (Fri Dec  2 08:00:00 2020)
; Activate: 20210101000000 (Sun Jan  1 08:00:00 2021)
...
```

Since the publish date is set to the morning of December 2, and our example scenario takes place on December 1, the next morning you will notice that your zone has gained a new DNSKEY record based on your new KSK, but with no corresponding RRSIG yet. Below is the abbreviated output - with shortened DNSKEY and RRSIG - when querying the authoritative name server, 192.168.1.13:

```
$ dig @192.168.1.13 example.com. DNSKEY +dnssec +multiline

...
;; ANSWER SECTION:
example.com.   300 IN DNSKEY 256 3 7 (
                AwEAAdYqAc...TiSlrma6Ef
```

(continues on next page)

(continued from previous page)

```
                ) ; ZSK; alg = NSEC3RSASHA1; key id = 29747
example.com.    300 IN DNSKEY 257 3 7 (
                AwEAAeTJ+w...O+Zy9j0m63
                ) ; KSK; alg = NSEC3RSASHA1; key id = 24828
example.com.    300 IN DNSKEY 257 3 7 (
                AwEAAc1BQN...Wdc0qoH21H
                ) ; KSK; alg = NSEC3RSASHA1; key id = 23550
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20201206125617 20201107115617 24828 example.com.
                4y1iPVJOrK...aC3iF9vgc= )
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20201206125617 20201107115617 29747 example.com.
                g/gfmPjr+y...rt/S/xjPo= )

...
```

Anytime after generating the DS record, you can upload it; it is not necessary to wait for the DNSKEY to be published in your zone, since this new KSK is not active yet. You can do it immediately after the new DS record has been generated on December 1, or you can wait until the next day after you have verified that the new DNSKEY record is added to the zone. Below are some screenshots from GoDaddy's web-based interface, used to add a new DS record.[8]

1. After logging in, click the green "Launch" button next to the domain name you want to manage.

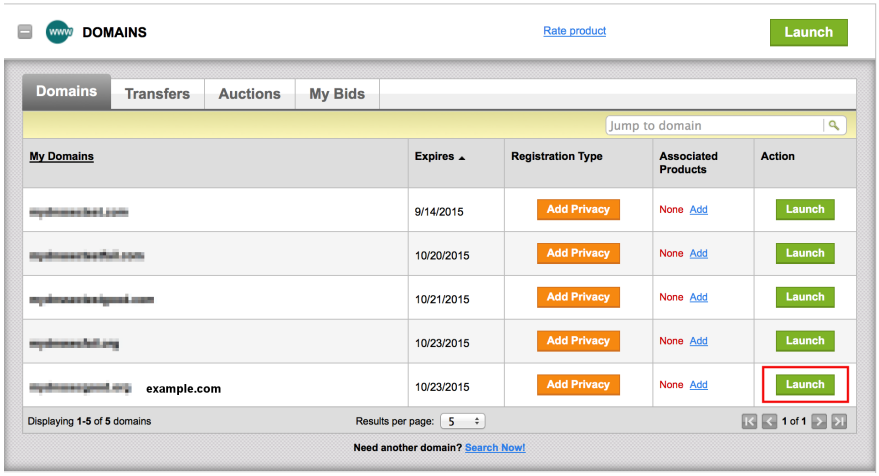



Fig. 7: Upload DS Record Step #1

2. Scroll down to the "DS Records" section and click "Manage."
3. A dialog appears, displaying the current key (24828). Click "Add DS Record."
4. Enter the Key ID, algorithm, digest type, and the digest, then click "Next."
5. Address any errors and click "Finish."
6. Both DS records are shown. Click "Save."

Finally, let's verify that the registrar has published the new DS record. This may take anywhere from a few minutes to a few days, depending on your parent zone. You can verify whether your parent zone has published the new DS record by querying for the DS record of your zone. In the example below, the Google public DNS server 8.8.8.8 is used:

[8] The screenshots were taken from GoDaddy's interface at the time the original version of this guide was published (2015). It may have changed since then.

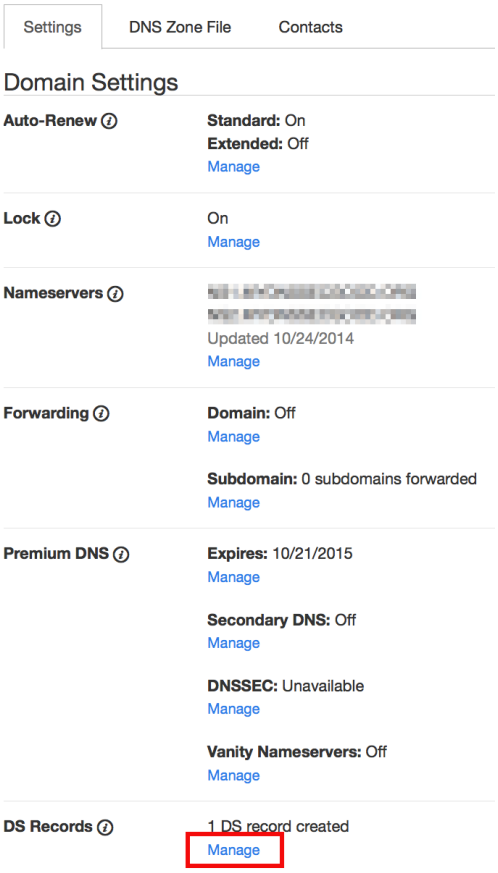



Fig. 8: Upload DS Record Step #2

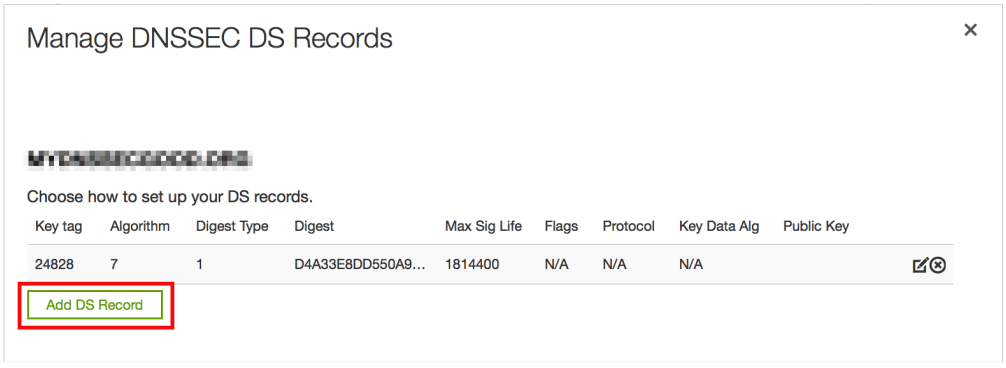



Fig. 9: Upload DS Record Step #3

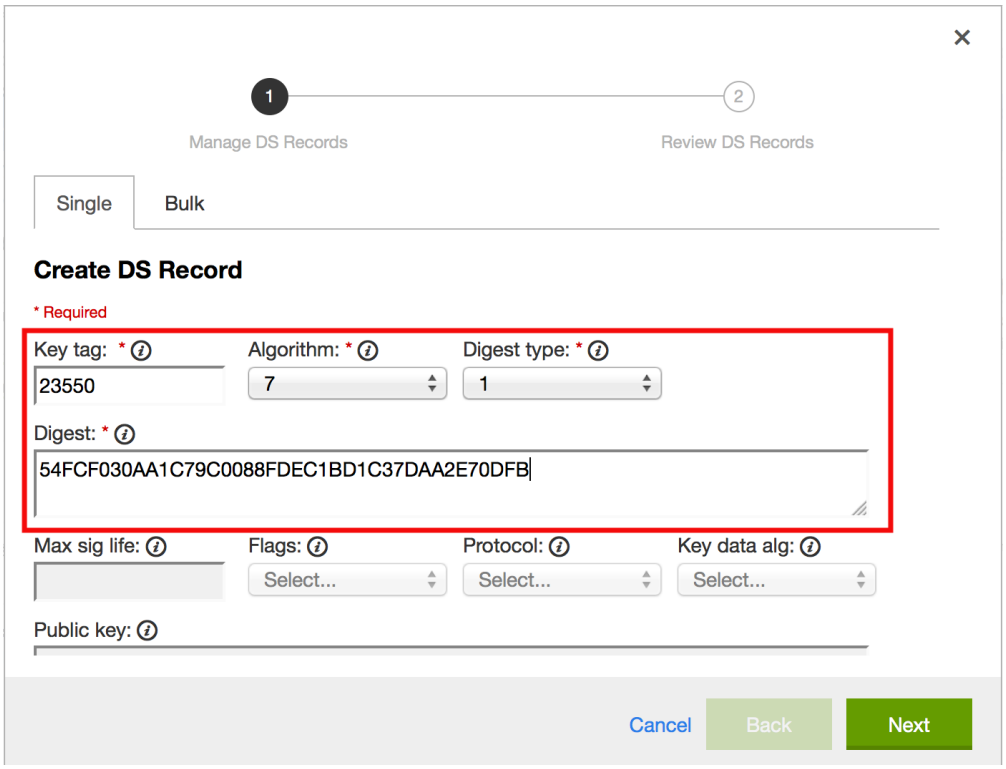



Fig. 10: Upload DS Record Step #4

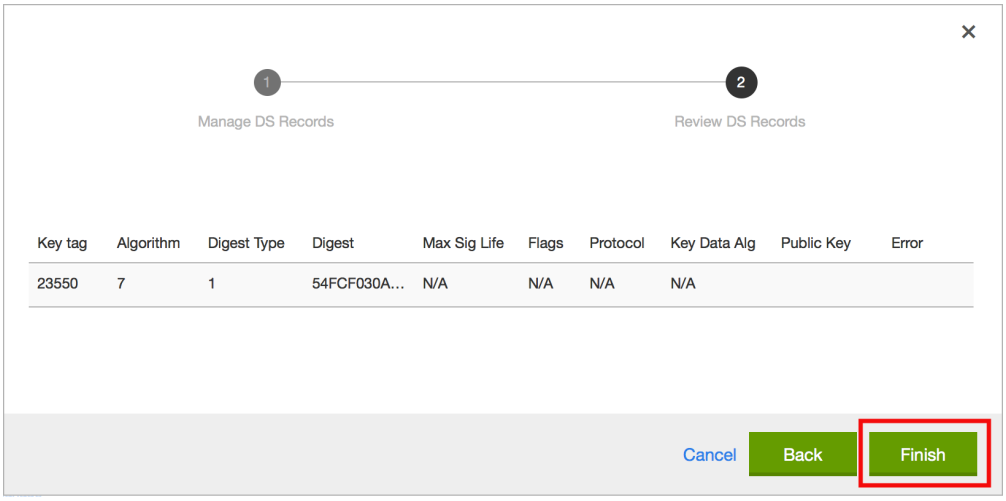



Fig. 11: Upload DS Record Step #5

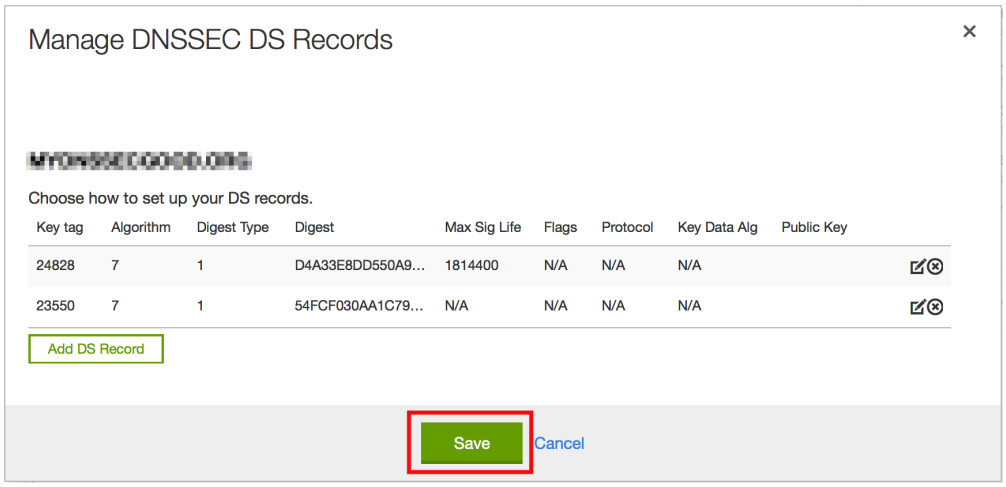



Fig. 12: Upload DS Record Step #6

```
$ dig @8.8.8.8 example.com. DS

...
;; ANSWER SECTION:
example.com.    21552   IN  DS  24828 7 1 D4A33E8DD550A9567B4C4971A34AD6C4B80A6AD3
example.com.    21552   IN  DS  23550 7 1 54FCF030AA1C79C0088FDEC1BD1C37DAA2E70DFB
```

You can also query your parent zone's authoritative name servers directly to see if these records have been published. DS records will not show up on your own authoritative zone, so you cannot query your own name servers for them. In this recipe, the parent zone is `.com`, so querying a few of the `.com` name servers is another appropriate verification.

### Day of KSK Rollover

If you have followed the examples in this document, as described in *Easy-Start Guide for Signing Authoritative Zones*, there is technically nothing you need to do manually on the actual day of the rollover. However, you should still keep an eye on the zone to make sure new signature(s) are being generated by the new KSK (23550 in this example). The easiest way is to query the authoritative name server 192.168.1.13 for the same DNSKEY and signatures, as you did a month ago:

```
$ dig @192.168.1.13 example.com. DNSKEY +dnssec +multiline

...
;; ANSWER SECTION:
example.com.    300 IN DNSKEY 256 3 7 (
                AwEAAdYqAc...TiSlrma6Ef
                ) ; ZSK; alg = NSEC3RSASHA1; key id = 29747
example.com.    300 IN DNSKEY 257 3 7 (
                AwEAAeTJ+w...O+Zy9j0m63
                ) ; KSK; alg = NSEC3RSASHA1; key id = 24828
example.com.    300 IN DNSKEY 257 3 7 (
                AwEAAc1BQN...Wdc0qoH21H
                ) ; KSK; alg = NSEC3RSASHA1; key id = 23550
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20210201074900 20210101064900 23550 mydnssecgood.org.
```

(continues on next page)

(continued from previous page)

```
                S6zTbBTfvU...Ib5eXkbtE= )
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20210105074900 20201206064900 29747 mydnssecgood.org.
                VY5URQA2/d...OVKr1+KX8= )
...
```

As you can see, the signature generated by the old KSK (24828) has disappeared, replaced by a new signature generated from the new KSK (23550).

#### One Month After KSK Rollover

While the removal of the old DNSKEY from the zone should be automated by *named*, the removal of the DS record is manual. You should make sure the old DNSKEY record is gone from your zone first, by querying for the DNSKEY records of the zone; this time we expect not to see the key with an ID of 24828:

```
$ dig @192.168.1.13 example.com. DNSKEY +dnssec +multiline

...
;; ANSWER SECTION:
example.com.    300 IN DNSKEY 256 3 7 (
                AwEAAdYqAc...TiSlrma6Ef
                ) ; ZSK; alg = NSEC3RSASHA1; key id = 29747
example.com.    300 IN DNSKEY 257 3 7 (
                AwEAAc1BQN...Wdc0qoH21H
                ) ; KSK; alg = NSEC3RSASHA1; key id = 23550
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20210208000000 20210105230000 23550 mydnssecgood.org.
                Qw9Em3dDok...bNCS7KISw= )
example.com.    300 IN RRSIG DNSKEY 7 2 300 (
                20210208000000 20210105230000 29747 mydnssecgood.org.
                OuelpIlpY9...XfsKupQgc= )
...
```

Since the key with the ID 24828 is gone, you can now remove the old DS record for that key from our parent zone. Be careful to remove the correct DS record. If you accidentally remove the new DS record(s) with key ID 23550, it could lead to a problem called "security lameness," as discussed in *Security Lameness*, and may cause users to be unable to resolve any names in the zone.

1. After logging in (again, GoDaddy.com in our example) and launching the domain, scroll down to the "DS Records" section and click Manage.
2. A dialog appears, displaying both keys (24828 and 23550). Use the far right-hand X button to remove key 24828.
3. Key 24828 now appears crossed out; click "Save" to complete the removal.

Congratulations, the KSK rollover is complete! As for the actual key files (ending in `.key` and `.private`), they may be deleted at this point, but they do not have to be.

### 11.8.3 NSEC and NSEC3

#### Migrating from NSEC to NSEC3

This recipe describes how to transition from using NSEC to NSEC3, as described in *Proof of Non-Existence (NSEC and NSEC3)*. This recipe assumes that the zones are already signed, and that *named* is configured according to the steps described in *Easy-Start Guide for Signing Authoritative Zones*.

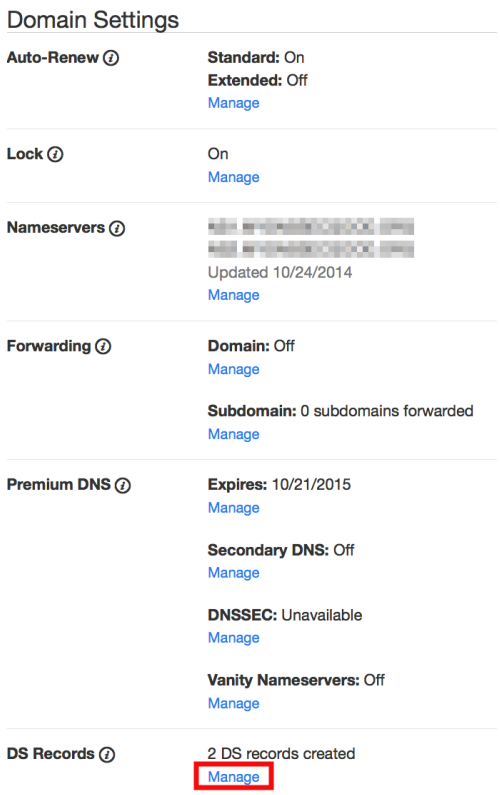



Fig. 13: Remove DS Record Step #1

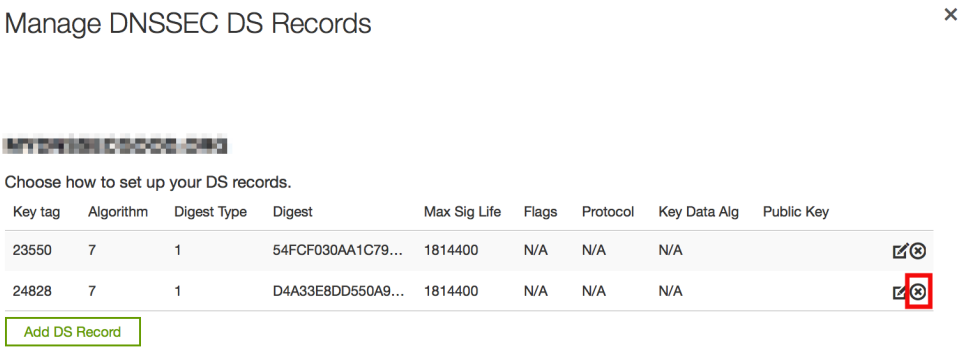



Fig. 14: Remove DS Record Step #2

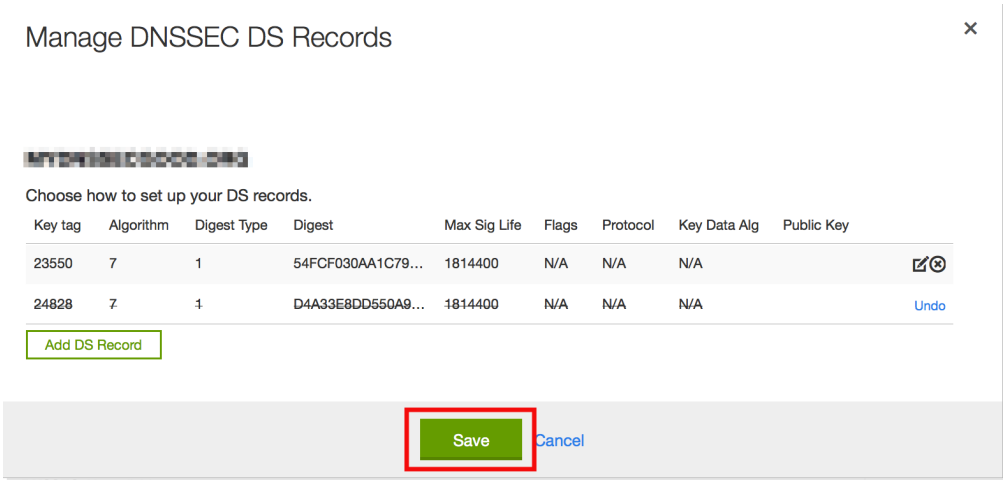



Fig. 15: Remove DS Record Step #3

**Warning**

If your zone is signed with RSASHA1 (algorithm 5), you cannot migrate to NSEC3 without also performing an algorithm rollover to RSASHA1-NSEC3-SHA1 (algorithm 7), as described in *Algorithm Rollovers*. This ensures that older validating resolvers that do not understand NSEC3 will fall back to treating the zone as unsecured (rather than "bogus"), as described in Section 2 of **RFC 5155**.

To enable NSEC3, update your *dnssec-policy* and add the desired NSEC3 parameters. The example below enables NSEC3 for zones with the `standard` DNSSEC policy, using 0 additional iterations, no opt-out, and a zero-length salt:

```
dnssec-policy "standard" {
    nsec3param iterations 0 optout no salt-length 0;
};
```

Then reconfigure the server with *rndc*. You can tell that it worked if you see the following debug log messages:

```
Oct 21 13:47:21 received control channel command 'reconfig'
Oct 21 13:47:21 zone example.com/IN (signed): zone_addnsec3chain(1,CREATE,0,-)
```

You can also verify that it worked by querying for a name that you know does not exist, and checking for the presence of the NSEC3 record. For example:

```
$ dig @192.168.1.13 thereisnowaythisexists.example.com. A +dnssec +multiline

...
5A03TL362CS8VSIH69CVA4MJIKRHFQH3.example.com. 300 IN NSEC3 1 0 0 - (
                TQ9QBEGA6CROHEOC8KIH1A2C06IVQ5ER
                NS SOA RRSIG DNSKEY NSEC3PARAM )
...
```

Our example used four parameters: 1, 0, 0, and -, in order. 1 represents the algorithm, 0 represents the opt-out flag, 0 represents the number of additional iterations, and - denotes no salt is used. To learn more about each of these parameters, please see *NSEC3PARAM*.

### Migrating from NSEC3 to NSEC

Migrating from NSEC3 back to NSEC is easy; just remove the *nsec3param* configuration option from your *dnssec-policy* and reconfigure the name server. You can tell that it worked if you see these messages in the log:

```
named[14093]: received control channel command 'reconfig'
named[14093]: zone example.com/IN: zone_addnsec3chain(1,REMOVE,0,-)
```

You can also query for a name that you know does not exist, and you should no longer see any traces of NSEC3 records.

```
$ dig @192.168.1.13 reieiergiuhewhiouwe.example.com. A +dnssec +multiline

...
example.com.        300 IN NSEC aaa.example.com. NS SOA RRSIG NSEC DNSKEY
...
ns1.example.com.    300 IN NSEC web.example.com. A RRSIG NSEC
...
```

### NSEC3 Opt-Out

This recipe discusses how to enable and disable NSEC3 opt-out, and how to show the results of each action. As discussed in *NSEC3 Opt-Out*, NSEC3 opt-out is a feature that can help conserve resources on parent zones with many delegations that have not yet been signed.

**Warning**

NSEC3 Opt-Out feature brings benefit only to _extremely_ large zones with lots of insecure delegations. It's use is counterproductive in all other cases as it decreases tamper-resistance of the zone and also decreases efficiency of resolver cache (see **RFC 8198**).

In other words, don't enable Opt-Out unless you are serving an equivalent of `com.` zone.

Because the NSEC3PARAM record does not keep track of whether opt-out is used, it is hard to check whether changes need to be made to the NSEC3 chain if the flag is changed. Similar to changing the NSEC3 salt, your best option is to change the value of `optout` together with another NSEC3 parameter, like `iterations`, and in a following step restore the `iterations` value.

For this recipe we assume the zone `example.com` has the following four entries (for this example, it is not relevant what record types these entries are):

- `ns1.example.com`
- `ftp.example.com`
- `www.example.com`
- `web.example.com`

And the zone `example.com` has five delegations to five subdomains, only one of which is signed and has a valid DS RRset:

- `aaa.example.com`, not signed
- `bbb.example.com`, signed
- `ccc.example.com`, not signed
- `ddd.example.com`, not signed

- `eee.example.com`, not signed

Before enabling NSEC3 opt-out, the zone `example.com` contains ten NSEC3 records; below is the list with the plain text name before the actual NSEC3 record:

- *aaa.example.com*: IFA1I3IE7EKCTPHM6R58URO3Q846I52M.example.com
- *bbb.example.com*: ROJUF3VJSJO6LQ2LC1DNSJ5GBAUJPVHE.example.com
- *ccc.example.com*: 0VPUT696LUVDPDS5NIHSHBH9KLV20V5K.example.com
- *ddd.example.com*: UHPBD5U4HRGB84MLC2NQOVEFNAKJU0CA.example.com
- *eee.example.com*: NF7I61FA4C2UEKPMEDSOC25FE0UJIMKT.example.com
- *ftp.example.com*: 8P15KCUAT1RHCSDN46HBQVPI5T532IN1.example.com
- *ns1.example.com*: GUFVRA2SFIO8RSFP7UO41E8AD1KR41FH.example.com
- *web.example.com*: CVQ4LA4ALPQIAO2H3N2RB6IR8UHM91E7.example.com
- *www.example.com*: MIFDNDT3NFF3OD53O7TLA1HRFF95JKUK.example.com
- *example.com*: ONIB9MGUB9H0RML3CDF5BGRJ59DKJHVK.example.com

We can enable NSEC3 opt-out with the following configuration, changing the `optout` configuration value from `no` to `yes`:

```
dnssec-policy "standard" {
    nsec3param iterations 0 optout yes salt-length 0;
};
```

After NSEC3 opt-out is enabled, the number of NSEC3 records is reduced. Notice that the unsigned delegations `aaa`, `ccc`, `ddd`, and `eee` no longer have corresponding NSEC3 records.

- *bbb.example.com*: ROJUF3VJSJO6LQ2LC1DNSJ5GBAUJPVHE.example.com
- *ftp.example.com*: 8P15KCUAT1RHCSDN46HBQVPI5T532IN1.example.com
- *ns1.example.com*: GUFVRA2SFIO8RSFP7UO41E8AD1KR41FH.example.com
- *web.example.com*: CVQ4LA4ALPQIAO2H3N2RB6IR8UHM91E7.example.com
- *www.example.com*: MIFDNDT3NFF3OD53O7TLA1HRFF95JKUK.example.com
- *example.com*: ONIB9MGUB9H0RML3CDF5BGRJ59DKJHVK.example.com

To undo NSEC3 opt-out, change the configuration again:

```
dnssec-policy "standard" {
    nsec3param iterations 0 optout no salt-length 0;
};
```

**Note**

NSEC3 hashes the plain text domain name, and we can compute our own hashes using the tool *nsec3hash*. For example, to compute the hashed name for `www.example.com` using the parameters we listed above, we can execute this command:

```
# nsec3hash - 1 0 www.example.com.
MIFDNDT3NFF3OD53O7TLA1HRFF95JKUK (salt=-, hash=1, iterations=0)
```

### 11.8.4 Reverting to Unsigned

This recipe describes how to revert from a signed zone (DNSSEC) back to an unsigned (DNS) zone.

Here is what *named.conf* looks like when it is signed:

```
zone "example.com" IN {
    type primary;
    file "db/example.com.db";
    dnssec-policy "default";
};
```

To indicate the reversion to unsigned, change the *dnssec-policy* line:

```
zone "example.com" IN {
    type primary;
    file "db/example.com.db";
    dnssec-policy "insecure";
};
```

Then use *rndc reload* to reload the zone.

The "insecure" policy is a built-in policy (like "default"). It makes sure the zone is still DNSSEC-maintained, to allow for a graceful transition to unsigned. It also publishes the CDS and CDNSKEY DELETE records automatically at the appropriate time.

If the parent zone allows management of DS records via CDS/CDNSKEY, as described in **RFC 8078**, the DS record should be removed from the parent automatically.

Otherwise, DS records can be removed via the registrar. Below is an example showing how to remove DS records using the GoDaddy web-based interface:

1. After logging in, click the green "Launch" button next to the domain name you want to manage.

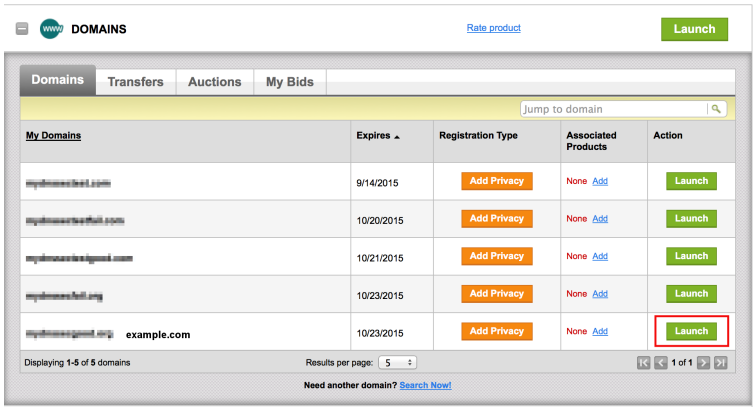



Fig. 16: Revert to Unsigned Step #1

2. Scroll down to the "DS Records" section and click Manage.

3. A dialog appears, displaying all current keys. Use the far right-hand X button to remove each key.

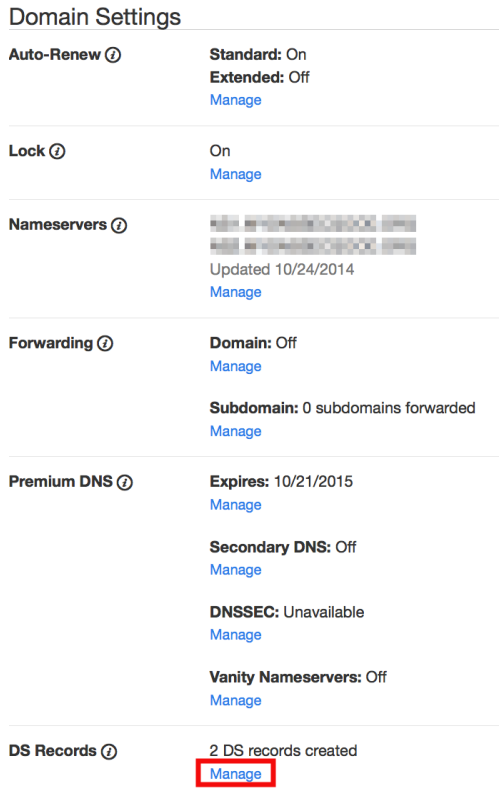



Fig. 17: Revert to Unsigned Step #2

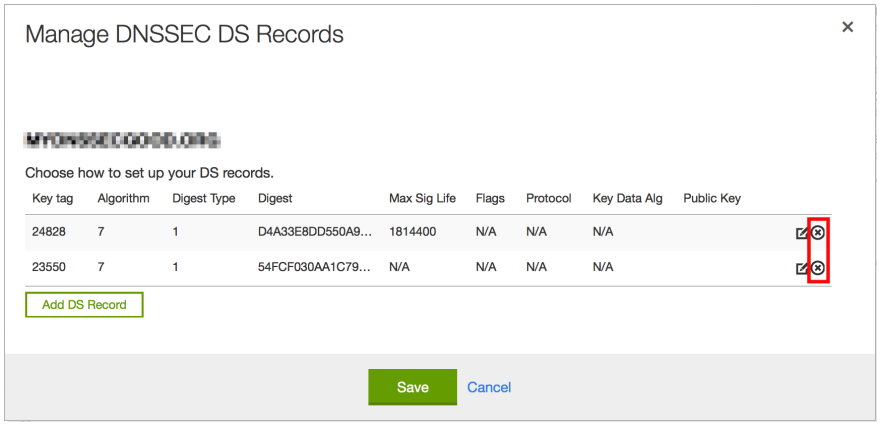



Fig. 18: Revert to Unsigned Step #3

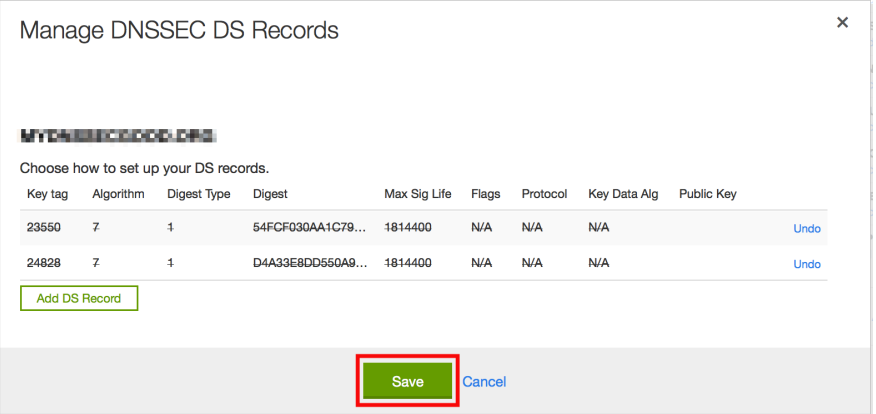



Fig. 19: Revert to Unsigned Step #4

4. Click Save.

When the DS records have been removed from the parent zone, use `rndc dnssec -checkds -key id withdrawn example.com` to tell `named` that the DS is removed, and the remaining DNSSEC records will be removed in a timely manner. Or, if parental agents are configured, the DNSSEC records will be automatically removed after BIND has seen that the parental agents no longer serve the DS RRset for this zone.

After a while, the zone is reverted back to the traditional, insecure DNS format. This can be verified by checking that all DNSKEY and RRSIG records have been removed from the zone.

The `dnssec-policy` line can then be removed from `named.conf` and the zone reloaded. The zone will no longer be subject to any DNSSEC maintenance.

## 11.9 Commonly Asked Questions

Below are some common questions and (hopefully) some answers that help.

**Do I need IPv6 to have DNSSEC?**
No. DNSSEC can be deployed without IPv6.

**Does DNSSEC encrypt my DNS traffic, so others cannot eavesdrop on my DNS queries?**
No. Although cryptographic keys and digital signatures are used in DNSSEC, they only provide authenticity and integrity, not privacy. Someone who sniffs network traffic can still see all the DNS queries and answers in plain text; DNSSEC just makes it very difficult for the eavesdropper to alter or spoof the DNS responses. For protection against eavesdropping, the preferred protocol is DNS-over-TLS. DNS-over-HTTPS can also do the job, but it is more complex.

**If I deploy DNS-over-TLS/HTTPS, can I skip deploying DNSSEC?**
No. DNS-over-encrypted-transport stops eavesdroppers on a network, but it does not protect against cache poisoning and answer manipulation in other parts of the DNS resolution chain. In other words, these technologies offer protection only for records when they are in transit between two machines; any compromised server can still redirect traffic elsewhere (or simply eavesdrop). However, DNSSEC provides integrity and authenticity for DNS *records*, even when these records are stored in caches and on disks.

**Does DNSSEC protect the communication between my laptop and my name server?**
Unfortunately, not at the moment. DNSSEC is designed to protect the communication between end clients (laptop) and name servers; however, there are few applications or stub resolver libraries as of mid-2020 that take advantage of this capability.

**Does DNSSEC secure zone transfers?**
No. You should consider using TSIG to secure zone transfers among your name servers.

**Does DNSSEC protect my network from malicious websites?**
DNSSEC makes it much more difficult for attackers to spoof DNS responses or perform cache poisoning. It cannot protect against users who visit a malicious website that an attacker owns and operates, or prevent users from mistyping a domain name; it will just become less likely that an attacker can hijack other domain names.

In other words, DNSSEC is designed to provide confidence that when a DNS response is received for www.company.com over port 53, it really came from Company's name servers and the answers are authentic. But that does not mean the web server a user visits over port 80 or port 443 is necessarily safe.

**If I enable DNSSEC validation, will it break DNS lookup, since most domain names do not yet use DNSSEC?**
No, DNSSEC is backwards-compatible to "standard" DNS. A DNSSEC-enabled validating resolver can still look up all of these domain names as it always has under standard DNS.

There are four (4) categories of responses (see **RFC 4035**):

**Secure**
Domains that have DNSSEC deployed correctly.

**Insecure**
Domains that have yet to deploy DNSSEC.

**Bogus**
Domains that have deployed DNSSEC but have done it incorrectly.

**Indeterminate**
Domains for which it is not possible to determine whether these domains use DNSSEC.

A DNSSEC-enabled validating resolver still resolves *Secure* and *Insecure*; only *Bogus* and *Indeterminate* result in a SERVFAIL. As of mid-2022, roughly one-third of users worldwide are using DNSSEC validation on their recursive name servers. Google public DNS (8.8.8.8) also has enabled DNSSEC validation.

**Do I need to have special client software to use DNSSEC?**
No. DNSSEC only changes the communication behavior among DNS servers, not between a DNS server (validating resolver) and a client (stub resolver). With DNSSEC validation enabled on your recursive server, if a domain name does not pass the checks, an error message (typically SERVFAIL) is returned to clients; to most client software today, it appears that the DNS query has failed or that the domain name does not exist.

**Since DNSSEC uses public key cryptography, do I need Public Key Infrastructure (PKI) in order to use DNSSEC?**
No, DNSSEC does not depend on an existing PKI. Public keys are stored within the DNS hierarchy; the trustworthiness of each zone is guaranteed by its parent zone, all the way back to the root zone. A copy of the trust anchor for the root zone is distributed with BIND 9.

**Do I need to purchase SSL certificates from a Certificate Authority (CA) to use DNSSEC?**
No. With DNSSEC, you generate and publish your own keys, and sign your own data as well. There is no need to pay someone else to do it for you.

**My parent zone does not support DNSSEC; can I still sign my zone?**
Technically, yes, but you will not get the full benefit of DNSSEC, as other validating resolvers are not able to validate your zone data. Without the DS record(s) in your parent zone, other validating resolvers treat your zone as an insecure (traditional) zone, and no actual verification is carried out. To the rest of the world, your zone still appears to be insecure, and it will continue to be insecure until your parent zone can host the DS record(s) for you and tell the rest of the world that your zone is signed.

**Is DNSSEC the same thing as TSIG?**
No. TSIG is typically used between primary and secondary name servers to secure zone transfers, while DNSSEC secures DNS lookup by validating answers. Even if you enable DNSSEC, zone transfers are still not validated; to

secure the communication between your primary and secondary name servers, consider setting up TSIG or similar secure channels.

**How are keys copied from primary to secondary server(s)?**

DNSSEC uses public cryptography, which results in two types of keys: public and private. The public keys are part of the zone data, stored as DNSKEY record types. Thus the public keys are synchronized from primary to secondary server(s) as part of the zone transfer. The private keys are not, and should not be, stored anywhere other than secured on the primary server. See *Key Storage* for more information on key storage options and considerations.

**Can I use the same key for multiple zones?**

Yes and no. Good security practice suggests that you should use unique key pairs for each zone, just as you should have different passwords for your email account, social media login, and online banking credentials. On a technical level, it is completely feasible to reuse a key, but multiple zones are at risk if one key pair is compromised. However, if you have hundreds or thousands of zones to administer, a single key pair for all might be less error-prone to manage. You may choose to use the same approach as with password management: use unique passwords for your bank accounts and shopping sites, but use a standard password for your not-very-important logins. First, categorize your zones: high-value zones (or zones that have specific key rollover requirements) get their own key pairs, while other, more "generic" zones can use a single key pair for easier management. Note that at present (mid-2020), fully automatic signing (using the `dnssec-policy` clause in your `named` configuration file) does not support reuse of keys except when the same zone appears in multiple views (see next question). To use the same key for multiple zones, sign your zones using semi-automatic signing. Each zone wishing to use the key should point to the same key directory.

**How do I sign the different instances of a zone that appears in multiple views?**

Add a `dnssec-policy` statement to each `zone` definition in the configuration file. To avoid problems when a single computer accesses different instances of the zone while information is still in its cache (e.g., a laptop moving from your office to a customer site), you should sign all instances with the same key. This means setting the same DNSSEC policy for all instances of the zone, and making sure that the key directory is the same for all instances of the zone.

**Will there be any problems if I change the DNSSEC policy for a zone?**

If you are using fully automatic signing, no. Just change the parameters in the `dnssec-policy` statement and reload the configuration file. `named` makes a smooth transition to the new policy, ensuring that your zone remains valid at all times.

CHAPTER

# TWELVE

# MANUAL PAGES

## 12.1 arpaname - translate IP addresses to the corresponding ARPA names

### 12.1.1 Synopsis

**arpaname** {*ipaddress* ...}

### 12.1.2 Description

**arpaname** translates IP addresses (IPv4 and IPv6) to the corresponding IN-ADDR.ARPA or IP6.ARPA names.

### 12.1.3 See Also

BIND 9 Administrator Reference Manual.

## 12.2 ddns-confgen - TSIG key generation tool

### 12.2.1 Synopsis

**ddns-confgen** [**-a** algorithm] [**-h**] [**-k** keyname] [**-q**] [**-s** name] [**-z** zone]

### 12.2.2 Description

**ddns-confgen** is an utility that generates keys for use in TSIG signing. The resulting keys can be used, for example, to secure dynamic DNS updates to a zone, or for the *rndc* command channel.

The key name can specified using *-k* parameter and defaults to `ddns-key`. The generated key is accompanied by configuration text and instructions that can be used with *nsupdate* and *named* when setting up dynamic DNS, including an example `update-policy` statement. (This usage is similar to the *rndc-confgen* command for setting up command-channel security.)

Note that *named* itself can configure a local DDNS key for use with *nsupdate -l*; it does this when a zone is configured with `update-policy local;`. **ddns-confgen** is only needed when a more elaborate configuration is required: for instance, if *nsupdate* is to be used from a remote system.

### 12.2.3 Options

**-a** `algorithm`

This option specifies the algorithm to use for the TSIG key. Available choices are: hmac-md5, hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, and hmac-sha512. The default is hmac-sha256. Options are case-insensitive, and the "hmac-" prefix may be omitted.

**-h**

This option prints a short summary of options and arguments.

**-k** keyname

This option specifies the key name of the DDNS authentication key. The default is `ddns-key` when neither the *-s* nor *-z* option is specified; otherwise, the default is `ddns-key` as a separate label followed by the argument of the option, e.g., `ddns-key.example.com`. The key name must have the format of a valid domain name, consisting of letters, digits, hyphens, and periods.

**-q**

This option enables quiet mode, which prints only the key, with no explanatory text or usage examples. This is essentially identical to *tsig-keygen*.

**-s** name

This option generates a configuration example to allow dynamic updates of a single hostname. The example *named.conf* text shows how to set an update policy for the specified name using the "name" nametype. The default key name is `ddns-key.name`. Note that the "self" nametype cannot be used, since the name to be updated may differ from the key name. This option cannot be used with the *-z* option.

**-z** zone

This option generates a configuration example to allow dynamic updates of a zone. The example *named.conf* text shows how to set an update policy for the specified zone using the "zonesub" nametype, allowing updates to all subdomain names within that zone. This option cannot be used with the *-s* option.

### 12.2.4 See Also

*nsupdate(1)*, *named.conf(5)*, *named(8)*, BIND 9 Administrator Reference Manual.

## 12.3 delv - DNS lookup and validation utility

### 12.3.1 Synopsis

**delv** [@server] [ [**-4**] | [**-6**] ] [**-a** anchor-file] [**-b** address] [**-c** class] [**-d** level] [**-i**] [**-m**] [**-p** port#] [**-q** name] [**-t** type] [**-x** addr] [name] [type] [class] [queryopt...]

**delv** [**-h**]

**delv** [**-v**]

**delv** [queryopt...] [query...]

### 12.3.2 Description

**delv** is a tool for sending DNS queries and validating the results, using the same internal resolver and validator logic as *named*.

**delv** sends to a specified name server all queries needed to fetch and validate the requested data; this includes the original requested query, subsequent queries to follow CNAME or DNAME chains, queries for DNSKEY, and DS records to establish a chain of trust for DNSSEC validation. It does not perform iterative resolution, but simulates the behavior of a name server configured for DNSSEC validating and forwarding.

By default, responses are validated using the built-in DNSSEC trust anchor for the root zone ("."). Records returned by **delv** are either fully validated or were not signed. If validation fails, an explanation of the failure is included in the output; the validation process can be traced in detail. Because **delv** does not rely on an external server to carry out validation, it can be used to check the validity of DNS responses in environments where local name servers may not be trustworthy.

Unless it is told to query a specific name server, **delv** tries each of the servers listed in `/etc/resolv.conf`. If no usable server addresses are found, **delv** sends queries to the localhost addresses (127.0.0.1 for IPv4, ::1 for IPv6).

When no command-line arguments or options are given, **delv** performs an NS query for "." (the root zone).

### 12.3.3 Simple Usage

A typical invocation of **delv** looks like:

```
delv @server name type
```

where:

**server**

is the name or IP address of the name server to query. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation. When the supplied `server` argument is a hostname, **delv** resolves that name before querying that name server (note, however, that this initial lookup is *not* validated by DNSSEC).

If no `server` argument is provided, **delv** consults `/etc/resolv.conf`; if an address is found there, it queries the name server at that address. If either of the *-4* or *-6* options is in use, then only addresses for the corresponding transport are tried. If no usable addresses are found, **delv** sends queries to the localhost addresses (127.0.0.1 for IPv4, ::1 for IPv6).

**name**

is the domain name to be looked up.

**type**

indicates what type of query is required - ANY, A, MX, etc. `type` can be any valid query type. If no `type` argument is supplied, **delv** performs a lookup for an A record.

### 12.3.4 Options

**-a** `anchor-file`

This option specifies a file from which to read an alternate DNSSEC root zone trust anchor.

By default, keys that do not match the root zone name (.) are ignored. If an alternate key name is desired, it can be specified using the *+root* option.

Note: When reading trust anchors, **delv** treats `trust-anchors`, `initial-key`, and `static-key` identically. That is, for a managed key, it is the *initial* key that is trusted; **RFC 5011** key management is not supported. **delv** does not consult the managed-keys database maintained by *named*. This means that if the default key built in to **delv** is revoked, **delv** must be updated to a newer version in order to continue validating.

**-b** `address`

This option sets the source IP address of the query to `address`. This must be a valid address on one of the host's network interfaces, or `0.0.0.0`, or `::`. An optional source port may be specified by appending `#<port>`

**-c** `class`

This option sets the query class for the requested data. Currently, only class "IN" is supported in **delv** and any other value is ignored.

**-d** `level`

This option sets the systemwide debug level to `level`. The allowed range is from 0 to 99. The default is 0 (no debugging). Debugging traces from **delv** become more verbose as the debug level increases. See the *+mtrace*, *+rtrace*, and *+vtrace* options below for additional debugging details.

**-h**

This option displays the **delv** help usage output and exits.

**-i**

This option sets insecure mode, which disables internal DNSSEC validation. (Note, however, that this does not set the CD bit on upstream queries. If the server being queried is performing DNSSEC validation, then it does not return invalid data; this can cause **delv** to time out. When it is necessary to examine invalid data to debug a DNSSEC problem, use *dig +cd*.)

**-m**

This option enables memory usage debugging.

**-p** port#

This option specifies a destination port to use for queries, instead of the standard DNS port number 53. This option is used with a name server that has been configured to listen for queries on a non-standard port number.

**-q** name

This option sets the query name to `name`. While the query name can be specified without using the *-q* option, it is sometimes necessary to disambiguate names from types or classes (for example, when looking up the name "ns", which could be misinterpreted as the type NS, or "ch", which could be misinterpreted as class CH).

**-t** type

This option sets the query type to `type`, which can be any valid query type supported in BIND 9 except for zone transfer types AXFR and IXFR. As with *-q*, this is useful to distinguish query-name types or classes when they are ambiguous. It is sometimes necessary to disambiguate names from types.

The default query type is "A", unless the *-x* option is supplied to indicate a reverse lookup, in which case it is "PTR".

**-v**

This option prints the **delv** version and exits.

**-x** addr

This option performs a reverse lookup, mapping an address to a name. `addr` is an IPv4 address in dotted-decimal notation, or a colon-delimited IPv6 address. When *-x* is used, there is no need to provide the `name` or `type` arguments; **delv** automatically performs a lookup for a name like `11.12.13.10.in-addr.arpa` and sets the query type to PTR. IPv6 addresses are looked up using nibble format under the IP6.ARPA domain.

**-4**

This option forces **delv** to only use IPv4.

**-6**

This option forces **delv** to only use IPv6.

### 12.3.5 Query Options

**delv** provides a number of query options which affect the way results are displayed, and in some cases the way lookups are performed.

Each query option is identified by a keyword preceded by a plus sign (+). Some keywords set or reset an option. These may be preceded by the string `no` to negate the meaning of that keyword. Other keywords assign values to options like the timeout interval. They have the form `+keyword=value`. The query options are:

**+cdflag, +nocdflag**

This option controls whether to set the CD (checking disabled) bit in queries sent by **delv**. This may be useful when troubleshooting DNSSEC problems from behind a validating resolver. A validating resolver blocks invalid responses, making it difficult to retrieve them for analysis. Setting the CD flag on queries causes the resolver to return invalid responses, which **delv** can then validate internally and report the errors in detail.

**+class, +noclass**

This option controls whether to display the CLASS when printing a record. The default is to display the CLASS.

**+hint**=FILE, **+nohint**

This option specifies a filename from which to load root hints; this will be used to find the root name servers when name server mode (`delv +ns`) is in use. If the option is not specified, built-in root hints will be used.

**+ns, +nons**

This option toggles name server mode. When this option is in use, the `delv` process instantiates a full recursive resolver, and uses that to look up the requested query name and type. Turning on this option also activates `+mtrace`, `+strace` and `+rtrace`, so that every iterative query will be logged, including the full response messages from each authoritatve server. These logged messages will be written to `stdout` rather than `stderr` as usual, so that the full trace can be captured more easily.

This is intended to be similar to the behavior of `dig +trace`, but because it uses the same code as `named`, it much more accurately replicates the behavior of a recursive name server with a cold cache that is processing a recursive query.

**+qmin**[=MODE], **+noqmin**

When used with `+ns`, this option enables QNAME minimization mode. Valid options of MODE are `relaxed` and `strict`. By default, QNAME minimization is disabled. If `+qmin` is specified but MODE is omitted, then `relaxed` mode will be used.

**+ttl, +nottl**

This option controls whether to display the TTL when printing a record. The default is to display the TTL.

**+rtrace, +nortrace**

This option toggles resolver fetch logging. This reports the name and type of each query sent by **delv** in the process of carrying out the resolution and validation process, including the original query and all subsequent queries to follow CNAMEs and to establish a chain of trust for DNSSEC validation.

This is equivalent to setting the debug level to 1 in the "resolver" logging category. Setting the systemwide debug level to 1 using the *-d* option produces the same output, but affects other logging categories as well.

**+mtrace, +nomtrace**

This option toggles logging of messages received. This produces a detailed dump of the responses received by **delv** in the process of carrying out the resolution and validation process.

This is equivalent to setting the debug level to 10 for the "packets" module of the "resolver" logging category. Setting the systemwide debug level to 10 using the *-d* option produces the same output, but affects other logging categories as well.

**+strace, +nostrace**

This option toggles logging of messages sent. This produces a detailed dump of the queries sent by **delv** in the process of carrying out the resolution and validation process. Turning on this option also activates `+mtrace`.

This is equivalent to setting the debug level to 11 for the "packets" module of the "resolver" logging category. Setting the systemwide debug level to 11 using the *-d* option produces the same output, but affects other logging categories as well.

**+vtrace, +novtrace**

This option toggles validation logging. This shows the internal process of the validator as it determines whether an answer is validly signed, unsigned, or invalid.

This is equivalent to setting the debug level to 3 for the "validator" module of the "dnssec" logging category. Setting the systemwide debug level to 3 using the *-d* option produces the same output, but affects other logging categories as well.

**+short, +noshort**

This option toggles between verbose and terse answers. The default is to print the answer in a verbose form.

**+comments, +nocomments**

This option toggles the display of comment lines in the output. The default is to print comments.

**+rrcomments, +norrcomments**

This option toggles the display of per-record comments in the output (for example, human-readable key information about DNSKEY records). The default is to print per-record comments.

**+crypto, +nocrypto**

This option toggles the display of cryptographic fields in DNSSEC records. The contents of these fields are unnecessary to debug most DNSSEC validation failures and removing them makes it easier to see the common failures. The default is to display the fields. When omitted, they are replaced by the string `[omitted]` or, in the DNSKEY case, the key ID is displayed as the replacement, e.g. `[ key id = value ]`.

**+restarts**

When name server mode (`delv +ns`) is in use, this option sets the maximum number of CNAME queries to follow before terminating resolution. This prevents `delv` from hanging in the event of a CNAME loop. The default is 11.

**+maxqueries**

This option specifies the maximum number of queries to send to resolve a name before giving up. The default is 50.

**+maxtotalqueries**

This option specifies the maximum number of queries to send to resolve a client request before giving up. The default is 200.

**+trust, +notrust**

This option controls whether to display the trust level when printing a record. The default is to display the trust level.

**+split**`[=W]`**, +nosplit**

This option splits long hex- or base64-formatted fields in resource records into chunks of `W` characters (where `W` is rounded up to the nearest multiple of 4). `+nosplit` or `+split=0` causes fields not to be split at all. The default is 56 characters, or 44 characters when multiline mode is active.

**+all, +noall**

This option sets or clears the display options *+comments*, *+rrcomments*, and *+trust* as a group.

**+multiline, +nomultiline**

This option prints long records (such as RRSIG, DNSKEY, and SOA records) in a verbose multi-line format with human-readable comments. The default is to print each record on a single line, to facilitate machine parsing of the **delv** output.

**+dnssec, +nodnssec**

This option indicates whether to display RRSIG records in the **delv** output. The default is to do so. Note that (unlike in *dig*) this does *not* control whether to request DNSSEC records or to validate them. DNSSEC records are always requested, and validation always occurs unless suppressed by the use of *-i* or *+noroot*.

**+root**`[=ROOT]`**, +noroot**

This option indicates whether to perform conventional DNSSEC validation, and if so, specifies the name of a trust anchor. The default is to validate using a trust anchor of "." (the root zone), for which there is a built-in key. If specifying a different trust anchor, then *-a* must be used to specify a file containing the key.

**+tcp, +notcp**

This option controls whether to use TCP when sending queries. The default is to use UDP unless a truncated response has been received.

**+unknownformat, +nounknownformat**

This option prints all RDATA in unknown RR-type presentation format (RFC 3597). The default is to print RDATA for known types in the type's presentation format.

**+yaml, +noyaml**

This option prints response data in YAML format.

### 12.3.6 Files

`/etc/resolv.conf`

### 12.3.7 See Also

*dig(1)*, *named(8)*, RFC 4034, RFC 4035, RFC 4431, RFC 5074, RFC 5155.

## 12.4 dig - DNS lookup utility

### 12.4.1 Synopsis

**dig** [@server] [**-b** address] [**-c** class] [**-f** filename] [**-k** filename] [**-m**] [**-p** port#] [**-q** name] [**-t** type] [**-v**] [**-x** addr] [**-y** [hmac:]name:key] [ [**-4**] | [**-6**] ] [name] [type] [class] [queryopt...]

**dig** [**-h**]

**dig** [global-queryopt...] [query...]

### 12.4.2 Description

**dig** is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use **dig** to troubleshoot DNS problems because of its flexibility, ease of use, and clarity of output. Other lookup tools tend to have less functionality than **dig**.

Although **dig** is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. A brief summary of its command-line arguments and options is printed when the *-h* option is given. The BIND 9 implementation of **dig** allows multiple lookups to be issued from the command line.

Unless it is told to query a specific name server, **dig** tries each of the servers listed in `/etc/resolv.conf`. If no usable server addresses are found, **dig** sends the query to the local host.

When no command-line arguments or options are given, **dig** performs an NS query for "." (the root).

It is possible to set per-user defaults for **dig** via `${HOME}/.digrc`. This file is read and any options in it are applied before the command-line arguments. The *-r* option disables this feature, for scripts that need predictable behavior.

The IN and CH class names overlap with the IN and CH top-level domain names. Either use the *-t* and *-c* options to specify the type and class, use the *-q* to specify the domain name, or use "IN." and "CH." when looking up these top-level domains.

### 12.4.3 Simple Usage

A typical invocation of **dig** looks like:

```
dig @server name type
```

where:

**server**

is the name or IP address of the name server to query. This can be an IPv4 address in dotted-decimal notation or an IPv6 address in colon-delimited notation. When the supplied `server` argument is a hostname, **dig** resolves that name before querying that name server.

If no `server` argument is provided, **dig** consults `/etc/resolv.conf`; if an address is found there, it queries the name server at that address. If either of the *-4* or *-6* options are in use, then only addresses for the corresponding transport are tried. If no usable addresses are found, **dig** sends the query to the local host. The reply from the name server that responds is displayed.

**name**

is the name of the resource record that is to be looked up.

**type**

indicates what type of query is required - ANY, A, MX, SIG, etc. `type` can be any valid query type. If no `type` argument is supplied, **dig** performs a lookup for an A record.

### 12.4.4 Options

**-4**

This option indicates that only IPv4 should be used.

**-6**

This option indicates that only IPv6 should be used.

**-b** `address[#port]`

This option sets the source IP address of the query. The `address` must be a valid address on one of the host's network interfaces, or "0.0.0.0" or "::". An optional port may be specified by appending `#port`.

**-c** `class`

This option sets the query class. The default `class` is IN; other classes are HS for Hesiod records or CH for Chaosnet records.

**-f** `file`

This option sets batch mode, in which **dig** reads a list of lookup requests to process from the given `file`. Each line in the file should be organized in the same way it would be presented as a query to **dig** using the command-line interface.

**-h**

Print a usage summary.

**-k** `keyfile`

This option tells **dig** to sign queries using TSIG or SIG(0) using a key read from the given file. Key files can be generated using *tsig-keygen*. When using TSIG authentication with **dig**, the name server that is queried needs to know the key and algorithm that is being used. In BIND, this is done by providing appropriate `key` and `server` statements in *named.conf* for TSIG and by looking up the KEY record in zone data for SIG(0).

**-m**

This option enables memory usage debugging.

**-p** port

This option sends the query to a non-standard port on the server, instead of the default port 53. This option is used to test a name server that has been configured to listen for queries on a non-standard port number.

**-q** name

This option specifies the domain name to query. This is useful to distinguish the `name` from other arguments.

**-r**

This option indicates that options from `${HOME}/.digrc` should not be read. This is useful for scripts that need predictable behavior.

**-t** type

This option indicates the resource record type to query, which can be any valid query type. If it is a resource record type supported in BIND 9, it can be given by the type mnemonic (such as `NS` or `AAAA`). The default query type is `A`, unless the *-x* option is supplied to indicate a reverse lookup. A zone transfer can be requested by specifying a type of AXFR. When an incremental zone transfer (IXFR) is required, set the `type` to `ixfr=N`. The incremental zone transfer contains all changes made to the zone since the serial number in the zone's SOA record was `N`.

All resource record types can be expressed as `TYPEnn`, where `nn` is the number of the type. If the resource record type is not supported in BIND 9, the result is displayed as described in **RFC 3597**.

**-u**

This option indicates that print query times should be provided in microseconds instead of milliseconds.

**-v**

This option prints the version number and exits.

**-x** addr

This option sets simplified reverse lookups, for mapping addresses to names. The `addr` is an IPv4 address in dotted-decimal notation, or a colon-delimited IPv6 address. When the *-x* option is used, there is no need to provide the `name`, `class`, and `type` arguments. **dig** automatically performs a lookup for a name like `94.2.0.192.in-addr.arpa` and sets the query type and class to PTR and IN respectively. IPv6 addresses are looked up using nibble format under the IP6.ARPA domain.

**-y** `[hmac:]keyname:secret`

This option signs queries using TSIG with the given authentication key. `keyname` is the name of the key, and `secret` is the base64-encoded shared secret. `hmac` is the name of the key algorithm; valid choices are `hmac-md5`, `hmac-sha1`, `hmac-sha224`, `hmac-sha256`, `hmac-sha384`, or `hmac-sha512`. If `hmac` is not specified, the default is `hmac-md5`; if MD5 was disabled, the default is `hmac-sha256`.

**Note**

Only the *-k* option should be used, rather than the *-y* option, because with *-y* the shared secret is supplied as a command-line argument in clear text. This may be visible in the output from `ps1` or in a history file maintained by the user's shell.

### 12.4.5 Query Options

**dig** provides a number of query options which affect the way in which lookups are made and the results displayed. Some of these set or reset flag bits in the query header, some determine which sections of the answer get printed, and others determine the timeout and retry strategies.

Each query option is identified by a keyword preceded by a plus sign (+). Some keywords set or reset an option; these may be preceded by the string `no` to negate the meaning of that keyword. Other keywords assign values to options, like the timeout interval. They have the form `+keyword=value`. Keywords may be abbreviated, provided the abbreviation is unambiguous; for example, *+cd* is equivalent to *+cdflag*. Query options are order sensitive. The query options are:

**+aaflag, +noaaflag**

This option is a synonym for *+aaonly*, *+noaaonly*.

**+aaonly, +noaaonly**

This option sets the `aa` flag in the query.

**+additional, +noadditional**

This option displays [or does not display] the additional section of a reply. The default is to display it.

**+adflag, +noadflag**

This option sets [or does not set] the AD (authentic data) bit in the query. This requests the server to return whether all of the answer and authority sections have been validated as secure, according to the security policy of the server. `AD=1` indicates that all records have been validated as secure and the answer is not from a OPT-OUT range. `AD=0` indicates that some part of the answer was insecure or not validated. This bit is set by default.

**+all, +noall**

This option sets or clears all display flags.

**+answer, +noanswer**

This option displays [or does not display] the answer section of a reply. The default is to display it.

**+authority, +noauthority**

This option displays [or does not display] the authority section of a reply. The default is to display it.

**+badcookie, +nobadcookie**

This option retries the lookup with a new server cookie if a BADCOOKIE response is received.

**+besteffort, +nobesteffort**

This option attempts to display the contents of messages which are malformed. The default is to not display malformed answers.

**+bufsize**`[=B]`

This option sets the UDP message buffer size advertised using EDNS0 to `B` bytes. The maximum and minimum sizes of this buffer are 65535 and 0, respectively. `+bufsize` restores the default buffer size.

**+cd, +cdflag, +nocdflag**

This option sets [or does not set] the CD (checking disabled) bit in the query. This requests the server to not perform DNSSEC validation of responses.

**+class, +noclass**

This option displays [or does not display] the CLASS when printing the record.

**+cmd, +nocmd**

This option toggles the printing of the initial comment in the output, identifying the version of **dig** and the query options that have been applied. This option always has a global effect; it cannot be set globally and then overridden on a per-lookup basis. The default is to print this comment.

**+coflag, +co, +nocoflag, +noco**

This option sets [or does not set] the CO (Compact denial of existence Ok) EDNS bit in the query. If set, it tells servers that Compact Denial of Existence responses are acceptable when replying to queries. The default is `+nocoflag`.

**+comments, +nocomments**

This option toggles the display of some comment lines in the output, with information about the packet header and OPT pseudosection, and the names of the response section. The default is to print these comments.

Other types of comments in the output are not affected by this option, but can be controlled using other command-line switches. These include *+cmd*, *+question*, *+stats*, and *+rrcomments*.

**+cookie**=####, **+nocookie**

This option sends [or does not send] a COOKIE EDNS option, with an optional value. Replaying a COOKIE from a previous response allows the server to identify a previous client. The default is `+cookie`.

`+cookie` is also set when *+trace* is set to better emulate the default queries from a nameserver.

**+crypto, +nocrypto**

This option toggles the display of cryptographic fields in DNSSEC records. The contents of these fields are unnecessary for debugging most DNSSEC validation failures and removing them makes it easier to see the common failures. The default is to display the fields. When omitted, they are replaced by the string `[omitted]` or, in the DNSKEY case, the key ID is displayed as the replacement, e.g. `[ key id = value ]`.

**+defname, +nodefname**

This option, which is deprecated, is treated as a synonym for *+search*, *+nosearch*.

**+dns64prefix, +nodns64prefix**

Lookup IPV4ONLY.ARPA AAAA and print any DNS64 prefixes found.

**+dnssec, +do, +nodnssec, +nodo**

This option requests that DNSSEC records be sent by setting the DNSSEC OK (DO) bit in the OPT record in the additional section of the query.

**+domain**=somename

This option sets the search list to contain the single domain `somename`, as if specified in a `domain` directive in `/etc/resolv.conf`, and enables search list processing as if the *+search* option were given.

**+edns**[=#], **+noedns**

This option specifies the EDNS version to query with. Valid values are 0 to 255. Setting the EDNS version causes an EDNS query to be sent. `+noedns` clears the remembered EDNS version. EDNS is set to 0 by default.

**+ednsflags**[=#], **+noednsflags**

This option sets the must-be-zero EDNS flags bits (Z bits) to the specified value. Decimal, hex, and octal encodings are accepted. Setting a named flag (e.g. DO, CO) is silently ignored. By default, no Z bits are set.

**+ednsnegotiation, +noednsnegotiation**

This option enables/disables EDNS version negotiation. By default, EDNS version negotiation is enabled.

**+ednsopt**[=code[:value]], **+noednsopt**

This option specifies the EDNS option with code point `code` and an optional payload of `value` as a hexadecimal string. `code` can be either an EDNS option name (for example, `NSID` or `ECS`) or an arbitrary numeric value. `+noednsopt` clears the EDNS options to be sent.

**+expire, +noexpire**

This option sends an EDNS Expire option.

**+fail, +nofail**

This option indicates that *named* should try [or not try] the next server if a SERVFAIL is received. The default is to not try the next server, which is the reverse of normal stub resolver behavior.

**+fuzztime**[=value], **+nofuzztime**

This option allows the signing time to be specified when generating signed messages. If a value is specified it is the seconds since 00:00:00 January 1, 1970 UTC ignoring leap seconds. If no value is specified 1646972129 (Fri 11 Mar 2022 04:15:29 UTC) is used. The default is `+nofuzztime` and the current time is used.

**+header-only, +noheader-only**

This option sends a query with a DNS header without a question section. The default is to add a question section. The query type and query name are ignored when this is set.

**+https**[=value], **+nohttps**

This option indicates whether to use DNS over HTTPS (DoH) when querying name servers. When this option is in use, the port number defaults to 443. The HTTP POST request mode is used when sending the query.

If `value` is specified, it will be used as the HTTP endpoint in the query URI; the default is `/dns-query`. So, for example, `dig @example.com +https` will use the URI `https://example.com/dns-query`.

**+https-get**[=value], **+nohttps-get**

Similar to *+https*, except that the HTTP GET request mode is used when sending the query.

**+https-post**[=value], **+nohttps-post**

Same as *+https*.

**+http-plain**[=value], **+nohttp-plain**

Similar to *+https*, except that HTTP queries will be sent over a non-encrypted channel. When this option is in use, the port number defaults to 80 and the HTTP request mode is POST.

**+http-plain-get**[=value], **+nohttp-plain-get**

Similar to *+http-plain*, except that the HTTP request mode is GET.

**+http-plain-post**[=value], **+nohttp-plain-post**

Same as *+http-plain*.

**+identify, +noidentify**

This option shows [or does not show] the IP address and port number that supplied the answer, when the *+short* option is enabled. If short form answers are requested, the default is not to show the source address and port number of the server that provided the answer.

**+idn, +noidn**

Enable or disable IDN processing. By default IDN is enabled for input query names, and for display when the output is a terminal.

You can also turn off **dig**'s IDN processing by setting the `IDN_DISABLE` environment variable.

**+ignore, +noignore**

This option ignores [or does not ignore] truncation in UDP responses instead of retrying with TCP. By default, TCP retries are performed.

**+keepalive, +nokeepalive**

This option sends [or does not send] an EDNS Keepalive option.

**+keepopen, +nokeepopen**

This option keeps [or does not keep] the TCP socket open between queries, and reuses it rather than creating a new TCP socket for each lookup. The default is `+nokeepopen`.

**+multiline, +nomultiline**

This option prints [or does not print] records, like the SOA records, in a verbose multi-line format with human-readable comments. The default is to print each record on a single line to facilitate machine parsing of the **dig** output.

**+ndots**=D

This option sets the number of dots (`D`) that must appear in `name` for it to be considered absolute. The default value is that defined using the `ndots` statement in `/etc/resolv.conf`, or 1 if no `ndots` statement is present. Names with fewer dots are interpreted as relative names, and are searched for in the domains listed in the `search` or `domain` directive in `/etc/resolv.conf` if *+search* is set.

**+nsid, +nonsid**

When enabled, this option includes an EDNS name server ID request when sending a query.

**+nssearch, +nonssearch**

When this option is set, **dig** attempts to find the authoritative name servers for the zone containing the name being looked up, and display the SOA record that each name server has for the zone. Addresses of servers that did not respond are also printed.

**+onesoa, +noonesoa**

When enabled, this option prints only one (starting) SOA record when performing an AXFR. The default is to print both the starting and ending SOA records.

**+opcode**=value, **+noopcode**

When enabled, this option sets (restores) the DNS message opcode to the specified value. The default value is QUERY (0).

**+padding**=value

This option pads the size of the query packet using the EDNS Padding option to blocks of value bytes. For example, +padding=32 causes a 48-byte query to be padded to 64 bytes. The default block size is 0, which disables padding; the maximum is 512. Values are ordinarily expected to be powers of two, such as 128; however, this is not mandatory. Responses to padded queries may also be padded, but only if the query uses TCP or DNS COOKIE.

**+proxy**[=src_addr[#src_port]-dst_addr[#dst_port]], **+noproxy**

When this option is set, **dig** adds PROXYv2 headers to the queries. When source and destination addresses are specified, the headers contain them and use the PROXY command. It means for the remote peer that the queries were sent on behalf of another node and that the PROXYv2 header reflects the original connection endpoints. The default source port is 0 and destination port is *53*.

For encrypted DNS transports, to prevent accidental information leakage, encryption is applied to the PROXYv2 headers: the headers are sent right after the handshake process has been completed.

For plain DNS transports, no encryption is applied to the PROXYv2 headers.

If the addressees are omitted, PROXYv2 headers, that use the LOCAL command set, are added instead. For the remote peer, that means that the queries were sent on purpose without being relayed, so the real connection endpoint addresses must be used.

**+proxy-plain**[=src_addr[#src_port]-dst_addr[#dst_port], **+noproxy-plain**

The same as +[no]proxy, but instructs dig to send PROXYv2 headers ahead of any encryption, before any handshake messages are sent. That makes **dig** behave exactly how it is described in the PROXY protocol specification, but not all software expects such behaviour.

Please consult the software documentation to find out if you need this option. (for example, dnsdist expects encrypted PROXYv2 headers sent over TLS when encryption is used, while HAProxy and many other software packages expect plain ones).

For plain DNS transports the option is effectively an alias for the +[no]proxy described above.

**+qid**=value

This option specifies the query ID to use when sending queries.

**+qr, +noqr**

This option toggles the display of the query message as it is sent. By default, the query is not printed.

**+question, +noquestion**

This option toggles the display of the question section of a query when an answer is returned. The default is to print the question section as a comment.

**+raflag, +noraflag**

This option sets [or does not set] the RA (Recursion Available) bit in the query. The default is +noraflag. This bit is ignored by the server for QUERY.

**+rdflag, +nordflag**

This option is a synonym for *+recurse*, *+norecurse*.

**+recurse, +norecurse**

This option toggles the setting of the RD (recursion desired) bit in the query. This bit is set by default, which means **dig** normally sends recursive queries. Recursion is automatically disabled when the *+nssearch* or *+trace* query option is used.

**+retry**=T

This option sets the number of times to retry UDP and TCP queries to server to T instead of the default, 2. Unlike *+tries*, this does not include the initial query.

**+rrcomments, +norrcomments**

This option toggles the display of per-record comments in the output (for example, human-readable key information about DNSKEY records). The default is not to print record comments unless multiline mode is active.

**+search, +nosearch**

This option uses [or does not use] the search list defined by the searchlist or domain directive in `resolv.conf`, if any. The search list is not used by default.

`ndots` from `resolv.conf` (default 1), which may be overridden by *+ndots*, determines whether the name is treated as relative and hence whether a search is eventually performed.

**+short, +noshort**

This option toggles whether a terse answer is provided. The default is to print the answer in a verbose form. This option always has a global effect; it cannot be set globally and then overridden on a per-lookup basis.

**+showbadcookie, +noshowbadcookie**

This option toggles whether to show the message containing the BADCOOKIE rcode before retrying the request or not. The default is to not show the messages.

**+showbadvers, +noshowbadvers**

This option toggles whether to show the message containing the BADVERS rcode before retrying the request or not. The default is to not show the messages.

**+showsearch, +noshowsearch**

This option performs [or does not perform] a search showing intermediate results.

**+split**=W

This option splits long hex- or base64-formatted fields in resource records into chunks of W characters (where W is rounded up to the nearest multiple of 4). `+nosplit` or `+split=0` causes fields not to be split at all. The default is 56 characters, or 44 characters when multiline mode is active.

**+stats, +nostats**

This option toggles the printing of statistics: when the query was made, the size of the reply, etc. The default behavior is to print the query statistics as a comment after each lookup.

**+subnet**=addr[/prefix-length], **+nosubnet**

This option sends [or does not send] an EDNS CLIENT-SUBNET option with the specified IP address or network prefix.

`dig +subnet=0.0.0.0/0`, or simply `dig +subnet=0` for short, sends an EDNS CLIENT-SUBNET option with an empty address and a source prefix-length of zero, which signals a resolver that the client's address information must *not* be used when resolving this query.

**+tcflag, +notcflag**

This option sets [or does not set] the TC (TrunCation) bit in the query. The default is `+notcflag`. This bit is ignored by the server for QUERY.

**+tcp, +notcp**

This option indicates whether to use TCP when querying name servers. The default behavior is to use UDP unless a type `any` or `ixfr=N` query is requested, in which case the default is TCP. AXFR queries always use TCP. To prevent retry over TCP when TC=1 is returned from a UDP query, use `+ignore`.

**+timeout**=T

This option sets the timeout for a query to `T` seconds. The default timeout is 5 seconds. An attempt to set `T` to less than 1 is silently set to 1.

**+tls, +notls**

This option indicates whether to use DNS over TLS (DoT) when querying name servers. When this option is in use, the port number defaults to 853.

**+tls-ca**[=file-name], **+notls-ca**

This option enables remote server TLS certificate validation for DNS transports, relying on TLS. Certificate authorities certificates are loaded from the specified PEM file (`file-name`). If the file is not specified, the default certificates from the global certificates store are used.

**+tls-certfile**=file-name, **+tls-keyfile**=file-name, **+notls-certfile, +notls-keyfile**

These options set the state of certificate-based client authentication for DNS transports, relying on TLS. Both certificate chain file and private key file are expected to be in PEM format. Both options must be specified at the same time.

**+tls-hostname**=hostname, **+notls-hostname**

This option makes **dig** use the provided hostname during remote server TLS certificate verification. Otherwise, the DNS server name is used. This option has no effect if *+tls-ca* is not specified.

**+trace, +notrace**

This option toggles tracing of the delegation path from the root name servers for the name being looked up. Tracing is disabled by default. When tracing is enabled, **dig** makes iterative queries to resolve the name being looked up. It follows referrals from the root servers, showing the answer from each server that was used to resolve the lookup.

If `@server` is also specified, it affects only the initial query for the root zone name servers.

*+dnssec* is set when *+trace* is set, to better emulate the default queries from a name server.

Note that the `delv +ns` option can also be used for tracing the resolution of a name from the root (see *delv*).

**+tries**=T

This option sets the number of times to try UDP and TCP queries to server to `T` instead of the default, 3. If `T` is less than or equal to zero, the number of tries is silently rounded up to 1.

**+ttlid, +nottlid**

This option displays [or does not display] the TTL when printing the record.

**+ttlunits, +nottlunits**

This option displays [or does not display] the TTL in friendly human-readable time units of `s`, `m`, `h`, `d`, and `w`, representing seconds, minutes, hours, days, and weeks. This implies *+ttlid*.

**+unknownformat, +nounknownformat**

This option prints all RDATA in unknown RR type presentation format (**RFC 3597**). The default is to print RDATA for known types in the type's presentation format.

**+vc, +novc**

This option uses [or does not use] TCP when querying name servers. This alternate syntax to *+tcp* is provided for backwards compatibility. The `vc` stands for "virtual circuit."

**+yaml, +noyaml**

When enabled, this option prints the responses (and, if *+qr* is in use, also the outgoing queries) in a detailed YAML format.

**+zflag, +nozflag**

This option sets [or does not set] the last unassigned DNS header flag in a DNS query. This flag is off by default.

### 12.4.6 Multiple Queries

The BIND 9 implementation of **dig** supports specifying multiple queries on the command line (in addition to supporting the *-f* batch file option). Each of those queries can be supplied with its own set of flags, options, and query options.

In this case, each `query` argument represents an individual query in the command-line syntax described above. Each consists of any of the standard options and flags, the name to be looked up, an optional query type and class, and any query options that should be applied to that query.

A global set of query options, which should be applied to all queries, can also be supplied. These global query options must precede the first tuple of name, class, type, options, flags, and query options supplied on the command line. Any global query options (except *+cmd* and *+short* options) can be overridden by a query-specific set of query options. For example:

```
dig +qr www.isc.org any -x 127.0.0.1 isc.org ns +noqr
```

shows how **dig** can be used from the command line to make three lookups: an ANY query for `www.isc.org`, a reverse lookup of 127.0.0.1, and a query for the NS records of `isc.org`. A global query option of *+qr* is applied, so that **dig** shows the initial query it made for each lookup. The final query has a local query option of *+noqr* which means that **dig** does not print the initial query when it looks up the NS records for `isc.org`.

### 12.4.7 Return Codes

**dig** return codes are:

**0**

DNS response received, including NXDOMAIN status

**1**

Usage error

**8**

Couldn't open batch file

**9**

No reply from server

**10**

Internal error

### 12.4.8 Files

`/etc/resolv.conf`

`${HOME}/.digrc`

### 12.4.9 Examples

Only display the IP address(es) for example.com:

```
dig +short example.com
```

Query the nameserver f.gtld-servers.net for example.com:

```
dig @f.gtld-servers.net example.com
```

Look up the TXT record for example.com:

```
dig txt example.com
```

Look up the hostname for an IP with reverse DNS:

```
dig -x 192.0.2.1
```

Display a much shorter output with just the name, record type, TTL, and value for each answer:

```
dig +noall +answer example.com
```

### 12.4.10 See Also

*delv(1)*, *host(1)*, *named(8)*, *dnssec-keygen(8)*, **RFC 1035**.

### 12.4.11 Bugs

There are probably too many query options.

## 12.5 dnssec-cds - change DS records for a child zone based on CDS/CDNSKEY

### 12.5.1 Synopsis

**dnssec-cds** [**-a** alg...] [**-c** class] [**-D**] {**-d** dsset-file} {**-f** child-file} [**-i**[extension]] [-s** start-time] [**-T** ttl] [**-u**] [**-v** level] [**-V**] {domain}

### 12.5.2 Description

The **dnssec-cds** command changes DS records at a delegation point based on CDS or CDNSKEY records published in the child zone. If both CDS and CDNSKEY records are present in the child zone, the CDS is preferred. This enables a child zone to inform its parent of upcoming changes to its key-signing keys (KSKs); by polling periodically with **dnssec-cds**, the parent can keep the DS records up-to-date and enable automatic rolling of KSKs.

Two input files are required. The *-f child-file* option specifies a file containing the child's CDS and/or CDNSKEY records, plus RRSIG and DNSKEY records so that they can be authenticated. The *-d path* option specifies the location of a file containing the current DS records. For example, this could be a `dsset-` file generated by *dnssec-signzone*, or the output of *dnssec-dsfromkey*, or the output of a previous run of **dnssec-cds**.

The **dnssec-cds** command uses special DNSSEC validation logic specified by **RFC 7344**. It requires that the CDS and/or CDNSKEY records be validly signed by a key represented in the existing DS records. This is typically the pre-existing KSK.

For protection against replay attacks, the signatures on the child records must not be older than they were on a previous run of **dnssec-cds**. Their age is obtained from the modification time of the `dsset-` file, or from the *-s* option.

To protect against breaking the delegation, **dnssec-cds** ensures that the DNSKEY RRset can be verified by every key algorithm in the new DS RRset, and that the same set of keys are covered by every DS digest type.

By default, replacement DS records are written to the standard output; with the *-i* option the input file is overwritten in place. The replacement DS records are the same as the existing records, when no change is required. The output can be empty if the CDS/CDNSKEY records specify that the child zone wants to be insecure.

> **Warning**
>
> Be careful not to delete the DS records when **dnssec-cds** fails!

Alternatively, :option`dnssec-cds -u` writes an *nsupdate* script to the standard output. The *-u* and *-i* options can be used together to maintain a `dsset-` file as well as emit an *nsupdate* script.

### 12.5.3 Options

**-a** `algorithm`

When converting CDS records to DS records, this option specifies the acceptable digest algorithms. This option can be repeated, so that multiple digest types are allowed. If none of the CDS records use an acceptable digest type, **dnssec-cds** will try to use CDNSKEY records instead; if there are no CDNSKEY records, it reports an error.

When converting CDNSKEY records to DS records, this option specifies the digest algorithm to use. It can be repeated, so that multiple DS records are created for each CDNSKEY records.

The algorithm must be one of SHA-1, SHA-256, or SHA-384. These values are case-insensitive, and the hyphen may be omitted. If no algorithm is specified, the default is SHA-256 only.

**-c** `class`

This option specifies the DNS class of the zones.

**-D**

This option generates DS records from CDNSKEY records if both CDS and CDNSKEY records are present in the child zone. By default CDS records are preferred.

**-d** `path`

This specifies the location of the parent DS records. The path can be the name of a file containing the DS records; if it is a directory, **dnssec-cds** looks for a `dsset-` file for the domain inside the directory.

To protect against replay attacks, child records are rejected if they were signed earlier than the modification time of the `dsset-` file. This can be adjusted with the *-s* option.

**-f** `child-file`

This option specifies the file containing the child's CDS and/or CDNSKEY records, plus its DNSKEY records and the covering RRSIG records, so that they can be authenticated.

The examples below describe how to generate this file.

**-i** `extension`

This option updates the `dsset-` file in place, instead of writing DS records to the standard output.

There must be no space between the *-i* and the extension. If no extension is provided, the old `dsset-` is discarded. If an extension is present, a backup of the old `dsset-` file is kept with the extension appended to its filename.

To protect against replay attacks, the modification time of the `dsset-` file is set to match the signature inception time of the child records, provided that it is later than the file's current modification time.

**-s** `start-time`

This option specifies the date and time after which RRSIG records become acceptable. This can be either an absolute or a relative time. An absolute start time is indicated by a number in YYYYMMDDHHMMSS notation;

20170827133700 denotes 13:37:00 UTC on August 27th, 2017. A time relative to the `dsset-` file is indicated with `-N`, which is N seconds before the file modification time. A time relative to the current time is indicated with `now+N`.

If no start-time is specified, the modification time of the `dsset-` file is used.

**-T** `ttl`

This option specifies a TTL to be used for new DS records. If not specified, the default is the TTL of the old DS records. If they had no explicit TTL, the new DS records also have no explicit TTL.

**-u**

This option writes an *nsupdate* script to the standard output, instead of printing the new DS reords. The output is empty if no change is needed.

Note: The TTL of new records needs to be specified: it can be done in the original `dsset-` file, with the *-T* option, or using the *nsupdate* `ttl` command.

**-V**

This option prints version information.

**-v** `level`

This option sets the debugging level. Level 1 is intended to be usefully verbose for general users; higher levels are intended for developers.

**domain**

This indicates the name of the delegation point/child zone apex.

### 12.5.4 Exit Status

The **dnssec-cds** command exits 0 on success, or non-zero if an error occurred.

If successful, the DS records may or may not need to be changed.

### 12.5.5 Examples

Before running *dnssec-signzone*, ensure that the delegations are up-to-date by running **dnssec-cds** on every `dsset-` file.

To fetch the child records required by **dnssec-cds**, invoke *dig* as in the script below. It is acceptable if the *dig* fails, since **dnssec-cds** performs all the necessary checking.

```
for f in dsset-*
do
    d=${f#dsset-}
    dig +dnssec +noall +answer $d DNSKEY $d CDNSKEY $d CDS |
    dnssec-cds -i -f /dev/stdin -d $f $d
done
```

When the parent zone is automatically signed by *named*, **dnssec-cds** can be used with *nsupdate* to maintain a delegation as follows. The `dsset-` file allows the script to avoid having to fetch and validate the parent DS records, and it maintains the replay attack protection time.

```
dig +dnssec +noall +answer $d DNSKEY $d CDNSKEY $d CDS |
dnssec-cds -u -i -f /dev/stdin -d $f $d |
nsupdate -l
```

### 12.5.6 See Also

*dig(1)*, *dnssec-settime(8)*, *dnssec-signzone(8)*, *nsupdate(1)*, BIND 9 Administrator Reference Manual, **RFC 7344**.

## 12.6 dnssec-dsfromkey - DNSSEC DS RR generation tool

### 12.6.1 Synopsis

**dnssec-dsfromkey** [ **-1** | **-2** | **-a** alg ] [ **-C** ] [**-T** TTL] [**-v** level] [**-K** directory] {keyfile}

**dnssec-dsfromkey** [ **-1** | **-2** | **-a** alg ] [ **-C** ] [**-T** TTL] [**-v** level] [**-c** class] [**-A**] {**-f** file} [dnsname]

**dnssec-dsfromkey** [ **-1** | **-2** | **-a** alg ] [ **-C** ] [**-T** TTL] [**-v** level] [**-c** class] [**-K** directory] {**-s**} {dnsname}

**dnssec-dsfromkey** [ **-h** | **-V** ]

### 12.6.2 Description

The **dnssec-dsfromkey** command outputs DS (Delegation Signer) resource records (RRs), or CDS (Child DS) RRs with the *-C* option.

By default, only KSKs are converted (keys with flags = 257). The *-A* option includes ZSKs (flags = 256). Revoked keys are never included.

The input keys can be specified in a number of ways:

By default, **dnssec-dsfromkey** reads a key file named in the format `Knnnn.+aaa+iiiii.key`, as generated by *dnssec-keygen*.

With the *-f file* option, **dnssec-dsfromkey** reads keys from a zone file or partial zone file (which can contain just the DNSKEY records).

With the *-s* option, **dnssec-dsfromkey** reads a `keyset-` file, as generated by *dnssec-keygen -C*.

### 12.6.3 Options

**-1**

This option is an abbreviation for *-a SHA1*. This digest is deprecated.

**-2**

This option is an abbreviation for *-a SHA-256*.

**-a** `algorithm`

This option specifies a digest algorithm to use when converting DNSKEY records to DS records. This option can be repeated, so that multiple DS records are created for each DNSKEY record.

The algorithm must be one of SHA-1 (deprecated), SHA-256, or SHA-384. These values are case-insensitive, and the hyphen may be omitted. If no algorithm is specified, the default is SHA-256.

**-A**

This option indicates that ZSKs are to be included when generating DS records. Without this option, only keys which have the KSK flag set are converted to DS records and printed. This option is only useful in *-f* zone file mode.

**-c** `class`

This option specifies the DNS class; the default is IN. This option is only useful in *-s* keyset or *-f* zone file mode.

**-C**

This option generates CDS records rather than DS records.

**-f** file

This option sets zone file mode, in which the final dnsname argument of **dnssec-dsfromkey** is the DNS domain name of a zone whose master file can be read from `file`. If the zone name is the same as `file`, then it may be omitted.

If `file` is `-`, then the zone data is read from the standard input. This makes it possible to use the output of the *dig* command as input, as in:

```
dig dnskey example.com | dnssec-dsfromkey -f - example.com
```

**-h**

This option prints usage information.

**-K** directory

This option tells BIND 9 to look for key files or `keyset-` files in `directory`.

**-s**

This option enables keyset mode, in which the final dnsname argument from **dnssec-dsfromkey** is the DNS domain name used to locate a `keyset-` file.

**-T** TTL

This option specifies the TTL of the DS records. By default the TTL is omitted.

**-v** level

This option sets the debugging level.

**-V**

This option prints version information.

### 12.6.4 Example

To build the SHA-256 DS RR from the `Kexample.com.+003+26160` keyfile, issue the following command:

```
dnssec-dsfromkey -2 Kexample.com.+003+26160
```

The command returns something similar to:

```
example.com. IN DS 26160 5 2 3A1EADA7A74B8D0BA86726B0C227AA85AB8BBD2B2004F41A868A54F0C5EA0B94
```

### 12.6.5 Files

The keyfile can be designated by the key identification `Knnnn.+aaa+iiiii` or the full file name `Knnnn.+aaa+iiiii.key`, as generated by *dnssec-keygen*.

The keyset file name is built from the `directory`, the string `keyset-`, and the `dnsname`.

### 12.6.6 Caveat

A keyfile error may return "file not found," even if the file exists.

### 12.6.7 See Also

*dnssec-keygen(8)*, *dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, **RFC 3658** (DS RRs), **RFC 4509** (SHA-256 for DS RRs), **RFC 6605** (SHA-384 for DS RRs), **RFC 7344** (CDS and CDNSKEY RRs).

## 12.7 dnssec-importkey - import DNSKEY records from external systems so they can be managed

### 12.7.1 Synopsis

**dnssec-importkey** [**-K** directory] [**-L** ttl] [**-P** date/offset] [**-P** sync date/offset] [**-D** date/offset] [**-D** sync date/offset] [**-h**] [**-v** level] [**-V**] {keyfile}

**dnssec-importkey** {**-f** filename} [**-K** directory] [**-L** ttl] [**-P** date/offset] [**-P** sync date/offset] [**-D** date/offset] [**-D** sync date/offset] [**-h**] [**-v** level] [**-V**] [dnsname]

### 12.7.2 Description

**dnssec-importkey** reads a public DNSKEY record and generates a pair of .key/.private files. The DNSKEY record may be read from an existing .key file, in which case a corresponding .private file is generated, or it may be read from any other file or from the standard input, in which case both .key and .private files are generated.

The newly created .private file does *not* contain private key data, and cannot be used for signing. However, having a .private file makes it possible to set publication (`-P`) and deletion (`-D`) times for the key, which means the public key can be added to and removed from the DNSKEY RRset on schedule even if the true private key is stored offline.

When using `dnssec-policy`, do not use **dnssec-importkey** to import key files that cannot be used for signing. In this case, simply publish the imported DNSKEY record in the zone, and make sure that the files are outside the configured `key-directory`.

### 12.7.3 Options

**-f** `filename`

This option indicates the zone file mode. Instead of a public keyfile name, the argument is the DNS domain name of a zone master file, which can be read from `filename`. If the domain name is the same as `filename`, then it may be omitted.

If `filename` is set to `"-"`, then the zone data is read from the standard input.

**-K** `directory`

This option sets the directory in which the key files are to reside.

**-L** `ttl`

This option sets the default TTL to use for this key when it is converted into a DNSKEY RR. This is the TTL used when the key is imported into a zone, unless there was already a DNSKEY RRset in place, in which case the existing TTL takes precedence. Setting the default TTL to `0` or `none` removes it from the key.

**-h**

This option emits a usage message and exits.

**-v** `level`

This option sets the debugging level.

**-V**

This option prints version information.

### 12.7.4 Timing Options

Dates can be expressed in the format YYYYMMDD or YYYYMMDDHHMMSS. (which is the format used inside key files), or 'Day Mon DD HH:MM:SS YYYY' (as printed by `dnssec-settime -p`), or UNIX epoch time (as printed by `dnssec-settime -up`), or the literal `now`.

The argument can be followed by + or – and an offset from the given time. The literal `now` can be omitted before an offset. The offset can be followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, so that it is computed in years (defined as 365 24-hour days, ignoring leap years), months (defined as 30 24-hour days), weeks, days, hours, or minutes, respectively. Without a suffix, the offset is computed in seconds.

To explicitly prevent a date from being set, use `none`, `never`, or `unset`.

All these formats are case-insensitive.

**-P** `date/offset`

This option sets the date on which a key is to be published to the zone. After that date, the key is included in the zone but is not used to sign it.

**sync** `date/offset`

This option sets the date on which CDS and CDNSKEY records that match this key are to be published to the zone.

**-D** `date/offset`

This option sets the date on which the key is to be deleted. After that date, the key is no longer included in the zone. (However, it may remain in the key repository.)

**sync** `date/offset`

This option sets the date on which the CDS and CDNSKEY records that match this key are to be deleted.

### 12.7.5 Files

A keyfile can be designed by the key identification `Knnnn.+aaa+iiiii` or the full file name `Knnnn.+aaa+iiiii.key`, as generated by *dnssec-keygen*.

### 12.7.6 See Also

*dnssec-keygen(8)*, *dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, **RFC 5011**.

## 12.8 dnssec-keyfromlabel - DNSSEC key generation tool

### 12.8.1 Synopsis

**dnssec-keyfromlabel** {**-l** label} [**-3**] [**-a** algorithm] [**-A** date/offset] [**-c** class] [**-D** date/offset] [**-D** sync date/offset] [**-E** engine] [**-f** flag] [**-G**] [**-I** date/offset] [**-i** interval] [**-k**] [**-K** directory] [**-L** ttl] [**-M** tag_min:tag_max] [**-n** nametype] [**-P** date/offset] [**-P** sync date/offset] [**-p** protocol] [**-R** date/offset] [**-S** key] [**-t** type] [**-v** level] [**-V**] [**-y**] {name}

### 12.8.2 Description

**dnssec-keyfromlabel** generates a pair of key files that reference a key object stored in a cryptographic hardware service module (HSM). The private key file can be used for DNSSEC signing of zone data as if it were a conventional signing key created by *dnssec-keygen*, but the key material is stored within the HSM and the actual signing takes place there.

The `name` of the key is specified on the command line. This must match the name of the zone for which the key is being generated.

### 12.8.3 Options

**-a** algorithm

This option selects the cryptographic algorithm. The value of algorithm must be one of RSASHA1 (deprecated), NSEC3RSASHA1 (deprecated), RSASHA256, RSASHA512, ECDSAP256SHA256, ECDSAP384SHA384, ED25519, or ED448.

These values are case-insensitive. In some cases, abbreviations are supported, such as ECDSA256 for ECDSAP256SHA256 and ECDSA384 for ECDSAP384SHA384. If RSASHA1 (deprecated) is specified along with the *-3* option, then NSEC3RSASHA1 (deprecated) is used instead.

This option is mandatory except when using the *-S* option, which copies the algorithm from the predecessory key.

Changed in version 9.12.0: The default value RSASHA1 (deprecated) for newly generated keys was removed.

**-3**

This option uses an NSEC3-capable algorithm to generate a DNSSEC key. If this option is used with an algorithm that has both NSEC and NSEC3 versions, then the NSEC3 version is used; for example, `dnssec-keygen -3a RSASHA1` specifies the NSEC3RSASHA1 (deprecated) algorithm.

**-E** engine

This option specifies the cryptographic hardware to use.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-l** label

This option specifies the label for a key pair in the crypto hardware.

When BIND 9 is built with OpenSSL-based PKCS#11 support, the label is an arbitrary string that identifies a particular key. It may be preceded by an optional OpenSSL engine name, followed by a colon, as in `pkcs11:keylabel`.

**-n** nametype

This option specifies the owner type of the key. The value of nametype must either be ZONE (for a DNSSEC zone key (KEY/DNSKEY)), HOST or ENTITY (for a key associated with a host (KEY)), USER (for a key associated with a user (KEY)), or OTHER (DNSKEY). These values are case-insensitive.

**-C**

This option enables compatibility mode, which generates an old-style key, without any metadata. By default, **dnssec-keyfromlabel** includes the key's creation date in the metadata stored with the private key; other dates may be set there as well, including publication date, activation date, etc. Keys that include this data may be incompatible with older versions of BIND; the *-C* option suppresses them.

**-c** class

This option indicates that the DNS record containing the key should have the specified class. If not specified, class IN is used.

**-f** flag

This option sets the specified flag in the flag field of the KEY/DNSKEY record. The only recognized flags are KSK (Key-Signing Key) and REVOKE.

**-G**

This option generates a key, but does not publish it or sign with it. This option is incompatible with *-P* and *-A*.

**-h**

This option prints a short summary of the options and arguments to **dnssec-keyfromlabel**.

**-K** directory

This option sets the directory in which the key files are to be written.

**-k**

This option generates KEY records rather than DNSKEY records.

**-L** `ttl`

This option sets the default TTL to use for this key when it is converted into a DNSKEY RR. This is the TTL used when the key is imported into a zone, unless there was already a DNSKEY RRset in place, in which case the existing TTL would take precedence. Setting the default TTL to `0` or `none` removes it.

**-M** `tag_min:tag_max`

This option sets the range of key tag values that `dnssec-keyfromlabel` will accept. If the key tag of the new key or the key tag of the revoked version of the new key is outside this range, the new key will be rejected. This is designed to be used when generating keys in a multi-signer scenario, where each operator is given a range of key tags to prevent collisions among different operators. The valid values for `tag_min` and `tag_max` are [0..65535]. The default allows all key tag values to be accepted.

**-p** `protocol`

This option sets the protocol value for the key. The protocol is a number between 0 and 255. The default is 3 (DNSSEC). Other possible values for this argument are listed in **RFC 2535** and its successors.

**-S** `key`

This option generates a key as an explicit successor to an existing key. The name, algorithm, size, and type of the key are set to match the predecessor. The activation date of the new key is set to the inactivation date of the existing one. The publication date is set to the activation date minus the prepublication interval, which defaults to 30 days.

**-t** `type`

This option indicates the type of the key. `type` must be one of AUTHCONF, NOAUTHCONF, NOAUTH, or NOCONF. The default is AUTHCONF. AUTH refers to the ability to authenticate data, and CONF to the ability to encrypt data.

**-v** `level`

This option sets the debugging level.

**-V**

This option prints version information.

**-y**

This option allows DNSSEC key files to be generated even if the key ID would collide with that of an existing key, in the event of either key being revoked. (This is only safe to enable if **RFC 5011** trust anchor maintenance is not used with either of the keys involved.)

### 12.8.4 Timing Options

Dates can be expressed in the format YYYYMMDD or YYYYMMDDHHMMSS (which is the format used inside key files), or 'Day Mon DD HH:MM:SS YYYY' (as printed by `dnssec-settime -p`), or UNIX epoch time (as printed by `dnssec-settime -up`), or the literal `now`.

The argument can be followed by `+` or `-` and an offset from the given time. The literal `now` can be omitted before an offset. The offset can be followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, so that it is computed in years (defined as 365 24-hour days, ignoring leap years), months (defined as 30 24-hour days), weeks, days, hours, or minutes, respectively. Without a suffix, the offset is computed in seconds.

To explicitly prevent a date from being set, use `none`, `never`, or `unset`.

All these formats are case-insensitive.

**-P** date/offset

This option sets the date on which a key is to be published to the zone. After that date, the key is included in the zone but is not used to sign it. If not set, and if the *-G* option has not been used, the default is the current date.

**sync** date/offset

This option sets the date on which CDS and CDNSKEY records that match this key are to be published to the zone.

**-A** date/offset

This option sets the date on which the key is to be activated. After that date, the key is included in the zone and used to sign it. If not set, and if the *-G* option has not been used, the default is the current date.

**-R** date/offset

This option sets the date on which the key is to be revoked. After that date, the key is flagged as revoked. It is included in the zone and is used to sign it.

**-I** date/offset

This option sets the date on which the key is to be retired. After that date, the key is still included in the zone, but it is not used to sign it.

**-D** date/offset

This option sets the date on which the key is to be deleted. After that date, the key is no longer included in the zone. (However, it may remain in the key repository.)

**sync** date/offset

This option sets the date on which the CDS and CDNSKEY records that match this key are to be deleted.

**-i** interval

This option sets the prepublication interval for a key. If set, then the publication and activation dates must be separated by at least this much time. If the activation date is specified but the publication date is not, the publication date defaults to this much time before the activation date; conversely, if the publication date is specified but not the activation date, activation is set to this much time after publication.

If the key is being created as an explicit successor to another key, then the default prepublication interval is 30 days; otherwise it is zero.

As with date offsets, if the argument is followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, the interval is measured in years, months, weeks, days, hours, or minutes, respectively. Without a suffix, the interval is measured in seconds.

### 12.8.5 Generated Key Files

When **dnssec-keyfromlabel** completes successfully, it prints a string of the form `Knnnn.+aaa+iiiii` to the standard output. This is an identification string for the key files it has generated.

- `nnnn` is the key name.
- `aaa` is the numeric representation of the algorithm.
- `iiiii` is the key identifier (or footprint).

**dnssec-keyfromlabel** creates two files, with names based on the printed string. `Knnnn.+aaa+iiiii.key` contains the public key, and `Knnnn.+aaa+iiiii.private` contains the private key.

The `.key` file contains a DNS KEY record that can be inserted into a zone file (directly or with an $INCLUDE statement).

The `.private` file contains algorithm-specific fields. For obvious security reasons, this file does not have general read permission.

### 12.8.6 See Also

*dnssec-keygen(8)*, *dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, **RFC 4034**, **RFC 7512**.

## 12.9 dnssec-keygen: DNSSEC key generation tool

### 12.9.1 Synopsis

**dnssec-keygen** [**-3**] [**-A** date/offset] [**-a** algorithm] [**-b** keysize] [**-C**] [**-c** class] [**-D** date/offset] [**-d** bits] [**-D** sync date/offset] [**-E** engine] [**-f** flag] [**-F**] [**-G**] [**-h**] [**-I** date/offset] [**-i** interval] [**-K** directory] [**-k** policy] [**-L** ttl] [**-l** file] [**-n** nametype] [**-M** tag_min:tag_max] [**-P** date/offset] [**-P** sync date/offset] [**-p** protocol] [**-q**] [**-R** date/offset] [**-S** key] [**-s** strength] [**-T** rrtype] [**-t** type] [**-V**] [**-v** level] {name}

### 12.9.2 Description

**dnssec-keygen** generates keys for DNSSEC (Secure DNS), as defined in **RFC 2535** and **RFC 4034**.

The `name` of the key is specified on the command line. For DNSSEC keys, this must match the name of the zone for which the key is being generated.

### 12.9.3 Options

**-3**

This option uses an NSEC3-capable algorithm to generate a DNSSEC key. If this option is used with an algorithm that has both NSEC and NSEC3 versions, then the NSEC3 version is selected; for example, `dnssec-keygen -3 -a RSASHA1` specifies the NSEC3RSASHA1 (deprecated) algorithm.

**-a** `algorithm`

This option selects the cryptographic algorithm. For DNSSEC keys, the value of `algorithm` must be one of RSASHA1 (deprecated), NSEC3RSASHA1 (deprecated), RSASHA256, RSASHA512, ECDSAP256SHA256, ECDSAP384SHA384, ED25519, or ED448.

These values are case-insensitive. In some cases, abbreviations are supported, such as ECDSA256 for ECDSAP256SHA256 and ECDSA384 for ECDSAP384SHA384. If RSASHA1 (deprecated) is specified along with the *-3* option, NSEC3RSASHA1 (deprecated) is used instead.

This parameter *must* be specified except when using the *-S* option, which copies the algorithm from the predecessor key.

In prior releases, HMAC algorithms could be generated for use as TSIG keys, but that feature was removed in BIND 9.13.0. Use *tsig-keygen* to generate TSIG keys.

**-b** `keysize`

This option specifies the number of bits in the key. The choice of key size depends on the algorithm used: RSA keys must be between 1024 and 4096 bits; Diffie-Hellman keys must be between 128 and 4096 bits. Elliptic curve algorithms do not need this parameter.

If the key size is not specified, some algorithms have pre-defined defaults. For example, RSA keys for use as DNSSEC zone-signing keys have a default size of 1024 bits; RSA keys for use as key-signing keys (KSKs, generated with *-f KSK*) default to 2048 bits.

**-C**

This option enables compatibility mode, which generates an old-style key, without any timing metadata. By default, **dnssec-keygen** includes the key's creation date in the metadata stored with the private key; other dates may be set there as well, including publication date, activation date, etc. Keys that include this data may be incompatible with older versions of BIND; the *-C* option suppresses them.

**-c** class

This option indicates that the DNS record containing the key should have the specified class. If not specified, class IN is used.

**-d** bits

This option specifies the key size in bits. For the algorithms RSASHA1, NSEC3RSASA1, RSASHA256, and RSASHA512 the key size must be between 1024 and 4096 bits; DH size is between 128 and 4096 bits. This option is ignored for algorithms ECDSAP256SHA256, ECDSAP384SHA384, ED25519, and ED448.

**-E** engine

This option specifies the cryptographic hardware to use, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually pkcs11).

**-f** flag

This option sets the specified flag in the flag field of the KEY/DNSKEY record. The only recognized flags are ZSK (Zone-Signing Key), KSK (Key-Signing Key) and REVOKE.

Note that ZSK is not a physical flag in the DNSKEY record, it is merely used to explicitly tell that you want to create a ZSK. Setting *-f* in conjunction with *-k* will result in generating keys that only match the given role set with this option.

**-F**

This options turns on FIPS (US Federal Information Processing Standards) mode if the underlying crytographic library supports running in FIPS mode.

**-G**

This option generates a key, but does not publish it or sign with it. This option is incompatible with *-P* and *-A*.

**-h**

This option prints a short summary of the options and arguments to **dnssec-keygen**.

**-K** directory

This option sets the directory in which the key files are to be written.

**-k** policy

This option creates keys for a specific dnssec-policy. If a policy uses multiple keys, **dnssec-keygen** generates multiple keys. This also creates a ".state" file to keep track of the key state.

This option creates keys according to the dnssec-policy configuration, hence it cannot be used at the same time as many of the other options that **dnssec-keygen** provides.

**-L** ttl

This option sets the default TTL to use for this key when it is converted into a DNSKEY RR. This is the TTL used when the key is imported into a zone, unless there was already a DNSKEY RRset in place, in which case the existing TTL takes precedence. If this value is not set and there is no existing DNSKEY RRset, the TTL defaults to the SOA TTL. Setting the default TTL to 0 or none is the same as leaving it unset.

**-l** file

This option provides a configuration file that contains a dnssec-policy statement (matching the policy set with *-k*).

**-M** tag_min:tag_max

This option sets the range of acceptable key tag values that dnssec-keygen will produce. If the key tag of the new key or the key tag of the revoked version of the new key is outside this range, the new key will be rejected and another new key will be generated. This is designed to be used when generating keys in a multi-signer scenario, where each operator is given a range of key tags to prevent collisions among different operators. The valid values

for `tag_min` and `tag_max` are [0..65535]. The default allows all key tag values to be produced. This option is ignored when `-k policy` is specified.

**-n** `nametype`

This option specifies the owner type of the key. The value of `nametype` must either be ZONE (for a DNSSEC zone key (KEY/DNSKEY)), HOST or ENTITY (for a key associated with a host (KEY)), USER (for a key associated with a user (KEY)), or OTHER (DNSKEY). These values are case-insensitive. The default is ZONE for DNSKEY generation.

**-p** `protocol`

This option sets the protocol value for the generated key, for use with *-T KEY*. The protocol is a number between 0 and 255. The default is 3 (DNSSEC). Other possible values for this argument are listed in **RFC 2535** and its successors.

**-q**

This option sets quiet mode, which suppresses unnecessary output, including progress indication. Without this option, when **dnssec-keygen** is run interactively to generate an RSA or DSA key pair, it prints a string of symbols to `stderr` indicating the progress of the key generation. A . indicates that a random number has been found which passed an initial sieve test; + means a number has passed a single round of the Miller-Rabin primality test; and a space ( ) means that the number has passed all the tests and is a satisfactory key.

**-S** `key`

This option creates a new key which is an explicit successor to an existing key. The name, algorithm, size, and type of the key are set to match the existing key. The activation date of the new key is set to the inactivation date of the existing one. The publication date is set to the activation date minus the prepublication interval, which defaults to 30 days.

**-s** `strength`

This option specifies the strength value of the key. The strength is a number between 0 and 15, and currently has no defined purpose in DNSSEC.

**-T** `rrtype`

This option specifies the resource record type to use for the key. `rrtype` must be either DNSKEY or KEY. The default is DNSKEY when using a DNSSEC algorithm, but it can be overridden to KEY for use with SIG(0).

**-t** `type`

This option indicates the type of the key for use with *-T KEY*. `type` must be one of AUTHCONF, NOAUTHCONF, NOAUTH, or NOCONF. The default is AUTHCONF. AUTH refers to the ability to authenticate data, and CONF to the ability to encrypt data.

**-V**

This option prints version information.

**-v** `level`

This option sets the debugging level.

### 12.9.4 Timing Options

Dates can be expressed in the format YYYYMMDD or YYYYMMDDHHMMSS (which is the format used inside key files), or 'Day Mon DD HH:MM:SS YYYY' (as printed by `dnssec-settime -p`), or UNIX epoch time (as printed by `dnssec-settime -up`), or the literal `now`.

The argument can be followed by + or – and an offset from the given time. The literal `now` can be omitted before an offset. The offset can be followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, so that it is computed in years (defined as 365 24-hour days, ignoring leap years), months (defined as 30 24-hour days), weeks, days, hours, or minutes, respectively. Without a suffix, the offset is computed in seconds.

To unset a date, use `none`, `never`, or `unset`.

**-P** date/offset

This option sets the date on which a key is to be published to the zone. After that date, the key is included in the zone but is not used to sign it. If not set, and if the *-G* option has not been used, the default is the current date.

**sync** date/offset

This option sets the date on which CDS and CDNSKEY records that match this key are to be published to the zone.

**-A** date/offset

This option sets the date on which the key is to be activated. After that date, the key is included in the zone and used to sign it. If not set, and if the *-G* option has not been used, the default is the current date. If set, and *-P* is not set, the publication date is set to the activation date minus the prepublication interval.

**-R** date/offset

This option sets the date on which the key is to be revoked. After that date, the key is flagged as revoked. It is included in the zone and is used to sign it.

**-I** date/offset

This option sets the date on which the key is to be retired. After that date, the key is still included in the zone, but it is not used to sign it.

**-D** date/offset

This option sets the date on which the key is to be deleted. After that date, the key is no longer included in the zone. (However, it may remain in the key repository.)

**sync** date/offset

This option sets the date on which the CDS and CDNSKEY records that match this key are to be deleted.

**-i** interval

This option sets the prepublication interval for a key. If set, then the publication and activation dates must be separated by at least this much time. If the activation date is specified but the publication date is not, the publication date defaults to this much time before the activation date; conversely, if the publication date is specified but not the activation date, activation is set to this much time after publication.

If the key is being created as an explicit successor to another key, then the default prepublication interval is 30 days; otherwise it is zero.

As with date offsets, if the argument is followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, the interval is measured in years, months, weeks, days, hours, or minutes, respectively. Without a suffix, the interval is measured in seconds.

### 12.9.5 Generated Keys

When **dnssec-keygen** completes successfully, it prints a string of the form `Knnnn.+aaa+iiiii` to the standard output. This is an identification string for the key it has generated.

- `nnnn` is the key name.
- `aaa` is the numeric representation of the algorithm.
- `iiiii` is the key identifier (or footprint).

**dnssec-keygen** creates two files, with names based on the printed string. `Knnnn.+aaa+iiiii.key` contains the public key, and `Knnnn.+aaa+iiiii.private` contains the private key.

The `.key` file contains a DNSKEY or KEY record. When a zone is being signed by *named* or *dnssec-signzone -S*, DNSKEY records are included automatically. In other cases, the `.key` file can be inserted into a zone file manually or with an `$INCLUDE` statement.

The `.private` file contains algorithm-specific fields. For obvious security reasons, this file does not have general read permission.

### 12.9.6 Example

To generate an ECDSAP256SHA256 zone-signing key for the zone `example.com`, issue the command:

```
dnssec-keygen -a ECDSAP256SHA256 example.com
```

The command prints a string of the form:

```
Kexample.com.+013+26160
```

In this example, **dnssec-keygen** creates the files `Kexample.com.+013+26160.key` and `Kexample.com.+013+26160.private`.

To generate a matching key-signing key, issue the command:

```
dnssec-keygen -a ECDSAP256SHA256 -f KSK example.com
```

### 12.9.7 See Also

*dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, RFC 2539, RFC 2845, RFC 4034.

## 12.10 dnssec-ksr - Create signed key response (SKR) files for offline KSK setups

### 12.10.1 Synopsis

**dnssec-ksr** [**-E** engine] [**-e** date/offset] [**-F**] [**-f** file] [**-h**] [**-i** date/offset] [**-K** directory] [**-k** policy] [**-l** file] [**-o**] [**-V**] [**-v** level] {command} {zone}

### 12.10.2 Description

The **dnssec-ksr** can be used to issue several commands that are needed to generate presigned RRsets for a zone where the private key file of the Key Signing Key (KSK) is typically offline. This requires Zone Signing Keys (ZSKs) to be pregenerated, and the DNSKEY, CDNSKEY, and CDS RRsets to be already signed in advance.

The latter is done by creating Key Signing Requests (KSRs) that can be imported to the environment where the KSK is available. Once there, this program can create Signed Key Responses (SKRs) that can be loaded by an authoritative DNS server.

### 12.10.3 Options

**-E** `engine`

This option specifies the cryptographic hardware to use, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-e** `date/offset`

This option sets the end date for which keys or SKRs need to be generated (depending on the command).

**-F**

This options turns on FIPS (US Federal Information Processing Standards) mode if the underlying crytographic library supports running in FIPS mode.

**-f**

This option sets the SKR file to be signed when issuing a `sign` command.

**-h**

This option prints a short summary of the options and arguments to **dnssec-ksr**.

**-i** date/offset

This option sets the start date for which keys or SKRs need to be generated (depending on the command).

**-K** directory

This option sets the directory in which the key files are to be read or written (depending on the command).

**-k** policy

This option sets the specific dnssec-policy for which keys need to be generated, or signed.

**-l** file

This option provides a configuration file that contains a dnssec-policy statement (matching the policy set with *-k*).

**-o**

Normally when pregenerating keys, ZSKs are created. When this option is set, create KSKs instead.

**-V**

This option prints version information.

**-v** level

This option sets the debugging level. Level 1 is intended to be usefully verbose for general users; higher levels are intended for developers.

command

The KSR command to be executed. See below for the available commands.

zone

The name of the zone for which the KSR command is being executed.

### 12.10.4 Commands

**keygen**

Pregenerate a number of keys, given a DNSSEC policy and an interval. The number of generated keys depends on the interval and the key lifetime.

**request**

Create a Key Signing Request (KSR), given a DNSSEC policy and an interval. This will generate a file with a number of key bundles, where each bundle contains the currently published ZSKs (according to the timing metadata).

**sign**

Sign a Key Signing Request (KSR), given a DNSSEC policy and an interval, creating a Signed Key Response (SKR). This will add the corresponding DNSKEY, CDS, and CDNSKEY records for the KSK that is being used for signing.

### 12.10.5 Exit Status

The **dnssec-ksr** command exits 0 on success, or non-zero if an error occurred.

### 12.10.6 Examples

When you need to generate ZSKs for the zone "example.com" for the next year, given a `dnssec-policy` named "mypolicy":

```
dnssec-ksr -i now -e +1y -k mypolicy -l named.conf keygen example.com
```

Creating a KSR for the same zone and period can be done with:

```
dnssec-ksr -i now -e +1y -k mypolicy -l named.conf request example.com > ksr.txt
```

Typically you would now transfer the KSR to the system that has access to the KSK.

Signing the KSR created above can be done with:

```
dnssec-ksr -i now -e +1y -k kskpolicy -l named.conf -f ksr.txt sign example.com
```

Make sure that the DNSSEC parameters in `kskpolicy` match those in `mypolicy`.

### 12.10.7 See Also

*dnssec-keygen(8)*, *dnssec-signzone(8)*, BIND 9 Administrator Reference Manual.

## 12.11 dnssec-revoke - set the REVOKED bit on a DNSSEC key

### 12.11.1 Synopsis

**dnssec-revoke** [**-hr**] [**-v** level] [**-V**] [**-K** directory] [**-E** engine] [**-f**] [**-R**] {keyfile}

### 12.11.2 Description

**dnssec-revoke** reads a DNSSEC key file, sets the REVOKED bit on the key as defined in **RFC 5011**, and creates a new pair of key files containing the now-revoked key.

### 12.11.3 Options

**-h**

This option emits a usage message and exits.

**-K** `directory`

This option sets the directory in which the key files are to reside.

**-r**

This option indicates to remove the original keyset files after writing the new keyset files.

**-v** `level`

This option sets the debugging level.

**-V**

This option prints version information.

**-E** `engine`

This option specifies the cryptographic hardware to use, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-f**

This option indicates a forced overwrite and causes **dnssec-revoke** to write the new key pair, even if a file already exists matching the algorithm and key ID of the revoked key.

**-R**

This option prints the key tag of the key with the REVOKE bit set, but does not revoke the key.

### 12.11.4 See Also

*dnssec-keygen(8)*, BIND 9 Administrator Reference Manual, **RFC 5011**.

## 12.12 dnssec-settime: set the key timing metadata for a DNSSEC key

### 12.12.1 Synopsis

**dnssec-settime** [**-f**] [**-K** directory] [**-L** ttl] [**-P** date/offset] [**-P** ds date/offset] [**-P** sync date/offset] [**-A** date/offset] [**-R** date/offset] [**-I** date/offset] [**-D** date/offset] [**-D** ds date/offset] [**-D** sync date/offset] [**-S** key] [**-i** interval] [**-h**] [**-V**] [**-v** level] [**-E** engine] {keyfile} [**-s**] [**-g** state] [**-d** state date/offset] [**-k** state date/offset] [**-r** state date/offset] [**-z** state date/offset]

### 12.12.2 Description

**dnssec-settime** reads a DNSSEC private key file and sets the key timing metadata as specified by the *-P*, *-A*, *-R*, *-I*, and *-D* options. The metadata can then be used by *dnssec-signzone* or other signing software to determine when a key is to be published, whether it should be used for signing a zone, etc.

If none of these options is set on the command line, **dnssec-settime** simply prints the key timing metadata already stored in the key.

When key metadata fields are changed, both files of a key pair (`Knnnn.+aaa+iiiii.key` and `Knnnn.+aaa+iiiii.private`) are regenerated.

Metadata fields are stored in the private file. A human-readable description of the metadata is also placed in comments in the key file. The private file's permissions are always set to be inaccessible to anyone other than the owner (mode 0600).

When working with state files, it is possible to update the timing metadata in those files as well with *-s*. With this option, it is also possible to update key states with *-d* (DS), *-k* (DNSKEY), *-r* (RRSIG of KSK), or *-z* (RRSIG of ZSK). Allowed states are HIDDEN, RUMOURED, OMNIPRESENT, and UNRETENTIVE.

The goal state of the key can also be set with *-g*. This should be either HIDDEN or OMNIPRESENT, representing whether the key should be removed from the zone or published.

It is NOT RECOMMENDED to manipulate state files manually, except for testing purposes.

### 12.12.3 Options

**-f**

This option forces an update of an old-format key with no metadata fields. Without this option, **dnssec-settime** fails when attempting to update a legacy key. With this option, the key is recreated in the new format, but with the original key data retained. The key's creation date is set to the present time. If no other values are specified, then the key's publication and activation dates are also set to the present time.

**-K** `directory`

This option sets the directory in which the key files are to reside.

**-L** ttl

This option sets the default TTL to use for this key when it is converted into a DNSKEY RR. This is the TTL used when the key is imported into a zone, unless there was already a DNSKEY RRset in place, in which case the existing TTL takes precedence. If this value is not set and there is no existing DNSKEY RRset, the TTL defaults to the SOA TTL. Setting the default TTL to `0` or `none` removes it from the key.

**-h**

This option emits a usage message and exits.

**-V**

This option prints version information.

**-v** level

This option sets the debugging level.

**-E** engine

This option specifies the cryptographic hardware to use, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

### 12.12.4 Timing Options

Dates can be expressed in the format YYYYMMDD or YYYYMMDDHHMMSS (which is the format used inside key files), or 'Day Mon DD HH:MM:SS YYYY' (as printed by `dnssec-settime -p`), or UNIX epoch time (as printed by `dnssec-settime -up`), or the literal `now`.

The argument can be followed by `+` or `-` and an offset from the given time. The literal `now` can be omitted before an offset. The offset can be followed by one of the suffixes `y`, `mo`, `w`, `d`, `h`, or `mi`, so that it is computed in years (defined as 365 24-hour days, ignoring leap years), months (defined as 30 24-hour days), weeks, days, hours, or minutes, respectively. Without a suffix, the offset is computed in seconds.

To unset a date, use `none`, `never`, or `unset`.

All these formats are case-insensitive.

**-P** date/offset

This option sets the date on which a key is to be published to the zone. After that date, the key is included in the zone but is not used to sign it.

**ds** date/offset

This option sets the date on which DS records that match this key have been seen in the parent zone.

**sync** date/offset

This option sets the date on which CDS and CDNSKEY records that match this key are to be published to the zone.

**-A** date/offset

This option sets the date on which the key is to be activated. After that date, the key is included in the zone and used to sign it.

**-R** date/offset

This option sets the date on which the key is to be revoked. After that date, the key is flagged as revoked. It is included in the zone and is used to sign it.

**-I** date/offset

This option sets the date on which the key is to be retired. After that date, the key is still included in the zone, but it is not used to sign it.

**-D** date/offset

This option sets the date on which the key is to be deleted. After that date, the key is no longer included in the zone. (However, it may remain in the key repository.)

**ds** date/offset

This option sets the date on which the DS records that match this key have been seen removed from the parent zone.

**sync** date/offset

This option sets the date on which the CDS and CDNSKEY records that match this key are to be deleted.

**-S** predecessor key

This option selects a key for which the key being modified is an explicit successor. The name, algorithm, size, and type of the predecessor key must exactly match those of the key being modified. The activation date of the successor key is set to the inactivation date of the predecessor. The publication date is set to the activation date minus the prepublication interval, which defaults to 30 days.

**-i** interval

This option sets the prepublication interval for a key. If set, then the publication and activation dates must be separated by at least this much time. If the activation date is specified but the publication date is not, the publication date defaults to this much time before the activation date; conversely, if the publication date is specified but not the activation date, activation is set to this much time after publication.

If the key is being created as an explicit successor to another key, then the default prepublication interval is 30 days; otherwise it is zero.

As with date offsets, if the argument is followed by one of the suffixes y, mo, w, d, h, or mi, the interval is measured in years, months, weeks, days, hours, or minutes, respectively. Without a suffix, the interval is measured in seconds.

### 12.12.5 Key State Options

To test dnssec-policy it may be necessary to construct keys with artificial state information; these options are used by the testing framework for that purpose, but should never be used in production.

Known key states are HIDDEN, RUMOURED, OMNIPRESENT, and UNRETENTIVE.

**-s**

This option indicates that when setting key timing data, the state file should also be updated.

**-g** state

This option sets the goal state for this key. Must be HIDDEN or OMNIPRESENT.

**-d** state date/offset

This option sets the DS state for this key as of the specified date, offset from the current date.

**-k** state date/offset

This option sets the DNSKEY state for this key as of the specified date, offset from the current date.

**-r** state date/offset

This option sets the RRSIG (KSK) state for this key as of the specified date, offset from the current date.

**-z** state date/offset

This option sets the RRSIG (ZSK) state for this key as of the specified date, offset from the current date.

### 12.12.6 Printing Options

**dnssec-settime** can also be used to print the timing metadata associated with a key.

**-u**

This option indicates that times should be printed in Unix epoch format.

**-p** C/P/Pds/Psync/A/R/I/D/Dds/Dsync/all

This option prints a specific metadata value or set of metadata values. The *-p* option may be followed by one or more of the following letters or strings to indicate which value or values to print: `C` for the creation date, `P` for the publication date, `Pds` for the DS publication date, ``Psync`` for the CDS and CDNSKEY publication date, `A` for the activation date, `R` for the revocation date, `I` for the inactivation date, `D` for the deletion date, `Dds` for the DS deletion date, and `Dsync` for the CDS and CDNSKEY deletion date. To print all of the metadata, use `all`.

### 12.12.7 See Also

*dnssec-keygen(8)*, *dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, **RFC 5011**.

## 12.13 dnssec-signzone - DNSSEC zone signing tool

### 12.13.1 Synopsis

**dnssec-signzone** [**-a**] [**-c** class] [**-d** directory] [**-D**] [**-E** engine] [**-e** end-time] [**-f** output-file] [**-F**] [**-g**] [**-G** **sync-records**] [**-h**] [**-i** interval] [**-I** input-format] [**-j** jitter] [**-J** filename] [**-K** directory] [**-k** key] [**-L** serial] [**-M** maxttl] [**-N** soa-serial-format] [**-o** origin] [**-O** output-format] [**-P**] [**-Q**] [**-q**] [**-R**] [**-S**] [**-s** start-time] [**-T** ttl] [**-t**] [**-u**] [**-v** level] [**-V**] [**-X** extended end-time] [**-x**] [**-z**] [**-3** salt] [**-H** iterations] [**-A**] {zonefile} [key...]

### 12.13.2 Description

**dnssec-signzone** signs a zone; it generates NSEC and RRSIG records and produces a signed version of the zone. The security status of delegations from the signed zone (that is, whether the child zones are secure) is determined by the presence or absence of a `keyset` file for each child zone.

### 12.13.3 Options

**-a**

This option verifies all generated signatures.

**-c** class

This option specifies the DNS class of the zone.

**-C**

This option sets compatibility mode, in which a `keyset-zonename` file is generated in addition to `dsset-zonename` when signing a zone, for use by older versions of **dnssec-signzone**.

**-d** directory

This option indicates the directory where BIND 9 should look for `dsset-` or `keyset-` files.

**-D**

This option indicates that only those record types automatically managed by **dnssec-signzone**, i.e., RRSIG, NSEC, NSEC3 and NSEC3PARAM records, should be included in the output. If smart signing (*-S*) is used, DNSKEY records are also included. The resulting file can be included in the original zone file with `$INCLUDE`. This option cannot be combined with *-O raw* or serial-number updating.

**-E** engine

This option specifies the hardware to use for cryptographic operations, such as a secure key store used for signing, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-F**

This options turns on FIPS (US Federal Information Processing Standards) mode if the underlying crytographic library supports running in FIPS mode.

**-g**

This option indicates that DS records for child zones should be generated from a `dsset-` or `keyset-` file. Existing DS records are removed.

**-G** sync-records

This option indicates which CDS and CDNSKEY records should be generated. `sync-records` is a comma-separated string with the following allowed items: `cdnskey`, and `cds:<digest-type>`, where `digest-type` is an allowed algorithm such as SHA-256 (2), or SHA-384 (4). Only works in combination with smart signing (`-S`).

**-J** filename

This option tells **dnssec-signzone** to read the journal from the given file when loading the zone file.

**-K** directory

This option specifies the directory to search for DNSSEC keys. If not specified, it defaults to the current directory.

**-k** key

This option tells BIND 9 to treat the specified key as a key-signing key, ignoring any key flags. This option may be specified multiple times.

**-M** maxttl

This option sets the maximum TTL for the signed zone. Any TTL higher than `maxttl` in the input zone is reduced to `maxttl` in the output. This provides certainty as to the largest possible TTL in the signed zone, which is useful to know when rolling keys. The maxttl is the longest possible time before signatures that have been retrieved by resolvers expire from resolver caches. Zones that are signed with this option should be configured to use a matching `max-zone-ttl` in *named.conf*. (Note: This option is incompatible with *-D*, because it modifies non-DNSSEC data in the output zone.)

**-s** start-time

This option specifies the date and time when the generated RRSIG records become valid. This can be either an absolute or relative time. An absolute start time is indicated by a number in YYYYMMDDHHMMSS notation; 20000530144500 denotes 14:45:00 UTC on May 30th, 2000. A relative start time is indicated by `+N`, which is N seconds from the current time. If no `start-time` is specified, the current time minus 1 hour (to allow for clock skew) is used.

**-e** end-time

This option specifies the date and time when the generated RRSIG records expire. As with `start-time`, an absolute time is indicated in YYYYMMDDHHMMSS notation. A time relative to the start time is indicated with `+N`, which is N seconds from the start time. A time relative to the current time is indicated with `now+N`. If no `end-time` is specified, 30 days from the start time is the default. `end-time` must be later than `start-time`.

**-X** extended end-time

This option specifies the date and time when the generated RRSIG records for the DNSKEY RRset expire. This is to be used in cases when the DNSKEY signatures need to persist longer than signatures on other records; e.g., when the private component of the KSK is kept offline and the KSK signature is to be refreshed manually.

As with `end-time`, an absolute time is indicated in YYYYMMDDHHMMSS notation. A time relative to the start time is indicated with `+N`, which is N seconds from the start time. A time relative to the current time is indicated

with `now+N`. If no `extended end-time` is specified, the value of `end-time` is used as the default. (`end-time`, in turn, defaults to 30 days from the start time.) `extended end-time` must be later than `start-time`.

**-f** `output-file`

This option indicates the name of the output file containing the signed zone. The default is to append `.signed` to the input filename. If `output-file` is set to `-`, then the signed zone is written to the standard output, with a default output format of `full`.

**-h**

This option prints a short summary of the options and arguments to **dnssec-signzone**.

**-V**

This option prints version information.

**-i** `interval`

This option indicates that, when a previously signed zone is passed as input, records may be re-signed. The `interval` option specifies the cycle interval as an offset from the current time, in seconds. If a RRSIG record expires after the cycle interval, it is retained; otherwise, it is considered to be expiring soon and it is replaced.

The default cycle interval is one quarter of the difference between the signature end and start times. So if neither `end-time` nor `start-time` is specified, **dnssec-signzone** generates signatures that are valid for 30 days, with a cycle interval of 7.5 days. Therefore, if any existing RRSIG records are due to expire in less than 7.5 days, they are replaced.

Note that the calculation of cycle interval is based upon the validity period of the replacement signatures that would be generated by `dnssec-signzone`, not on the valid lifetimes of the input RRSIGs being considered for pre-expiry replacement.

**-I** `input-format`

This option sets the format of the input zone file. Possible formats are `text` (the default), and `raw`. This option is primarily intended to be used for dynamic signed zones, so that the dumped zone file in a non-text format containing updates can be signed directly. This option is not useful for non-dynamic zones.

**-j** `jitter`

When signing a zone with a fixed signature lifetime, all RRSIG records issued at the time of signing expire simultaneously. If the zone is incrementally signed, i.e., a previously signed zone is passed as input to the signer, all expired signatures must be regenerated at approximately the same time. The `jitter` option specifies a jitter window that is used to randomize the signature expire time, thus spreading incremental signature regeneration over time.

Signature lifetime jitter also, to some extent, benefits validators and servers by spreading out cache expiration, i.e., if large numbers of RRSIGs do not expire at the same time from all caches, there is less congestion than if all validators need to refetch at around the same time.

**-L** `serial`

When writing a signed zone to "raw" format, this option sets the "source serial" value in the header to the specified `serial` number. (This is expected to be used primarily for testing purposes.)

**-n** `ncpus`

This option specifies the number of threads to use. By default, one thread is started for each detected CPU.

**-N** `soa-serial-format`

This option sets the SOA serial number format of the signed zone. Possible formats are `keep` (the default), `increment`, `unixtime`, and `date`.

**keep**

This format indicates that the SOA serial number should not be modified.

**increment**
This format increments the SOA serial number using **RFC 1982** arithmetic.

**unixtime**
This format sets the SOA serial number to the number of seconds since the beginning of the Unix epoch, unless the serial number is already greater than or equal to that value, in which case it is simply incremented by one.

**date**
This format sets the SOA serial number to today's date, in YYYYMMDDNN format, unless the serial number is already greater than or equal to that value, in which case it is simply incremented by one.

**-o** `origin`

This option sets the zone origin. If not specified, the name of the zone file is assumed to be the origin.

**-O** `output-format`

This option sets the format of the output file containing the signed zone. Possible formats are `text` (the default), which is the standard textual representation of the zone; `full`, which is text output in a format suitable for processing by external scripts; and `raw` and `raw=N`, which store the zone in binary formats for rapid loading by *named*. `raw=N` specifies the format version of the raw zone file: if N is 0, the raw file can be read by any version of *named*; if N is 1, the file can be read by release 9.9.0 or higher. The default is 1.

**-P**

This option disables post-sign verification tests.

The post-sign verification tests ensure that for each algorithm in use there is at least one non-revoked self-signed KSK key, that all revoked KSK keys are self-signed, and that all records in the zone are signed by the algorithm. This option skips these tests.

**-Q**

This option removes signatures from keys that are no longer active.

Normally, when a previously signed zone is passed as input to the signer, and a DNSKEY record has been removed and replaced with a new one, signatures from the old key that are still within their validity period are retained. This allows the zone to continue to validate with cached copies of the old DNSKEY RRset. The *-Q* option forces **dnssec-signzone** to remove signatures from keys that are no longer active. This enables ZSK rollover using the procedure described in **RFC 6781 Section 4.1.1.1** ("Pre-Publish Zone Signing Key Rollover").

**-q**

This option enables quiet mode, which suppresses unnecessary output. Without this option, when **dnssec-signzone** is run it prints three pieces of information to standard output: the number of keys in use; the algorithms used to verify the zone was signed correctly and other status information; and the filename containing the signed zone. With the option that output is suppressed, leaving only the filename.

**-R**

This option removes signatures from keys that are no longer published.

This option is similar to *-Q*, except it forces **dnssec-signzone** to remove signatures from keys that are no longer published. This enables ZSK rollover using the procedure described in **RFC 6781 Section 4.1.1.2** ("Double Signature Zone Signing Key Rollover").

**-S**

This option enables smart signing, which instructs **dnssec-signzone** to search the key repository for keys that match the zone being signed, and to include them in the zone if appropriate.

When a key is found, its timing metadata is examined to determine how it should be used, according to the following rules. Each successive rule takes priority over the prior ones:

If no timing metadata has been set for the key, the key is published in the zone and used to sign the zone.

If the key's publication date is set and is in the past, the key is published in the zone.

If the key's activation date is set and is in the past, the key is published (regardless of publication date) and used to sign the zone.

If the key's revocation date is set and is in the past, and the key is published, then the key is revoked, and the revoked key is used to sign the zone.

If either the key's unpublication or deletion date is set and in the past, the key is NOT published or used to sign the zone, regardless of any other metadata.

If the key's sync publication date is set and is in the past, synchronization records (type CDS and/or CDNSKEY) are created.

If the key's sync deletion date is set and is in the past, synchronization records (type CDS and/or CDNSKEY) are removed.

**-T** `ttl`

This option specifies a TTL to be used for new DNSKEY records imported into the zone from the key repository. If not specified, the default is the TTL value from the zone's SOA record. This option is ignored when signing without *-S*, since DNSKEY records are not imported from the key repository in that case. It is also ignored if there are any pre-existing DNSKEY records at the zone apex, in which case new records' TTL values are set to match them, or if any of the imported DNSKEY records had a default TTL value. In the event of a conflict between TTL values in imported keys, the shortest one is used.

**-t**

This option prints statistics at completion.

**-u**

This option updates the NSEC/NSEC3 chain when re-signing a previously signed zone. With this option, a zone signed with NSEC can be switched to NSEC3, or a zone signed with NSEC3 can be switched to NSEC or to NSEC3 with different parameters. Without this option, **dnssec-signzone** retains the existing chain when re-signing.

**-v** `level`

This option sets the debugging level.

**-x**

This option indicates that BIND 9 should only sign the DNSKEY, CDNSKEY, and CDS RRsets with key-signing keys, and should omit signatures from zone-signing keys.

**-z**

This option indicates that BIND 9 should ignore the KSK flag on keys when determining what to sign. This causes KSK-flagged keys to sign all records, not just the DNSKEY RRset.

**-3** `salt`

This option generates an NSEC3 chain with the given hex-encoded salt. A dash (-) can be used to indicate that no salt is to be used when generating the NSEC3 chain.

> **Note**
>
> `-3 -` is the recommended configuration. Adding salt provides no practical benefits. See **RFC 9276**.

**-H** `iterations`

This option indicates that, when generating an NSEC3 chain, BIND 9 should use this many iterations. The default is 0.

> **Warning**
>
> Values greater than 0 cause interoperability issues and also increase the risk of CPU-exhausting DoS attacks. See **RFC 9276**.

**-A**

This option indicates that, when generating an NSEC3 chain, BIND 9 should set the OPTOUT flag on all NSEC3 records and should not generate NSEC3 records for insecure delegations.

> **Warning**
>
> Do not use this option unless all its implications are fully understood. This option is intended only for extremely large zones (comparable to `com.`) with sparse secure delegations. See **RFC 9276**.

**-AA**

This option turns the OPTOUT flag off for all records. This is useful when using the *-u* option to modify an NSEC3 chain which previously had OPTOUT set.

**zonefile**

This option sets the file containing the zone to be signed.

**key**

This option specifies which keys should be used to sign the zone. If no keys are specified, the zone is examined for DNSKEY records at the zone apex. If these records are found and there are matching private keys in the current directory, they are used for signing.

### 12.13.4 Example

The following command signs the `example.com` zone with the ECDSAP256SHA256 key generated by *dnssec-keygen* (Kexample.com.+013+17247). Because the *-S* option is not being used, the zone's keys must be in the master file (`db.example.com`). This invocation looks for `dsset` files in the current directory, so that DS records can be imported from them (*-g*).

```
% dnssec-signzone -g -o example.com db.example.com \
Kexample.com.+013+17247
db.example.com.signed
%
```

In the above example, **dnssec-signzone** creates the file `db.example.com.signed`. This file should be referenced in a zone statement in the *named.conf* file.

This example re-signs a previously signed zone with default parameters. The private keys are assumed to be in the current directory.

```
% cp db.example.com.signed db.example.com
% dnssec-signzone -o example.com db.example.com
db.example.com.signed
%
```

### 12.13.5 See Also

*dnssec-keygen(8)*, BIND 9 Administrator Reference Manual, **RFC 4033**, **RFC 6781**.

## 12.14 dnssec-verify - DNSSEC zone verification tool

### 12.14.1 Synopsis

**dnssec-verify** [**-c** class] [**-E** engine] [**-I** input-format] [**-J** filename] [**-o** origin] [**-q**] [**-v** level] [**-V**] [**-x**] [**-z**] {zonefile}

### 12.14.2 Description

**dnssec-verify** verifies that a zone is fully signed for each algorithm found in the DNSKEY RRset for the zone, and that the NSEC/NSEC3 chains are complete.

### 12.14.3 Options

**-c** `class`

This option specifies the DNS class of the zone.

**-E** `engine`

This option specifies the cryptographic hardware to use, when applicable.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-I** `input-format`

This option sets the format of the input zone file. Possible formats are `text` (the default) and `raw`. This option is primarily intended to be used for dynamic signed zones, so that the dumped zone file in a non-text format containing updates can be verified independently. This option is not useful for non-dynamic zones.

**-J** `filename`

This option tells **dnssec-verify** to read the journal from the given file when loading the zone file.

**-o** `origin`

This option indicates the zone origin. If not specified, the name of the zone file is assumed to be the origin.

**-v** `level`

This option sets the debugging level.

**-V**

This option prints version information.

**-q**

This option sets quiet mode, which suppresses output. Without this option, when **dnssec-verify** is run it prints to standard output the number of keys in use, the algorithms used to verify the zone was signed correctly, and other status information. With this option, all non-error output is suppressed, and only the exit code indicates success.

**-x**

This option verifies only that the DNSKEY RRset is signed with key-signing keys. Without this flag, it is assumed that the DNSKEY RRset is signed by all active keys. When this flag is set, it is not an error if the DNSKEY RRset is not signed by zone-signing keys. This corresponds to the *-x option in dnssec-signzone*.

**-z**

This option indicates that the KSK flag on the keys should be ignored when determining whether the zone is correctly signed. Without this flag, it is assumed that there is a non-revoked, self-signed DNSKEY with the KSK flag set for each algorithm, and that RRsets other than DNSKEY RRset are signed with a different DNSKEY without the KSK flag set.

With this flag set, BIND 9 only requires that for each algorithm, there be at least one non-revoked, self-signed DNSKEY, regardless of the KSK flag state, and that other RRsets be signed by a non-revoked key for the same algorithm that includes the self-signed key; the same key may be used for both purposes. This corresponds to the *-z option in dnssec-signzone*.

**zonefile**

This option indicates the file containing the zone to be signed.

### 12.14.4 See Also

*dnssec-signzone(8)*, BIND 9 Administrator Reference Manual, **RFC 4033**.

## 12.15 dnstap-read - print dnstap data in human-readable form

### 12.15.1 Synopsis

**dnstap-read** [**-m**] [**-p**] [**-x**] [**-y**] {file}

### 12.15.2 Description

**dnstap-read** reads `dnstap` data from a specified file and prints it in a human-readable format. By default, `dnstap` data is printed in a short summary format, but if the *-y* option is specified, a longer and more detailed YAML format is used.

### 12.15.3 Options

**-m**

This option indicates trace memory allocations, and is used for debugging memory leaks.

**-p**

This option prints the text form of the DNS message that was encapsulated in the `dnstap` frame, after printing the `dnstap` data.

**-t**

This option prints long timestamps with millisecond precision.

**-x**

This option prints a hex dump of the wire form of the DNS message that was encapsulated in the `dnstap` frame, after printing the `dnstap` data.

**-y**

This option prints `dnstap` data in a detailed YAML format.

### 12.15.4 See Also

*named(8)*, *rndc(8)*, BIND 9 Administrator Reference Manual.

## 12.16 filter-aaaa.so - filter AAAA in DNS responses when A is present

### 12.16.1 Synopsis

**plugin query** "filter-aaaa.so" [{ parameters }];

### 12.16.2 Description

**filter-aaaa.so** is a query plugin module for *named*, enabling *named* to omit some IPv6 addresses when responding to clients.

Until BIND 9.12, this feature was implemented natively in *named* and enabled with the `filter-aaaa` ACL and the `filter-aaaa-on-v4` and `filter-aaaa-on-v6` options. These options are no longer available in *named.conf* but can be passed as parameters to the `filter-aaaa.so` plugin, for example:

```
plugin query "filter-aaaa.so" {
        filter-aaaa-on-v4 yes;
        filter-aaaa-on-v6 yes;
        filter-aaaa { 192.0.2.1; 2001:db8:2::1; };
};
```

This module is intended to aid transition from IPv4 to IPv6 by withholding IPv6 addresses from DNS clients which are not connected to the IPv6 Internet, when the name being looked up has an IPv4 address available. Use of this module is not recommended unless absolutely necessary.

Note: This mechanism can erroneously cause other servers not to give AAAA records to their clients. If a recursing server with both IPv6 and IPv4 network connections queries an authoritative server using this mechanism via IPv4, it is denied AAAA records even if its client is using IPv6.

### 12.16.3 Options

**filter-aaaa**
: This option specifies a list of client addresses for which AAAA filtering is to be applied. The default is `any`.

**filter-aaaa-on-v4**
: If set to `yes`, this option indicates that the DNS client is at an IPv4 address, in `filter-aaaa`. If the response does not include DNSSEC signatures, then all AAAA records are deleted from the response. This filtering applies to all responses, not only authoritative ones.

  If set to `break-dnssec`, then AAAA records are deleted even when DNSSEC is enabled. As suggested by the name, this causes the response to fail to verify, because the DNSSEC protocol is designed to detect deletions.

  This mechanism can erroneously cause other servers not to give AAAA records to their clients. If a recursing server with both IPv6 and IPv4 network connections queries an authoritative server using this mechanism via IPv4, it is denied AAAA records even if its client is using IPv6.

**filter-aaaa-on-v6**
: This option is identical to `filter-aaaa-on-v4`, except that it filters AAAA responses to queries from IPv6 clients instead of IPv4 clients. To filter all responses, set both options to `yes`.

### 12.16.4 See Also

BIND 9 Administrator Reference Manual.

# 12.17 host - DNS lookup utility

## 12.17.1 Synopsis

**host** [**-aACdlnrsTUwv**] [**-c** class] [**-N** ndots] [**-p** port] [**-R** number] [**-t** type] [**-W** wait] [**-m** flag] [ [**-4**] | [**-6**] ] [**-v**] [**-V**] {name} [server]

## 12.17.2 Description

**host** is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa. When no arguments or options are given, **host** prints a short summary of its command-line arguments and options.

`name` is the domain name that is to be looked up. It can also be a dotted-decimal IPv4 address or a colon-delimited IPv6 address, in which case **host** by default performs a reverse lookup for that address. `server` is an optional argument which is either the name or IP address of the name server that **host** should query instead of the server or servers listed in `/etc/resolv.conf`.

## 12.17.3 Options

**-4**

This option specifies that only IPv4 should be used for query transport. See also the *-6* option.

**-6**

This option specifies that only IPv6 should be used for query transport. See also the *-4* option.

**-a**

The *-a* ("all") option is normally equivalent to *-v -t ANY*. It also affects the behavior of the *-l* list zone option.

**-A**

The *-A* ("almost all") option is equivalent to *-a*, except that RRSIG, NSEC, and NSEC3 records are omitted from the output.

**-c** `class`

This option specifies the query class, which can be used to lookup HS (Hesiod) or CH (Chaosnet) class resource records. The default class is IN (Internet).

**-C**

This option indicates that *named* should check consistency, meaning that **host** queries the SOA records for zone `name` from all the listed authoritative name servers for that zone. The list of name servers is defined by the NS records that are found for the zone.

**-d**

This option prints debugging traces, and is equivalent to the *-v* verbose option.

**-l**

This option tells *named* to list the zone, meaning the **host** command performs a zone transfer of zone `name` and prints out the NS, PTR, and address records (A/AAAA).

Together, the *-l -a* options print all records in the zone.

**-N** `ndots`

This option specifies the number of dots (`ndots`) that have to be in `name` for it to be considered absolute. The default value is that defined using the `ndots` statement in `/etc/resolv.conf`, or 1 if no `ndots` statement is present. Names with fewer dots are interpreted as relative names, and are searched for in the domains listed in the `search` or `domain` directive in `/etc/resolv.conf`.

**-p** port

This option specifies the port to query on the server. The default is 53.

**-r**

This option specifies a non-recursive query; setting this option clears the RD (recursion desired) bit in the query. This means that the name server receiving the query does not attempt to resolve `name`. The *-r* option enables **host** to mimic the behavior of a name server by making non-recursive queries, and expecting to receive answers to those queries that can be referrals to other name servers.

**-R** number

This option specifies the number of retries for UDP queries. If `number` is negative or zero, the number of retries is silently set to 1. The default value is 1, or the value of the `attempts` option in `/etc/resolv.conf`, if set.

**-s**

This option tells *named* *not* to send the query to the next nameserver if any server responds with a SERVFAIL response, which is the reverse of normal stub resolver behavior.

**-t** type

This option specifies the query type. The `type` argument can be any recognized query type: CNAME, NS, SOA, TXT, DNSKEY, AXFR, etc.

When no query type is specified, **host** automatically selects an appropriate query type. By default, it looks for A, AAAA, MX, and HTTPS records. If the *-C* option is given, queries are made for SOA records. If `name` is a dotted-decimal IPv4 address or colon-delimited IPv6 address, **host** queries for PTR records.

If a query type of IXFR is chosen, the starting serial number can be specified by appending an equals sign (=), followed by the starting serial number, e.g., *-t IXFR=12345678*.

**-T, -U**

This option specifies TCP or UDP. By default, **host** uses UDP when making queries; the *-T* option makes it use a TCP connection when querying the name server. TCP is automatically selected for queries that require it, such as zone transfer (AXFR) requests. Type `ANY` queries default to TCP, but can be forced to use UDP initially via *-U*.

**-m** flag

This option sets memory usage debugging: the flag can be `record`, `usage`, or `trace`. The *-m* option can be specified more than once to set multiple flags.

**-v**

This option sets verbose output, and is equivalent to the *-d* debug option. Verbose output can also be enabled by setting the `debug` option in `/etc/resolv.conf`.

**-V**

This option prints the version number and exits.

**-w**

This option sets "wait forever": the query timeout is set to the maximum possible. See also the *-W* option.

**-W** wait

This options sets the length of the wait timeout, indicating that *named* should wait for up to `wait` seconds for a reply. If `wait` is less than 1, the wait interval is set to 1 second.

By default, **host** waits for 5 seconds for UDP responses and 10 seconds for TCP connections. These defaults can be overridden by the `timeout` option in `/etc/resolv.conf`.

See also the *-w* option.

### 12.17.4 IDN Support

If **host** has been built with IDN (internationalized domain name) support, it can accept and display non-ASCII domain names. **host** appropriately converts character encoding of a domain name before sending a request to a DNS server or displaying a reply from the server. To turn off IDN support, define the `IDN_DISABLE` environment variable. IDN support is disabled if the variable is set when **host** runs.

### 12.17.5 Files

`/etc/resolv.conf`

### 12.17.6 See Also

*dig(1)*, *named(8)*.

## 12.18 mdig - DNS pipelined lookup utility

### 12.18.1 Synopsis

**mdig** {@server} [**-f** filename] [**-h**] [**-v**] [ [**-4**] | [**-6**] ] [**-m**] [**-b** address] [**-p** port#] [**-c** class] [**-t** type] [**-i**] [**-x** addr] [plusopt...]

**mdig** {**-h**}

**mdig** [@server] {global-opt...} { {local-opt...} {query} ...}

### 12.18.2 Description

**mdig** is a multiple/pipelined query version of *dig*: instead of waiting for a response after sending each query, it begins by sending all queries. Responses are displayed in the order in which they are received, not in the order the corresponding queries were sent.

**mdig** options are a subset of the *dig* options, and are divided into "anywhere options," which can occur anywhere, "global options," which must occur before the query name (or they are ignored with a warning), and "local options," which apply to the next query on the command line.

The `@server` option is a mandatory global option. It is the name or IP address of the name server to query. (Unlike *dig*, this value is not retrieved from `/etc/resolv.conf`.) It can be an IPv4 address in dotted-decimal notation, an IPv6 address in colon-delimited notation, or a hostname. When the supplied `server` argument is a hostname, **mdig** resolves that name before querying the name server.

**mdig** provides a number of query options which affect the way in which lookups are made and the results displayed. Some of these set or reset flag bits in the query header, some determine which sections of the answer get printed, and others determine the timeout and retry strategies.

Each query option is identified by a keyword preceded by a plus sign (+). Some keywords set or reset an option. These may be preceded by the string `no` to negate the meaning of that keyword. Other keywords assign values to options like the timeout interval. They have the form `+keyword=value`.

### 12.18.3 Anywhere Options

**-f**

This option makes **mdig** operate in batch mode by reading a list of lookup requests to process from the file `filename`. The file contains a number of queries, one per line. Each entry in the file should be organized in the same way they would be presented as queries to **mdig** using the command-line interface.

**-h**

This option causes **mdig** to print detailed help information, with the full list of options, and exit.

**-v**

This option causes **mdig** to print the version number and exit.

### 12.18.4 Global Options

**-4**

This option forces **mdig** to only use IPv4 query transport.

**-6**

This option forces **mdig** to only use IPv6 query transport.

**-b** address

This option sets the source IP address of the query to address. This must be a valid address on one of the host's network interfaces or "0.0.0.0" or "::". An optional port may be specified by appending "#<port>"

**-m**

This option enables memory usage debugging.

**-p** port#

This option is used when a non-standard port number is to be queried. port# is the port number that **mdig** sends its queries to, instead of the standard DNS port number 53. This option is used to test a name server that has been configured to listen for queries on a non-standard port number.

The global query options are:

**+additional, +noadditional**

This option displays [or does not display] the additional section of a reply. The default is to display it.

**+all, +noall**

This option sets or clears all display flags.

**+answer, +noanswer**

This option displays [or does not display] the answer section of a reply. The default is to display it.

**+authority, +noauthority**

This option displays [or does not display] the authority section of a reply. The default is to display it.

**+besteffort, +nobesteffort**

This option attempts to display [or does not display] the contents of messages which are malformed. The default is to not display malformed answers.

**+burst**

This option delays queries until the start of the next second.

**+cl, +nocl**

This option displays [or does not display] the CLASS when printing the record.

**+comments, +nocomments**

This option toggles the display of comment lines in the output. The default is to print comments.

**+continue, +nocontinue**

This option toggles continuation on errors (e.g. timeouts).

**+crypto, +nocrypto**

This option toggles the display of cryptographic fields in DNSSEC records. The contents of these fields are unnecessary to debug most DNSSEC validation failures and removing them makes it easier to see the common failures. The default is to display the fields. When omitted, they are replaced by the string "[omitted]"; in the DNSKEY case, the key ID is displayed as the replacement, e.g., `[ key id = value ]`.

**+multiline, +nomultiline**

This option toggles printing of records, like the SOA records, in a verbose multi-line format with human-readable comments. The default is to print each record on a single line, to facilitate machine parsing of the **mdig** output.

**+question, +noquestion**

This option prints [or does not print] the question section of a query when an answer is returned. The default is to print the question section as a comment.

**+rrcomments, +norrcomments**

This option toggles the display of per-record comments in the output (for example, human-readable key information about DNSKEY records). The default is not to print record comments unless multiline mode is active.

**+short, +noshort**

This option provides [or does not provide] a terse answer. The default is to print the answer in a verbose form.

**+split**=W

This option splits long hex- or base64-formatted fields in resource records into chunks of `W` characters (where `W` is rounded up to the nearest multiple of 4). `+nosplit` or `+split=0` causes fields not to be split. The default is 56 characters, or 44 characters when multiline mode is active.

**+tcp, +notcp**

This option uses [or does not use] TCP when querying name servers. The default behavior is to use UDP.

**+ttlid, +nottlid**

This option displays [or does not display] the TTL when printing the record.

**+ttlunits, +nottlunits**

This option displays [or does not display] the TTL in friendly human-readable time units of "s", "m", "h", "d", and "w", representing seconds, minutes, hours, days, and weeks. This implies +ttlid.

**+vc, +novc**

This option uses [or does not use] TCP when querying name servers. This alternate syntax to *+tcp* is provided for backwards compatibility. The `vc` stands for "virtual circuit".

### 12.18.5 Local Options

**-c** class

This option sets the query class to `class`. It can be any valid query class which is supported in BIND 9. The default query class is "IN".

**-t** type

This option sets the query type to `type`. It can be any valid query type which is supported in BIND 9. The default query type is "A", unless the *-x* option is supplied to indicate a reverse lookup with the "PTR" query type.

**-x** addr

Reverse lookups - mapping addresses to names - are simplified by this option. `addr` is an IPv4 address in dotted-decimal notation, or a colon-delimited IPv6 address. **mdig** automatically performs a lookup for a query name like `11.12.13.10.in-addr.arpa` and sets the query type and class to PTR and IN respectively. By default, IPv6 addresses are looked up using nibble format under the IP6.ARPA domain.

The local query options are:

**+aaflag, +noaaflag**

This is a synonym for *+aaonly*, *+noaaonly*.

**+aaonly, +noaaonly**

This sets the `aa` flag in the query.

**+adflag, +noadflag**

This sets [or does not set] the AD (authentic data) bit in the query. This requests the server to return whether all of the answer and authority sections have all been validated as secure, according to the security policy of the server. AD=1 indicates that all records have been validated as secure and the answer is not from a OPT-OUT range. AD=0 indicates that some part of the answer was insecure or not validated. This bit is set by default.

**+bufsize**=B

This sets the UDP message buffer size advertised using EDNS0 to `B` bytes. The maximum and minimum sizes of this buffer are 65535 and 0 respectively. Values outside this range are rounded up or down appropriately. Values other than zero cause a EDNS query to be sent.

**+cdflag, +nocdflag**

This sets [or does not set] the CD (checking disabled) bit in the query. This requests the server to not perform DNSSEC validation of responses.

**+cookie**=####, **+nocookie**

This sends [or does not send] a COOKIE EDNS option, with an optional value. Replaying a COOKIE from a previous response allows the server to identify a previous client. The default is `+nocookie`.

**+dnssec, +nodnssec**

This requests that DNSSEC records be sent by setting the DNSSEC OK (DO) bit in the OPT record in the additional section of the query.

**+edns**[=#], **+noedns**

This specifies [or does not specify] the EDNS version to query with. Valid values are 0 to 255. Setting the EDNS version causes an EDNS query to be sent. `+noedns` clears the remembered EDNS version. EDNS is set to 0 by default.

**+ednsflags**[=#], **+noednsflags**

This sets the must-be-zero EDNS flag bits (Z bits) to the specified value. Decimal, hex, and octal encodings are accepted. Setting a named flag (e.g. DO) is silently ignored. By default, no Z bits are set.

**+ednsopt**[=code[:value]], **+noednsopt**

This specifies [or does not specify] an EDNS option with code point `code` and an optional payload of `value` as a hexadecimal string. `+noednsopt` clears the EDNS options to be sent.

**+expire, +noexpire**

This toggles sending of an EDNS Expire option.

**+nsid, +nonsid**

This toggles inclusion of an EDNS name server ID request when sending a query.

**+recurse, +norecurse**

This toggles the setting of the RD (recursion desired) bit in the query. This bit is set by default, which means **mdig** normally sends recursive queries.

**+retry**=T

This sets the number of times to retry UDP queries to server to `T` instead of the default, 2. Unlike *+tries*, this does not include the initial query.

**+subnet**=addr[/prefix-length], **+nosubnet**

This sends [or does not send] an EDNS Client Subnet option with the specified IP address or network prefix.

**mdig +subnet=0.0.0.0/0, or simply mdig +subnet=0**

This sends an EDNS client-subnet option with an empty address and a source prefix-length of zero, which signals a resolver that the client's address information must *not* be used when resolving this query.

**+timeout**=T

This sets the timeout for a query to T seconds. The default timeout is 5 seconds for UDP transport and 10 for TCP. An attempt to set T to less than 1 results in a query timeout of 1 second being applied.

**+tries**=T

This sets the number of times to try UDP queries to server to T instead of the default, 3. If T is less than or equal to zero, the number of tries is silently rounded up to 1.

**+udptimeout**=T

This sets the timeout between UDP query retries to T.

**+unknownformat, +nounknownformat**

This prints [or does not print] all RDATA in unknown RR-type presentation format (see **RFC 3597**). The default is to print RDATA for known types in the type's presentation format.

**+yaml, +noyaml**

This toggles printing of the responses in a detailed YAML format.

**+zflag, +nozflag**

This sets [or does not set] the last unassigned DNS header flag in a DNS query. This flag is off by default.

### 12.18.6 See Also

*dig(1)*, **RFC 1035**.

## 12.19 named-checkconf - named configuration file syntax checking tool

### 12.19.1 Synopsis

**named-checkconf** [**-achjklvz**] [**-p** [**-x** ]] [**-t** directory] {filename}

### 12.19.2 Description

**named-checkconf** checks the syntax, but not the semantics, of a *named* configuration file. The file, along with all files included by it, is parsed and checked for syntax errors. If no file is specified, /etc/named.conf is read by default.

Note: files that *named* reads in separate parser contexts, such as rndc.conf or rndc.key, are not automatically read by **named-checkconf**. Configuration errors in these files may cause *named* to fail to run, even if **named-checkconf** was successful. However, **named-checkconf** can be run on these files explicitly.

### 12.19.3 Options

**-a**

Don't check the *dnssec-policy*'s DNSSEC key algorithms against those supported by the crypto provider. This is useful when checking a *named.conf* intended to be run on another machine with possibly a different set of supported DNSSEC key algorithms.

**-h**
This option prints the usage summary and exits.

**-j**
When loading a zonefile, this option instructs *named* to read the journal if it exists.

**-k**
Check the *dnssec-policy*'s DNSSEC keys against the key files in the *key-directory*. This is useful when checking a *named.conf* to ensure a DNSSEC policy matches the existing keys.

**-l**
This option lists all the configured zones. Each line of output contains the zone name, class (e.g. IN), view, and type (e.g. primary or secondary).

**-c**
This option specifies that only the "core" configuration should be checked. This suppresses the loading of plugin modules, and causes all parameters to `plugin` statements to be ignored.

**-i**
This option ignores warnings on deprecated options.

**-p**
This option prints out the *named.conf* and included files in canonical form if no errors were detected. See also the *-x* option.

**-t** `directory`
This option instructs *named* to chroot to `directory`, so that `include` directives in the configuration file are processed as if run by a similarly chrooted *named*.

**-v**
This option prints the version of the **named-checkconf** program and exits.

**-x**
When printing the configuration files in canonical form, this option obscures shared secrets by replacing them with strings of question marks (`?`). This allows the contents of *named.conf* and related files to be shared - for example, when submitting bug reports - without compromising private data. This option cannot be used without *-p*.

**-z**
This option performs a test load of all zones of type `primary` found in *named.conf*.

**filename**
This indicates the name of the configuration file to be checked. If not specified, it defaults to `/etc/named.conf`.

### 12.19.4 Return Values

**named-checkconf** returns an exit status of 1 if errors were detected and 0 otherwise.

### 12.19.5 See Also

*named(8)*, *named-checkzone(8)*, BIND 9 Administrator Reference Manual.

## 12.20 named-checkzone - zone file validation tool

### 12.20.1 Synopsis

**named-checkzone** [**-d**] [**-h**] [**-j**] [**-q**] [**-v**] [**-c** class] [**-C** mode] [**-f** format] [**-F** format] [**-J** filename] [**-i** mode] [**-k** mode] [**-m** mode] [**-M** mode] [**-n** mode] [**-l** ttl] [**-L** serial] [**-o** filename] [**-r** mode] [**-s** style] [**-S** mode] [**-t** directory] [**-T** mode]

[**-w** directory] [**-D**] [**-W** mode] {zonename} {filename}

## 12.20.2 Description

**named-checkzone** checks the syntax and integrity of a zone file. It performs the same checks as *named* does when loading a zone. This makes **named-checkzone** useful for checking zone files before configuring them into a name server.

## 12.20.3 Options

**-d**

This option enables debugging.

**-h**

This option prints the usage summary and exits.

**-q**

This option sets quiet mode, which only sets an exit code to indicate successful or failed completion.

**-v**

This option prints the version of the **named-checkzone** program and exits.

**-j**

When loading a zone file, this option tells *named* to read the journal if it exists. The journal file name is assumed to be the zone file name with the string `.jnl` appended.

**-J** `filename`

When loading the zone file, this option tells *named* to read the journal from the given file, if it exists. This implies *-j*.

**-c** `class`

This option specifies the class of the zone. If not specified, `IN` is assumed.

**-C** `mode`

This option controls check mode on zone files when loading. Possible modes are `check-svcb:fail` and `check-svcb:ignore`.

`check-svcb:fail` turns on additional checks on `_dns` SVCB records and `check-svcb:ignore` disables these checks. The default is `check-svcb:fail`.

**-i** `mode`

This option performs post-load zone integrity checks. Possible modes are `full` (the default), `full-sibling`, `local`, `local-sibling`, and `none`.

Mode `full` checks that MX records refer to A or AAAA records (both in-zone and out-of-zone hostnames). Mode `local` only checks MX records which refer to in-zone hostnames.

Mode `full` checks that SRV records refer to A or AAAA records (both in-zone and out-of-zone hostnames). Mode `local` only checks SRV records which refer to in-zone hostnames.

Mode `full` checks that delegation NS records refer to A or AAAA records (both in-zone and out-of-zone hostnames). It also checks that glue address records in the zone match those advertised by the child. Mode `local` only checks NS records which refer to in-zone hostnames or verifies that some required glue exists, i.e., when the name server is in a child zone.

Modes `full-sibling` and `local-sibling` disable sibling glue checks, but are otherwise the same as `full` and `local`, respectively.

Mode `none` disables the checks.

**-f** format

This option specifies the format of the zone file. Possible formats are `text` (the default), and `raw`.

**-F** format

This option specifies the format of the output file specified. For **named-checkzone**, this does not have any effect unless it dumps the zone contents.

Possible formats are `text` (the default), which is the standard textual representation of the zone, and `raw` and `raw=N`, which store the zone in a binary format for rapid loading by *named*. `raw=N` specifies the format version of the raw zone file: if `N` is 0, the raw file can be read by any version of *named*; if N is 1, the file can only be read by release 9.9.0 or higher. The default is 1.

**-k** mode

This option performs `check-names` checks with the specified failure mode. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-l** ttl

This option sets a maximum permissible TTL for the input file. Any record with a TTL higher than this value causes the zone to be rejected. This is similar to using the `max-zone-ttl` option in *named.conf*.

**-L** serial

When compiling a zone to `raw` format, this option sets the "source serial" value in the header to the specified serial number. This is expected to be used primarily for testing purposes.

**-m** mode

This option specifies whether MX records should be checked to see if they are addresses. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-M** mode

This option checks whether a MX record refers to a CNAME. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-n** mode

This option specifies whether NS records should be checked to see if they are addresses. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-o** filename

This option writes the zone output to `filename`. If `filename` is `-`, then the zone output is written to standard output.

**-r** mode

This option checks for records that are treated as different by DNSSEC but are semantically equal in plain DNS. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-s** style

This option specifies the style of the dumped zone file. Possible styles are `full` (the default) and `relative`. The `full` format is most suitable for processing automatically by a separate script. The relative format is more human-readable and is thus suitable for editing by hand. This does not have any effect unless it dumps the zone contents. It also does not have any meaning if the output format is not text.

**-S** mode

This option checks whether an SRV record refers to a CNAME. Possible modes are `fail`, `warn` (the default), and `ignore`.

**-t** directory

This option tells *named* to chroot to `directory`, so that `include` directives in the configuration file are processed as if run by a similarly chrooted *named*.

**-T** mode

This option checks whether Sender Policy Framework (SPF) records exist and issues a warning if an SPF-formatted TXT record is not also present. Possible modes are `warn` (the default) and `ignore`.

**-w** directory

This option instructs *named* to chdir to `directory`, so that relative filenames in master file `$INCLUDE` directives work. This is similar to the directory clause in *named.conf*.

**-D**

This option dumps the zone file in canonical format.

**-W** mode

This option specifies whether to check for non-terminal wildcards. Non-terminal wildcards are almost always the result of a failure to understand the wildcard matching algorithm (**RFC 4592**). Possible modes are `warn` (the default) and `ignore`.

**zonename**

This indicates the domain name of the zone being checked.

**filename**

This is the name of the zone file.

### 12.20.4 Return Values

**named-checkzone** returns an exit status of 1 if errors were detected and 0 otherwise.

### 12.20.5 See Also

*named(8)*, *named-checkconf(8)*, *named-compilezone(8)*, **RFC 1035**, BIND 9 Administrator Reference Manual.

## 12.21 named-compilezone - zone file converting tool

### 12.21.1 Synopsis

**named-compilezone** [**-d**] [**-h**] [**-j**] [**-q**] [**-v**] [**-c** class] [**-C** mode] [**-f** format] [**-F** format] [**-J** filename] [**-i** mode] [**-k** mode] [**-m** mode] [**-M** mode] [**-n** mode] [**-l** ttl] [**-L** serial] [**-r** mode] [**-s** style] [**-S** mode] [**-t** directory] [**-T** mode] [**-w** directory] [**-D**] [**-W** mode] {**-o** filename} {zonename} {filename}

### 12.21.2 Description

**named-compilezone** checks the syntax and integrity of a zone file, and dumps the zone contents to a specified file in a specified format.

Unlike **named-checkzone**, zone contents are not strictly checked by default. If the output is to be used as an actual zone file to be loaded by *named*, then the check levels should be manually configured to be at least as strict as those specified in the *named* configuration file.

Running **named-checkzone** on the input prior to compiling will ensure that the zone compiles with the default requirements of *named*.

### 12.21.3 Options

**-d**

This option enables debugging.

**-h**

This option prints the usage summary and exits.

**-q**

This option sets quiet mode, which only sets an exit code to indicate successful or failed completion.

**-v**

This option prints the version of the *named-checkzone* program and exits.

**-j**

When loading a zone file, this option tells *named* to read the journal if it exists. The journal file name is assumed to be the zone file name with the string `.jnl` appended.

**-J** `filename`

When loading the zone file, this option tells *named* to read the journal from the given file, if it exists. This implies *-j*.

**-c** `class`

This option specifies the class of the zone. If not specified, `IN` is assumed.

**-C** `mode`

This option controls check mode on zone files when loading. Possible modes are `check-svcb:fail` and `check-svcb:ignore`.

`check-svcb:fail` turns on additional checks on `_dns` SVCB records and `check-svcb:ignore` disables these checks. The default is `check-svcb:ignore`.

**-i** `mode`

This option performs post-load zone integrity checks. Possible modes are `full`, `full-sibling`, `local`, `local-sibling`, and `none` (the default).

Mode `full` checks that MX records refer to A or AAAA records (both in-zone and out-of-zone hostnames). Mode `local` only checks MX records which refer to in-zone hostnames.

Mode `full` checks that SRV records refer to A or AAAA records (both in-zone and out-of-zone hostnames). Mode `local` only checks SRV records which refer to in-zone hostnames.

Mode `full` checks that delegation NS records refer to A or AAAA records (both in-zone and out-of-zone hostnames). It also checks that glue address records in the zone match those advertised by the child. Mode `local` only checks NS records which refer to in-zone hostnames or verifies that some required glue exists, i.e., when the name server is in a child zone.

Modes `full-sibling` and `local-sibling` disable sibling glue checks, but are otherwise the same as `full` and `local`, respectively.

Mode `none` disables the checks.

**-f** `format`

This option specifies the format of the zone file. Possible formats are `text` (the default), and `raw`.

**-F** `format`

This option specifies the format of the output file specified. For *named-checkzone*, this does not have any effect unless it dumps the zone contents.

Possible formats are `text` (the default), which is the standard textual representation of the zone, and `raw` and `raw=N`, which store the zone in a binary format for rapid loading by *named*. `raw=N` specifies the format version of the raw zone file: if `N` is 0, the raw file can be read by any version of *named*; if N is 1, the file can only be read by release 9.9.0 or higher. The default is 1.

**-k** mode

This option performs `check-names` checks with the specified failure mode. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-l** ttl

This option sets a maximum permissible TTL for the input file. Any record with a TTL higher than this value causes the zone to be rejected. This is similar to using the `max-zone-ttl` option in *named.conf*.

**-L** serial

When compiling a zone to `raw` format, this option sets the "source serial" value in the header to the specified serial number. This is expected to be used primarily for testing purposes.

**-m** mode

This option specifies whether MX records should be checked to see if they are addresses. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-M** mode

This option checks whether a MX record refers to a CNAME. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-n** mode

This option specifies whether NS records should be checked to see if they are addresses. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-o** filename

This option writes the zone output to `filename`. If `filename` is `-`, then the zone output is written to standard output. This is mandatory for **named-compilezone**.

**-r** mode

This option checks for records that are treated as different by DNSSEC but are semantically equal in plain DNS. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-s** style

This option specifies the style of the dumped zone file. Possible styles are `full` (the default) and `relative`. The `full` format is most suitable for processing automatically by a separate script. The relative format is more human-readable and is thus suitable for editing by hand.

**-S** mode

This option checks whether an SRV record refers to a CNAME. Possible modes are `fail`, `warn`, and `ignore` (the default).

**-t** directory

This option tells *named* to chroot to `directory`, so that `include` directives in the configuration file are processed as if run by a similarly chrooted *named*.

**-T** mode

This option checks whether Sender Policy Framework (SPF) records exist and issues a warning if an SPF-formatted TXT record is not also present. Possible modes are `warn` and `ignore` (the default).

**-w** directory

This option instructs *named* to chdir to `directory`, so that relative filenames in master file `$INCLUDE` directives work. This is similar to the directory clause in *named.conf*.

**-D**

This option dumps the zone file in canonical format. This is always enabled for **named-compilezone**.

**-W** mode

This option specifies whether to check for non-terminal wildcards. Non-terminal wildcards are almost always the result of a failure to understand the wildcard matching algorithm (**RFC 4592**). Possible modes are `warn` and `ignore` (the default).

**zonename**

This indicates the domain name of the zone being checked.

**filename**

This is the name of the zone file.

### 12.21.4 Return Values

**named-compilezone** returns an exit status of 1 if errors were detected and 0 otherwise.

### 12.21.5 See Also

*named(8)*, *named-checkconf(8)*, *named-checkzone(8)*, **RFC 1035**, BIND 9 Administrator Reference Manual.

## 12.22 named-journalprint - print zone journal in human-readable form

### 12.22.1 Synopsis

**named-journalprint** [-c serial] [**-dux**] {journal}

### 12.22.2 Description

**named-journalprint** scans the contents of a zone journal file, printing it in a human-readable form, or, optionally, converting it to a different journal file format.

Journal files are automatically created by *named* when changes are made to dynamic zones (e.g., by *nsupdate*). They record each addition or deletion of a resource record, in binary format, allowing the changes to be re-applied to the zone when the server is restarted after a shutdown or crash. By default, the name of the journal file is formed by appending the extension `.jnl` to the name of the corresponding zone file.

**named-journalprint** converts the contents of a given journal file into a human-readable text format. Each line begins with `add` or `del`, to indicate whether the record was added or deleted, and continues with the resource record in master-file format.

The `-c` (compact) option provides a mechanism to reduce the size of a journal by removing (most/all) transactions prior to the specified serial number. Note: this option *must not* be used while *named* is running, and can cause data loss if the zone file has not been updated to contain the data being removed from the journal. Use with extreme caution.

The `-x` option causes additional data about the journal file to be printed at the beginning of the output and before each group of changes.

The `-u` (upgrade) and `-d` (downgrade) options recreate the journal file with a modified format version. The existing journal file is replaced. `-d` writes out the journal in the format used by versions of BIND up to 9.16.11; `-u` writes it out in the format used by versions since 9.16.13. (9.16.12 is omitted due to a journal-formatting bug in that release.) Note that these options *must not* be used while *named* is running.

### 12.22.3 See Also

*named(8)*, *nsupdate(1)*, BIND 9 Administrator Reference Manual.

## 12.23 named-nzd2nzf - convert an NZD database to NZF text format

### 12.23.1 Synopsis

**named-nzd2nzf** {filename}

### 12.23.2 Description

**named-nzd2nzf** converts an NZD database to NZF format and prints it to standard output. This can be used to review the configuration of zones that were added to *named* via *rndc addzone*. It can also be used to restore the old file format when rolling back from a newer version of BIND to an older version.

### 12.23.3 Arguments

**filename**
: This is the name of the `.nzd` file whose contents should be printed.

### 12.23.4 See Also

BIND 9 Administrator Reference Manual.

## 12.24 named-rrchecker - syntax checker for individual DNS resource records

### 12.24.1 Synopsis

**named-rrchecker** [**-h**] [**-o** origin] [**-p**] [**-u**] [**-C**] [**-T**] [**-P**]

### 12.24.2 Description

**named-rrchecker** reads a single DNS resource record (RR) from standard input and checks whether it is syntactically correct.

**The input format is a minimal subset of the DNS zone file format. The entire input must be:**
: CLASS TYPE RDATA

- Input must not start with an owner (domain) name
- The *CLASS* field is mandatory (typically `IN`).
- The *TTL* field **must not** be present.
- RDATA format is specific to each RRTYPE.
- Leading and trailing whitespace in each field is ignored.

Format details can be found in **RFC 1035 Section 5.1** under `<rr>` specification. **RFC 3597** format is also accepted in any of the input fields. See *Examples*.

### 12.24.3 Options

**-o** `origin`
: This option specifies the origin to be used when interpreting names in the record: it defaults to root (.). The specified origin is always taken as an absolute name.

**-p**

This option prints out the resulting record in canonical form. If there is no canonical form defined, the record is printed in **RFC 3597** unknown record format.

**-u**

This option prints out the resulting record in **RFC 3597** unknown record format.

**-C, -T, -P**

These options do not read input. They print out known classes, standard types, and private type mnemonics. Each item is printed on a separate line. The resulting list of private types may be empty

**-h**

This option prints out the help menu.

### 12.24.4 Examples

Pay close attention to the `echo` command line options *-e* and *-n*, as they affect whitespace in the input to `named-rrchecker`.

**echo -n 'IN A 192.0.2.1' | named-rrchecker**

- Valid input is in **RFC 1035** format with no newline at the end of the input.
- Return code 0.

**echo -e '\n \n IN\tA 192.0.2.1 \t \n\n ' | named-rrchecker -p**

- Valid input with leading and trailing whitespace.
- Output: `IN A 192.0.2.1`
- Leading and trailing whitespace is not part of the output.

#### Relative names and origin

**echo 'IN CNAME target' | named-rrchecker -p**

- Valid input with a relative name as the CNAME target.
- Output: `IN CNAME target.`
- Relative name *target* from the input is converted to an absolute name using the default origin `.` (root).

**echo 'IN CNAME target' | named-rrchecker -p -o origin.test**

- Valid input with a relative name as the CNAME target.
- Output: `IN CNAME target.origin.test.`
- Relative name *target* from the input is converted to an absolute name using the specified origin `origin.test`

**echo 'IN CNAME target.' | named-rrchecker -p -o origin.test**

- Valid input with an absolute name as the CNAME target.
- Output: `IN CNAME target.`
- The specified origin has no influence if *target* from the input is already absolute.

### Special characters

Special characters allowed in zone files by RFC 1035 Section 5.1 are accepted.

**echo 'IN CNAME t\097r\get\.' | named-rrchecker -p -o origin.test**

- Valid input with backslash escapes.
- Output: `IN CNAME target\..origin.test.`
- `\097` denotes an ASCII value in decimal, which, in this example, is the character `a`.
- `\g` is converted to a plain `g` because the `g` character does not have a special meaning and so the `\` prefix does nothing in this case.
- `\.` denotes a literal ASCII dot (here as a part of the CNAME target name). Special meaning of `.` as the DNS label separator was disabled by the preceding `\` prefix.

**echo 'IN CNAME @' | named-rrchecker -p -o origin.test**

- Valid input with `@` used as a reference to the specified origin.
- Output: `IN CNAME origin.test.`

**echo 'IN CNAME \@' | named-rrchecker -p -o origin.test**

- Valid input with a literal `@` character (escaped).
- Output: `IN CNAME \@.origin.test.`

**echo 'IN CNAME prefix.@' | named-rrchecker -p -o origin.test**

- Valid input with `@` used as a reference to the specifed origin.
- Output: `IN CNAME prefix.\@.origin.test.`
- `@` has special meaning only if it is free-standing.

**echo 'IN A 192.0.2.1; comment' | named-rrchecker -p**

- Valid input with a trailing comment. Note the lack of whitespace before the start of the comment.
- Output: `IN A 192.0.2.1`

For multi-line examples see the next section.

### Multi-token records

**echo -e 'IN TXT two words \n' | named-rrchecker -p**

- Valid TXT RR with two unquoted words and trailing whitespace.
- Output: `IN TXT "two" "words"`
- Two unquoted words in the input are treated as two <*character-string*>s per RFC 1035 Section 3.3.14.
- Trailing whitespace is omitted from the last <*character-string*>.

**echo -e 'IN TXT "two words" \n' | named-rrchecker -p**

- Valid TXT RR with one *character-string* and trailing whitespace.
- Output: `IN TXT "two words"`

**echo -e 'IN TXT "problematic newline\n"' | named-rrchecker -p**

- Invalid input - the closing `"` is not detected before the end of the line.

**echo 'IN TXT "with newline\010"' | named-rrchecker -p**

- Valid input with an escaped newline character inside *character-string*.
- Output: `IN TXT "with newline\010"`

**echo -e 'IN TXT ( two\nwords )' | named-rrchecker -p**

- Valid multi-line input with line continuation allowed inside optional parentheses in the RDATA field.
- Output: `IN TXT "two" "words"`

**echo -e 'IN TXT ( two\nwords ; misplaced comment )' | named-rrchecker -p**

- Invalid input - comments, starting with ";", are ignored by the parser, so the closing parenthesis should be before the semicolon.

**echo -e 'IN TXT ( two\nwords ; a working comment\n )' | named-rrchecker -p**

- Valid input - the comment is terminated with a newline.
- Output: `IN TXT "two" "words"`

**echo 'IN HTTPS 1 . alpn="h2,h3"' | named-rrchecker -p**

- Valid HTTPS record
- Output: `IN HTTPS 1 . alpn="h2,h3"`

**echo -e 'IN HTTPS ( 1 \n . \n alpn="dot")port=853' | named-rrchecker -p**

- Valid HTTPS record with individual sub-fields split across multiple lines using **RFC 1035 Section 5.1** parentheses syntax to group data that crosses a line boundary.
- Note the missing whitespace between the closing parenthesis and adjacent tokens.
- Output: `IN HTTPS 1 . alpn="dot" port=853`

### Unknown type handling

**echo 'IN A 192.0.2.1' | named-rrchecker -u**

- Valid input in **RFC 1035** format.
- Output in **RFC 3957** format: `CLASS1 TYPE1 \# 4 C0000201`

**echo 'CLASS1 TYPE1 \# 4 C0000201' | named-rrchecker -p**

- Valid input in **RFC 3597** format.
- Output in **RFC 1035** format: `IN A 192.0.2.1`

**echo 'IN A \# 4 C0000201' | named-rrchecker -p**

- Valid input with class and type in **RFC 1035** format and rdata in **RFC 3597** format.
- Output in **RFC 1035** format: `IN A 192.0.2.1`

**echo 'IN HTTPS 1 . key3=\001\000' | named-rrchecker -p**

- Valid input with **RFC 9460** syntax for an unknown *key3* field. Syntax `\001\000` produces two octets with values 1 and 0, respectively.
- Output: `IN HTTPS 1 . port=256`
- *key3* matches the standardized key name *port*.
- Octets 1 and 0 were decoded as integer values in big-endian encoding.

**echo 'IN HTTPS 1 . key3=\001' | named-rrchecker -p**

- Invalid input - the length of the value for *key3* (i.e. port) does not match the known standard format for that parameter in the SVCB RRTYPE.

**echo 'IN HTTPS 1 . port=\001\000' | named-rrchecker -p**

- Invalid input - the key *port*, when specified using its standard mnemonic name, **must** use standard key-specific syntax.

#### Meta values

**echo 'IN AXFR' | named-rrchecker**

- Invalid input - AXFR is a meta type, not a genuine RRTYPE.

**echo 'ANY A 192.0.2.1' | named-rrchecker**

- Invalid input - ANY is meta class, not a true class.

**echo 'A 192.0.2.1' | named-rrchecker**

- Invalid input - the class field is missing, so the parser would try and fail to interpret the RRTYPE A as the class.

### 12.24.5 Return Codes

**0**
The whole input was parsed as one syntactically valid resource record.

**1**
The input is not a syntactically valid resource record, or the given type is not supported, or either/both class and type are meta-values, which should not appear in zone files.

### 12.24.6 See Also

**RFC 1034**, **RFC 1035**, **RFC 3957**, *named(8)*.

## 12.25 named.conf - configuration file for named

### 12.25.1 Synopsis

**named.conf**

### 12.25.2 Description

named.conf is the configuration file for *named*.

For complete documentation about the configuration statements, please refer to the Configuration Reference section in the BIND 9 Administrator Reference Manual.

Statements are enclosed in braces and terminated with a semi-colon. Clauses in the statements are also semi-colon terminated. The usual comment styles are supported:

C style: /* */

C++ style: // to end of line

Unix style: # to end of line

```
acl <string> { <address_match_element>; ... }; // may occur multiple times

controls {
        inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ] allow
↪{ <address_match_element>; ... } [ keys { <string>; ... } ] [ read-only <boolean>␣
↪]; // may occur multiple times
        unix <quoted_string> perm <integer> owner <integer> group <integer> [ keys {
↪<string>; ... } ] [ read-only <boolean> ]; // may occur multiple times
}; // may occur multiple times

dlz <string> {
        database <string>;
        search <boolean>;
}; // may occur multiple times

dnssec-policy <string> {
        cdnskey <boolean>;
        cds-digest-types { <string>; ... };
        dnskey-ttl <duration>;
        inline-signing <boolean>;
        keys { ( csk | ksk | zsk ) [ key-directory | key-store <string> ] lifetime
↪<duration_or_unlimited> algorithm <string> [ tag-range <integer> <integer> ] [
↪<integer> ]; ... };
        manual-mode <boolean>;
        max-zone-ttl <duration>;
        nsec3param [ iterations <integer> ] [ optout <boolean> ] [ salt-length
↪<integer> ];
        offline-ksk <boolean>;
        parent-ds-ttl <duration>;
        parent-propagation-delay <duration>;
        publish-safety <duration>;
        purge-keys <duration>;
        retire-safety <duration>;
        signatures-jitter <duration>;
        signatures-refresh <duration>;
        signatures-validity <duration>;
        signatures-validity-dnskey <duration>;
        zone-propagation-delay <duration>;
}; // may occur multiple times

dyndb <string> <quoted_string> { <unspecified-text> }; // may occur multiple times

http <string> {
        endpoints { <quoted_string>; ... };
        listener-clients <integer>;
        streams-per-connection <integer>;
}; // may occur multiple times

key <string> {
        algorithm <string>;
        secret <string>;
}; // may occur multiple times
```

(continues on next page)

(continued from previous page)

```
key-store <string> {
        directory <string>;
        pkcs11-uri <quoted_string>;
}; // may occur multiple times

logging {
        category <string> { <string>; ... }; // may occur multiple times
        channel <string> {
                buffered <boolean>;
                file <quoted_string> [ versions ( unlimited | <integer> ) ] [ size
↪<size> ] [ suffix ( increment | timestamp ) ];
                null;
                print-category <boolean>;
                print-severity <boolean>;
                print-time ( iso8601 | iso8601-utc | local | <boolean> );
                severity <log_severity>;
                stderr;
                syslog [ <syslog_facility> ];
        }; // may occur multiple times
};

managed-keys { <string> ( static-key | initial-key | static-ds | initial-ds )
↪<integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times,␣
↪deprecated

options {
        allow-new-zones <boolean>;
        allow-notify { <address_match_element>; ... };
        allow-proxy { <address_match_element>; ... }; // experimental
        allow-proxy-on { <address_match_element>; ... }; // experimental
        allow-query { <address_match_element>; ... };
        allow-query-cache { <address_match_element>; ... };
        allow-query-cache-on { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-recursion { <address_match_element>; ... };
        allow-recursion-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6␣
↪( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        answer-cookie <boolean>;
        attach-cache <string>;
        auth-nxdomain <boolean>;
        automatic-interface-scan <boolean>;
        avoid-v4-udp-ports { <portrange>; ... }; // deprecated
        avoid-v6-udp-ports { <portrange>; ... }; // deprecated
        bindkeys-file <quoted_string>; // test only
        blackhole { <address_match_element>; ... };
        catalog-zones { zone <string> [ default-primaries [ port <integer> ] [ source␣
```

(continues on next page)

(continued from previous page)

```
↪( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> |
↪<ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key
↪<string> ] [ tls <string> ]; ... } ] [ zone-directory <quoted_string> ] [ in-memory
↪<boolean> ] [ min-update-interval <duration> ]; ... };
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( primary | master | secondary | slave | response ) ( fail | warn␣
↪| ignore ); // may occur multiple times
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        clients-per-query <integer>;
        cookie-algorithm ( siphash24 );
        cookie-secret <string>; // may occur multiple times
        deny-answer-addresses { <address_match_element>; ... } [ except-from {
↪<string>; ... } ];
        deny-answer-aliases { <string>; ... } [ except-from { <string>; ... } ];
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        directory <quoted_string>;
        disable-algorithms <string> { <string>; ... }; // may occur multiple times
        disable-ds-digests <string> { <string>; ... }; // may occur multiple times
        disable-empty-zone <string>; // may occur multiple times
        dns64 <netprefix> {
                break-dnssec <boolean>;
                clients { <address_match_element>; ... };
                exclude { <address_match_element>; ... };
                mapped { <address_match_element>; ... };
                recursive-only <boolean>;
                suffix <ipv6_address>;
        }; // may occur multiple times
        dns64-contact <string>;
        dns64-server <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnsrps-enable <boolean>; // not configured
        dnsrps-library <quoted_string>; // not configured
        dnsrps-options { <unspecified-text> }; // not configured
        dnssec-accept-expired <boolean>;
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-must-be-secure <string> <boolean>; // may occur multiple times,␣
↪deprecated
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        dnssec-validation ( yes | no | auto );
        dnstap { ( all | auth | client | forwarder | resolver | update ) [ ( query |␣
↪response ) ]; ... }; // not configured
```

(continues on next page)

(continued from previous page)

```
        dnstap-identity ( <quoted_string> | none | hostname ); // not configured
        dnstap-output ( file | unix ) <quoted_string> [ size ( unlimited | <size> ) ]␣
↪[ versions ( unlimited | <integer> ) ] [ suffix ( increment | timestamp ) ]; // not␣
↪configured
        dnstap-version ( <quoted_string> | none ); // not configured
        dual-stack-servers [ port <integer> ] { ( <quoted_string> [ port <integer> ]␣
↪| <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ); ... };
        dump-file <quoted_string>;
        edns-udp-size <integer>;
        empty-contact <string>;
        empty-server <string>;
        empty-zones-enable <boolean>;
        fetch-quota-params <integer> <fixedpoint> <fixedpoint> <fixedpoint>;
        fetches-per-server <integer> [ ( drop | fail ) ];
        fetches-per-zone <integer> [ ( drop | fail ) ];
        flush-zones-on-shutdown <boolean>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        fstrm-set-buffer-hint <integer>; // not configured
        fstrm-set-flush-timeout <integer>; // not configured
        fstrm-set-input-queue-size <integer>; // not configured
        fstrm-set-output-notify-threshold <integer>; // not configured
        fstrm-set-output-queue-model ( mpsc | spsc ); // not configured
        fstrm-set-output-queue-size <integer>; // not configured
        fstrm-set-reopen-interval <duration>; // not configured
        geoip-directory ( <quoted_string> | none );
        heartbeat-interval <integer>; // deprecated
        hostname ( <quoted_string> | none );
        http-listener-clients <integer>;
        http-port <integer>;
        http-streams-per-connection <integer>;
        https-port <integer>;
        interface-interval <duration>;
        ipv4only-contact <string>;
        ipv4only-enable <boolean>;
        ipv4only-server <string>;
        ixfr-from-differences ( primary | master | secondary | slave | <boolean> );
        keep-response-order { <address_match_element>; ... }; // obsolete
        key-directory <quoted_string>;
        lame-ttl <duration>;
        listen-on [ port <integer> ] [ proxy <string> ] [ tls <string> ] [ http
↪<string> ] { <address_match_element>; ... }; // may occur multiple times
        listen-on-v6 [ port <integer> ] [ proxy <string> ] [ tls <string> ] [ http
↪<string> ] { <address_match_element>; ... }; // may occur multiple times
        lmdb-mapsize <sizeval>;
        managed-keys-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        match-mapped-addresses <boolean>;
        max-cache-size ( default | unlimited | <sizeval> | <percentage> );
        max-cache-ttl <duration>;
```

(continues on next page)

(continued from previous page)

```
        max-clients-per-query <integer>;
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-ncache-ttl <duration>;
        max-query-count <integer>;
        max-query-restarts <integer>;
        max-records <integer>;
        max-records-per-type <integer>;
        max-recursion-depth <integer>;
        max-recursion-queries <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-rsa-exponent-size <integer>;
        max-stale-ttl <duration>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-udp-size <integer>;
        max-validation-failures-per-fetch <integer>; // experimental
        max-validations-per-fetch <integer>; // experimental
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        memstatistics <boolean>;
        memstatistics-file <quoted_string>;
        message-compression <boolean>;
        min-cache-ttl <duration>;
        min-ncache-ttl <duration>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        minimal-any <boolean>;
        minimal-responses ( no-auth | no-auth-recursive | <boolean> );
        multi-master <boolean>;
        new-zones-directory <quoted_string>;
        no-case-compress { <address_match_element>; ... };
        nocookie-udp-size <integer>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-rate <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        nta-lifetime <duration>;
        nta-recheck <duration>;
        nxdomain-redirect <string>;
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        pid-file ( <quoted_string> | none );
        port <integer>;
```

(continues on next page)

(continued from previous page)

```
        preferred-glue <string>;
        prefetch <integer> [ <integer> ];
        provide-ixfr <boolean>;
        qname-minimization ( strict | relaxed | disabled | off );
        query-source [ address ] ( <ipv4_address> | * | none );
        query-source-v6 [ address ] ( <ipv6_address> | * | none );
        querylog <boolean>;
        rate-limit {
                all-per-second <integer>;
                errors-per-second <integer>;
                exempt-clients { <address_match_element>; ... };
                ipv4-prefix-length <integer>;
                ipv6-prefix-length <integer>;
                log-only <boolean>;
                max-table-size <integer>;
                min-table-size <integer>;
                nodata-per-second <integer>;
                nxdomains-per-second <integer>;
                qps-scale <integer>;
                referrals-per-second <integer>;
                responses-per-second <integer>;
                slip <integer>;
                window <integer>;
        };
        recursing-file <quoted_string>;
        recursion <boolean>;
        recursive-clients <integer>;
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-server-cookie <boolean>;
        resolver-query-timeout <integer>;
        resolver-use-dns64 <boolean>;
        response-padding { <address_match_element>; ... } block-size <integer>;
        response-policy { zone <string> [ add-soa <boolean> ] [ log <boolean> ] [␣
↪max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ policy ( cname |␣
↪disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only
↪<quoted_string> ) ] [ recursive-only <boolean> ] [ nsip-enable <boolean> ] [␣
↪nsdname-enable <boolean> ] [ ede <string> ]; ... } [ add-soa <boolean> ] [␣
↪break-dnssec <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval
↪<duration> ] [ min-ns-dots <integer> ] [ nsip-wait-recurse <boolean> ] [␣
↪nsdname-wait-recurse <boolean> ] [ qname-wait-recurse <boolean> ] [ recursive-only
↪<boolean> ] [ servfail-until-ready <boolean> ] [ nsip-enable <boolean> ] [␣
↪nsdname-enable <boolean> ] [ dnsrps-enable <boolean> ] [ dnsrps-options {
↪<unspecified-text> } ];
        responselog <boolean>;
        reuseport <boolean>;
        root-key-sentinel <boolean>;
        rrset-order { [ class <string> ] [ type <string> ] [ name <quoted_string> ]
↪<string> <string>; ... };
        secroots-file <quoted_string>;
        send-cookie <boolean>;
```

(continues on next page)

(continued from previous page)

```
        serial-query-rate <integer>;
        serial-update-method ( date | increment | unixtime );
        server-id ( <quoted_string> | none | hostname );
        servfail-ttl <duration>;
        session-keyalg <string>;
        session-keyfile ( <quoted_string> | none );
        session-keyname <string>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        sig0checks-quota <integer>; // experimental
        sig0checks-quota-exempt { <address_match_element>; ... }; // experimental
        sig0key-checks-limit <integer>;
        sig0message-checks-limit <integer>;
        sortlist { <address_match_element>; ... }; // deprecated
        stale-answer-client-timeout ( disabled | off | <integer> );
        stale-answer-enable <boolean>;
        stale-answer-ttl <duration>;
        stale-cache-enable <boolean>;
        stale-refresh-time <duration>;
        startup-notify-rate <integer>;
        statistics-file <quoted_string>;
        synth-from-dnssec <boolean>;
        tcp-advertised-timeout <integer>;
        tcp-clients <integer>;
        tcp-idle-timeout <integer>;
        tcp-initial-timeout <integer>;
        tcp-keepalive-timeout <integer>;
        tcp-listen-queue <integer>;
        tcp-receive-buffer <integer>;
        tcp-reuse-timeout <integer>;
        tcp-send-buffer <integer>;
        tkey-domain <quoted_string>; // obsolete
        tkey-gssapi-credential <quoted_string>; // deprecated
        tkey-gssapi-keytab <quoted_string>;
        tls-port <integer>;
        transfer-format ( many-answers | one-answer );
        transfer-message-size <integer>;
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        transfers-in <integer>;
        transfers-out <integer>;
        transfers-per-ns <integer>;
        trust-anchor-telemetry <boolean>;
        try-tcp-refresh <boolean>;
        udp-receive-buffer <integer>;
        udp-send-buffer <integer>;
        update-check-ksk <boolean>; // obsolete
        update-quota <integer>;
        use-v4-udp-ports { <portrange>; ... }; // deprecated
        use-v6-udp-ports { <portrange>; ... }; // deprecated
```

(continues on next page)

(continued from previous page)

```
        v6-bias <integer>;
        validate-except { <string>; ... };
        version ( <quoted_string> | none );
        zero-no-soa-ttl <boolean>;
        zero-no-soa-ttl-cache <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};

plugin ( query ) <string> [ { <unspecified-text> } ]; // may occur multiple times

remote-servers <string> [ port <integer> ] [ source ( <ipv4_address> | * ) ] [␣
↪source-v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port
↪<integer> ] | <ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string>␣
↪]; ... }; // may occur multiple times

server <netprefix> {
        bogus <boolean>;
        edns <boolean>;
        edns-udp-size <integer>;
        edns-version <integer>;
        keys <server_key>;
        max-udp-size <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        padding <integer>;
        provide-ixfr <boolean>;
        query-source [ address ] ( <ipv4_address> | * );
        query-source-v6 [ address ] ( <ipv6_address> | * );
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-cookie <boolean>;
        send-cookie <boolean>;
        tcp-keepalive <boolean>;
        tcp-only <boolean>;
        transfer-format ( many-answers | one-answer );
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        transfers <integer>;
}; // may occur multiple times

statistics-channels {
        inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | * ) ] [␣
↪allow { <address_match_element>; ... } ]; // may occur multiple times
}; // may occur multiple times

tls <string> {
        ca-file <quoted_string>;
        cert-file <quoted_string>;
        cipher-suites <string>;
        ciphers <string>;
        dhparam-file <quoted_string>;
```

(continues on next page)

(continued from previous page)

```
        key-file <quoted_string>;
        prefer-server-ciphers <boolean>;
        protocols { <string>; ... };
        remote-hostname <quoted_string>;
        session-tickets <boolean>;
}; // may occur multiple times

trust-anchors { <string> ( static-key | initial-key | static-ds | initial-ds )
↪<integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times

trusted-keys { <string> <integer> <integer> <integer> <quoted_string>; ... }; // may␣
↪occur multiple times, deprecated

view <string> [ <class> ] {
        allow-new-zones <boolean>;
        allow-notify { <address_match_element>; ... };
        allow-proxy { <address_match_element>; ... }; // experimental
        allow-proxy-on { <address_match_element>; ... }; // experimental
        allow-query { <address_match_element>; ... };
        allow-query-cache { <address_match_element>; ... };
        allow-query-cache-on { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-recursion { <address_match_element>; ... };
        allow-recursion-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6␣
↪( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        attach-cache <string>;
        auth-nxdomain <boolean>;
        catalog-zones { zone <string> [ default-primaries [ port <integer> ] [ source␣
↪( <ipv4_address> | * ) ] [ source-v6 ( <ipv6_address> | * ) ] { ( <server-list> |
↪<ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ) [ key
↪<string> ] [ tls <string> ]; ... } ] [ zone-directory <quoted_string> ] [ in-memory
↪<boolean> ] [ min-update-interval <duration> ]; ... };
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( primary | master | secondary | slave | response ) ( fail | warn␣
↪| ignore ); // may occur multiple times
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        clients-per-query <integer>;
        deny-answer-addresses { <address_match_element>; ... } [ except-from {
↪<string>; ... } ];
```

(continues on next page)

(continued from previous page)

```
        deny-answer-aliases { <string>; ... } [ except-from { <string>; ... } ];
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        disable-algorithms <string> { <string>; ... }; // may occur multiple times
        disable-ds-digests <string> { <string>; ... }; // may occur multiple times
        disable-empty-zone <string>; // may occur multiple times
        dlz <string> {
                database <string>;
                search <boolean>;
        }; // may occur multiple times
        dns64 <netprefix> {
                break-dnssec <boolean>;
                clients { <address_match_element>; ... };
                exclude { <address_match_element>; ... };
                mapped { <address_match_element>; ... };
                recursive-only <boolean>;
                suffix <ipv6_address>;
        }; // may occur multiple times
        dns64-contact <string>;
        dns64-server <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnsrps-enable <boolean>; // not configured
        dnsrps-options { <unspecified-text> }; // not configured
        dnssec-accept-expired <boolean>;
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-must-be-secure <string> <boolean>; // may occur multiple times,␣
↪deprecated
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        dnssec-validation ( yes | no | auto );
        dnstap { ( all | auth | client | forwarder | resolver | update ) [ ( query |␣
↪response ) ]; ... }; // not configured
        dual-stack-servers [ port <integer> ] { ( <quoted_string> [ port <integer> ]␣
↪| <ipv4_address> [ port <integer> ] | <ipv6_address> [ port <integer> ] ); ... };
        dyndb <string> <quoted_string> { <unspecified-text> }; // may occur multiple␣
↪times
        edns-udp-size <integer>;
        empty-contact <string>;
        empty-server <string>;
        empty-zones-enable <boolean>;
        fetch-quota-params <integer> <fixedpoint> <fixedpoint> <fixedpoint>;
        fetches-per-server <integer> [ ( drop | fail ) ];
        fetches-per-zone <integer> [ ( drop | fail ) ];
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        ipv4only-contact <string>;
        ipv4only-enable <boolean>;
        ipv4only-server <string>;
        ixfr-from-differences ( primary | master | secondary | slave | <boolean> );
```

(continues on next page)

```
        key <string> {
                algorithm <string>;
                secret <string>;
        }; // may occur multiple times
        key-directory <quoted_string>;
        lame-ttl <duration>;
        lmdb-mapsize <sizeval>;
        managed-keys { <string> ( static-key | initial-key | static-ds | initial-ds )
↪<integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times,␣
↪deprecated
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        match-clients { <address_match_element>; ... };
        match-destinations { <address_match_element>; ... };
        match-recursive-only <boolean>;
        max-cache-size ( default | unlimited | <sizeval> | <percentage> );
        max-cache-ttl <duration>;
        max-clients-per-query <integer>;
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-ncache-ttl <duration>;
        max-query-count <integer>;
        max-query-restarts <integer>;
        max-records <integer>;
        max-records-per-type <integer>;
        max-recursion-depth <integer>;
        max-recursion-queries <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-stale-ttl <duration>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-udp-size <integer>;
        max-validation-failures-per-fetch <integer>; // experimental
        max-validations-per-fetch <integer>; // experimental
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        message-compression <boolean>;
        min-cache-ttl <duration>;
        min-ncache-ttl <duration>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        minimal-any <boolean>;
        minimal-responses ( no-auth | no-auth-recursive | <boolean> );
        multi-master <boolean>;
        new-zones-directory <quoted_string>;
        no-case-compress { <address_match_element>; ... };
        nocookie-udp-size <integer>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
```

(continues on next page)

(continued from previous page)

```
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        nta-lifetime <duration>;
        nta-recheck <duration>;
        nxdomain-redirect <string>;
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        plugin ( query ) <string> [ { <unspecified-text> } ]; // may occur multiple␣
↪times
        preferred-glue <string>;
        prefetch <integer> [ <integer> ];
        provide-ixfr <boolean>;
        qname-minimization ( strict | relaxed | disabled | off );
        query-source [ address ] ( <ipv4_address> | * | none );
        query-source-v6 [ address ] ( <ipv6_address> | * | none );
        rate-limit {
                all-per-second <integer>;
                errors-per-second <integer>;
                exempt-clients { <address_match_element>; ... };
                ipv4-prefix-length <integer>;
                ipv6-prefix-length <integer>;
                log-only <boolean>;
                max-table-size <integer>;
                min-table-size <integer>;
                nodata-per-second <integer>;
                nxdomains-per-second <integer>;
                qps-scale <integer>;
                referrals-per-second <integer>;
                responses-per-second <integer>;
                slip <integer>;
                window <integer>;
        };
        recursion <boolean>;
        request-expire <boolean>;
        request-ixfr <boolean>;
        request-nsid <boolean>;
        require-server-cookie <boolean>;
        resolver-query-timeout <integer>;
        resolver-use-dns64 <boolean>;
        response-padding { <address_match_element>; ... } block-size <integer>;
        response-policy { zone <string> [ add-soa <boolean> ] [ log <boolean> ] [␣
↪max-policy-ttl <duration> ] [ min-update-interval <duration> ] [ policy ( cname |␣
↪disabled | drop | given | no-op | nodata | nxdomain | passthru | tcp-only
↪<quoted_string> ) ] [ recursive-only <boolean> ] [ nsip-enable <boolean> ] [␣
↪nsdname-enable <boolean> ] [ ede <string> ]; ... } [ add-soa <boolean> ] [␣
↪break-dnssec <boolean> ] [ max-policy-ttl <duration> ] [ min-update-interval
↪<duration> ] [ min-ns-dots <integer> ] [ nsip-wait-recurse <boolean> ] [␣
↪nsdname-wait-recurse <boolean> ] [ qname-wait-recurse <boolean> ] [ recursive-only
↪<boolean> ] [ servfail-until-ready <boolean> ] [ nsip-enable <boolean> ] [␣
```

(continues on next page)

(continued from previous page)

```
↪nsdname-enable <boolean> ] [ dnsrps-enable <boolean> ] [ dnsrps-options {
↪<unspecified-text> } ];
        root-key-sentinel <boolean>;
        rrset-order { [ class <string> ] [ type <string> ] [ name <quoted_string> ]
↪<string> <string>; ... };
        send-cookie <boolean>;
        serial-update-method ( date | increment | unixtime );
        server <netprefix> {
                bogus <boolean>;
                edns <boolean>;
                edns-udp-size <integer>;
                edns-version <integer>;
                keys <server_key>;
                max-udp-size <integer>;
                notify-source ( <ipv4_address> | * );
                notify-source-v6 ( <ipv6_address> | * );
                padding <integer>;
                provide-ixfr <boolean>;
                query-source [ address ] ( <ipv4_address> | * );
                query-source-v6 [ address ] ( <ipv6_address> | * );
                request-expire <boolean>;
                request-ixfr <boolean>;
                request-nsid <boolean>;
                require-cookie <boolean>;
                send-cookie <boolean>;
                tcp-keepalive <boolean>;
                tcp-only <boolean>;
                transfer-format ( many-answers | one-answer );
                transfer-source ( <ipv4_address> | * );
                transfer-source-v6 ( <ipv6_address> | * );
                transfers <integer>;
        }; // may occur multiple times
        servfail-ttl <duration>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        sig0key-checks-limit <integer>;
        sig0message-checks-limit <integer>;
        sortlist { <address_match_element>; ... }; // deprecated
        stale-answer-client-timeout ( disabled | off | <integer> );
        stale-answer-enable <boolean>;
        stale-answer-ttl <duration>;
        stale-cache-enable <boolean>;
        stale-refresh-time <duration>;
        synth-from-dnssec <boolean>;
        transfer-format ( many-answers | one-answer );
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        trust-anchor-telemetry <boolean>;
        trust-anchors { <string> ( static-key | initial-key | static-ds | initial-ds␣
↪) <integer> <integer> <integer> <quoted_string>; ... }; // may occur multiple times
```

(continues on next page)

(continued from previous page)

```
        trusted-keys { <string> <integer> <integer> <integer> <quoted_string>; ... };␣
↪// may occur multiple times, deprecated
        try-tcp-refresh <boolean>;
        update-check-ksk <boolean>; // obsolete
        v6-bias <integer>;
        validate-except { <string>; ... };
        zero-no-soa-ttl <boolean>;
        zero-no-soa-ttl-cache <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
}; // may occur multiple times
```

Any of these zone statements can also be set inside the view statement.

```
zone <string> [ <class> ] {
        type primary;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6␣
↪( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        check-dup-records ( fail | warn | ignore );
        check-integrity <boolean>;
        check-mx ( fail | warn | ignore );
        check-mx-cname ( fail | warn | ignore );
        check-names ( fail | warn | ignore );
        check-sibling <boolean>;
        check-spf ( warn | ignore );
        check-srv-cname ( fail | warn | ignore );
        check-svcb <boolean>;
        check-wildcard <boolean>;
        checkds ( explicit | <boolean> );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        dlz <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-policy <string>;
        dnssec-secure-to-insecure <boolean>; // obsolete
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        file <quoted_string>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        inline-signing <boolean>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
```

(continues on next page)

(continued from previous page)

```
        key-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        parental-agents [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        serial-update-method ( date | increment | unixtime );
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        update-check-ksk <boolean>; // obsolete
        update-policy ( local | { ( deny | grant ) <string> ( 6to4-self | external |␣
↪krb5-self | krb5-selfsub | krb5-subdomain | krb5-subdomain-self-rhs | ms-self | ms-
↪selfsub | ms-subdomain | ms-subdomain-self-rhs | name | self | selfsub | selfwild |␣
↪subdomain | tcp-self | wildcard | zonesub ) [ <string> ] <rrtypelist>; ... } );
        zero-no-soa-ttl <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

```
zone <string> [ <class> ] {
        type secondary;
        allow-notify { <address_match_element>; ... };
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6␣
↪( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        check-names ( fail | warn | ignore );
        checkds ( explicit | <boolean> );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
```

(continues on next page)

(continued from previous page)

```
↪deprecated
        dlz <string>;
        dnskey-sig-validity <integer>; // obsolete
        dnssec-dnskey-kskonly <boolean>; // obsolete
        dnssec-loadkeys-interval <integer>;
        dnssec-policy <string>;
        dnssec-update-mode ( maintain | no-resign ); // obsolete
        file <quoted_string>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        inline-signing <boolean>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
        key-directory <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        notify-to-soa <boolean>;
        nsec3-test-zone <boolean>; // test only
        parental-agents [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-
↪v6 ( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        parental-source ( <ipv4_address> | * );
        parental-source-v6 ( <ipv6_address> | * );
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 (
↪<ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        request-expire <boolean>;
        request-ixfr <boolean>;
        sig-signing-nodes <integer>;
        sig-signing-signatures <integer>;
        sig-signing-type <integer>;
```

(continues on next page)

(continued from previous page)

```
        sig-validity-interval <integer> [ <integer> ]; // obsolete
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        try-tcp-refresh <boolean>;
        update-check-ksk <boolean>; // obsolete
        zero-no-soa-ttl <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

```
zone <string> [ <class> ] {
        type mirror;
        allow-notify { <address_match_element>; ... };
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        allow-transfer [ port <integer> ] [ transport <string> ] {
↪<address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        also-notify [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6␣
↪( <ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        check-names ( fail | warn | ignore );
        database <string>;
        file <quoted_string>;
        ixfr-from-differences <boolean>;
        journal <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-ixfr-ratio ( unlimited | <percentage> );
        max-journal-size ( default | unlimited | <sizeval> );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        notify ( explicit | master-only | primary-only | <boolean> );
        notify-defer <integer>;
        notify-delay <integer>;
        notify-source ( <ipv4_address> | * );
        notify-source-v6 ( <ipv6_address> | * );
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 (
↪<ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        request-expire <boolean>;
        request-ixfr <boolean>;
```

(continues on next page)

(continued from previous page)

```
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        try-tcp-refresh <boolean>;
        zero-no-soa-ttl <boolean>;
        zone-statistics ( full | terse | none | <boolean> );
};
```

```
zone <string> [ <class> ] {
        type forward;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
};
```

```
zone <string> [ <class> ] {
        type hint;
        check-names ( fail | warn | ignore );
        file <quoted_string>;
};
```

```
zone <string> [ <class> ] {
        type redirect;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        dlz <string>;
        file <quoted_string>;
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-records <integer>;
        max-records-per-type <integer>;
        max-types-per-name <integer>;
        max-zone-ttl ( unlimited | <duration> ); // deprecated
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 (
↪<ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        zone-statistics ( full | terse | none | <boolean> );
};
```

```
zone <string> [ <class> ] {
        type static-stub;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        max-records <integer>;
        max-records-per-type <integer>;
        max-types-per-name <integer>;
        server-addresses { ( <ipv4_address> | <ipv6_address> ); ... };
        server-names { <string>; ... };
        zone-statistics ( full | terse | none | <boolean> );
```

(continues on next page)

(continued from previous page)

```
};
```

```
zone <string> [ <class> ] {
        type stub;
        allow-query { <address_match_element>; ... };
        allow-query-on { <address_match_element>; ... };
        check-names ( fail | warn | ignore );
        database <string>;
        dialup ( notify | notify-passive | passive | refresh | <boolean> ); //␣
↪deprecated
        file <quoted_string>;
        forward ( first | only );
        forwarders [ port <integer> ] [ tls <string> ] { ( <ipv4_address> |
↪<ipv6_address> ) [ port <integer> ] [ tls <string> ]; ... };
        masterfile-format ( raw | text );
        masterfile-style ( full | relative );
        max-records <integer>;
        max-records-per-type <integer>;
        max-refresh-time <integer>;
        max-retry-time <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-time-in <integer>;
        max-types-per-name <integer>;
        min-refresh-time <integer>;
        min-retry-time <integer>;
        min-transfer-rate-in <integer> <integer>;
        multi-master <boolean>;
        primaries [ port <integer> ] [ source ( <ipv4_address> | * ) ] [ source-v6 (
↪<ipv6_address> | * ) ] { ( <server-list> | <ipv4_address> [ port <integer> ] |
↪<ipv6_address> [ port <integer> ] ) [ key <string> ] [ tls <string> ]; ... };
        transfer-source ( <ipv4_address> | * );
        transfer-source-v6 ( <ipv6_address> | * );
        zone-statistics ( full | terse | none | <boolean> );
};
```

```
zone <string> [ <class> ] {
        in-view <string>;
};
```

### 12.25.3 Files

`/etc/named.conf`

### 12.25.4 See Also

*named(8)*, *named-checkconf(8)*, *rndc(8)*, *rndc-confgen(8)*, *tsig-keygen(8)*, BIND 9 Administrator Reference Manual.

## 12.26 named - Internet domain name server

### 12.26.1 Synopsis

**named** [ [**-4**] | [**-6**] ] [**-c** config-file] [**-C**] [**-d** debug-level] [**-D** string] [**-E** engine-name] [**-f**] [**-g**] [**-L** logfile] [**-M** option] [**-m** flag] [**-n** #cpus] [**-p** port] [**-s**] [**-t** directory] [**-u** user] [**-v**] [**-V**] ]

### 12.26.2 Description

**named** is a Domain Name System (DNS) server, part of the BIND 9 distribution from ISC. For more information on the DNS, see **RFC 1033**, **RFC 1034**, and **RFC 1035**.

When invoked without arguments, **named** reads the default configuration file `/etc/named.conf`, reads any initial data, and listens for queries.

### 12.26.3 Options

**-4**

This option tells **named** to use only IPv4, even if the host machine is capable of IPv6. *-4* and *-6* are mutually exclusive.

**-6**

This option tells **named** to use only IPv6, even if the host machine is capable of IPv4. *-4* and *-6* are mutually exclusive.

**-c** `config-file`

This option tells **named** to use `config-file` as its configuration file instead of the default, `/etc/named.conf`. To ensure that the configuration file can be reloaded after the server has changed its working directory due to to a possible `directory` option in the configuration file, `config-file` should be an absolute pathname.

**-C**

This option prints out the default built-in configuration and exits.

NOTE: This is for debugging purposes only and is not an accurate representation of the actual configuration used by *named* at runtime.

**-d** `debug-level`

This option sets the daemon's debug level to `debug-level`. Debugging traces from **named** become more verbose as the debug level increases.

**-D** `string`

This option specifies a string that is used to identify a instance of **named** in a process listing. The contents of `string` are not examined.

**-E** `engine-name`

When applicable, this option specifies the hardware to use for cryptographic operations, such as a secure key store used for signing.

When BIND 9 is built with OpenSSL, this needs to be set to the OpenSSL engine identifier that drives the cryptographic accelerator or hardware service module (usually `pkcs11`).

**-f**

This option runs the server in the foreground (i.e., do not daemonize).

**-F**

This options turns on FIPS (US Federal Information Processing Standards) mode if the underlying crytographic library supports running in FIPS mode.

**-g**

This option runs the server in the foreground and forces all logging to `stderr`.

**-L** `logfile`

This option sets the log to the file `logfile` by default, instead of the system log.

**-M** `option`

This option sets the default (comma-separated) memory context options. The possible flags are:

- `fill`: fill blocks of memory with tag values when they are allocated or freed, to assist debugging of memory problems; this is the implicit default if **named** has been compiled with `--enable-developer`.
- `nofill`: disable the behavior enabled by `fill`; this is the implicit default unless **named** has been compiled with `--enable-developer`.

**-m** `flag`

This option turns on memory usage debugging flags. Possible flags are `usage`, `trace` and `record`. These correspond to the `ISC_MEM_DEBUGXXXX` flags described in `<isc/mem.h>`.

**-n** `#cpus`

This option creates `#cpus` worker threads to take advantage of multiple CPUs. If not specified, **named** tries to determine the number of CPUs present and creates one thread per CPU. If it is unable to determine the number of CPUs, a single worker thread is created.

**-p** `value`

This option specifies the port(s) on which the server will listen for queries. If `value` is of the form `<portnum>` or `dns=<portnum>`, the server will listen for DNS queries on `portnum`; if not not specified, the default is port 53. If `value` is of the form `tls=<portnum>`, the server will listen for TLS queries on `portnum`; the default is 853. If `value` is of the form `https=<portnum>`, the server will listen for HTTPS queries on `portnum`; the default is 443. If `value` is of the form `http=<portnum>`, the server will listen for HTTP queries on `portnum`; the default is 80.

**-s**

This option writes memory usage statistics to `stdout` on exit.

**Note**

This option is mainly of interest to BIND 9 developers and may be removed or changed in a future release.

**-t** `directory`

This option tells **named** to chroot to `directory` after processing the command-line arguments, but before reading the configuration file.

**Warning**

This option should be used in conjunction with the *-u* option, as chrooting a process running as root doesn't enhance security on most systems; the way `chroot` is defined allows a process with root privileges to escape a chroot jail.

**-U** `#listeners`
: This option has been removed. Attempts to use it now result in a warning.

**-u** `user`
: This option sets the setuid to `user` after completing privileged operations, such as creating sockets that listen on privileged ports.

> **Note**
>
> On Linux, **named** uses the kernel's capability mechanism to drop all root privileges except the ability to `bind` to a privileged port and set process resource limits. Unfortunately, this means that the *-u* option only works when **named** is run on kernel 2.2.18 or later, or kernel 2.3.99-pre3 or later, since previous kernels did not allow privileges to be retained after `setuid`.

**-v**
: This option reports the version number and exits.

**-V**
: This option reports the version number, build options, supported cryptographics algorithms, and exits.

**-X** `lock-file`
: This option has been removed and using it will cause a fatal error.

### 12.26.4 Signals

In routine operation, signals should not be used to control the nameserver; *rndc* should be used instead.

**SIGHUP**
: This signal forces a reload of the server.

**SIGINT, SIGTERM**
: These signals shut down the server.

The result of sending any other signals to the server is undefined.

### 12.26.5 Configuration

The **named** configuration file is too complex to describe in detail here. A complete description is provided in the BIND 9 Administrator Reference Manual.

**named** inherits the `umask` (file creation mode mask) from the parent process. If files created by **named**, such as journal files, need to have custom permissions, the `umask` should be set explicitly in the script used to start the **named** process.

### 12.26.6 Files

**/etc/named.conf**
: The default configuration file.

**/run/named.pid**
: The default process-id file.

### 12.26.7 See Also

**RFC 1033**, **RFC 1034**, **RFC 1035**, *named-checkconf(8)*, *named-checkzone(8)*, *rndc(8)*, *named.conf(5)*, BIND 9 Administrator Reference Manual.

## 12.27 nsec3hash - generate NSEC3 hash

### 12.27.1 Synopsis

**nsec3hash** {salt} {algorithm} {iterations} {domain}

**nsec3hash -r** {algorithm} {flags} {iterations} {salt} {domain}

### 12.27.2 Description

**nsec3hash** generates an NSEC3 hash based on a set of NSEC3 parameters. This can be used to check the validity of NSEC3 records in a signed zone.

If this command is invoked as `nsec3hash -r`, it takes arguments in order, matching the first four fields of an NSEC3 record followed by the domain name: `algorithm`, `flags`, `iterations`, `salt`, `domain`. This makes it convenient to copy and paste a portion of an NSEC3 or NSEC3PARAM record into a command line to confirm the correctness of an NSEC3 hash.

### 12.27.3 Arguments

**salt**

This is the salt provided to the hash algorithm.

**algorithm**

This is a number indicating the hash algorithm. Currently the only supported hash algorithm for NSEC3 is SHA-1, which is indicated by the number 1; consequently "1" is the only useful value for this argument.

**flags**

This is provided for compatibility with NSEC3 record presentation format, but is ignored since the flags do not affect the hash.

**iterations**

This is the number of additional times the hash should be performed.

**domain**

This is the domain name to be hashed.

### 12.27.4 See Also

BIND 9 Administrator Reference Manual, **RFC 5155**.

## 12.28 nslookup - query Internet name servers interactively

### 12.28.1 Synopsis

**nslookup** [-option] [name | -] [server]

### 12.28.2 Description

**nslookup** is a program to query Internet domain name servers. **nslookup** has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode prints just the name and requested information for a host or domain.

### 12.28.3 Arguments

Interactive mode is entered in the following cases:

a. when no arguments are given (the default name server is used);

b. when the first argument is a hyphen (-) and the second argument is the host name or Internet address of a name server.

Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

Options can also be specified on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information, with an initial timeout of 10 seconds, type:

```
nslookup -query=hinfo  -timeout=10
```

The `-version` option causes **nslookup** to print the version number and immediately exit.

### 12.28.4 Interactive Commands

**host [server]**
: This command looks up information for *host* using the current default server or using `server`, if specified. If *host* is an Internet address and the query type is A or PTR, the name of the host is returned. If *host* is a name and does not have a trailing period (`.`), the search list is used to qualify the name.

  To look up a host not in the current domain, append a period to the name.

**server domain|lserver domain**
: These commands change the default server to `domain`; `lserver` uses the initial server to look up information about `domain`, while `server` uses the current default server. If an authoritative answer cannot be found, the names of servers that might have the answer are returned.

**root**
: This command is not implemented.

**finger**
: This command is not implemented.

**ls**
: This command is not implemented.

**view**
: This command is not implemented.

**help**
: This command is not implemented.

**?**
: This command is not implemented.

**exit**
: This command exits the program.

**set keyword[=value]**
: This command is used to change state information that affects the lookups. Valid keywords are:

  **all**
  : This keyword prints the current values of the frequently used options to `set`. Information about the current default server and host is also printed.

  **class=value**
  : This keyword changes the query class to one of:

**IN**
the Internet class

**CH**
the Chaos class

**HS**
the Hesiod class

**ANY**
wildcard

The class specifies the protocol group of the information. The default is `IN`; the abbreviation for this keyword is `cl`.

**nodebug**
This keyword turns on or off the display of the full response packet, and any intermediate response packets, when searching. The default for this keyword is `nodebug`; the abbreviation for this keyword is `[no]deb`.

**nod2**
This keyword turns debugging mode on or off. This displays more about what nslookup is doing. The default is `nod2`.

**domain=name**
This keyword sets the search list to `name`.

**nosearch**
If the lookup request contains at least one period, but does not end with a trailing period, this keyword appends the domain names in the domain search list to the request until an answer is received. The default is `search`.

**port=value**
This keyword changes the default TCP/UDP name server port to `value` from its default, port 53. The abbreviation for this keyword is `po`.

**querytype=value | type=value**
This keyword changes the type of the information query to `value`. The defaults are A and then AAAA; the abbreviations for these keywords are `q` and `ty`.

Please note that it is only possible to specify one query type. Only the default behavior looks up both when an alternative is not specified.

**norecurse**
This keyword tells the name server to query other servers if it does not have the information. The default is `recurse`; the abbreviation for this keyword is `[no]rec`.

**ndots=number**
This keyword sets the number of dots (label separators) in a domain that disables searching. Absolute names always stop searching.

**retry=number**
This keyword sets the number of retries to `number`.

**timeout=number**
This keyword changes the initial timeout interval to wait for a reply to `number`, in seconds.

**novc**
This keyword indicates that a virtual circuit should always be used when sending requests to the server. `novc` is the default.

**nofail**
This keyword tries the next nameserver if a nameserver responds with SERVFAIL or a referral (nofail), or terminates the query (fail) on such a response. The default is `nofail`.

### 12.28.5 Return Values

**nslookup** returns with an exit status of 1 if any query failed, and 0 otherwise.

### 12.28.6 IDN Support

If **nslookup** has been built with IDN (internationalized domain name) support, it can accept and display non-ASCII domain names. **nslookup** appropriately converts character encoding of a domain name before sending a request to a DNS server or displaying a reply from the server. To turn off IDN support, define the `IDN_DISABLE` environment variable. IDN support is disabled if the variable is set when **nslookup** runs, or when the standard output is not a tty.

### 12.28.7 Files

`/etc/resolv.conf`

### 12.28.8 See Also

*dig(1)*, *host(1)*, *named(8)*.

## 12.29 nsupdate - dynamic DNS update utility

### 12.29.1 Synopsis

**nsupdate** [**-d**] [**-D**] [**-i**] [**-L** level] [ [**-g**] | [**-o**] | [**-l**] | [**-y** [hmac:]keyname:secret] | [**-k** keyfile] ] [ [**-S**] [**-K** tlskeyfile] [**-E** tlscertfile] [**-A** tlscafile] [**-H** tlshostname] [-O] ] [**-t** timeout] [**-u** udptimeout] [**-r** udpretries] [**-v**] [**-T**] [**-P**] [**-V**] [ [**-4**] | [**-6**] ] [filename]

### 12.29.2 Description

**nsupdate** is used to submit Dynamic DNS Update requests, as defined in **RFC 2136**, to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via **nsupdate** or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with **nsupdate** must be in the same zone. Requests are sent to the zone's primary server, which is identified by the MNAME field of the zone's SOA record.

Transaction signatures can be used to authenticate the Dynamic DNS updates. These use the TSIG resource record type described in **RFC 2845**, the SIG(0) record described in **RFC 2535** and **RFC 2931**, or GSS-TSIG as described in **RFC 3645**.

TSIG relies on a shared secret that should only be known to **nsupdate** and the name server. For instance, suitable `key` and `server` statements are added to `/etc/named.conf` so that the name server can associate the appropriate secret key and algorithm with the IP address of the client application that is using TSIG authentication. *ddns-confgen* can generate suitable configuration fragments. **nsupdate** uses the *-y* or *-k* options to provide the TSIG shared secret; these options are mutually exclusive.

SIG(0) uses public key cryptography. To use a SIG(0) key, the public key must be stored in a KEY record in a zone served by the name server.

GSS-TSIG uses Kerberos credentials. Standard GSS-TSIG mode is switched on with the *-g* flag. A non-standards-compliant variant of GSS-TSIG used by Windows 2000 can be switched on with the *-o* flag.

### 12.29.3 Options

**-4**

This option sets use of IPv4 only.

**-6**

This option sets use of IPv6 only.

**-A** `tlscafile`

This option specifies the file of the certificate authorities (CA) certificates (in PEM format) in order to verify the remote server TLS certificate when using DNS-over-TLS (DoT), to achieve Strict or Mutual TLS. When used, it will override the certificates from the global certificates store, which are otherwise used by default when *-S* is enabled. This option can not be used in conjuction with *-O*, and it implies *-S*.

**-C**

Overrides the default *resolv.conf* file. This is only intended for testing.

**-d**

This option sets debug mode, which provides tracing information about the update requests that are made and the replies received from the name server.

**-D**

This option sets extra debug mode.

**-E** `tlscertfile`

This option sets the certificate(s) file for authentication for the DNS-over-TLS (DoT) transport to the remote server. The certificate chain file is expected to be in PEM format. This option implies *-S*, and can only be used with *-K*.

**-g**

This option enables standard GSS-TSIG mode.

**-H** `tlshostname`

This option makes **nsupdate** use the provided hostname during remote server TLS certificate verification. Otherwise, the DNS server name is used. This option implies *-S*.

**-i**

This option forces interactive mode, even when standard input is not a terminal.

**-k** `keyfile`

This option indicates the file containing the TSIG authentication key. Keyfiles may be in two formats: a single file containing a *named.conf*-format `key` statement, which may be generated automatically by *ddns-confgen*; or a pair of files whose names are of the format `K{name}.+157.+{random}.key` and `K{name}.+157.+{random}.private`, which can be generated by *dnssec-keygen*. The *-k* option can also be used to specify a SIG(0) key used to authenticate Dynamic DNS update requests. In this case, the key specified is not an HMAC-MD5 key.

**-K** `tlskeyfile`

This option sets the key file for authenticated encryption for the DNS-over-TLS (DoT) transport with the remote server. The private key file is expected to be in PEM format. This option implies *-S*, and can only be used with *-E*.

**-l**

This option sets local-host only mode, which sets the server address to localhost (disabling the `server` so that the server address cannot be overridden). Connections to the local server use a TSIG key found in `/run/session.key`, which is automatically generated by *named* if any local `primary` zone has set `update-policy` to `local`. The location of this key file can be overridden with the *-k* option.

**-L** level

This option sets the logging debug level. If zero, logging is disabled.

**-o**

This option is deprecated. Previously, it enabled a non-standards-compliant variant of GSS-TSIG that was used by Windows 2000. Since that OS is now long past its end of life, this option is now treated as a synonym for *-g*.

**-O**

This option enables Opportunistic TLS. When used, the remote peer's TLS certificate will not be verified. This option should be used for debugging purposes only, and it is not recommended to use it in production. This option can not be used in conjuction with *-A*, and it implies *-S*.

**-p** port

This option sets the port to use for connections to a name server. The default is 53.

**-P**

This option prints the list of private BIND-specific resource record types whose format is understood by **nsupdate**. See also the *-T* option.

**-r** udpretries

This option sets the number of UDP retries. The default is 3. If zero, only one update request is made.

**-S**

This option indicates whether to use DNS-over-TLS (DoT) when querying name servers specified by `server servername port` syntax in the input file, and the primary server discovered through a SOA request. When the *-K* and *-E* options are used, then the specified TLS client certificate and private key pair are used for authentication (Mutual TLS). This option implies *-v*.

**-t** timeout

This option sets the maximum time an update request can take before it is aborted. The default is 300 seconds. If zero, the timeout is disabled for TCP mode. For UDP mode, the option *-u* takes precedence over this option, unless the option *-u* is set to zero, in which case the interval is computed from the *-t* timeout interval and the number of UDP retries. For UDP mode, the timeout can not be disabled, and will be rounded up to 1 second in case if both *-t* and *-u* are set to zero.

**-T**

This option prints the list of IANA standard resource record types whose format is understood by **nsupdate**. **nsupdate** exits after the lists are printed. The *-T* option can be combined with the *-P* option.

Other types can be entered using `TYPEXXXXX` where `XXXXX` is the decimal value of the type with no leading zeros. The rdata, if present, is parsed using the UNKNOWN rdata format, (<backslash> <hash> <space> <length> <space> <hexstring>).

**-u** udptimeout

This option sets the UDP retry interval. The default is 3 seconds. If zero, the interval is computed from the timeout interval and number of UDP retries.

**-v**

This option specifies that TCP should be used even for small update requests. By default, **nsupdate** uses UDP to send update requests to the name server unless they are too large to fit in a UDP request, in which case TCP is used. TCP may be preferable when a batch of update requests is made.

**-V**

This option prints the version number and exits.

**-y** [hmac:]keyname:secret

This option sets the literal TSIG authentication key. `keyname` is the name of the key, and `secret` is the base64 encoded shared secret. `hmac` is the name of the key algorithm; valid choices are `hmac-md5`, `hmac-sha1`, `hmac-sha224`, `hmac-sha256`, `hmac-sha384`, or `hmac-sha512`. If `hmac` is not specified, the default is `hmac-md5`, or if MD5 was disabled, `hmac-sha256`.

NOTE: Use of the *-y* option is discouraged because the shared secret is supplied as a command-line argument in clear text. This may be visible in the output from ps1 or in a history file maintained by the user's shell.

### 12.29.4 Input Format

**nsupdate** reads input from `filename` or standard input. Each command is supplied on exactly one line of input. Some commands are for administrative purposes; others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates are rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are either present or missing from the zone. A blank input line (or the `send` command) causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats and their meanings are as follows:

**server servername port**

This command sends all dynamic update requests to the name server `servername`. When no server statement is provided, **nsupdate** sends updates to the primary server of the correct zone. The MNAME field of that zone's SOA record identify the primary server for that zone. `port` is the port number on `servername` where the dynamic update requests are sent. If no port number is specified, the default DNS port number of 53 is used.

> **Note**
>
> This command has no effect when GSS-TSIG is in use.

**local address port**

This command sends all dynamic update requests using the local `address`. When no local statement is provided, **nsupdate** sends updates using an address and port chosen by the system. `port` can also be used to force requests to come from a specific port. If no port number is specified, the system assigns one.

**zone zonename**

This command specifies that all updates are to be made to the zone `zonename`. If no `zone` statement is provided, **nsupdate** attempts to determine the correct zone to update based on the rest of the input.

**class classname**

This command specifies the default class. If no `class` is specified, the default class is `IN`.

**ttl seconds**

This command specifies the default time-to-live, in seconds, for records to be added. The value `none` clears the default TTL.

**key hmac:keyname secret**

This command specifies that all updates are to be TSIG-signed using the `keyname-secret` pair. If `hmac` is specified, it sets the signing algorithm in use. The default is `hmac-md5`; if MD5 was disabled, the default is `hmac-sha256`. The `key` command overrides any key specified on the command line via *-y* or *-k*.

**gsstsig**

This command uses GSS-TSIG to sign the updates. This is equivalent to specifying *-g* on the command line.

**oldgsstsig**
This command is deprecated and will be removed in a future release. Previously, it caused `nsupdate` to use the Windows 2000 version of GSS-TSIG to sign updates. It is now treated as a synonym for `gsstsig`.

**realm [realm_name]**
When using GSS-TSIG, this command specifies the use of `realm_name` rather than the default realm in `krb5.conf`. If no realm is specified, the saved realm is cleared.

**check-names [boolean]**
This command turns on or off check-names processing on records to be added. Check-names has no effect on prerequisites or records to be deleted. By default check-names processing is on. If check-names processing fails, the record is not added to the UPDATE message.

**check-svbc [boolean]**
This command turns on or off check-svcb processing on records to be added. Check-svcb has no effect on prerequisites or records to be deleted. By default check-svcb processing is on. If check-svcb processing fails, the record is not added to the UPDATE message.

**lease time [keytime]**
Set the EDNS Update Lease (UL) option to value to `time` and optionally also set the key lease time to `keytime` in seconds. If `time` is `none` the lease times are cleared.

**prereq nxdomain domain-name**
This command requires that no resource record of any type exist with the name `domain-name`.

**prereq yxdomain domain-name**
This command requires that `domain-name` exist (as at least one resource record, of any type).

**prereq nxrrset domain-name class type**
This command requires that no resource record exist of the specified `type`, `class`, and `domain-name`. If `class` is omitted, IN (Internet) is assumed.

**prereq yxrrset domain-name class type**
This command requires that a resource record of the specified `type`, `class` and `domain-name` exist. If `class` is omitted, IN (internet) is assumed.

**prereq yxrrset domain-name class type data**
With this command, the `data` from each set of prerequisites of this form sharing a common `type`, `class`, and `domain-name` are combined to form a set of RRs. This set of RRs must exactly match the set of RRs existing in the zone at the given `type`, `class`, and `domain-name`. The `data` are written in the standard text representation of the resource record's RDATA.

Note RDATA which is empty (e.g. APL with an zero length rdata) needs to be entered using `\# 0` form.

**update delete domain-name ttl class type data**
This command deletes any resource records named `domain-name`. If `type` and `data` are provided, only matching resource records are removed. The Internet class is assumed if `class` is not supplied. The `ttl` is ignored, and is only allowed for compatibility.

Note RDATA which is empty (e.g. APL with an zero length rdata) needs to be entered using `\# 0` form.

**update add domain-name ttl class type data**
This command adds a new resource record with the specified `ttl`, `class`, and `data`.

Note RDATA which is empty (e.g. APL with an zero length rdata) needs to be entered using `\# 0` form.

**show**
This command displays the current message, containing all of the prerequisites and updates specified since the last send.

**send**
This command sends the current message. This is equivalent to entering a blank line.

**answer**
This command displays the answer.

**debug**
This command turns on debugging.

**version**
This command prints the version number.

**help**
This command prints a list of commands.

Lines beginning with a semicolon (;) are comments and are ignored.

### 12.29.5 Examples

The examples below show how **nsupdate** can be used to insert and delete resource records from the `example.com` zone. Notice that the input in each example contains a trailing blank line, so that a group of commands is sent as one dynamic update request to the primary name server for `example.com`.

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
```

Any A records for `oldhost.example.com` are deleted, and an A record for `newhost.example.com` with IP address 172.16.1.1 is added. The newly added record has a TTL of 1 day (86400 seconds).

```
# nsupdate
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
```

The prerequisite condition tells the name server to verify that there are no resource records of any type for `nickname.example.com`. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in **RFC 1034** that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in **RFC 2535** to allow CNAMEs to have RRSIG, DNSKEY, and NSEC records.)

### 12.29.6 Files

**/etc/resolv.conf**
Used to identify the default name server

**/run/session.key**
Sets the default TSIG key for use in local-only mode

**K{name}.+157.+{random}.key**
Base-64 encoding of the HMAC-MD5 key created by *dnssec-keygen*.

**K{name}.+157.+{random}.private**
Base-64 encoding of the HMAC-MD5 key created by *dnssec-keygen*.

### 12.29.7 See Also

**RFC 2136**, **RFC 3007**, **RFC 2104**, **RFC 2845**, **RFC 1034**, **RFC 2535**, **RFC 2931**, *named(8)*, *dnssec-keygen(8)*, *tsig-keygen(8)*.

### 12.29.8 Bugs

The TSIG key is redundantly stored in two separate files. This is a consequence of **nsupdate** using the DST library for its cryptographic operations, and may change in future releases.

## 12.30 rndc-confgen - rndc key generation tool

### 12.30.1 Synopsis

**rndc-confgen** [**-a**] [**-A** algorithm] [**-b** keysize] [**-c** keyfile] [**-h**] [**-k** keyname] [**-p** port] [**-s** address] [**-t** chrootdir] [**-u** user]

### 12.30.2 Description

**rndc-confgen** generates configuration files for *rndc*. It can be used as a convenient alternative to writing the *rndc.conf* file and the corresponding `controls` and `key` statements in *named.conf* by hand. Alternatively, it can be run with the *-a* option to set up a `rndc.key` file and avoid the need for a *rndc.conf* file and a `controls` statement altogether.

### 12.30.3 Options

**-a**

This option sets automatic *rndc* configuration, which creates a file `/etc/rndc.key` that is read by both *rndc* and *named* on startup. The `rndc.key` file defines a default command channel and authentication key allowing *rndc* to communicate with *named* on the local host with no further configuration.

If a more elaborate configuration than that generated by *rndc-confgen -a* is required, for example if rndc is to be used remotely, run **rndc-confgen** without the *-a* option and set up *rndc.conf* and *named.conf* as directed.

**-A** `algorithm`

This option specifies the algorithm to use for the TSIG key. Available choices are: hmac-md5, hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, and hmac-sha512. The default is hmac-sha256.

**-b** `keysize`

This option specifies the size of the authentication key in bits. The size must be between 1 and 512 bits; the default is the hash size.

**-c** `keyfile`

This option is used with the *-a* option to specify an alternate location for `rndc.key`.

**-h**

This option prints a short summary of the options and arguments to **rndc-confgen**.

**-k** `keyname`

This option specifies the key name of the *rndc* authentication key. This must be a valid domain name and will be sanitized using the domain name semantics. The default is `rndc-key`.

**-p** `port`

This option specifies the command channel port where *named* listens for connections from *rndc*. The default is 953.

**-q**

This option prevets printing the written path in automatic configuration mode.

**-s** address

This option specifies the IP address where *named* listens for command-channel connections from *rndc*. The default is the loopback address 127.0.0.1.

**-t** chrootdir

This option is used with the *-a* option to specify a directory where *named* runs chrooted. An additional copy of the `rndc.key` is written relative to this directory, so that it is found by the chrooted *named*.

**-u** user

This option is used with the *-a* option to set the owner of the generated `rndc.key` file. If *-t* is also specified, only the file in the chroot area has its owner changed.

### 12.30.4 Examples

To allow *rndc* to be used with no manual configuration, run:

```
rndc-confgen -a
```

To print a sample *rndc.conf* file and the corresponding `controls` and `key` statements to be manually inserted into *named.conf*, run:

**rndc-confgen**

### 12.30.5 See Also

*rndc(8)*, *rndc.conf(5)*, *named(8)*, BIND 9 Administrator Reference Manual.

## 12.31 rndc.conf - rndc configuration file

### 12.31.1 Synopsis

**rndc.conf**

### 12.31.2 Description

**rndc.conf** is the configuration file for *rndc*, the BIND 9 name server control utility. This file has a similar structure and syntax to *named.conf*. Statements are enclosed in braces and terminated with a semi-colon. Clauses in the statements are also semi-colon terminated. The usual comment styles are supported:

C style: /* */

C++ style: // to end of line

Unix style: # to end of line

**rndc.conf** is much simpler than *named.conf*. The file uses three statements: an options statement, a server statement, and a key statement.

The `options` statement contains five clauses. The `default-server` clause is followed by the name or address of a name server. This host is used when no name server is given as an argument to *rndc*. The `default-key` clause is followed by the name of a key, which is identified by a `key` statement. If no `keyid` is provided on the rndc command line, and no `key` clause is found in a matching `server` statement, this default key is used to authenticate the server's commands and responses. The `default-port` clause is followed by the port to connect to on the remote name server. If no `port` option is provided on the rndc command line, and no `port` clause is found in a matching `server` statement,

this default port is used to connect. The `default-source-address` and `default-source-address-v6` clauses can be used to set the IPv4 and IPv6 source addresses respectively.

After the `server` keyword, the server statement includes a string which is the hostname or address for a name server. The statement has three possible clauses: `key`, `port`, and `addresses`. The key name must match the name of a key statement in the file. The port number specifies the port to connect to. If an `addresses` clause is supplied, these addresses are used instead of the server name. Each address can take an optional port. If an `source-address` or `source-address-v6` is supplied, it is used to specify the IPv4 and IPv6 source address, respectively.

The `key` statement begins with an identifying string, the name of the key. The statement has two clauses. `algorithm` identifies the authentication algorithm for *rndc* to use; currently only HMAC-MD5 (for compatibility), HMAC-SHA1, HMAC-SHA224, HMAC-SHA256 (default), HMAC-SHA384, and HMAC-SHA512 are supported. This is followed by a secret clause which contains the base-64 encoding of the algorithm's authentication key. The base-64 string is enclosed in double quotes.

There are two common ways to generate the base-64 string for the secret. The BIND 9 program *rndc-confgen* can be used to generate a random key, or the `mmencode` program, also known as `mimencode`, can be used to generate a base-64 string from known input. `mmencode` does not ship with BIND 9 but is available on many systems. See the Example section for sample command lines for each.

### 12.31.3 Example

```
options {
  default-server  localhost;
  default-key     samplekey;
};
```

```
server localhost {
  key             samplekey;
};
```

```
server testserver {
  key     testkey;
  addresses   { localhost port 5353; };
};
```

```
key samplekey {
  algorithm       hmac-sha256;
  secret          "6FMfj43Osz4lyb24OIe2iGEz9lf1llJO+lz";
};
```

```
key testkey {
  algorithm   hmac-sha256;
  secret      "R3HI8P6BKw9ZwXwN3VZKuQ==";
};
```

In the above example, *rndc* by default uses the server at localhost (127.0.0.1) and the key called "samplekey". Commands to the localhost server use the "samplekey" key, which must also be defined in the server's configuration file with the same name and secret. The key statement indicates that "samplekey" uses the HMAC-SHA256 algorithm and its secret clause contains the base-64 encoding of the HMAC-SHA256 secret enclosed in double quotes.

If *rndc -s testserver* is used, then *rndc* connects to the server on localhost port 5353 using the key "testkey".

To generate a random secret with *rndc-confgen*:

*rndc-confgen*

A complete **rndc.conf** file, including the randomly generated key, is written to the standard output. Commented-out `key` and `controls` statements for *named.conf* are also printed.

To generate a base-64 secret with `mmencode`:

```
echo "known plaintext for a secret" | mmencode
```

### 12.31.4 Name Server Configuration

The name server must be configured to accept rndc connections and to recognize the key specified in the **rndc.conf** file, using the controls statement in *named.conf*. See the sections on the `controls` statement in the BIND 9 Administrator Reference Manual for details.

### 12.31.5 See Also

*rndc(8)*, *rndc-confgen(8)*, *mmencode(1)*, BIND 9 Administrator Reference Manual.

## 12.32 rndc - name server control utility

### 12.32.1 Synopsis

**rndc** [**-b** source-address] [**-c** config-file] [**-k** key-file] [**-s** server] [**-p** port] [**-q**] [**-r**] [**-V**] [**-y** server_key] [[**-4**] | [**-6**]] {command}

### 12.32.2 Description

**rndc** controls the operation of a name server. If **rndc** is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

**rndc** communicates with the name server over a TCP connection, sending commands authenticated with digital signatures. In the current versions of **rndc** and *named*, the only supported authentication algorithms are HMAC-MD5 (for compatibility), HMAC-SHA1, HMAC-SHA224, HMAC-SHA256 (default), HMAC-SHA384, and HMAC-SHA512. They use a shared secret on each end of the connection, which provides TSIG-style authentication for the command request and the name server's response. All commands sent over the channel must be signed by a server_key known to the server.

**rndc** reads a configuration file to determine how to contact the name server and decide what algorithm and key it should use.

### 12.32.3 Options

**-4**

This option indicates use of IPv4 only.

**-6**

This option indicates use of IPv6 only.

**-b** `source-address`

This option indicates `source-address` as the source address for the connection to the server. Multiple instances are permitted, to allow setting of both the IPv4 and IPv6 source addresses.

**-c** `config-file`

This option indicates `config-file` as the configuration file instead of the default, `/etc/rndc.conf`.

**-k** key-file

This option indicates key-file as the key file instead of the default, /etc/rndc.key. The key in /etc/rndc.key is used to authenticate commands sent to the server if the config-file does not exist.

**-s** server

server is the name or address of the server which matches a server statement in the configuration file for **rndc**. If no server is supplied on the command line, the host named by the default-server clause in the options statement of the **rndc** configuration file is used.

**-p** port

This option instructs BIND 9 to send commands to TCP port port instead of its default control channel port, 953.

**-q**

This option sets quiet mode, where message text returned by the server is not printed unless there is an error.

**-r**

This option instructs **rndc** to print the result code returned by *named* after executing the requested command (e.g., ISC_R_SUCCESS, ISC_R_FAILURE, etc.).

**-t** timeout

This option sets the idle timeout period for **rndc** to timeout seconds. The default is 60 seconds, and the maximum settable value is 86400 seconds (1 day). If set to 0, there is no timeout.

**-V**

This option enables verbose logging.

**-y** server_key

This option indicates use of the key server_key from the configuration file. For control message validation to succeed, server_key must be known by *named* with the same algorithm and secret string. If no server_key is specified, **rndc** first looks for a key clause in the server statement of the server being used, or if no server statement is present for that host, then in the default-key clause of the options statement. Note that the configuration file contains shared secrets which are used to send authenticated control commands to name servers, and should therefore not have general read or write access.

### 12.32.4 Commands

A list of commands supported by **rndc** can be seen by running **rndc** without arguments.

Currently supported commands are:

**addzone** zone [class [view]] configuration

This command adds a zone while the server is running. This command requires the allow-new-zones option to be set to yes. The configuration string specified on the command line is the zone configuration text that would ordinarily be placed in *named.conf*.

The configuration is saved in a file called viewname.nzf (or, if *named* is compiled with liblmdb, an LMDB database file called viewname.nzd). viewname is the name of the view, unless the view name contains characters that are incompatible with use as a file name, in which case a cryptographic hash of the view name is used instead. When *named* is restarted, the file is loaded into the view configuration so that zones that were added can persist after a restart.

This sample addzone command adds the zone example.com to the default view:

```
rndc addzone example.com '{ type primary; file "example.com.db"; };'
```

(Note the brackets around and semi-colon after the zone configuration text.)

See also *rndc delzone* and *rndc modzone*.

**delzone** [-clean] zone [class [view]]

This command deletes a zone while the server is running.

If the -clean argument is specified, the zone's master file (and journal file, if any) are deleted along with the zone. Without the -clean option, zone files must be deleted manually. (If the zone is of type secondary or stub, the files needing to be removed are reported in the output of the rndc delzone command.)

If the zone was originally added via rndc addzone, then it is removed permanently. However, if it was originally configured in *named.conf*, then that original configuration remains in place; when the server is restarted or reconfigured, the zone is recreated. To remove it permanently, it must also be removed from *named.conf*.

See also *rndc addzone* and *rndc modzone*.

**dnssec**

The following commands allow you to interact with the "dnssec-policy" of a given zone.

**-checkds** [-key id [-alg algorithm]] [-when time] (published | withdrawn) zone [class [view]]

This command informs *named* that the DS for a specified zone's key-signing key (KSK) has been confirmed to be published in, or withdrawn from, the parent zone. This is required in order to complete a KSK rollover. The -key id and -alg algorithm arguments can be used to specify a particular KSK, if necessary; if there is only one key acting as a KSK for the zone, these arguments can be omitted. The time of publication or withdrawal for the DS is set to the current time by default, but can be overridden to a specific time with the argument -when time, where time is expressed in YYYYMMDDHHMMSS notation.

**-rollover** -key id [-alg algorithm] [-when time] zone [class [view]]

This command allows you to schedule key rollover for a specific key (overriding the original key lifetime). The -key id and -alg algorithm arguments specify which key to roll. The time to start the rollover can be set with -when time, where time is expressed in YYYYMMDDHHMMSS. If not set the rollover will start immediately.

**-status** [-v] zone [class [view]]

This command shows the DNSSEC signing state for the specified zone. Adding -v also lists no longer used keys and shows the key states of the keys.

**-step** zone [class [view]]

This command sends a signal to an instance of *named* for a zone configured with dnssec-policy in manual mode, telling it to continue with the operations that had previously been blocked but logged. This gives the human operator a chance to review the log messages, understand what will happen next and then, using rndc dnssec -step, to inform *named* to proceed to the next stage.

**dnstap** (-reopen | -roll [number])

This command closes and re-opens DNSTAP output files.

rndc dnstap -reopen allows the output file to be renamed externally, so that *named* can truncate and re-open it.

rndc dnstap -roll causes the output file to be rolled automatically, similar to log files. The most recent output file has ".0" appended to its name; the previous most recent output file is moved to ".1", and so on. If number is specified, then the number of backup log files is limited to that number.

**dumpdb** [-all | -cache | -zones | -adb | -bad | -expired | -fail] [view ...]

This command dumps the server's caches (default) and/or zones to the dump file for the specified views. If no view is specified, all views are dumped. (See the dump-file option in the BIND 9 Administrator Reference Manual.)

**fetchlimit** [view]

This command dumps a list of servers that are currently being rate-limited as a result of fetches-per-server settings, and a list of domain names that are currently being rate-limited as a result of fetches-per-zone settings.

**flush**

This command flushes the server's cache.

**flushname** name [view]

This command flushes the given name from the view's DNS cache and, if applicable, from the view's nameserver address database, bad server cache, and SERVFAIL cache.

**flushtree** name [view]

This command flushes the given name, and all of its subdomains, from the view's DNS cache, address database, bad server cache, and SERVFAIL cache.

**freeze** [zone [class [view]]]

This command suspends updates to a dynamic zone. If no zone is specified, then all zones are suspended. This allows manual edits to be made to a zone normally updated by dynamic update, and causes changes in the journal file to be synced into the master file. All dynamic update attempts are refused while the zone is frozen.

See also *rndc thaw*.

**halt** [-p]

This command stops the server immediately. Recent changes made through dynamic update or IXFR are not saved to the master files, but are rolled forward from the journal files when the server is restarted. If -p is specified, *named*'s process ID is returned. This allows an external process to determine when *named* has completed halting.

See also *rndc stop*.

**skr** -import file zone [class [view]]

This command allows you to import a SKR file for the specified zone, to support offline KSK signing.

**loadkeys** [zone [class [view]]]

This command fetches all DNSSEC keys for the given zone from the key directory. If they are within their publication period, they are merged into the zone's DNSKEY RRset. Unlike *rndc sign*, however, the zone is not immediately re-signed by the new keys, but is allowed to incrementally re-sign over time.

This command requires that the zone be configured with a dnssec-policy.

**managed-keys** (status | refresh | sync | destroy) [class [view]]

This command inspects and controls the "managed-keys" database which handles **RFC 5011** DNSSEC trust anchor maintenance. If a view is specified, these commands are applied to that view; otherwise, they are applied to all views.

- When run with the status keyword, this prints the current status of the managed-keys database.
- When run with the refresh keyword, this forces an immediate refresh query to be sent for all the managed keys, updating the managed-keys database if any new keys are found, without waiting the normal refresh interval.
- When run with the sync keyword, this forces an immediate dump of the managed-keys database to disk (in the file managed-keys.bind or (viewname.mkeys). This synchronizes the database with its journal file, so that the database's current contents can be inspected visually.
- When run with the destroy keyword, the managed-keys database is shut down and deleted, and all key maintenance is terminated. This command should be used only with extreme caution.

  Existing keys that are already trusted are not deleted from memory; DNSSEC validation can continue after this command is used. However, key maintenance operations cease until *named* is restarted or reconfigured, and all existing key maintenance states are deleted.

  Running *rndc reconfig* or restarting *named* immediately after this command causes key maintenance to be reinitialized from scratch, just as if the server were being started for the first time. This is primarily

intended for testing, but it may also be used, for example, to jumpstart the acquisition of new keys in the event of a trust anchor rollover, or as a brute-force repair for key maintenance problems.

**memprof** [(on | off | dump)]

This command controls memory profiling. To have any effect, *named* must be built with jemalloc, the library have profiling support enabled and run with the `prof:true` allocator configuration. (either via `MALLOC_CONF` or `/etc/malloc.conf`)

The `prof_active:false` option is recommended to ensure the profiling overhead does not affect *named* when not needed.

The `on` and `off` options will start and stop the jemalloc memory profiling respectively. When run with the *dump* option, *named* will dump the profile to the working directory. The name will be chosen automatically by jemalloc.

**modzone** zone [class [view]] configuration

This command modifies the configuration of a zone while the server is running. This command requires the `allow-new-zones` option to be set to `yes`. As with `addzone`, the configuration string specified on the command line is the zone configuration text that would ordinarily be placed in *named.conf*.

If the zone was originally added via *rndc addzone*, the configuration changes are recorded permanently and are still in effect after the server is restarted or reconfigured. However, if it was originally configured in *named.conf*, then that original configuration remains in place; when the server is restarted or reconfigured, the zone reverts to its original configuration. To make the changes permanent, it must also be modified in *named.conf*.

See also *rndc addzone* and *rndc delzone*.

**notify** zone [class [view]]

This command resends NOTIFY messages for the zone.

**notrace**

This command sets the server's debugging level to 0.

See also *rndc trace*.

**nta** [(-class class | -dump | -force | -remove | -lifetime duration)] domain [view]

This command sets a DNSSEC negative trust anchor (NTA) for `domain`, with a lifetime of `duration`. The default lifetime is configured in *named.conf* via the `nta-lifetime` option, and defaults to one hour. The lifetime cannot exceed one week.

A negative trust anchor selectively disables DNSSEC validation for zones that are known to be failing because of misconfiguration rather than an attack. When data to be validated is at or below an active NTA (and above any other configured trust anchors), *named* aborts the DNSSEC validation process and treats the data as insecure rather than bogus. This continues until the NTA's lifetime has elapsed.

NTAs persist across restarts of the *named* server. The NTAs for a view are saved in a file called `name.nta`, where `name` is the name of the view; if it contains characters that are incompatible with use as a file name, a cryptographic hash is generated from the name of the view.

An existing NTA can be removed by using the `-remove` option.

An NTA's lifetime can be specified with the `-lifetime` option. TTL-style suffixes can be used to specify the lifetime in seconds, minutes, or hours. If the specified NTA already exists, its lifetime is updated to the new value. Setting `lifetime` to zero is equivalent to `-remove`.

If `-dump` is used, any other arguments are ignored and a list of existing NTAs is printed. Note that this may include NTAs that are expired but have not yet been cleaned up.

Normally, *named* periodically tests to see whether data below an NTA can now be validated (see the `nta-recheck` option in the Administrator Reference Manual for details). If data can be validated, then the NTA is regarded as no longer necessary and is allowed to expire early. The `-force` parameter overrides this behavior and forces an NTA to persist for its entire lifetime, regardless of whether data could be validated if the NTA were not present.

The view class can be specified with `-class`. The default is class `IN`, which is the only class for which DNSSEC is currently supported.

All of these options can be shortened, i.e., to `-l`, `-r`, `-d`, `-f`, and `-c`.

Unrecognized options are treated as errors. To refer to a domain or view name that begins with a hyphen, use a double-hyphen (--) on the command line to indicate the end of options.

**querylog** `[(on | off)]`

This command enables or disables query logging. For backward compatibility, this command can also be used without an argument to toggle query logging on and off.

Query logging can also be enabled by explicitly directing the `queries category` to a `channel` in the `logging` section of *named.conf*, or by specifying `querylog yes;` in the `options` section of *named.conf*.

**reconfig**

This command reloads the configuration file and loads new zones, but does not reload existing zone files even if they have changed. This is faster than a full *rndc reload* when there is a large number of zones, because it avoids the need to examine the modification times of the zone files.

**recursing**

This command dumps the list of queries *named* is currently recursing on, and the list of domains to which iterative queries are currently being sent.

The first list includes all unique clients that are waiting for recursion to complete, including the query that is awaiting a response and the timestamp (seconds since the Unix epoch) of when named started processing this client query.

The second list comprises of domains for which there are active (or recently active) fetches in progress. It reports the number of active fetches for each domain and the number of queries that have been passed (allowed) or dropped (spilled) as a result of the `fetches-per-zone` limit. (Note: these counters are not cumulative over time; whenever the number of active fetches for a domain drops to zero, the counter for that domain is deleted, and the next time a fetch is sent to that domain, it is recreated with the counters set to zero).

**refresh** `zone [class [view]]`

This command schedules zone maintenance for the given zone.

**reload**

This command reloads the configuration file and zones.

**zone** `[class [view]]`

If a zone is specified, this command reloads only the given zone. If no zone is specified, the reloading happens asynchronously.

**reset-stats** `<counter-name ...>`

This command resets the requested statistics counters.

At least one counter name must be provided. Currently the following counters are supported: `recursive-high-water`, `tcp-high-water`.

**responselog** `[on | off]`

This command enables or disables response logging. For backward compatibility, this command can also be used without an argument to toggle response logging on and off.

Unlike query logging, response logging cannot be enabled by explicitly directing the `responses category` to a `channel` in the `logging` section of *named.conf*, but it can still be enabled by specifying `responselog yes;` in the `options` section of *named.conf*.

**retransfer** [-force] zone [class [view]]

This command retransfers the given secondary zone from the primary server.

If the zone is configured to use inline-signing, the signed version of the zone is discarded; after the retransfer of the unsigned version is complete, the signed version is regenerated with new signatures. With the optional -force argument provided if there is an ongoing zone transfer it will be aborted before a new zone transfer is scheduled.

**scan**

This command scans the list of available network interfaces for changes, without performing a full *rndc reconfig* or waiting for the interface-interval timer.

**secroots** [-] [view ...]

This command dumps the security roots (i.e., trust anchors configured via trust-anchors, or the managed-keys or trusted-keys statements [both deprecated], or dnssec-validation auto) and negative trust anchors for the specified views. If no view is specified, all views are dumped. Security roots indicate whether they are configured as trusted keys, managed keys, or initializing managed keys (managed keys that have not yet been updated by a successful key refresh query).

If the first argument is -, then the output is returned via the **rndc** response channel and printed to the standard output. Otherwise, it is written to the secroots dump file, which defaults to named.secroots, but can be overridden via the secroots-file option in *named.conf*.

See also *rndc managed-keys*.

**serve-stale** (on | off | reset | status) [class [view]]

This command enables, disables, resets, or reports the current status of the serving of stale answers as configured in *named.conf*.

If serving of stale answers is disabled by rndc-serve-stale off, then it remains disabled even if *named* is reloaded or reconfigured. rndc serve-stale reset restores the setting as configured in *named.conf*.

rndc serve-stale status reports whether caching and serving of stale answers is currently enabled or disabled. It also reports the values of stale-answer-ttl and max-stale-ttl.

**showzone** zone [class [view]]

If the server is configured with allow-new-zones set to yes, then this command prints the configuration of a running zone.

See also *rndc addzone*, *rndc modzone*. and *rndc delzone*.

**sign** zone [class [view]]

This command fetches all DNSSEC keys for the given zone from the key directory (see the key-directory option in the BIND 9 Administrator Reference Manual). If they are within their publication period, they are merged into the zone's DNSKEY RRset. If the DNSKEY RRset is changed, then the zone is automatically re-signed with the new key set. This will replace signatures of inactive keys with signatures from active keys, and update signatures that expire within the refresh interval.

This command requires that the zone be configured with a dnssec-policy.

See also *rndc loadkeys*.

**signing** [(-list | -clear keyid/algorithm | -clear all | -nsec3param (parameters | none) | -serial value) zone [class [view]]

This command lists, edits, or removes the DNSSEC signing-state records for the specified zone. The status of ongoing DNSSEC operations, such as signing or generating NSEC3 chains, is stored in the zone in the form of DNS resource records of type sig-signing-type. rndc signing -list converts these records into a human-readable form, indicating which keys are currently signing or have finished signing the zone, and which NSEC3 chains are being created or removed.

`rndc signing -clear` can remove a single key (specified in the same format that `rndc signing -list` uses to display it), or all keys. In either case, only completed keys are removed; any record indicating that a key has not yet finished signing the zone is retained.

`rndc signing -nsec3param` sets the NSEC3 parameters for a zone. This is the only supported mechanism for using NSEC3 with `inline-signing` zones. Parameters are specified in the same format as an NSEC3PARAM resource record: `hash algorithm`, `flags`, `iterations`, and `salt`, in that order.

Currently, the only defined value for `hash algorithm` is `1`, representing SHA-1. The `flags` may be set to `0` or `1`, depending on whether the opt-out bit in the NSEC3 chain should be set. `iterations` defines the number of additional times to apply the algorithm when generating an NSEC3 hash. The `salt` is a string of data expressed in hexadecimal, a hyphen (`-`) if no salt is to be used, or the keyword `auto`, which causes *named* to generate a random 64-bit salt.

The only recommended configuration is `rndc signing -nsec3param 1 0 0 - zone`, i.e. no salt, no additional iterations, no opt-out.

> **Warning**
>
> Do not use extra iterations, salt, or opt-out unless all their implications are fully understood. A higher number of iterations causes interoperability problems and opens servers to CPU-exhausting DoS attacks.

`rndc signing -nsec3param none` removes an existing NSEC3 chain and replaces it with NSEC.

`rndc signing -serial value` sets the serial number of the zone to `value`. If the value would cause the serial number to go backwards, it is rejected. The primary use of this parameter is to set the serial number on inline signed zones.

**stats**

This command writes server statistics to the statistics file. (See the `statistics-file` option in the BIND 9 Administrator Reference Manual.)

**status**

This command displays the status of the server. Note that the number of zones includes the internal `bind/CH` zone and the default `./IN` hint zone, if there is no explicit root zone configured.

**stop** `-p`

This command stops the server, making sure any recent changes made through dynamic update or IXFR are first saved to the master files of the updated zones. If `-p` is specified, *named*'s process ID is returned. This allows an external process to determine when *named* has completed stopping.

See also *rndc halt*.

**sync** `-clean [zone [class [view]]]`

This command syncs changes in the journal file for a dynamic zone to the master file. If the "-clean" option is specified, the journal file is also removed. If no zone is specified, then all zones are synced.

**tcp-timeouts** `[initial idle keepalive advertised]`

When called without arguments, this command displays the current values of the `tcp-initial-timeout`, `tcp-idle-timeout`, `tcp-keepalive-timeout`, and `tcp-advertised-timeout` options. When called with arguments, these values are updated. This allows an administrator to make rapid adjustments when under a denial-of-service (DoS) attack. See the descriptions of these options in the BIND 9 Administrator Reference Manual for details of their use.

**thaw** `[zone [class [view]]]`

This command enables updates to a frozen dynamic zone. If no zone is specified, then all frozen zones are enabled. This causes the server to reload the zone from disk, and re-enables dynamic updates after the load has

completed. After a zone is thawed, dynamic updates are no longer refused. If the zone has changed and the `ixfr-from-differences` option is in use, the journal file is updated to reflect changes in the zone. Otherwise, if the zone has changed, any existing journal file is removed. If no zone is specified, the reloading happens asynchronously.

See also *rndc freeze*.

**trace** [level]

If no level is specified, this command increments the server's debugging level by one.

**level**

If specified, this command sets the server's debugging level to the provided value.

See also *rndc notrace*.

**validation** (on | off | status) [view ...]

This command enables, disables, or checks the current status of DNSSEC validation. By default, validation is enabled.

The cache is flushed when validation is turned on or off to avoid using data that might differ between states.

**zonestatus** zone [class [view]]

This command displays the current status of the given zone, including the master file name and any include files from which it was loaded, when it was most recently loaded, the current serial number, the number of nodes, whether the zone supports dynamic updates, whether the zone is DNSSEC signed, whether it uses automatic DNSSEC key management or inline signing, and the scheduled refresh or expiry times for the zone.

See also *rndc showzone*.

**rndc** commands that specify zone names, such as *reload retransfer*, or *zonestatus*, can be ambiguous when applied to zones of type `redirect`. Redirect zones are always called `.`, and can be confused with zones of type `hint` or with secondary copies of the root zone. To specify a redirect zone, use the special zone name `-redirect`, without a trailing period. (With a trailing period, this would specify a zone called "-redirect".)

### 12.32.5 Limitations

There is currently no way to provide the shared secret for a `server_key` without using the configuration file.

Several error messages could be clearer.

### 12.32.6 See Also

*rndc.conf(5)*, *rndc-confgen(8)*, *named(8)*, *named.conf(5)*, BIND 9 Administrator Reference Manual.

## 12.33 tsig-keygen - TSIG key generation tool

### 12.33.1 Synopsis

**tsig-keygen** [**-a** algorithm] [**-h**] [name]

### 12.33.2 Description

**tsig-keygen** is an utility that generates keys for use with TSIG (Transaction Signatures) as defined in **RFC 2845**. The resulting keys can be used, for example, to secure dynamic DNS updates to a zone, or for the *rndc* command channel.

A domain name can be specified on the command line to be used as the name of the generated key. If no name is specified, the default is `tsig-key`.

### 12.33.3 Options

**-a** algorithm

This option specifies the algorithm to use for the TSIG key. Available choices are: hmac-md5, hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, and hmac-sha512. The default is hmac-sha256. Options are case-insensitive, and the "hmac-" prefix may be omitted.

**-h**

This option prints a short summary of options and arguments.

### 12.33.4 See Also

*nsupdate(1)*, *named.conf(5)*, *named(8)*, BIND 9 Administrator Reference Manual.

CHAPTER

# THIRTEEN

# GENERAL DNS REFERENCE INFORMATION

## 13.1 Requests for Comment (RFCs)

Specification documents for the Internet protocol suite, including the DNS, are published as part of the Request for Comments (RFCs) series of technical notes. The standards themselves are defined by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). RFCs can be viewed online at: https://www.rfc-editor.org/.

While reading RFCs, please keep in mind that **not all RFCs are standards**, and also that the validity of documents does change over time. Every RFC needs to be interpreted in the context of other documents.

BIND 9 strives for strict compliance with IETF standards. To the best of our knowledge, BIND 9 complies with the following RFCs, with the caveats and exceptions listed in the numbered notes below. Many of these RFCs were written by current or former ISC staff members. The list is non-exhaustive.

Some of these RFCs, though DNS-related, are not concerned with implementing software.

### 13.1.1 Protocol Specifications

**RFC 1034** - P. Mockapetris. *Domain Names — Concepts and Facilities.* November 1987.

**RFC 1035** - P. Mockapetris. *Domain Names — Implementation and Specification.* November 1987.[1] [2]

**RFC 1183** - C. F. Everhart, L. A. Mamakos, R. Ullmann, P. Mockapetris. *New DNS RR Definitions.* October 1990.

**RFC 1521** - N. Borenstein, N. Freed - *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies.* September 1993.[17]

**RFC 1706** - B. Manning and R. Colella. *DNS NSAP Resource Records.* October 1994.

**RFC 1712** - C. Farrell, M. Schulze, S. Pleitner, and D. Baldoni. *DNS Encoding of Geographical Location.* November 1994.

**RFC 1876** - C. Davis, P. Vixie, T. Goodwin, and I. Dickinson. *A Means for Expressing Location Information in the Domain Name System.* January 1996.

**RFC 1982** - R. Elz and R. Bush. *Serial Number Arithmetic.* August 1996.

**RFC 1995** - M. Ohta. *Incremental Zone Transfer in DNS.* August 1996.

**RFC 1996** - P. Vixie. *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY).* August 1996.

**RFC 2136** - P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE).* April 1997.

**RFC 2163** - A. Allocchio. *Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM).* January 1998.

[1] Queries to zones that have failed to load return SERVFAIL rather than a non-authoritative response. This is considered a feature.
[2] CLASS ANY queries are not supported. This is considered a feature.
[17] Only the Base 64 encoding specification is supported.

**RFC 2181** - R. Elz and R. Bush. *Clarifications to the DNS Specification.* July 1997.

**RFC 2230** - R. Atkinson. *Key Exchange Delegation Record for the DNS.* November 1997.

**RFC 2308** - M. Andrews. *Negative Caching of DNS Queries (DNS NCACHE).* March 1998.

**RFC 2539** - D. Eastlake, 3rd. *Storage of Diffie-Hellman Keys in the Domain Name System (DNS).* March 1999.

**RFC 2782** - A. Gulbrandsen, P. Vixie, and L. Esibov. *A DNS RR for Specifying the Location of Services (DNS SRV).* February 2000.

**RFC 2930** - D. Eastlake, 3rd. *Secret Key Establishment for DNS (TKEY RR).* September 2000.

**RFC 2931** - D. Eastlake, 3rd. *DNS Request and Transaction Signatures (SIG(0)s).* September 2000.[3]

**RFC 3007** - B. Wellington. *Secure Domain Name System (DNS) Dynamic Update.* November 2000.

**RFC 3110** - D. Eastlake, 3rd. *RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS).* May 2001.

**RFC 3123** - P. Koch. *A DNS RR Type for Lists of Address Prefixes (APL RR).* June 2001.

**RFC 3225** - D. Conrad. *Indicating Resolver Support of DNSSEC.* December 2001.

**RFC 3226** - O. Gudmundsson. *DNSSEC and IPv6 A6 Aware Server/Resolver Message Size Requirements.* December 2001.

**RFC 3363** - R. Bush, A. Durand, B. Fink, O. Gudmundsson, and T. Hain. *Representing Internet Protocol Version 6 (IPv6) Addresses in the Domain Name System (DNS).* August 2002.[15]

**RFC 3403** - M. Mealling. *Dynamic Delegation Discovery System (DDDS). Part Three: The Domain Name System (DNS) Database.* October 2002.

**RFC 3492** - A. Costello. *Punycode: A Bootstring Encoding of Unicode for Internationalized Domain Names in Applications (IDNA).* March 2003.[18]

**RFC 3493** - R. Gilligan, S. Thomson, J. Bound, J. McCann, and W. Stevens. *Basic Socket Interface Extensions for IPv6.* March 2003.

**RFC 3496** - A. G. Malis and T. Hsiao. *Protocol Extension for Support of Asynchronous Transfer Mode (ATM) Service Class-aware Multiprotocol Label Switching (MPLS) Traffic Engineering.* March 2003.

**RFC 3596** - S. Thomson, C. Huitema, V. Ksinant, and M. Souissi. *DNS Extensions to Support IP Version 6.* October 2003.

**RFC 3597** - A. Gustafsson. *Handling of Unknown DNS Resource Record (RR) Types.* September 2003.

**RFC 3645** - S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead, and R. Hall. *Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG).* October 2003.

**RFC 4025** - M. Richardson. *A Method for Storing IPsec Keying Material in DNS.* March 2005.

**RFC 4033** - R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *DNS Security Introduction and Requirements.* March 2005.

**RFC 4034** - R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *Resource Records for the DNS Security Extensions.* March 2005.

**RFC 4035** - R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *Protocol Modifications for the DNS Security Extensions.* March 2005.

**RFC 4255** - J. Schlyter and W. Griffin. *Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints.* January 2006.

[3] When receiving a query signed with a SIG(0), the server is only able to verify the signature if it has the key in its local authoritative data; it cannot do recursion or validation to retrieve unknown keys.

[15] Section 4 is ignored.

[18] BIND 9 requires `--with-libidn2` to enable entry of IDN labels within dig, host, and nslookup at compile time. ACE labels are supported everywhere with or without `--with-libidn2`.

**RFC 4343** - D. Eastlake, 3rd. *Domain Name System (DNS) Case Insensitivity Clarification.* January 2006.

**RFC 4398** - S. Josefsson. *Storing Certificates in the Domain Name System (DNS).* March 2006.

**RFC 4470** - S. Weiler and J. Ihren. *Minimally covering NSEC Records and DNSSEC On-line Signing.* April 2006.[6]

**RFC 4509** - W. Hardaker. *Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs).* May 2006.

**RFC 4592** - E. Lewis. *The Role of Wildcards in the Domain Name System.* July 2006.

**RFC 4635** - D. Eastlake, 3rd. *HMAC SHA (Hashed Message Authentication Code, Secure Hash Algorithm) TSIG Algorithm Identifiers.* August 2006.

**RFC 4701** - M. Stapp, T. Lemon, and A. Gustafsson. *A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR).* October 2006.

**RFC 4955** - D. Blacka. *DNS Security (DNSSEC) Experiments.* July 2007.[7]

**RFC 5001** - R. Austein. *DNS Name Server Identifier (NSID) Option.* August 2007.

**RFC 5011** - M. StJohns. *Automated Updates of DNS Security (DNSSEC) Trust Anchors.*

**RFC 5155** - B. Laurie, G. Sisson, R. Arends, and D. Blacka. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence.* March 2008.

**RFC 5205** - P. Nikander and J. Laganier. *Host Identity Protocol (HIP) Domain Name System (DNS) Extension.* April 2008.

**RFC 5452** - A. Hubert and R. van Mook. *Measures for Making DNS More Resilient Against Forged Answers.* January 2009.[8]

**RFC 5702** - J. Jansen. *Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC.* October 2009.

**RFC 5891** - J. Klensin. *Internationalized Domain Names in Applications (IDNA): Protocol.* August 2010Page 520, 18

**RFC 5936** - E. Lewis and A. Hoenes, Ed. *DNS Zone Transfer Protocol (AXFR).* June 2010.

**RFC 5952** - S. Kawamura and M. Kawashima. *A Recommendation for IPv6 Address Text Representation.* August 2010.

**RFC 6052** - C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li. *IPv6 Addressing of IPv4/IPv6 Translators.* October 2010.

**RFC 6147** - M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum. *DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers.* April 2011.[9]

**RFC 6604** - D. Eastlake, 3rd. *xNAME RCODE and Status Bits Clarification.* April 2012.

**RFC 6605** - P. Hoffman and W. C. A. Wijngaards. *Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC.* April 2012.[10]

**RFC 6672** - S. Rose and W. Wijngaards. *DNAME Redirection in the DNS.* June 2012.

**RFC 6698** - P. Hoffman and J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA.* August 2012.

**RFC 6725** - S. Rose. *DNS Security (DNSSEC) DNSKEY Algorithm IANA Registry Updates.* August 2012.[11]

**RFC 6742** - RJ Atkinson, SN Bhatti, U. St. Andrews, and S. Rose. *DNS Resource Records for the Identifier-Locator Network Protocol (ILNP).* November 2012.

[6] Minimally Covering NSEC records are accepted but not generated.
[7] BIND 9 interoperates with correctly designed experiments.
[8] `named` only uses ports to extend the ID space; addresses are not used.
[9] Section 5.5 does not match reality. `named` uses the presence of DO=1 to detect if validation may be occurring. CD has no bearing on whether validation occurs.
[10] Compliance is conditional on the OpenSSL library being linked against a supporting ECDSA.
[11] RSAMD5 support has been removed. See **RFC 8624**.

**RFC 6840** - S. Weiler, Ed., and D. Blacka, Ed. *Clarifications and Implementation Notes for DNS Security (DNSSEC).* February 2013.[12]

**RFC 6891** - J. Damas, M. Graff, and P. Vixie. *Extension Mechanisms for DNS (EDNS(0)).* April 2013.

**RFC 7043** - J. Abley. *Resource Records for EUI-48 and EUI-64 Addresses in the DNS.* October 2013.

**RFC 7050** - T. Savolainen, J. Korhonen, and D. Wing. *Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis.* November 2013.[20]

**RFC 7208** - S. Kitterman. *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1.* April 2014.

**RFC 7314** - M. Andrews. *Extension Mechanisms for DNS (EDNS) EXPIRE Option.* July 2014.

**RFC 7344** - W. Kumari, O. Gudmundsson, and G. Barwood. *Automating DNSSEC Delegation Trust Maintenance.* September 2014.[13]

**RFC 7477** - W. Hardaker. *Child-to-Parent Synchronization in DNS.* March 2015.

**RFC 7553** - P. Faltstrom and O. Kolkman. *The Uniform Resource Identifier (URI) DNS Resource Record.* June 2015.

**RFC 7583** - S. Morris, J. Ihren, J. Dickinson, and W. Mekking. *DNSSEC Key Rollover Timing Considerations.* October 2015.

**RFC 7766** - J. Dickinson, S. Dickinson, R. Bellis, A. Mankin, and D. Wessels. *DNS Transport over TCP - Implementation Requirements.* March 2016.

**RFC 7828** - P. Wouters, J. Abley, S. Dickinson, and R. Bellis. *The edns-tcp-keepalive EDNS0 Option.* April 2016.

**RFC 7830** - A. Mayrhofer. *The EDNS(0) Padding Option.* May 2016.[14]

**RFC 7858** - Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. *Specification for DNS over Transport Layer Security (TLS).* May 2016.[21]

**RFC 7929** - P. Wouters. *DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP.* August 2016.

**RFC 8078** - O. Gudmundsson and P. Wouters. *Managing DS Records from the Parent via CDS/CDNSKEY.* March 2017.[22]

**RFC 8080** - O. Sury and R. Edmonds. *Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC.* February 2017.

**RFC 8484** - P. Hoffman and P. McManus. *DNS Queries over HTTPS (DoH).* October 2018.[21]

**RFC 8509** - G. Huston, J. Damas, W. Kumari. *A Root Key Trust Anchor Sentinel for DNSSEC.* December 2018.

**RFC 8624** - P. Wouters and O. Sury. *Algorithm Implementation Requirements and Usage Guidance for DNSSEC.* June 2019.

**RFC 8659** - P. Hallam-Baker, R. Stradling, and J. Hoffman-Andrews. *DNS Certification Authority Authorization (CAA) Resource Record.* November 2019.

**RFC 8880** - S. Cheshire and D. Schinazi. *Special Use Domain Name 'ipv4only.arpa'.* August 2020.

**RFC 8945** - F. Dupont, S. Morris, P. Vixie, D. Eastlake 3rd, O. Gudmundsson, and B. Wellington. *Secret Key Transaction Authentication for DNS (TSIG).* November 2020.

**RFC 9103** - W. Toorop, S. Dickinson, S. Sahib, P. Aras, and A. Mankin. *DNS Zone Transfer over TLS.* August 2021.[23]

[12] Section 5.9 - Always set CD=1 on queries. This is *not* done, as it prevents DNSSEC from working correctly through another recursive server. When talking to a recursive server, the best algorithm is to send CD=0 and then send CD=1 iff SERVFAIL is returned, in case the recursive server has a bad clock and/or bad trust anchor. Alternatively, one can send CD=1 then CD=0 on validation failure, in case the recursive server is under attack or there is stale/bogus authoritative data.

[20] **RFC 7050** is updated by **RFC 8880**.

[13] Updating of parent zones is not yet implemented.

[14] `named` does not currently encrypt DNS requests, so the PAD option is accepted but not returned in responses.

[21] Forwarding DNS queries over encrypted transports is not supported yet.

[22] Updating of parent zones is not yet implemented.

[23] Strict TLS and Mutual TLS authentication mechanisms are not supported yet.

**RFC 9432** - P. van Dijk, L. Peltan, O. Sury, W. Toorop, C.R. Monshouwer, P. Thomassen, A. Sargsyan. *DNS Catalog Zones.* July 2023.

**RFC 9460** - B. Schwartz, M. Bishop and E. Nygren, *Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records).* November 2023.[24]

### 13.1.2 Best Current Practice RFCs

**RFC 2219** - M. Hamilton and R. Wright. *Use of DNS Aliases for Network Services.* October 1997.

**RFC 2317** - H. Eidnes, G. de Groot, and P. Vixie. *Classless IN-ADDR.ARPA Delegation.* March 1998.

**RFC 2606** - D. Eastlake, 3rd and A. Panitz. *Reserved Top Level DNS Names.* June 1999.[16]

**RFC 3901** - A. Durand and J. Ihren. *DNS IPv6 Transport Operational Guidelines.* September 2004.

**RFC 5625** - R. Bellis. *DNS Proxy Implementation Guidelines.* August 2009.

**RFC 6303** - M. Andrews. *Locally Served DNS Zones.* July 2011.

**RFC 7793** - M. Andrews. *Adding 100.64.0.0/10 Prefixes to the IPv4 Locally-Served DNS Zones Registry.* May 2016.

**RFC 8906** - M. Andrews and R. Bellis. *A Common Operational Problem in DNS Servers: Failure to Communicate.* September 2020.

**RFC 9276** - W. Hardaker and V. Dukhovni. *Guidance for NSEC3 Parameter Settings.* August 2022.

### 13.1.3 For Your Information

**RFC 1101** - P. Mockapetris. *DNS Encoding of Network Names and Other Types.* April 1989.

**RFC 1123** - R. Braden. *Requirements for Internet Hosts - Application and Support.* October 1989.

**RFC 1535** - E. Gavron. *A Security Problem and Proposed Correction With Widely Deployed DNS Software.* October 1993.

**RFC 1536** - A. Kumar, J. Postel, C. Neuman, P. Danzig, and S. Miller. *Common DNS Implementation Errors and Suggested Fixes.* October 1993.

**RFC 1912** - D. Barr. *Common DNS Operational and Configuration Errors.* February 1996.

**RFC 2874** - M. Crawford and C. Huitema. *DNS Extensions to Support IPv6 Address Aggregation and Renumbering.* July 2000.[4]

**RFC 3833** - D. Atkins and R. Austein. *Threat Analysis of the Domain Name System (DNS).* August 2004.

**RFC 4074** - Y. Morishita and T. Jinmei. *Common Misbehavior Against DNS Queries for IPv6 Addresses.* June 2005.

**RFC 4294** - J. Loughney, Ed. - *IPv6 Node Requirements.* April 2006.[19]

**RFC 4431** - M. Andrews and S. Weiler. *The DNSSEC Lookaside Validation (DLV) DNS Resource Record.* February 2006.[5]

**RFC 4892** - S. Woolf and D. Conrad. *Requirements for a Mechanism Identifying a Name Server Instance.* June 2007.

**RFC 6781** - O. Kolkman, W. Mekking, and R. Gieben. *DNSSEC Operational Practices, Version 2.* December 2012.

**RFC 7129** - R. Gieben and W. Mekking. *Authenticated Denial of Existence in the DNS.* February 2014.

**RFC 8749** - W. Mekking and D. Mahoney. *Moving DNSSEC Lookaside Validation (DLV) to Historic Status.* March 2020.

[24] Additional section processing is not supported for HTTPS and SVCB records.
[16] This does not apply to DNS server implementations.
[4] Compliance is with loading and serving of A6 records only. A6 records were moved to the experimental category by **RFC 3363**.
[19] Section 5.1 - DNAME records are fully supported.
[5] Compliance is with loading and serving of DLV records only. DLV records were moved to the historic category by **RFC 8749**.

## 13.2 Notes

## 13.3 Internet Drafts

Internet Drafts (IDs) are rough-draft working documents of the Internet Engineering Task Force (IETF). They are, in essence, RFCs in the preliminary stages of development. Implementors are cautioned not to regard IDs as archival, and they should not be quoted or cited in any formal documents unless accompanied by the disclaimer that they are "works in progress." IDs have a lifespan of six months, after which they are deleted unless updated by their authors.

CHAPTER

# FOURTEEN

# RELEASE NOTES

**Contents**


- *Release Notes*
  - *Introduction*
  - *Supported Platforms*
  - *Download*
  - *Known Issues*
  - *Notes for BIND 9.20.29*
    - *Security Fixes*
    - *Feature Changes*
    - *Bug Fixes*
  - *Notes for BIND 9.20.28*
  - *Notes for BIND 9.20.27*
    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
  - *Notes for BIND 9.20.26*
    - *Security Fixes*
    - *Removed Features*
    - *Bug Fixes*
  - *Notes for BIND 9.20.25*
  - *Notes for BIND 9.20.24*
    - *Removed Features*
    - *Feature Changes*
    - *Bug Fixes*
  - *Notes for BIND 9.20.23*

    - *Security Fixes*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.22*
    - *Security Fixes*
    - *Bug Fixes*
- *Notes for BIND 9.20.21*
    - *Security Fixes*
    - *Bug Fixes*
- *Notes for BIND 9.20.20*
    - *Security Fixes*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.19*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.18*
    - *Security Fixes*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.17*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.16*
    - *Bug Fixes*
- *Notes for BIND 9.20.15*
    - *Security Fixes*
    - *New Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.14*
- *Notes for BIND 9.20.13*
    - *New Features*
    - *Removed Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.12*

    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.11*
    - *Security Fixes*
    - *New Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.10*
    - *New Features*
    - *Removed Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.9*
    - *Security Fixes*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.8*
    - *New Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.7*
    - *New Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.6*
    - *New Features*
    - *Bug Fixes*
- *Notes for BIND 9.20.5*
    - *Security Fixes*
    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.4*
    - *New Features*
    - *Removed Features*
    - *Feature Changes*
    - *Bug Fixes*
- *Notes for BIND 9.20.3*

    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
    - *Known Issues*
  - *Notes for BIND 9.20.2*
    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
    - *Known Issues*
  - *Notes for BIND 9.20.1*
    - *New Features*
    - *Feature Changes*
    - *Bug Fixes*
    - *Known Issues*
  - *Notes for BIND 9.20.0*
    - *New Features*
    - *Removed Features*
    - *Deprecated Features*
    - *Feature Changes*
    - *Bug Fixes*
    - *Known Issues*
  - *License*
  - *End of Life*
  - *Thank You*


## 14.1 Introduction

BIND 9.20 (Extended Support Version) is a stable branch, suitable for production use. This document summarizes significant changes since the last production release on the 9.18 branch. Please see the *Changelog* file for a more detailed list of changes and bug fixes.

## 14.2 Supported Platforms

See the *Supported Platforms* section in the *Resource Requirements* chapter.

## 14.3 Download

The latest versions of BIND 9 software can always be found at https://www.isc.org/download/. There you will find additional information about each release, and source code.

## 14.4 Known Issues

The list of known issues affecting the latest version in the 9.20 branch can be found at https://gitlab.isc.org/isc-projects/bind9/-/wikis/Known-Issues-in-BIND-9.20

## 14.5 Notes for BIND 9.20.29

### 14.5.1 Security Fixes

- Prevent excessive CPU use validating crafted DNSSEC responses. **(CVE-2026-19668)**

  A malicious authoritative server could serve a securely delegated zone whose DS and DNSKEY records carried many distinct key tags but no valid match, forcing a validating resolver into excessive key-tag matching and high CPU use for every query. This work is now bounded by the per-query validation limit (*max-validations-per-fetch*).

  ISC would like to thank Zuyao Xu and Xiang Li of the All-in-One Security and Privacy Laboratory, Nankai University, for bringing this vulnerability to our attention. **[GL #5349]**

- Require a TSIG on every message of incoming zone transfers. **(CVE-2026-19033)**

  Previously, *named* accepted TSIG-signed zone transfers in which some messages were unsigned, and processed those messages before the next signature could vouch for them. It now requires a TSIG on every message of an incoming AXFR or IXFR; all modern nameservers already sign every message, so no change is expected in practice. **[GL #6062]**

- Prevent a DNSSEC downgrade of secure delegations via unrelated NSEC3 records. **(CVE-2026-77119)**

  A validating resolver could be tricked into treating a secure delegation as unsigned and accepting forged answers for names beneath it, if an attacker could inject responses to its queries. Such forged proofs are now rejected. **[GL #6234]**

- Prevent forged DNSSEC-validated NXDOMAIN responses. **(CVE-2026-19941)**

  A validating resolver could accept a signed NSEC record from an unrelated zone as proof that a wildcard did not exist. An on-path attacker or malicious forwarder controlling a signed zone could therefore forge an authenticated NXDOMAIN response for a name that should resolve through a wildcard. The wildcard-denial and name-nonexistence proofs are now required to be signed by the same zone. **[GL #6253]**

- DNS64 with *break-dnssec* could cause an assertion failure. **(CVE-2026-19666)**

  When a *dns64* statement is configured with `break-dnssec yes;` and its `exclude` list matches some but not all of the addresses in an AAAA RRset, *named* removes the excluded addresses from the answer instead of synthesizing new ones. If the answer being filtered had been cached together with a proof that the queried name did not exist — which is what a wildcard match produces — *named* terminated with an assertion failure.

  Only recursive resolvers are affected, and only when `break-dnssec yes;` is in use; the answer has to come from the cache, so a server that is only authoritative cannot reach this.

  ISC would like to thank Rintaro Kawasugi for bringing this vulnerability to our attention. **[GL #6301]**

- Reject oversized negative cache records. **(CVE-2026-19667)**

A single crafted response from a server could make a resolver cache a malformed negative entry and then terminate with an assertion failure when reading it back. Only recursive resolvers are affected, on a default configuration.

ISC would like to thank Rintaro Kawasugi for bringing this vulnerability to our attention. **[GL #6302]**

- Prevent resolver crash with cached DNSSEC proofs. **(CVE-2026-19662)**

  Under certain timing conditions, concurrent recursive queries could cause `named` to crash when cached DNSSEC NOQNAME proof data was replaced while still in use. Cached proof data is now retained until all queries using it have completed.

  ISC would like to thank Samy Medjahed/Ap4sh for bringing this vulnerability to our attention. **[GL #6333]**

- Discard repeated SOA, CNAME, and DNAME records when parsing DNS messages. **(CVE-2026-75029)**

  A DNS message could carry the same SOA, CNAME, or DNAME record many times, and `named` kept every copy while parsing it. With name compression those copies took up far more memory internally than in the message itself, and every later processing step had to handle all of them. Only the first copy of such a record is now kept; identical repeats are discarded. **[GL #6335]**

- Fix an unauthenticated crash on HTTPS using SIG(0). **(CVE-2026-77692)**

  A specifically crafted HTTPS query using SIG(0) as authentication could crash `named` if the client closed the connection before `named` actually verified the signature. This is now fixed.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #6343]**

- Cached HTTPS/SVCB aliases could exhaust resolver CPU. **(CVE-2026-81736)**

  A recursive resolver that had cached a large set of interlinked HTTPS or SVCB records in alias form could be driven to do an excessive amount of work assembling a single response, because it followed every cached alias target when building the additional section. A client permitted to use recursion, together with an attacker-controlled zone used to plant the records, could repeat small queries to consume enough CPU to delay or deny service to other clients. The amount of additional processing done for one query is now bounded.

  ISC would like to thank Henrique Pereira for bringing this vulnerability to our attention. **[GL #6347]**

- Prevent TKEY queries from terminating `named` without global options. **(CVE-2026-76163)**

  The `named` process could terminate unexpectedly when a remote client sent a TKEY query and the configuration did not include a global `options` statement. This has been fixed.

  ISC would like to thank Owais Lone (thesecguy) for bringing this vulnerability to our attention. **[GL #6357]**

- Out-of-zone records in a zone database could be served as authoritative. **(CVE-2026-78301)**

  When a zone database contained records for names outside the zone — such as a delegation above the zone apex, left behind by a secondary that had accepted out-of-zone data from its primary — the server could treat them as authoritative and answer queries for names inside the zone with that out-of-zone data instead of the zone's own. A server that was also a resolver could follow such a delegation and cache the answers of the server it named, affecting names outside the configured zone. Zone database lookups are now confined to names at or below the zone's origin.

  ISC would like to thank Henrique Pereira for bringing this vulnerability to our attention. **[GL #6361]**

- Fix crash on wildcard answers carrying both NSEC and NSEC3 proofs. **(CVE-2026-80274)**

  When a wildcard answer arrived with both NSEC and NSEC3 records at the name proving that the queried name did not exist, the resolver could pick different records when caching the answer and when retrieving the proof, depending on the order in which the authoritative server sent them. This could terminate `named` with an assertion failure, fail the query with SERVFAIL, or serve a denial record other than the one that had been verified. The resolver now caches and serves the same denial record it accepted when the answer was received.

  ISC would like to thank hythyt for bringing this vulnerability to our attention. **[GL #6369]**

- Following HTTPS/SVCB aliases could leak resolver cache memory. **(CVE-2026-81563)**

  When a recursive server answered a query for an HTTPS or SVCB record in alias form and the alias target had more than 13 records, the target records were pinned in the cache permanently instead of being released once the answer was sent. A remote party who could make the server follow such aliases to a steady stream of fresh names could grow the cache beyond the configured *max-cache-size* until the server was unable to resolve unrelated names. The records are now released correctly.

  ISC would like to thank Samy Medjahed/Ap4sh for bringing this vulnerability to our attention. **[GL #6374]**

### 14.5.2 Feature Changes

- Reject oversized and malformed DNSKEY records up front.

  Oversized RSA key material in a DNSKEY record was only rejected after it had been converted, allocating memory proportional to the record size. Such records are now rejected before conversion, as are Ed25519 and Ed448 keys with trailing bytes that were previously silently ignored. **[GL #4537]**

### 14.5.3 Bug Fixes

- Prevent a crash when using both *dns64* and `filter-a`.

  An assertion failure was possible when using both *dns64* and the `filter-a` plugin simultaneously; this has been fixed. **[GL #5979]**

- Stop passing UDP client addresses to *update-policy* `external` helpers.

  Dynamic update rules of type `external` delegate the authorization decision to an external helper daemon. The client address field in the helper request is only meant to carry TCP client addresses, which cannot easily be spoofed, but UDP client addresses were passed as well, so the helper could base its decision on an untrustworthy address. For updates arriving over UDP, the helper request no longer includes a client address. **[GL #6061]**

- Missing required NSEC3 for delegation not detected.

  A missing required NSEC3 record for an insecure delegation in a non-opt-out range was not being detected. This has been fixed. **[GL #6063]**

- Tighten EUI48 and EUI64 text parsing.

  Malformed EUI48 and EUI64 records could be accepted. This has been fixed. **[GL #6082]**

- GeoIP ACL state could be stale or wrong after reload.

  Previously, *named* cached GeoIP information after looking it up, but the cached information was not invalidated when the GeoIP database was reloaded, so it could continue to be used. Existing cached GeoIP information is now invalidated as part of the reloading process. **[GL #6083]**

- Honor DNSSEC policy key tag ranges.

  When a *dnssec-policy* configured a non-default `tag-range`, *dnssec-keygen* and *dnssec-ksr* could accept generated keys outside that range. Both tools now honor the configured minimum and maximum key tags. **[GL #6091]**

- Fix a double free in *mdig* when EDNS options are specified.

  The *mdig* utility could terminate with a double free when EDNS options were specified on the command line. This has been fixed. **[GL #6095]**

- Fix a crash when an IXFR falls back to AXFR with updates still pending.

  When a secondary zone received an incremental transfer (IXFR) and the primary then caused *named* to fall back to a full transfer (AXFR) while some of the already-received incremental changes were still waiting to be applied,

*named* could later crash when that transfer finished. The pending changes are now discarded correctly before the AXFR retry. **[GL #6114]**

- Fix DS requests to parental agents over TLS.

  TLS configuration for parental agents was being ignored when sending DS requests. This has been fixed. **[GL #6135]**

- Fix the *rndc-confgen -q* (quiet) option.

  The command-line parsing in *rndc-confgen* was broken, so *rndc-confgen -q* did not work. This has been fixed. **[GL #6187]**

- Enforce query ACLs for redirect zones and searched DLZs.

  Queries answered from redirect zones or searched DLZ databases did not consistently honor *allow-query* and *allow-query-on*, potentially exposing restricted DNS data to excluded clients or through excluded listening addresses. These ACLs are now enforced before redirect or DLZ data is returned. **[GL #6251] [GL #6252]**

- Check `asnum` validity in GeoIP ACLs.

  The validity of autonomous system (AS) numbers is now checked at configuration time when parsing GeoIP ACLs that use `asnum` elements.

  `asnum` values start with an optional case-insensitive `AS` prefix, followed only by decimal digits, with no spaces or other extraneous characters. The value represented cannot exceed 2^32. **[GL #6255]**

- Fix a crash on *remote-servers* lists that reference themselves.

  Since 9.21.16 and 9.20.17, a *remote-servers*, *primaries*, `masters`, or *parental-agents* list that referenced itself, directly or through another list, made *named* crash on startup or reconfiguration. Such references are again skipped and the remaining entries in the list are used, as in earlier versions. **[GL #6287]**

- A record from outside a response policy zone could crash *named*.

  A response policy zone transferred from a primary can contain a record whose name lies outside the zone. Such a record could terminate *named* with an assertion failure, both when it arrived and again at every startup afterwards, because a secondary keeps it in its own copy of the zone. Records like this are now rejected and logged; previously one could also silently create a policy entry for an unrelated name. **[GL #6304]**

- Invalid *key-store* configuration could abort the DNSSEC tools.

  An invalid *key-store* block named `key-directory` in a configuration file could abort the DNSSEC tools. This has been fixed. **[GL #6313]**

- NSEC signature set could bypass the secure-delegation check.

  When proving that a delegation was insecure, the validator bounded an NSEC record's authority by the signer of whichever RRSIG happened to come first in the record's signature set, rather than the signature that actually verified. A grandparent NSEC padded with an extra, unverifiable signature could therefore pass the check that kept such proofs from reaching below a signed child zone. The validator now requires every signature on the NSEC to name the same signer and refuses proofs whose signature set is malformed or larger than *max-validations-per-fetch* allows. **[GL #6321]**

- Fix a possible *nsupdate* issue when using GSS-TSIG.

  The *nsupdate* process could terminate unexpectedly when using the GSS-TSIG mode executed with the *nsupdate -g* option. This has been fixed. **[GL #6325]**

- Fix a crash with a single-element `geoip` sortlist.

  If *named* was configured with a single-element *sortlist* containing a `geoip` ACL element, any matching query triggered an assertion failure. This has been fixed. **[GL #6342]**

- Prevent out-of-bailiwick CNAMEs from evicting cached records.

  A recursive resolver could remove valid cached records when a DNS response contained an out-of-bailiwick CNAME with the same owner name. Out-of-bailiwick data is now discarded before it can modify the cache. **[GL #6345]**

- Restore periodic cleanup of stale resolver address data.

  Stale resolver address data could remain cached until memory pressure or an explicit flush. The cleanup interval has been corrected, so stale entries are removed periodically again. **[GL #6346]**

- Fix *named-checkconf*/*named* crash with malformed key name.

  When a primary/remote-server key name was malformed, *named-checkconf* and *named* were both crashing (after warning about the invalid key name). This is now fixed. **[GL #6362]**

- Prevent resolver crashes while processing DNS over TCP.

  Recursive resolvers could terminate with an assertion failure while processing DNS responses over TCP under sustained traffic. The failure was observed on resolvers configured globally with `forward only;`, but the same transport path is also used by iterative resolution. This has been fixed. **[GL !12537]**

## 14.6 Notes for BIND 9.20.28

> **Note**
>
> The BIND 9.20.28 release was withdrawn after the discovery of a regression in it during pre-release testing.

## 14.7 Notes for BIND 9.20.27

### 14.7.1 New Features

- Disclose active Negative Trust Anchors with Extended DNS Error 33.

  A Negative Trust Anchor (**RFC 7646**) turns off DNSSEC validation for a domain, so a name that would normally fail validation resolves instead. *named* now marks such answers with Extended DNS Error code 33, "Negative Trust Anchor", so operators can see at a glance when a response came back only because an NTA was in effect. **[GL #6268]**

### 14.7.2 Feature Changes

- Speed up RPZ policy zone updates.

  RPZ updates used to be applied one small step at a time, adding overhead on large policy zones. Updates are now applied as a single batch, improving update performance for large RPZ zones, at the cost of no longer overlapping with concurrent updates. **[GL #5787] [GL #6270]**

### 14.7.3 Bug Fixes

- Ensure NSEC authority does not cross zonecut boundary.

  When using a cached NSEC record to prove that a delegation is insecure, *named* now checks that the signer name in the corresponding RRSIG is not above a known secure delegation point. This prevents a signed namespace from being downgraded to insecure using an NSEC record from the grandparent zone. **[GL #5967]**

- Treat an unusable NSEC3 chain as a verification failure.

When transferring in a mirror zone, DNSSEC verification could incorrectly succeed when the zone had an invalid NSEC3PARAM record, leading to subsequent validation failures. This has been fixed. **[GL #6136]**

- Treat non-canonical RPZ prefixes as any other failure.

  RPZ prefixes that were not encoded in canonical form did not work. They are now handled in the same way as any other encoding error. **[GL #6043]**

- Negative caching stopped working with stale-answer-client-timeout set to `0`.

  Negative answers were re-fetched on every query instead of once they actually expired, effectively disabling negative caching. This has been fixed. **[GL #6245]**

- An unterminated OpenSSL private-key `Label:` field could be read past its parser buffer.

  The `Label:` field in a `.private` key file is now checked for length and NUL-termination. Malformed files are rejected. **[GL #6193]**

- Restore SMF support on Solaris and illumos. **[GL #6096]**

- Fix compilation on GNU/Hurd. **[GL #6285]**

- *dig +yaml* was producing invalid YAML when a lookup failed.

  When no server could be reached, *dig* printed its plain-text startup banner ahead of the YAML output, making the result unparsable. *dig* no longer does this and correctly reflects options such as `+nocmd`, `+short` and `+yaml`, regardless of where they appear on the command line. **[GL #1230]**

- Properly prevent TSIG generation command line injection attacks.

  When key names are generated with *rndc-confgen*, *tsig-keygen* and *ddns-confgen*, special characters must be escaped to ensure that the configuration is parsed correctly. **[GL #6071]**

- Fix a potential heap bounds overflow write in *dnssec-signzone*.

  It was possible for *dnssec-signzone* to overflow array bounds while signing. This has been fixed. **[GL #6076]**

- Fix crashes on invalid DNSTAP input in *dnstap-read*.

  Malformed DNSTAP files could trigger a NULL pointer dereference or an out-of-bounds memory read in *dnstap-read*. This has been fixed. **[GL #6077] [GL #6124]**

## 14.8 Notes for BIND 9.20.26

### 14.8.1 Security Fixes

- Correct verification of NSEC3 signer name. **(CVE-2026-10723)**

  Previously, *named* accepted child-zone NSEC3 records where the first label equaled the hash of the parent zone as valid parent-zone closest encloser proofs. This has been fixed.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for bringing this vulnerability to our attention. **[GL #5874]**

- Malformed DNSKEY records could trigger an assertion. **(CVE-2026-10822)**

  Previously, `dns_name_fromwire()` did not honor the record boundary when reading names from the wire, allowing malformed records to be accepted when they should not have been. In particular, malformed DNSKEY records could trigger an assertion failure when being printed. This has been fixed. **[GL #6004]**

- Fix handling of RPZ CNAME expansion that returns too-long name. **(CVE-2026-11331)**

  Previously, if the expansion of a wildcard CNAME RPZ policy resulted in a name that exceeded the length limit, a self-referential CNAME and the original address record were returned, allowing the policy to be bypassed. In

branches up to 9.20 this also left query processing in an inconsistent state, which could trigger an assertion failure. *named* now returns a YXDOMAIN response, without the address.

ISC would like to thank Laith Mash'al (0xmshal) for bringing this vulnerability to our attention. **[GL #5856]**

- Prevent excessive validation work from crafted negative responses. **(CVE-2026-11605)**

  Previously, a validating resolver could be made to perform a large amount of DNSSEC validation work in response to a single answer, consuming excessive CPU. A malicious authoritative server could trigger this by returning a signed negative answer (NXDOMAIN or NODATA) padded with many denial-of-existence proof records, which the resolver continued to verify beyond its per-query validation limit. It now enforces that limit on negative answers and returns SERVFAIL once the limit is reached. **[GL #4463]**

- Prevent cache exhaustion under sustained attack. **(CVE-2026-11622)**

  Cache memory could become exhausted with expired entries whose memory was not released, due to a sustained attack on the same DNS name that prevented the cleanup. This has been fixed. **[GL #4760]**

- Stop accepting invalid signed wildcard records. **(CVE-2026-11721)**

  Signed wildcard responses in which the Labels field in the RRSIG record was less than the number of labels in the Signer Name field were being incorrectly accepted. This in turn broke *synth-from-dnssec*, which depends on such records being correctly validated. This has been fixed.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for bringing this vulnerability to our attention. **[GL #5871]**

- Do not assert for some specific CNAME and DNAME queries. **(CVE-2026-12617)**

  A bug in the resolver's handling of certain cached DNAME and CNAME responses could cause *named* to trigger an assertion failure and exit. An attacker controlling a domain name and the authoritative DNS server it was hosted on could exploit this behavior to cause a denial-of-service. This has been fixed.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for bringing this vulnerability to our attention. **[GL #5946]**

- Prevent crash from malformed NSEC/NSEC3 response. **(CVE-2026-13204)**

  An assertion could be triggered by an improperly signed NOQNAME proof. This has been fixed.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for bringing this vulnerability to our attention. **[GL #5985]**

- Fix DNSSEC validation bypass via out-of-zone NSEC Next Field. **(CVE-2026-13321)**

  Previously, a malicious zone with out-of-zone NSEC next-owner names could cause a DNSSEC-validating resolver to cache such a record and, if *synth-from-dnssec* was enabled, to generate negative answers for any zone that was covered by the range. This has been fixed.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for bringing this vulnerability to our attention. **[GL #5873]**

- Reclaim memory promptly when DNSSEC validations are canceled.

  When a resolver is flooded with queries that require DNSSEC validation - for example during a random-subdomain attack - many of those validations are canceled before they complete. Previously, a canceled validation still kept its place in the internal work queue and held the associated response in memory until that queued work eventually ran, so memory could climb sharply under sustained load. The internal work queue is now dropped as soon as the validation is canceled, releasing the memory it was holding. **[GL #4760]**

### 14.8.2 Removed Features

- Remove the secondary validator in `query.c`.

  Previously, when the additional section of a response was being populated, if cached data was found with pending trust, it would be opportunistically validated. The code implementing this validation was not quite formally correct. Rather than fixing it, the code has been removed: RRsets with pending trust are now omitted from responses. **[GL #5966] [GL #5968] [GL #5972]**

### 14.8.3 Bug Fixes

- Fix a bug in DNS UPDATE processing with *inline-signing* enabled.

  In rare cases the *named* process could terminate unexpectedly when processing authorized DNS UPDATE messages in quick succession that were updating a zone with *inline-signing* enabled. This has been fixed. **[GL #5816]**

- Properly detect private records before copying.

  Previously, an assertion was triggered when trying to copy a private record to a buffer for modification. *named* now extends the private type detection and copies the contents after rejecting invalid private records. **[GL #5857]**

- Tighten referral DS acceptance.

  Previously, *named* accepted DS records for sibling zones when it shouldn't have. This has been fixed. **[GL #5870]**

- Don't synthesize negative responses with pending NSEC.

  If an NSEC record has not yet been validated and is cached with trust pending, *named* no longer uses it to synthesize negative responses. **[GL #5872] [GL #5887] [GL #5977]**

- Check that an NSEC signer is at or above the name to be validated.

  A check has been added to ensure that an NSEC record being used as a proof of nonexistence for a given name is not signed by a name lower in the DNS hierarchy than the one in question. **[GL #5876]**

- Don't evict DNSSEC-validated cache data on a CD=1 NXDOMAIN.

  When a client sent a query with the checking-disabled (CD) bit set and the answer was NXDOMAIN, the resolver cached that unvalidated negative response and discarded any DNSSEC-validated records it already held for the same name, even though the validated data was more trustworthy. A single such response - including a forged one - could flush validated records from the cache and force the resolver to fetch them again. The resolver now checks the trust level of the existing data first and leaves the cache unchanged when it is already validated. **[GL #5877]**

- Fix a *deny-answer-aliases* configuration bypass issue.

  It was possible to use a maliciously crafted authoritative zone to make a *named* resolver synthesize a DNAME "alias" that should have been rejected by the configured *deny-answer-aliases* option. This has been fixed. **[GL #5930]**

- Reject external referrals from forwarders.

  Under a `forward first;` policy in a forwarding zone, *named* could accept NS records above the forward zone apex from negative responses. This has been fixed. **[GL #5937]**

- Fix a zone transfer over TLS (XoT) issue when using the opportunistic TLS mode.

  The *named* process, running as a secondary DNS server and configured to transfer a zone from a primary server using an encrypted XoT transport in opportunistic TLS mode (i.e. without peer certificate/hostname validation), could terminate unexpectedly when the TLS ALPN negotiation with the primary server was unsuccessful. This has been fixed. **[GL #5957]**

- Unvalidated opt-out NSEC3 could be accepted in insecurity proof.

  When determining whether an insecure delegation was legitimate, NSEC3 opt-out records which had not yet passed validation could be used. This has been fixed. **[GL #5970]**

- Check wildcard signer and NOQNAME signer match.

  A positive wildcard answer, and the NSEC3 proof that the requested name doesn't exist in the zone, must both be from the same zone. Otherwise, an NSEC3 from an ancestor zone could be used to interfere with validation.

  *named* now retrieves the signer name from a wildcard response's signature. An NSEC3 record cannot be used as a NOQNAME proof for the wildcard unless it exactly matches the name one level above the NSEC3. **[GL #5971]**

- Fix CNAME resolution failure caused by a cached SERVFAIL response.

  Under certain circumstances, a cached SERVFAIL response could incorrectly prevent successful resolution of a CNAME target. This could cause resolution failures to persist until the cached SERVFAIL entry expired, even when the CNAME target itself was otherwise resolvable. This has been fixed. **[GL #5983]**

- Reject unsupported RSA DNSKEY shapes during DNSSEC validation.

  An authoritative server publishing an RSA DNSKEY with an unusually large modulus or an exotic public exponent could make each DNSSEC signature check on a validating recursive resolver noticeably more expensive than for a normally sized key. Such DNSKEYs are now treated as invalid. **[GL #6008]**

- Fix a bug in GeoIP2 string matching.

  When using GeoIP2 ACLs (see *acl*), *named* could incorrectly match a name using a sub-string instead of the full name match. This has been fixed. **[GL #6019]**

- Fix DNS-over-HTTPS (DoH) quota configuration issue.

  The *http-listener-clients* and *http-streams-per-connection* configuration options could be truncated to smaller values (or to `0`, which means unlimited) when very large configuration values in excess of `65535` were used. It is very unlikely that such large values were used in production, and the default values for the affected options are `300` and `100`, respectively. This has been fixed. **[GL #6021]**

- Truncated reply to a TSIG query no longer stalls the resolver.

  When an upstream server returned a truncated reply to a query that *named* had signed with TSIG, the resolver could keep waiting for a follow-up UDP packet that never arrived, stalling the query until it hit the *resolver-query-timeout* and the client received no answer. *named* now treats any reply it cannot authenticate as an immediate failure and returns SERVFAIL right away as a defense in depth. **[GL #6028]**

- Ignore updates removing DNSKEY RRset with class ANY.

  When a dynamic update is received that removes the DNSKEY (or CDNSKEY, or CDS) RRset, *named* now removes all records except the ones that are in use for signing the zone. **[GL #6045]**

- Ignore 0-byte reads in the TCP read callback.

  Callbacks for libuv stream reads do not signal zero-length reads as a failure signal but rather as EAGAIN/EWOULDBLOCK. This could trigger an assertion when a zero-length read was pushed onto a PROXYv2 endpoint that had not yet processed the headers, as it expected a non-NULL region of positive length. **[GL #6140]**

- Only print per-zone glue stats when *zone-statistics* is set to `full`.

  The code printing query statistics was ignoring the *zone-statistics* option. This has been fixed. **[GL #6164]**

- CDS/CDNSKEY records were not removed when re-configuring the server.

  When on an *rndc reconfig* the DNSSEC policy changes such that it changes the expected CDNSKEY and/or CDS records in the zone, the RRset should be updated accordingly. This did not happen when removing digests from the configuration, or setting `cdnskey no;`. This has been fixed. **[GL #6166]**

- Fix a crash when querying an empty non-terminal in a wildcard zone in RBTDB.

  A query for an empty non-terminal in a wildcard zone served from the RBT zone database could abort *named* with an assertion failure. It now returns the correct NODATA answer. **[GL #6170]**

- Stop reusing outgoing TCP connections the peer has already closed.

  Previously, *named* could hand a new query to an idle forwarder/upstream TCP or TLS connection that the peer had already closed, causing the query to fail (and CLOSE-WAIT sockets to pile up). Idle reused connections are now watched, so a close is noticed and the connection is dropped instead of reused. A new *tcp-reuse-timeout* option controls how long an idle outgoing connection is kept open for reuse (default 5 seconds). **[GL #6171]**

- Fix DNSSEC validation failures for names under an apex DNAME.

  DNSSEC validation could fail with SERVFAIL for names covered by a DNAME at the apex of a signed zone, unless the zone's keys were already validated in the cache. This regression was introduced by the recent fix for resolver stalls on CNAME responses to DS queries, and has now been addressed. **[GL #6176]**

## 14.9 Notes for BIND 9.20.25

**Note**

The BIND 9.20.25 release was withdrawn after the discovery of a regression in a security fix in it during pre-release testing.

## 14.10 Notes for BIND 9.20.24

### 14.10.1 Removed Features

- Remove ineffective TCP fallback after repeated UDP timeouts.

  When an authoritative server failed to respond to two consecutive UDP queries, *named* marked the next retry as TCP but still sent it over UDP, producing misleading dnstap records. The ineffective retry path has been removed; a corrected TCP fallback will be restored in future BIND 9 versions. **[GL #5529]**

### 14.10.2 Feature Changes

- Fall back to TCP on receipt of a UDP response with a mismatched query ID.

  BIND used to wait silently for the correct DNS message ID on a UDP fetch, even after receiving a response from the expected server with the wrong ID, leaving room for off-path spoofing attempts to keep guessing within that window. The resolver now retries the fetch over TCP on the first such response, and a new `MismatchTCP` statistics counter tracks how often the fallback fires. **[GL #5449]**

- Limit the number of glue records cached from a referral.

  When a delegation response contained many glue addresses per listed nameserver, all of them were cached without a per-nameserver bound, inflating resolver cache memory beyond what resolution could ever use. The cache now keeps at most 20 IPv4 and 20 IPv6 glue addresses per nameserver from a delegation. **[GL #5701]**

- Fix a resolver stall on a CNAME response to a DS query.

  A validating resolver could stall for about twelve seconds and then return SERVFAIL when an authoritative server answered a DS query with a CNAME. Such responses are now rejected promptly, so the query fails quickly instead of hanging. **[GL #5878]**

### 14.10.3 Bug Fixes

- The resolver now removes other RRsets at the same name when caching a CNAME.

  When an RRset is in stale cache and the authoritative server changes the record type to CNAME, the resolver fails to refresh the stale cache. This has been fixed. **[GL #5302]**

- Fix *nxdomain-redirect* combined with *dns64*.

  When a resolver was configured with both *nxdomain-redirect* and *dns64* in the same view, an AAAA query for a nonexistent name could abort *named*. The combination failed whenever the redirect zone held A records but no AAAA records. The server now serves the empty AAAA response from the redirect zone as-is, instead of attempting DNS64 synthesis on top of it. **[GL #5789]**

- Fix DNS64 owner case after DNAME restart.

  When BIND 9 was configured to use DNS64 and encountered a DNAME redirect, it could end up using freed memory for the DNS response owner name. This caused the response to contain corrupted data. This fix ensures the correct owner name is used when constructing the synthesized response after a DNAME redirect. **[GL #5934]**

- Clear REDIRECT flag when it isn't needed.

  When *nxdomain-redirect* is in use, and a recursive query is used to get the redirected answer, a flag is set to distinguish it from a normal recursive response. Previously, that flag was left set afterward, which could trigger an assertion if a normal recursive query was sent later on behalf of the same client: for example, because the *filter-aaaa* plugin was in use. This has been fixed. **[GL #5936]**

- Disable output escaping in `bind9.xsl`.

  The statistics charts were not displaying on some browsers. This has been fixed. **[GL #5990]**

- Fix crash on badly configured secondary signer.

  A badly configured secondary signer that was missing the `file` entry caused the server to crash, rather than to reject the configuration. This has been fixed. **[GL #5993]**

- Fix a possible crash on concurrent TKEY DELETE for the same key.

  On a server configured with *tkey-gssapi-keytab*, an authenticated peer could crash *named* by sending two TKEY DELETE requests for the same dynamic key in rapid succession. This has been fixed. **[GL #6001]**

- Reject RRSIG records covering meta-types.

  A recursive resolver could accept and cache an RRSIG record whose Type-Covered field named a meta-type (ANY, AXFR, IXFR, MAILA, MAILB), even though no real RRset of those types ever exists. Such records are now rejected by the DNS message parser. **[GL #6002]**

## 14.11 Notes for BIND 9.20.23

### 14.11.1 Security Fixes

- Limit resolver server list size. **(CVE-2026-3592)**

  When resolving a domain with many nameservers that shared overlapping IP addresses (e.g., 10 NS records all pointing at the same set of addresses), BIND could previously waste time querying duplicate addresses and build up excessively large server lists. Addresses in the resolver's server list are now deduplicated so that each unique IP is only queried once per resolution attempt, regardless of how many NS records point to it. The number of addresses stored per nameserver name is also now capped at six (combined A and AAAA), preventing memory and CPU overhead from domains with unusually large NS/glue sets.

  ISC would like to thank Shuhan Zhang from Tsinghua University for reporting this issue. **[GL #5641]**

- Fix GSS-API resource leak. **(CVE-2026-3039)**

  A memory leak was fixed where each GSS-API TKEY negotiation leaked a security context inside the GSS library. An unauthenticated attacker could exhaust server memory by sending repeated TKEY queries to a server with `tkey-gssapi-keytab` configured. The leaked memory was allocated by the GSS library, bypassing BIND's memory accounting.

  Multi-round GSS-API negotiation (GSS_S_CONTINUE_NEEDED) is now rejected, as BIND never supported it correctly and Kerberos/SPNEGO completes in a single round.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5752]**

- Disable recursion, UPDATE, and NOTIFY for non-IN views. **(CVE-2026-5946)**

  Recursion, dynamic updates (UPDATE), and zone change notifications (NOTIFY) are now disabled for views with a class other than IN (such as CHAOS or HESIOD); authoritative service for non-IN zones (e.g. version.bind in class CHAOS) continues to work as before. Servers configured with `recursion yes;` in a non-IN view log a warning at startup, and `named-checkconf` flags the same condition. UPDATE and NOTIFY messages that specify the meta-classes ANY or NONE in the question section are now rejected with FORMERR.

  This addresses a set of closely related security issues collectively identified as CVE-2026-5946. ISC would like to thank Mcsky23 for bringing these issues to our attention. **[GL #5784]**

- Avoid unbounded recursion loop. **(CVE-2026-5950)**

  A bug during bad server handling could cause the resolver to enter an infinite loop, continuously sending queries to an upstream server with no exit condition, until the resolver query timeout was hit. This has been fixed.

  ISC would like to thank Billy Baraja (BielraX) for bringing this issue to our attention. **[GL #5804]**

- Fix crash in resolver when SIG(0)-signed responses are received under load. **(CVE-2026-5947)**

  A resolver could crash when handling a SIG(0)-signed response if the matching client query was cancelled while signature verification was still in progress — for example, when the recursive-clients quota was exhausted. This has been fixed.

  ISC would like to thank Naoki Wakamatsu for bringing this vulnerability to our attention. **[GL #5819]**

- Fix use-after-free error in DNS-over-HTTPS when processing HTTP/2 SETTINGS frames. **(CVE-2026-3593)**

  Previously, a use-after-free vulnerability in the DNS-over-HTTPS implementation could cause `named` to crash when a client sent a flood of HTTP/2 SETTINGS frames while a DoH response was being written. This affected servers with DoH (DNS-over-HTTPS) enabled and has been fixed.

  ISC would like to thank Naresh Kandula Parmar (Nottiboy) for reporting this. **[GL #5755]**

- Fix outgoing zone transfers' quota issue.

  Unauthorized clients could consume the entire outgoing zone-transfer quota and block authorized zone transfer clients. This has been fixed. **[GL #3589]**

### 14.11.2 Feature Changes

- Fix CPU spikes and slow queries when cache approaches memory limit.

  Cache cleanup is now spread probabilistically to avoid CPU usage spikes and a drop in query throughput. **[GL #5891]**

### 14.11.3 Bug Fixes

- Use the zone file's basename as origin in DNSSEC tools.

  In *dnssec-signzone* and *dnssec-verify*, when the zone origin is not specified using the `-o` parameter, the default behavior is to try to sign using the zone's file name as the origin. So, for example, `dnssec-signzone -S example.com` will work, so long as the file name matches the zone name.

  This now also works if the zone is in a different directory. For example, `dnssec-signzone -S zones/example.com` will set the origin value to `example.com`. **[GL #5678]**

- Fix a possible race condition during zone transfers.

  The *named* process could terminate unexpectedly when processing an IXFR message during a zone transfer. This has been fixed. **[GL #5767]**

- Fix *named* crash when processing SIG records in dynamic updates.

  Previously, *named* could abort if a client sent a dynamic update containing a SIG record (the legacy signature type) to a zone configured with an update-policy. The function *dns_db_findrdataset* had an incorrect requirements prerequisite that prevented SIG records from being looked up, which was triggered as part of processing an UPDATE request and could be triggered remotely by any client permitted to send updates. This has been fixed by ensuring that SIG records are handled consistently with RRSIG records during update processing. **[GL #5818]**

- Fix zone verification of NSEC3 signed zones.

  Previously, when computing the compressed bitmap during verification of an NSEC3-signed zone, an undersized buffer was used that resulted in an out-of-bounds write if there were too many active windows in the bitmap. This impacted the mirror zones which are NSEC3-signed, *dnssec-signzone* and *dnssec-verify*. This has been fixed. **[GL #5834]**

- Prevent a crash when using both *dns64* and *filter-aaaa*.

  An assertion failure could be triggered if both *dns64* and the *filter-aaaa* plugin were in use simultaneously. This happened if the plugin triggered a second recursion process, which then attempted to store DNS64 state information in a pointer that had already been set by the original recursion process. This has been fixed. **[GL #5854]**

- Fixed an assertion failure when processing catalog zones.

  If a TXT record containing an invalid name TSIG key name was found when processing a catalog zone member's primaries definition, `dns_name_free` was incorrectly called, triggering an assertion. This has been fixed. **[GL #5858]**

- Prevent malicious DNSSEC zones from exhausting validator CPU.

  A DNSSEC-signed zone could publish a DNSKEY with an unusually large RSA public exponent and force any validator resolving names in that zone to spend disproportionate CPU verifying signatures. The validator now rejects such DNSKEYs, matching the limit already applied to keys read from files or HSMs. **[GL #5881]**

- Fix *rndc-confgen* aborting on HMAC-SHA-384/512 keys above 512 bits.

  *rndc-confgen* (with either `-A hmac-sha384` or `-A hmac-sha512`) previously documented a `-b` range of 1..1024, but any value above 512 aborted on hardened builds instead of producing a key. The full advertised range now works. **[GL #5903]**

- Prevent crafted queries from degrading RRL performance.

  With response rate limiting enabled, an attacker sending queries from many spoofed source addresses could steer entries into the same slot of the internal rate-limit table and slow down query processing on the affected server. The table now uses a per-process keyed hash so the placement of entries cannot be predicted or influenced from the network. **[GL #5906]**

- Prevent rare *named* crash when notifies are cancelled.

  Under heavy load, *named* could occasionally crash when a queued outbound notify or zone refresh was cancelled at the moment it was being sent — for example, while a zone was being reloaded or removed. The race that caused the crash is now prevented. **[GL #5915]**

- Stop *delv* from aborting on a malformed query name.

  *delv* previously aborted with SIGABRT instead of exiting cleanly when given a query name that failed wire-format conversion (e.g. a label longer than 63 octets). After this change *delv* prints the parse error and exits with a normal failure code. **[GL #5916]**

- Fix a crash when reconfiguring while an NTA is being rechecked.

  Previously, if *named* was reconfigured or shut down while a negative trust anchor was being rechecked against authoritative servers, the in-flight recheck could outlive the view that owned it and cause *named* to crash. This has been fixed. **[GL #5938]**

- Fix a bug in *allow-query*/*allow-transfer* catalog zone custom properties.

  The *named* process could terminate unexpectedly when processing a catalog zone with an invalid *allow-query* or *allow-transfer* custom property (i.e. having a non-APL type) coexisting with the valid property. This has been fixed. **[GL #5941]**

- Fix a memory leak issue in catalog zones.

  The *named* process could leak small amounts of memory when processing a catalog zone entry which had defined custom primary servers with TSIG keys, if both the regular `primaries` custom property syntax and the legacy alternative syntax (`masters`) were used at the same time. This has been fixed. **[GL #5943]**

- Fix suppressed missing-glue check in *named-checkzone*.

  *named-checkzone* and *named-checkconf -z* silently skipped the missing-glue check for any NS name that had already triggered an extra-AAAA-glue warning, so zones missing required A glue could pass validation and be deployed with broken delegations. **[GL !11899]**

- Implement seamless outgoing TCP connection reuse.

  The resolver can and will reuse outgoing TCP connections to the same host, as recommended by **RFC 7766**. This prevents a whole class of attacks that abuse the fact that establishing a TCP connection is expensive and it is fairly easy to deplete the outgoing TCP ports by putting them into `TIME_WAIT` state.

  The number of pipelined queries per connection is capped at 256 to limit the impact of a connection drop. **[GL !11845]**

- Reject record sets too large to serve in DNS.

  When BIND was asked to store a record set whose total size exceeded what fit in a DNS message, it would allocate memory and build the structure, then fail later at response time. Such oversized record sets are now rejected at the time of storage with an error, avoiding wasted work on data that can never be served. **[GL !11963]**

## 14.12 Notes for BIND 9.20.22

### 14.12.1 Security Fixes

- Fix crash when reconfiguring zone update policy during active updates.

  We fixed a crash that could occur when running *rndc reconfig* to change a zone's update policy (e.g., from *allow-update* to *update-policy*) while DNS UPDATE requests were being processed for that zone.

  ISC would like to thank Vitaly Simonovich for bringing this issue to our attention. **[GL #5817]**

### 14.12.2 Bug Fixes

- Fix intermittent `named` crashes during asynchronous zone operations.

  Asynchronous zone loading and dumping operations occasionally dispatched tasks to the wrong internal event loop. This threading violation triggered internal safety assertions that abruptly terminated `named`. Strict loop affinity is now enforced for these tasks, ensuring they execute on their designated threads and preventing the crashes. **[GL #4882]**

- Count temporal problems with DNSSEC validation as attempts.

  After the KeyTrap vulnerability **(CVE-2023-50387)**, any temporal DNSSEC errors were originally hard errors that caused validation failures, even if the records had another valid signature. This has been changed; RRSIGs outside of the inception and expiration time are not counted as hard errors. However, these errors were not even counted as validation attempts, so an excessive number of expired RRSIGs would cause some non-cryptographic extra work for the validator. This has been fixed and the temporal errors are now correctly counted as validation attempts. **[GL #5760]**

- Fix a possible deadlock in RPZ processing.

  The `named` process could hang when processing a maliciously crafted update for a response policy zone (RPZ). This has been fixed. **[GL #5775]**

- Fix a crash triggered by `rndc modzone` on a zone from a configuration file.

  Calling `rndc modzone` on a zone that was configured in the configuration file caused a crash. This has been fixed. **[GL #5800]**

- Fix the processing of empty catalog zone ACLs.

  The `named` process could terminate unexpectedly when processing a catalog zone ACL in an APL resource record that was completely empty. This has been fixed. **[GL #5801]**

- Fix a crash triggered by `rndc modzone` on zone that already existed in NZF file.

  Calling `rndc modzone` didn't work properly for a zone that was configured in the configuration file. It could crash if BIND 9 was built without LMDB or if there was already an NZF file for the zone. This has been fixed. **[GL #5826]**

- Fix potential resource leak during resolver error handling.

  Under specific error conditions during query processing, resources were not being properly released, which could eventually lead to unnecessary memory consumption for the server. A potential resource leak in the resolver has been fixed. **[GL !11658]**

## 14.13 Notes for BIND 9.20.21

### 14.13.1 Security Fixes

- Fix unbounded NSEC3 iterations when validating referrals to unsigned delegations. **(CVE-2026-1519)**

  DNSSEC-signed zones may contain high iteration-count NSEC3 records, which prove that certain delegations are insecure. Previously, a validating resolver encountering such a delegation processed these iterations up to the number given, which could be a maximum of 65,535. This has been addressed by introducing a processing limit, set at 50. Now, if such an NSEC3 record is encountered, the delegation will be treated as insecure.

  ISC would like to thank Samy Medjahed/Ap4sh for bringing this vulnerability to our attention. **[GL #5708]**

- Fix memory leaks in code preparing DNSSEC proofs of non-existence. **(CVE-2026-3104)**

An attacker controlling a DNSSEC-signed zone could trigger a memory leak in the logic preparing DNSSEC proofs of non-existence, by creating more than *max-records-per-type* RRSIGs for NSEC records. These memory leaks have been fixed.

ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5742]**

- Prevent a crash in code processing queries containing a TKEY record. **(CVE-2026-3119)**

  The *named* process could terminate unexpectedly when processing a correctly signed query containing a TKEY record. This has been fixed.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5748]**

- Fix a stack use-after-return flaw in SIG(0) handling code. **(CVE-2026-3591)**

  A stack use-after-return flaw in SIG(0) handling code could enable ACL bypass and/or assertion failures in certain circumstances. This flaw has been fixed.

  ISC would like to thank Mcsky23 for bringing this vulnerability to our attention. **[GL #5754]**

### 14.13.2 Bug Fixes

- Fix the handling of *key* statements defined inside views.

  A recent change introduced in BIND 9.20.17 hardened the *key* name check when used in *primaries*, to immediately reject the configuration if the key was not defined (rather than only checking whether the key name was correctly formed). However, that change introduced a regression that prevented the use of a *key* defined in a view. This has now been fixed. **[GL #5761]**

## 14.14 Notes for BIND 9.20.20

### 14.14.1 Security Fixes

- Fix a use-after-free error in `dns_client_resolve()` triggered by a DNAME response.

  This issue only affected the *delv* tool and it has now been fixed.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5728]**

### 14.14.2 Feature Changes

- Record query time for all dnstap responses.

  Not all DNS responses had the query time set in their corresponding dnstap messages. This has been fixed. **[GL #3695]**

- Optimize TCP source port selection on Linux.

  Enable the `IP_LOCAL_PORT_RANGE` socket option on the outgoing TCP sockets to allow faster selection of the source <address,port> tuple for different destination <address,port> tuples, when nearing over 70-80% of the source port utilization. **[GL !11569]**

### 14.14.3 Bug Fixes

- Fix an assertion failure triggered by non-minimal IXFRs.

  Processing an IXFR that included an RRset whose contents were not changed by the transfer triggered an assertion failure. This has been fixed. **[GL #5759]**

- Fix a crash when retrying a NOTIFY over TCP.

  Furthermore, do not attempt to retry over TCP at all if the source address is not available. **[GL #5457]**

- Fetch loop detection improvements.

  Fix a case where an in-domain nameserver with expired glue would fail to resolve. **[GL #5588]**

- Randomize nameserver selection.

  Since BIND 9.20.17, when selecting nameserver addresses to be looked up, `named` selected them in DNSSEC order from the start of the NS RRset. This could lead to a resolution failure despite there being an address that could be resolved using the other nameserver names. `named` now randomizes the order in which nameserver addresses are looked up. **[GL #5695] [GL #5745]**

- Fix dnstap logging of forwarded queries. **[GL #5724]**

- A stale answer could have been served in case of multiple upstream failures when following CNAME chains. This has been fixed. **[GL #5751]**

- Fail DNSKEY validation when supported but invalid DS is found.

  A regression was introduced in BIND 9.20.6 when adding the EDE code for unsupported DNSKEY and DS algorithms. When the parent had both supported and unsupported algorithms in the DS record, the validator would treat the supported DS algorithm as insecure instead of bogus when validating DNSKEY records. This has no security impact, as the rest of the child zone correctly ends with bogus status, but it is incorrect and thus the regression has been fixed. **[GL #5757]**

- Importing an invalid SKR file might corrupt stack memory.

  If an administrator imported an invalid SKR file, the local stack in the import function might overflow. This could lead to a memory corruption on the stack and ultimately a server crash. This has been fixed. **[GL #5758]**

- Return FORMERR for queries with the EDNS Client Subnet FAMILY field set to 0.

  **RFC 7871** only defines families 1 (IPv4) and 2 (IPv6), and requires FORMERR to be returned for all unknown families. Queries with the EDNS Client Subnet FAMILY field set to 0 now elicit responses with RCODE=FORMERR. **[GL !11565]**

## 14.15 Notes for BIND 9.20.19

### 14.15.1 Feature Changes

- Update requirements for system test suite.

  Python 3.10 or newer is now required for running the system test suite. The required Python packages and their version requirements are now tracked in the file *bin/tests/system/requirements.txt*. **[GL #5690] [GL #5614]**

### 14.15.2 Bug Fixes

- Fix inbound IXFR performance regression.

  Very large inbound IXFR transfers were much slower compared to BIND 9.18. The performance was improved by adding specialized logic to handle IXFR transfers. **[GL #5442]**

- Make catalog zone names and member zones' entry names case-insensitive. **[GL #5693]**

- Fix implementation of BRID and HHIT record types. **[GL #5710]**

- Fix implementation of DSYNC record type. **[GL #5711]**

- Fix response policy and catalog zones to work with *$INCLUDE* directive.

  Reloading a RPZ or a catalog zone could have failed when *$INCLUDE* was in use. **[GL #5714]**

## 14.16 Notes for BIND 9.20.18

### 14.16.1 Security Fixes

- Fix incorrect length checks for BRID and HHIT records. **(CVE-2025-13878)**

  Malformed BRID and HHIT records could trigger an assertion failure. This has been fixed.

  ISC would like to thank Vlatko Kosturjak from Marlink Cyber for bringing this vulnerability to our attention. **[GL #5616]**

### 14.16.2 Feature Changes

- Add more information to the `rndc recursing` output about fetches.

  This adds more information about active fetches, for debugging and diagnostic purposes. **[GL !11305]**

### 14.16.3 Bug Fixes

- Make DNSSEC key rollovers more robust.

  A manual rollover when the zone was in an invalid DNSSEC state caused predecessor keys to be removed too quickly. Additional safeguards to prevent this have been added: DNSSEC records are not removed from the zone until the underlying state machine has moved back into a valid DNSSEC state. **[GL #5458]**

- Fix a catalog zone issue, where member zones could fail to load.

  A catalog zone member zone could fail to load in some rare cases, when the internally generated zone configuration string exceeded 512 bytes. That condition by itself was not enough for the issue to arise, but it was necessary. This could happen if, for example, the catalog zone's default primary servers list contained a large number of items. This has been fixed. **[GL #5658]**

- Allow glue in delegations with QTYPE=ANY.

  When a query for type ANY triggered a delegation response, all additional data was omitted from the response, including mandatory glue. This has been fixed. **[GL #5659]**

- Fix slow speed when signing a large delegation zone with NSEC3 opt-out.

  BIND 9.20+ took much longer signing a large delegation zone with NSEC3 opt-out compared to version 9.18. This has been fixed. **[GL #5672]**

- Reconfiguring an NSEC3 opt-out zone to NSEC caused the zone to be invalid.

  A zone that was signed with NSEC3, had opt-out enabled, and was then reconfigured to use NSEC, was published with missing NSEC records. This has been fixed. **[GL #5679]**

- Fix a possible catalog zone issue during reconfiguration.

  The `named` process could terminate unexpectedly during reconfiguration when a catalog zone update was taking place at the same time. This has been fixed. **[GL !11366]**

- Fix the charts in the statistics channel.

  The charts in the statistics channel could sometimes fail to render in the browser and were completely disabled for Mozilla-based browsers, for historical reasons. This has been fixed. **[GL !11018]**

## 14.17 Notes for BIND 9.20.17

### 14.17.1 Feature Changes

- Reduce the number of outgoing queries.

  Reduce the number of outgoing queries when resolving the nameservers for delegation points. This helps a DNS resolver with a cold cache resolve client queries with complex delegation chains and redirections. **[GL !11148]**

- Provide more information when memory allocation fails.

  BIND now provides more information about the failure when memory allocation fails. **[GL !11272]**

### 14.17.2 Bug Fixes

- Adding NSEC3 opt-out records could leave invalid records in chain.

  When creating an NSEC3 opt-out chain, a node in the chain could be removed too soon. The previous NSEC3 would therefore not be found, resulting in invalid NSEC3 records being left in the zone. This has been fixed. **[GL #5671]**

- Fix spurious timeouts while resolving names.

  Sometimes, loops in the resolving process (e.g., to resolve or validate `ns1.example.com`, we need to resolve `ns1.example.com`) were not properly detected, leading to a spurious 10-second delay. This has been fixed, and such loops are properly detected. **[GL #3033] [GL #5578]**

- Fix bug where zone switches from NSEC3 to NSEC after retransfer.

  When a zone was re-transferred but the zone journal on an inline-signing secondary was out of sync, the zone could fall back to using NSEC records instead of NSEC3. This has been fixed. **[GL #5527]**

- `AMTRELAY` type 0 presentation format handling was wrong.

  **RFC 8777** specifies a placeholder value of `.` for the gateway field when the gateway type is 0 (no gateway). This was not being checked for, nor was it emitted when displaying the record. This has been corrected.

  Instances of this record will need the placeholder period added to them when upgrading. **[GL #5639]**

- Fix parsing bug in *`remote-servers`* with key or TLS.

  The *`remote-servers`* clause enables the following pattern using a named `server-list`:

```
remote-servers a { 1.2.3.4; ... };
remote-servers b { a key foo; };
```

  However, such a configuration was wrongly rejected, with an `unexpected token 'foo'` error. This configuration is now accepted. **[GL #5646]**

- Fix DoT reconfigure/reload bug in the resolver.

  If client-side TLS transport was in use (for example, when forwarding queries to a DoT server), *`named`* could terminate unexpectedly when reconfiguring or reloading. This has been fixed. **[GL #5653]**

## 14.18 Notes for BIND 9.20.16

### 14.18.1 Bug Fixes

- Skip unsupported algorithms when looking for a signing key.

  A mix of supported and unsupported DNSSEC algorithms in the same zone could cause validation failures. Unsupported algorithms are now ignored when looking for signing keys. **[GL #5622]**

- Fix *dnssec-keygen* key collision checking for KEY RRtype keys.

  The *dnssec-keygen* utility program failed to detect possible KEY ID collisions with existing keys generated using the non-default `-T KEY` option (e.g., for `SIG(0)`). This has been fixed. **[GL #5506]**

- *dnssec-verify* now uses exit code 1 when failing due to illegal options.

  Previously, *dnssec-verify* exited with code 0 if the options could not be parsed. This has been fixed. **[GL #5574]**

- Prevent assertion failures of *dig* when a server is specified before the `-b` option.

  Previously, *dig* could exit with an assertion failure when a server was specified before the *dig -b* option. This has been fixed. **[GL #5609]**

- Skip buffer allocations if not logging.

  Previously, we allocated a 2KB buffer for IXFR change logging, regardless of the log level.

  This results in a 28% speedup in some scenarios.

## 14.19 Notes for BIND 9.20.15

### 14.19.1 Security Fixes

- DNSSEC validation fails if matching but invalid DNSKEY is found. **(CVE-2025-8677)**

  Previously, if a matching but cryptographically invalid key was encountered during DNSSEC validation, the key was skipped and not counted towards validation failures. *named* now treats such DNSSEC keys as hard failures and the DNSSEC validation fails immediately, instead of continuing with the next DNSKEYs in the RRset.

  ISC would like to thank Zuyao Xu and Xiang Li from the All-in-One Security and Privacy Laboratory at Nankai University for bringing this vulnerability to our attention. **[GL #5343]**

- Address various spoofing attacks. **(CVE-2025-40778)**

  Previously, several issues could be exploited to poison a DNS cache with spoofed records for zones which were not DNSSEC-signed or if the resolver was configured to not do DNSSEC validation. These issues were assigned CVE-2025-40778 and have now been fixed.

  As an additional layer of protection, *named* no longer accepts DNAME records or extraneous NS records in the AUTHORITY section unless these are received via spoofing-resistant transport (TCP, UDP with DNS cookies, TSIG, or SIG(0)).

  ISC would like to thank Yuxiao Wu, Yunyi Zhang, Baojun Liu, and Haixin Duan from Tsinghua University for bringing this vulnerability to our attention. **[GL #5414]**

- Cache-poisoning due to weak pseudo-random number generator. **(CVE-2025-40780)**

  It was discovered during research for an upcoming academic paper that a xoshiro128** internal state can be recovered by an external 3rd party, allowing the prediction of UDP ports and DNS IDs in outgoing queries. This could lead to an attacker spoofing the DNS answers with great efficiency and poisoning the DNS cache.

  The internal random generator has been changed to a cryptographically secure pseudo-random generator.

  ISC would like to thank Prof. Amit Klein and Omer Ben Simhon from Hebrew University of Jerusalem for bringing this vulnerability to our attention. **[GL #5484]**

### 14.19.2 New Features

- Add *dnssec-policy* keys configuration check to *named-checkconf*.

  A new option *-k* was added to *named-checkconf* that allows checking the *dnssec-policy keys* configuration against the configured key stores. If the found key files are not in sync with the given *dnssec-policy*, the check will fail.

  This is useful to run before migrating to *dnssec-policy*. **[GL #5486]**

### 14.19.3 Bug Fixes

- Missing DNSSEC information when CD bit is set in query.

  The RRSIGs for glue records were not being cached correctly for CD=1 queries. This has been fixed. **[GL #5502]**

- *rndc sign* during ZSK rollover will now replace signatures.

  When performing a ZSK rollover, if the new DNSKEY is omnipresent, the *rndc sign* command now signs the zone completely with the successor key, replacing all zone signatures from the predecessor key with new ones. **[GL #5483]**

- Use signer name when disabling DNSSEC algorithms.

  *disable-algorithms* could cause DNSSEC validation failures when the parent zone was signed with the algorithms that were being disabled for the child zone. This has been fixed; *disable-algorithms* now works on a whole-of-zone basis.

  If the zone's name is at or below the *disable-algorithms* name the algorithm is disabled for that zone, using deepest match when there are multiple *disable-algorithms* clauses. **[GL #5165]**

- Preserve cache when reload fails and reload the server again.

  This fixes an issue where failing to reconfigure/reload the server would fail to preserve the views' caches for subsequent server reconfigurations/reloads. **[GL #5523]**

## 14.20 Notes for BIND 9.20.14

> **Note**
>
> The BIND 9.20.14 release was withdrawn after the discovery of a regression in a security fix in it during pre-release testing.

## 14.21 Notes for BIND 9.20.13

### 14.21.1 New Features

- Add a new option `manual-mode` to *dnssec-policy*.

  When enabled, *named* will not modify DNSSEC keys or key states automatically. The proposed change will be logged and only after manual confirmation with `rndc dnssec -step` will the modification be made. **[GL #4606]**

- Add a new option `servfail-until-ready` to *response-policy* zones.

  By default, when *named* is started, it starts answering queries before all response policy zones are completely loaded and processed. This new option instructs *named* to respond with SERVFAIL until all the response policy zones are processed and ready. Note that if one or more response policy zones fail to load, *named* starts responding to queries according to those zones that did load.

Note, that enabling this option has no effect when a DNS Response Policy Service (DNSRPS) interface is used. **[GL #5222]**

- Support for parsing HHIT and BRID records has been added.

  **[GL #5444]**

### 14.21.2 Removed Features

- Deprecate the *tkey-gssapi-credential* statement.

  The *tkey-gssapi-keytab* statement allows GSS-TSIG to be set up in a simpler and more reliable way than using the *tkey-gssapi-credential* statement and setting environment variables (e.g. KRB5_KTNAME). Therefore, the *tkey-gssapi-credential* statement has been deprecated; *tkey-gssapi-keytab* should be used instead.

  For configurations currently using a combination of both *tkey-gssapi-keytab* *and* *tkey-gssapi-credential*, the latter should be dropped and the keytab pointed to by *tkey-gssapi-keytab* should now only contain the credential previously specified by *tkey-gssapi-credential*. **[GL #4204]**

- Obsolete the "tkey-domain" statement.

  Mark the tkey-domain statement as obsolete because it has not had any effect on server behavior since support for TKEY Mode 2 (Diffie-Hellman) was removed (in BIND 9.20.0). **[GL #4204]**

### 14.21.3 Bug Fixes

- Prevent spurious SERVFAILs for certain 0-TTL resource records.

  Under certain circumstances, BIND 9 can return SERVFAIL when updating existing entries in the cache with new NS, A, AAAA, or DS records that have a TTL of zero. **[GL #5294]**

- Fix unexpected termination if *catalog-zones* had undefined default-primaries.

  The issue manifested only if the server was reloaded or reconfigured twice. **[GL #5494]**

## 14.22 Notes for BIND 9.20.12

### 14.22.1 New Features

- Support for parsing DSYNC records has been added.

  These records are used for discovering the receiver endpoint for DNS notification messages. For more information, see draft-ietf-dnsop-generalized-notify-09. **[GL #5440]**

### 14.22.2 Feature Changes

- Add deprecation warnings for RSASHA1, RSASHA1-NSEC3SHA1, and DS digest type 1.

  RSASHA1 and RSASHA1-NSEC-SHA1 DNSKEY algorithms have been deprecated by the IETF and should no longer be used for DNSSEC. DS digest type 1 (SHA1) has also been deprecated in BIND 9. Validators are now expected to treat these algorithms and digest as unknown, resulting in some zones being treated as insecure when they were previously treated as secure. Warnings have been added to *named* and tools when these algorithms and this digest are being used for signing.

  Zones signed with RSASHA1 or RSASHA1-NSEC-SHA1 should be migrated to a different DNSKEY algorithm.

  Zones with DS or CDS records with digest type 1 (SHA1) should be updated to use a different digest type (e.g. SHA256) and the digest type 1 records should be removed. **[GL #5358]**

### 14.22.3 Bug Fixes

- Stale RRsets in a CNAME chain were not always refreshed.

  Previously, with serve-stale enabled and a CNAME chain that contained a stale RRset, the refresh query didn't always properly refresh the stale RRsets. This has been fixed. **[GL #5243]**

- Add RPZ extended DNS error for zones with a CNAME override policy configured.

  Previously, when the zone was configured with a CNAME override policy, or the response policy zone contained a wildcard CNAME, the extended DNS error code was not added. This has been fixed. **[GL #5342]**

- Fix *dig* issues.

  When used with the `+keepopen` option, *dig* could terminate unexpectedly in rare situations. Additionally, *dig* could hang and fail to shutdown properly when interrupted during a query. These have been fixed. **[GL #5381]**

- Log dropped or slipped responses in the `query-errors` category.

  Responses which were dropped or slipped because of Response Rate Limiting (RRL) were logged in the `rate-limit` category instead of the `query-errors` category, as documented in the ARM. This has been fixed. **[GL #5388]**

- *synth-from-dnssec* was not working in some scenarios.

  Aggressive use of DNSSEC-Validated cache with NSEC was not working in scenarios when no parent NSEC was in cache. This has been fixed. **[GL #5422]**

- Clean enough memory when adding new ADB names/entries under memory pressure.

  The ADB memory cleaning is opportunistic even when BIND is under memory pressure (in the overmem condition). *named* now ensures that the assigned memory limit is not exceeded by releasing twice the amount of memory allocated for each new ADB name/entry when under memory pressure. **[GL !10637]**

- Prevent spurious validation failures.

  Under rare circumstances, validation could fail if multiple clients simultaneously iterated the same set of DNSSEC signatures. This has been fixed. **[GL #3014]**

## 14.23 Notes for BIND 9.20.11

### 14.23.1 Security Fixes

- Fix a possible assertion failure when *stale-answer-client-timeout* is set to `0`.

  In specific circumstances the *named* resolver process could exit with an assertion failure when stale answers were enabled and the *stale-answer-client-timeout* configuration option was set to `0`. This has been fixed. **(CVE-2025-40777) [GL #5372]**

### 14.23.2 New Features

- Add support for the CO flag to *dig*.

  Add support for Compact Denial of Existence to *dig*. This includes showing the CO (Compact Answers OK) flag when displaying messages and adding an option to set the CO flag when making queries (*dig +coflag*). **[GL #5319]**

### 14.23.3 Bug Fixes

- Correct the default *interface-interval* from 60s to 60m.

  When the *interface-interval* parser was changed from a `uint32` parser to a duration parser, the default value stayed at plain number `60` which now means 60 seconds instead of 60 minutes. The documentation also incorrectly states that the value is in minutes. That has been fixed. **[GL #5246]**

- Fix a *purge-keys* bug when using multiple views of a zone.

  Previously, when a DNSSEC key was purged by one zone view, other zone views would return an error about missing key files. This has been fixed. **[GL #5315]**

- Use IPv6 queries in *delv +ns*.

  *delv +ns* invokes the same code to perform name resolution as *named*, but it neglected to set up an IPv6 dispatch object first. Consequently, it was behaving more like *named -4*. It now sets up dispatch objects for both address families, and performs resolver queries to both IPv4 and IPv6 addresses, except when one of the address families has been suppressed by using *delv -4* or *delv -6*. **[GL #5352]**

## 14.24 Notes for BIND 9.20.10

### 14.24.1 New Features

- Implement a new *notify-defer* configuration option.

  This new option sets a delay (in seconds) to wait before sending a set of `NOTIFY` messages for a zone. Whenever a `NOTIFY` message is ready to be sent, sending is deferred for this duration. This option should not be confused with the *notify-delay* option. The default is 0 seconds. **[GL #5259]**

### 14.24.2 Removed Features

- Implement the systemd notification protocol manually to remove dependency on libsystemd.

### 14.24.3 Bug Fixes

- Fix zone deletion issue.

  A secondary zone could initiate a new zone transfer from the primary server after it had been already deleted from the secondary server, and before the internal garbage collection was activated to clean it up completely. This has been fixed. **[GL #5291]**

- Fix a zone refresh bug.

  A secondary zone could fail to further refresh with new versions of the zone from a primary server if *named* was reconfigured during the SOA request step of an ongoing zone transfer. This has been fixed. **[GL #5307]**

## 14.25 Notes for BIND 9.20.9

### 14.25.1 Security Fixes

- Prevent an assertion failure when processing TSIG algorithm.

  DNS messages that included a Transaction Signature (TSIG) containing an invalid value in the algorithm field caused *named* to crash with an assertion failure. This has been fixed. **(CVE-2025-40775) [GL #5300]**

### 14.25.2 Feature Changes

- Return DNS COOKIE and NSID with BADVERS.

  This change allows the client to identify a server that returns a BADVERS response and to provide a DNS SERVER COOKIE to be included in the resent request. **[GL #5235]**

- Disable separate memory context for libxml2 memory allocations on macOS.

  As of macOS Sequoia 15.4, custom memory allocation functions are no longer supported by the system-wide version of libxml2. This prevents tracking libxml2 memory allocations in a separate *named* memory context, so the latter has been disabled on macOS; the system allocator is now directly used for libxml2 memory allocations on that operating system. **[GL #5268]**

- Use Jinja2 templates in system tests.

  *python-jinja2* is now required to run system tests. **[GL #4938]**

### 14.25.3 Bug Fixes

- Revert NSEC3 closest encloser lookup improvements.

  The performance improvements for NSEC3 closest encloser lookups that were restored in BIND 9.20.8 turned out to cause incorrect NSEC3 records to be returned in nonexistence proofs and were therefore reverted again. **[GL #5292]**

- Fix EDNS YAML output in *dig*.

  *dig* was producing invalid YAML when displaying some EDNS options. This has been corrected.

  Several other improvements have been made to the display of EDNS option data:

  - The correct name is now used for the UPDATE-LEASE option, which was previously displayed as `UL`, and it is split into separate `LEASE` and `LEASE-KEY` components in YAML mode.
  - Human-readable durations are now displayed as comments in YAML mode so as not to interfere with machine parsing.
  - KEY-TAG options are now displayed as an array of integers in YAML mode.
  - EDNS COOKIE options are displayed as separate `CLIENT` and `SERVER` components, and cookie STATUS is a retrievable variable in YAML mode.

  **[GL #5014]**

- Fix RDATA checks for PRIVATEOID keys.

  In PRIVATEOID keys, the key data begins with a length byte followed by an ASN.1 object identifier that indicates the cryptographic algorithm to use. Previously, the length byte was not accounted for when checking the contents of keys and signatures, which could have led to interoperability problems with any zones signed using PRIVATEOID. This has been fixed. **[GL #5270]**

- Fix a serve-stale issue with a delegated zone.

  Even with *stale-answer-client-timeout* set to `0`, stale responses were not returned immediately for names in domains delegated from authoritative zones configured on the resolver. This has been fixed. **[GL #5275]**

## 14.26 Notes for BIND 9.20.8

### 14.26.1 New Features

- Add support for EDE 20 (Not Authoritative)

Support was added for EDE codes 20 (Not Authoritative) when the client requests recursion (RD) but the server has recursion disabled.

**RFC 8914** indicates that EDE 20 should also be returned if the client does not have the RD bit set (and recursion is needed), but it does not apply for BIND; BIND would try to resolve from the "deepest" referral in the AUTHORITY section. For example, if the client asks for `www.isc.org/A` but the server only knows the root domain, it will return NOERROR but no answer for `www.isc.org/A`, just the list of other servers to ask. **[GL #1836]**

- Add support for EDE 7 and EDE 8.

  Support was added for EDE codes 7 (Signature Expired) and 8 (Signature Not Yet Valid), which might occur during DNSSEC validation. **[GL #2715]**

- *dig* can now display the received BADVERS message during negotiation.

  *dig +showbadvers* now displays the received BADVERS message and continues the EDNS version negotiation. Previously, to see the BADVERS message *dig +noednsnegotiation* had to be specified, which terminated the EDNS negotiation. Additionally, the specified EDNS value (`+edns=value`) is now used when making all the initial queries with *dig +trace*, i.e. EDNS version negotiation is performed with each server when performing the trace. **[GL #5234]**

- Add an *rndc* command to reset some statistics counters.

  The new *rndc reset-stats* command allows some statistics counters to be reset during runtime. At the moment only two "high-water" counters are supported, so the ability to reset them after the initial peaks during the server's "warm-up" phase may be useful for some operators. **[GL #5251]**

### 14.26.2 Bug Fixes

- Restore NSEC3 closest-encloser lookup improvements.

  A performance improvement for finding the closest encloser when generating authoritative responses from NSEC3 zones was previously reverted after a bug was found that could trigger an assertion failure. (**[GL #4460]**, **[GL #4950]**, and **[GL #5108]**) The bug has now been fixed, and the performance improvement has been restored. **[GL #5204]**

- Stop caching lack of EDNS support.

  *named* could falsely learn that a server did not support EDNS when a spoofed response was received; that subsequently prevented DNSSEC lookups from being made. This has been fixed. **[GL #3949]** **[GL #5066]**

- Fix resolver statistics counters for timed-out responses.

  When query responses timed out, the resolver could incorrectly increase the regular response counters, even if no response was received. This has been fixed. **[GL #5193]**

- Nested DNS validation could cause an assertion failure.

  When multiple nested DNS validations were destroyed out of order, the EDE context could be freed before all EDE codes were copied, which could cause an assertion failure. This has been fixed. **[GL #5213]**

- Wait for memory reclamation to finish in *named-checkconf*.

  Previously, when *named-checkzone* loaded the zone to the QP database, the delayed memory reclamation could cause an assertion check on exit. This has been fixed. **[GL #5220]**

- Ensure *max-clients-per-query* is at least *clients-per-query*.

  If the *max-clients-per-query* option is set to a lower value than *clients-per-query*, the value is adjusted to match *clients-per-query*. **[GL #5224]**

- Fix write after free in validator code.

Raw integer pointers were being used for the validator's nvalidations and nfails values, but the memory holding them could be freed while they were still being used. Reference counted counters are now used instead. **[GL #5239]**

- Don't enforce NOAUTH/NOCONF flags in DNSKEYs.

  All DNSKEY keys are able to authenticate. The `DNS_KEYTYPE_NOAUTH` (and `DNS_KEYTYPE_NOCONF`) flags were defined for the KEY rdata type, and are not applicable to DNSKEY. Previously, however, because the DNSKEY implementation was built on top of KEY, the `_NOAUTH` flag prevented authentication in DNSKEYs as well. This has been corrected. **[GL #5240]**

- Fix several small DNSSEC timing issues.

  The following small issues related to *dnssec-policy* have been fixed:

  - In some cases the key manager inside BIND 9 would run every hour, while in other cases it would run less often.
  - While `CDS` and `CDNSKEY` records will be removed correctly from the zone when the corresponding `DS` record needs to be updated, the expected timing metadata for when this should happen was never set.
  - There were a couple of cases where the safety intervals were added inappropriately, delaying key rollovers longer than necessary.
  - Identical keys in a *dnssec-policy* may have been retired inappropriately. Note that having *keys* with identical properties is discouraged in all cases.

  **[GL #5242]**

- Fix inconsistency in CNAME/DNAME handling during resolution.

  Previously, in some cases, the resolver could return rdatasets of type CNAME or DNAME without the result code being set to `DNS_R_CNAME` or `DNS_R_DNAME`. This could trigger an assertion failure. This has been fixed. **[GL #5201]**

## 14.27 Notes for BIND 9.20.7

### 14.27.1 New Features

- Implement the *min-transfer-rate-in* configuration option.

  A new option *min-transfer-rate-in* has been added to the view and zone configurations. It can abort incoming zone transfers that run very slowly due to network-related issues, for example. The default value is 10240 bytes in five minutes. **[GL #3914]**

- Add HTTPS record query to *host* command line tool.

  The *host* command was extended to also query for the HTTPS RR type by default.

- Implement *sig0key-checks-limit* and *sig0message-checks-limit*.

  Previously, a hard-coded limitation of a maximum of two key or message verification checks was introduced when checking a message's `SIG(0)` signature, to protect against possible DoS attacks. Two as a maximum was chosen so that more than a single key should only be required during key rotations, and in that case two keys are enough. It later became apparent that there are other use cases where even more keys are required; see the related GitLab issue for examples.

  This change introduces two new configuration options for the views: *sig0key-checks-limit* and *sig0message-checks-limit*. They define how many keys can be checked to find a matching key, and how many message verifications are allowed to take place once a matching key has been found. The former provides slightly less "expensive" key parsing operations and defaults to 16. The latter protects against expensive cryptographic operations when there are keys with colliding tags and algorithm numbers; the default is 2. **[GL #5050]**

### 14.27.2 Bug Fixes

- Fix *dual-stack-servers* configuration option.

  The *dual-stack-servers* configuration option was not working as expected; the specified servers were not being used when they should have been, leading to resolution failures. This has been fixed. **[GL #5019]**

- Fix a data race causing a permanent active client increase.

  Previously, a data race could cause a newly created fetch context for a new client to be used before it had been fully initialized, which would cause the query to become stuck; queries for the same data would be either paused indefinitely or dropped because of the *clients-per-query* limit. This has been fixed. **[GL #5053]**

- Fix deferred validation of unsigned DS and DNSKEY records.

  When processing a query with the "checking disabled" bit set (CD=1), *named* stores the invalidated result in the cache, marked "pending". When the same query is sent with CD=0, the cached data is validated and either accepted as an answer, or ejected from the cache as invalid. This deferred validation was not attempted for DS and DNSKEY records if they had no cached signatures, causing spurious validation failures. The deferred validation is now completed in this scenario.

  Also, if deferred validation fails, the data is now re-queried to find out whether the zone has been corrected since the invalid data was cached. **[GL #5066]**

- Fix RPZ race condition during a reconfiguration.

  With RPZ in use, *named* could terminate unexpectedly because of a race condition when a reconfiguration command was received using *rndc*. This has been fixed. **[GL #5146]**

- "CNAME and other data check" not applied to all types.

  An incorrect optimization caused "CNAME and other data" errors not to be detected if certain types were at the same node as a CNAME. This has been fixed. **[GL #5150]**

- Relax private DNSKEY and RRSIG constraints.

  DNSKEY, KEY, RRSIG, and SIG constraints have been relaxed to allow empty key and signature material after the algorithm identifier for PRIVATEOID and PRIVATEDNS. It is arguable whether this falls within the expected use of these types, as no key material is shared and the signatures are ineffective, but these are private algorithms and they can be totally insecure. **[GL #5167]**

- Remove NSEC/DS/NSEC3 RRSIG check from `dns_message_parse()`.

  Previously, when parsing responses, *named* incorrectly rejected responses without matching RRSIG records for NSEC/DS/NSEC3 records in the authority section. This rejection, if appropriate, should have been left for the validator to determine and has been fixed. **[GL #5185]**

- Fix TTL issue with ANY queries processed through RPZ "passthru".

  Answers to an "ANY" query which were processed by the RPZ "passthru" policy had the response-policy's `max-policy-ttl` value unexpectedly applied. This has been fixed. **[GL #5187]**

- *dnssec-signzone* needs to check for a NULL key when setting offline.

  *dnssec-signzone* could dereference a NULL key pointer when resigning a zone. This has been fixed. **[GL #5192]**

- Fix a bug in the statistics channel when querying zone transfer information.

  When querying zone transfer information from the statistics channel, there was a rare possibility that *named* could terminate unexpectedly if a zone transfer was in a state when transferring from all the available primary servers had failed earlier. This has been fixed. **[GL #5198]**

- Fix assertion failure when dumping recursing clients.

  Previously, if a new counter was added to the hash table while dumping recursing clients via the *rndc recursing* command, and *fetches-per-zone* was enabled, an assertion failure could occur. This has been fixed. **[GL #5200]**

- Dump the active resolver fetches from `dns_resolver_dumpfetches()`

  Previously, active resolver fetches were only dumped when the *fetches-per-zone* configuration option was enabled. Now, active resolver fetches are dumped along with the number of *clients-per-query* counters per resolver fetch.

## 14.28 Notes for BIND 9.20.6

### 14.28.1 New Features

- Adds support for EDE code 1 and 2.

  Support was added for EDE codes 1 and 2, which might occur during DNSSEC validation in the case of an unsupported RRSIG algorithm or DNSKEY digest. **[GL #2715]**

- Add an *rndc* command to toggle jemalloc profiling.

  The new command is *rndc memprof*; the memory profiling status is also reported inside *rndc status*. The status shows whether *named* can toggle memory profiling, and whether the server is built with jemalloc. **[GL #4759]**

- Add support for multiple extended DNS errors.

  The Extended DNS Error (EDE) mechanism may raise errors during a DNS resolution. *named* is now able to add up to three EDE codes in a DNS response. If there are duplicate error codes, only the first one is part of the DNS response. **[GL #5085]**

- Print the expiration time of stale records.

  BIND now prints the expiration time of any stale RRsets in the cache dump.

### 14.28.2 Bug Fixes

- Recently expired records could be returned with a timestamp in future.

  Under rare circumstances, an RRSet that expired at the time of the query could be returned with a TTL in the future. This has been fixed.

  As a side effect, the expiration time of expired RRSets is no longer returned in a cache dump. **[GL #5094]**

- YAML string not terminated in negative response in delv.

  **[GL #5098]**

- Fix a bug in *dnssec-signzone* related to keys being offline.

  When *dnssec-signzone* was called on an already-signed zone and the private key file was unavailable, a signature that needed to be refreshed was dropped without being able to generate a replacement. This has been fixed. **[GL #5126]**

- Apply the memory limit only to ADB database items.

  Under heavy load, a resolver could exhaust the memory available for storing the information in the Address Database (ADB), effectively discarding previously stored information in the ADB. The memory used to retrieve and provide information from the ADB is no longer subject to the same memory limits that are applied to the Address Database. **[GL #5127]**

- Avoid unnecessary locking in the zone/cache database.

  Lock contention among many worker threads referring to the same database node at the same time is now prevented. This improves zone and cache database performance for any heavily contended database nodes. **[GL #5130]**

- Fix reporting of Extended DNS Error 22 (No Reachable Authority).

  This error code was previously not reported in some applicable situations. This has been fixed. **[GL #5137]**

## 14.29 Notes for BIND 9.20.5

### 14.29.1 Security Fixes

- DNS-over-HTTPS flooding fixes. **(CVE-2024-12705)**

  Fix DNS-over-HTTPS implementation issues that arise under heavy query load. Optimize resource usage for `named` instances that accept queries over DNS-over-HTTPS.

  Previously, `named` processed all incoming HTTP/2 data at once, which could overwhelm the server, especially when dealing with clients that sent requests but did not wait for responses. That has been fixed. Now, `named` handles HTTP/2 data in smaller chunks and throttles reading until the remote side reads the response data. It also throttles clients that send too many requests at once.

  In addition, `named` now evaluates excessive streams opened by clients that include no DNS data, which is considered "flooding." It logs these clients and drops connections from them. **[GL #4795]**

  In some cases, `named` could leave DNS-over-HTTPS connections in the *CLOSE_WAIT* state indefinitely. That has also been fixed. **[GL #5083]**

  ISC would like to thank Jean-François Billaud for his assistance with investigating this issue.

- Limit additional section processing for large RDATA sets. **(CVE-2024-11187)**

  When answering queries, don't add data to the additional section if the answer has more than 13 names in the RDATA. This limits the number of lookups into the database(s) during a single client query, reducing the query-processing load. **[GL #5034]**

  ISC would like to thank Toshifumi Sakaguchi for bringing this vulnerability to our attention.

### 14.29.2 New Features

- Add Extended DNS Error Code 22 - No Reachable Authority.

  When the resolver is trying to query an authoritative server and eventually times out, a SERVFAIL answer is given to the client. Add the Extended DNS Error Code 22 - No Reachable Authority to the response. **[GL #2268]**

- Add a new option to configure the maximum number of outgoing queries per client request.

  The configuration option `max-query-count` sets how many outgoing queries per client request are allowed. The existing `max-recursion-queries` value is the number of permissible queries for a single name and is reset on every CNAME redirection. This new option is a global limit on the client request. The default is 200.

  The default for `max-recursion-queries` is changed from 32 to 50. This allows `named` to send a few more queries while looking up a single name. **[GL #4980] [GL #4921]**

- Use the Server Name Indication (SNI) extension for all outgoing TLS connections.

  This improves compatibility with other DNS server software. **[GL #5099]**

### 14.29.3 Feature Changes

- Performance optimization for NSEC3 lookups introduced in BIND 9.20.2 was reverted to avoid risks associated with a complex code change. **[GL #5108]**
- The configuration clauses `parental-agents` and `primaries` are renamed to *remote-servers*.

  The top blocks `primaries` and `parental-agents` are no longer preferred and should be renamed to *remote-servers*. The zone statements *parental-agents* and *primaries* are still used, and may refer to any *remote-servers* top block. **[GL #4544]**
- Add *none* parameter to *query-source* and *query-source-v6* to disable IPv4 or IPv6 upstream queries but allow listening to queries from clients on IPv4 or IPv6. **[GL #4981]**

### 14.29.4 Bug Fixes

- Fix *nsupdate* hang when processing a large update.

  To mitigate DNS flood attacks over a single TCP connection, throttle the connection when the other side does not read the data. Throttling should only occur on server-side sockets, but erroneously also happened for *nsupdate*, which acts as a client. When *nsupdate* started throttling the connection, it never attempted to read again. This has been fixed. **[GL #4910]**
- Fix possible assertion failure when reloading server while processing update policy rules. **[GL #5006]**
- Preserve cache across reconfig when using *attach-cache*.

  When the *attach-cache* option is used in the `options` block with an arbitrary name, it causes all views to use the same cache. Previously, this configuration caused the cache to be deleted and a new cache to be created every time the server was reconfigured. This has been fixed. **[GL #5061]**
- Resolve the spurious drops in performance due to glue cache.

  For performance reasons, the returned glue records are cached on the first use. The current implementation could randomly cause a performance drop and increased memory use. This has been fixed. **[GL #5064]**
- Fix *dnssec-signzone* signing non-DNSKEY RRsets with revoked keys.

  *dnssec-signzone* was using revoked keys for signing RRsets other than DNSKEY. This has been corrected. **[GL #5070]**
- Fix improper handling of unknown directives in `resolv.conf`.

  The line after an unknown directive in `resolv.conf` could accidentally be skipped, potentially affecting *dig*, *host*, *nslookup*, *nsupdate*, or *delv*. This has been fixed. **[GL #5084]**
- Fix response policy zones and catalog zones with an `$INCLUDE` statement defined.

  Response policy zones (RPZ) and catalog zones were not working correctly if they had an `$INCLUDE` statement defined. This has been fixed. **[GL #5111]**

## 14.30 Notes for BIND 9.20.4

### 14.30.1 New Features

- Update built-in `bind.keys` file with the new 2025 IANA root key.

  Add an *initial-ds* entry to `bind.keys` for the new root key, ID 38696, which is scheduled for publication in January 2025. **[GL #4896]**

### 14.30.2 Removed Features

- Move contributed DLZ modules into a separate repository. DLZ modules should not be used except in testing.

  The DLZ modules were not maintained, the DLZ interface itself is going to be scheduled for removal, and the DLZ interface is blocking. Any module that blocks the query to the `database` blocks the whole server.

  The DLZ modules now live in https://gitlab.isc.org/isc-projects/dlz-modules repository. **[GL #4865]**

### 14.30.3 Feature Changes

- `dnssec-ksr` now supports KSK rollovers.

  The tool now allows for KSK generation, as well as planned KSK rollovers. When signing a bundle from a Key Signing Request (KSR), only the key that is active in that time frame is used for signing. Also, the CDS and CDNSKEY records are now added and removed at the correct time. **[GL #4697] [GL #4705]**

- Print **RFC 7314**: EXPIRE option in transfer summary. **[GL #5013]**

- Emit more helpful log messages for exceeding `max-records-per-type`.

  The new log message is emitted when adding or updating an RRset fails due to exceeding the `max-records-per-type` limit. The log includes the owner name and type, corresponding zone name, and the limit value. It will be emitted on loading a zone file, inbound zone transfer (both AXFR and IXFR), handling a DDNS update, or updating a cache DB. It's especially helpful in the case of zone transfer, since the secondary side doesn't have direct access to the offending zone data.

  It could also be used for `max-types-per-name`, but this change doesn't implement it yet as it's much less likely to happen in practice.

- Harden key management when key files have become unavailable.

  Prior to doing key management, BIND 9 will check if the key files on disk match the expected keys. If key files for previously observed keys have become unavailable, this will prevent the internal key manager from running.

### 14.30.4 Bug Fixes

- Use TLS for notifies if configured to do so.

  Notifies configured to use TLS will now be sent over TLS, instead of plain text UDP or TCP. Also, failing to load the TLS configuration for `notify` now results in an error. **[GL #4821]**

- *{&dns}* is as valid as *{?dns}* in a SVCB's dohpath.

  `dig` failed to parse a valid *SVCB* record with a *dohpath* URI template containing a *{&dns}*, like *dohpath=/some/path?key=value{&dns}"*. **[GL #4922]**

- Fix NSEC3 closest encloser lookup for names with empty non-terminals.

  A previous performance optimization for finding the NSEC3 closest encloser when generating authoritative responses could cause servers to return incorrect NSEC3 records in some cases. This has been fixed. **[GL #4950]**

- `recursive-clients` statement with value 0 triggered an assertion failure.

  BIND 9.20.0 broke *recursive-clients 0;*. This has now been fixed. **[GL #4987]**

- Parsing of hostnames in `rndc.conf` was broken.

  When DSCP support was removed, parsing of hostnames in `rndc.conf` was accidentally broken, resulting in an assertion failure. This has been fixed. **[GL #4991]**

- `dig` options of the form *[+-]option=<value>* failed to display the value on the printed command line. This has been fixed. **[GL #4993]**

- Provide more visibility into TLS configuration errors by logging *SSL_CTX_use_certificate_chain_file()* and *SSL_CTX_use_PrivateKey_file()* errors individually. **[GL #5008]**
- Fix a race condition when canceling ADB find which could cause an assertion failure. **[GL #5024]**
- SERVFAIL cache memory cleaning is now more aggressive; it no longer consumes a lot of memory if the server encounters many SERVFAILs at once. **[GL #5025]**
- Fix trying the next primary XoT server when the previous one was marked as unreachable.

  In some cases *named* failed to try the next primary server in the *primaries* list when the previous one was marked as unreachable. This has been fixed. **[GL #5038]**

## 14.31 Notes for BIND 9.20.3

### 14.31.1 New Features

- Log query response status to the query log.

  Log a query response summary using the new `responses` category. Logging can be controlled via the *responselog* option and via *rndc responselog*. **[GL #459]**
- Added WALLET type.

  Add the new record type WALLET (262). This provides a mapping from a domain name to a cryptographic currency wallet. Multiple mappings can exist if multiple records exist. **[GL #4947]**

### 14.31.2 Feature Changes

- Set logging category for `notify`/`xfer-in`-related messages.

  Some `notify` and `xfer-in`-related log messages were logged at the "general" category level instead of their own category. This has been fixed. **[GL #2730]**
- Allow IXFR-to-AXFR fallback on `DNS_R_TOOMANYRECORDS`.

  This change allows fallback from an IXFR failure to AXFR when the reason is `DNS_R_TOOMANYRECORDS`. **[GL #4928]**

### 14.31.3 Bug Fixes

- Fix a statistics channel counter bug when "forward only" zones are used.

  When resolving a zone with a "forward only" policy, and finding out that all the forwarders were marked as "bad", the "ServerQuota" counter of the statistics channel was incorrectly increased. This has been fixed. **[GL #1793]**
- Fix a bug in the static-stub implementation.

  Static-stub addresses and addresses from other sources were being mixed together, resulting in static-stub queries going to addresses not specified in the configuration, or alternatively, static-stub addresses being used instead of the correct server addresses. **[GL #4850]**
- Don't allow *statistics-channels* if libxml2 and libjson-c are not configured.

  When BIND 9 is not configured with the libxml2 and libjson-c libraries, the use of the *statistics-channels* option is a fatal error. **[GL #4895]**
- Separate DNSSEC validation from long-running tasks.

  Split CPU-intensive and long-running tasks into separate threadpools in a way that the long-running tasks - like RPZ, catalog zone processing, or zone file operations - don't block CPU-intensive operations like DNSSEC validations. **[GL #4898]**

- Fix an assertion failure when processing access control lists.

  The *named* process could terminate unexpectedly when processing ACLs. This has been fixed. **[GL #4908]**

- Fix a bug in Offline KSK using a ZSK with an unlimited lifetime.

  If the ZSK had an unlimited lifetime, the timing metadata `Inactive` and `Delete` could not be found and were treated as an error, preventing the zone from being signed. This has been fixed. **[GL #4914]**

- Limit the outgoing UDP send queue size.

  If the operating system UDP queue got full and the outgoing UDP sending started to be delayed, BIND 9 could exhibit memory spikes as it tried to enqueue all the outgoing UDP messages. It now tries to deliver the outgoing UDP messages synchronously; if that fails, it drops the outgoing DNS message that would get queued up and then timeout on the client side. **[GL #4930]**

- Do not set `SO_INCOMING_CPU`.

  Remove the `SO_INCOMING_CPU` setting as kernel scheduling performs better without constraints. **[GL #4936]**

- Fix the *rndc dumpdb* command's error reporting.

  The *rndc dumpdb* command was not reporting errors that occurred when *named* started up the database dump process. This has been fixed. **[GL #4944]**

- Fix long-running incoming transfers.

  Incoming transfers that took longer than 30 seconds would stop reading from the TCP stream and the incoming transfer would be indefinitely stuck, causing BIND 9 to hang during shutdown.

  This has been fixed, and the *max-transfer-time-in* and *max-transfer-idle-in* timeouts are now honored. **[GL #4949]**

- Fix an assertion failure when receiving DNS responses over TCP.

  When matching the received Query ID in the TCP connection, an invalid Query ID could cause an assertion failure. This has been fixed. **[GL #4952]**

### 14.31.4 Known Issues

- There are no new known issues with this release. See *above* for a list of all known issues affecting this BIND 9 branch.

## 14.32 Notes for BIND 9.20.2

### 14.32.1 New Features

- Support for Offline KSK implemented.

  Add a new configuration option *offline-ksk* to enable Offline KSK key management. Signed Key Response (SKR) files created with *dnssec-ksr* (or other programs) can now be imported into *named* with the new *rndc skr -import* command. Rather than creating new DNSKEY, CDS, and CDNSKEY records and generating signatures covering these types, these records are loaded from the currently active bundle from the imported SKR.

  The implementation is loosely based on draft-icann-dnssec-keymgmt-01.txt. **[GL #1128]**

- Print the full path of the working directory in startup log messages.

  *named* now prints its initial working directory during startup, and the changed working directory when loading or reloading its configuration file, if it has a valid *directory* option defined. **[GL #4731]**

- Support a restricted key tag range when generating new keys.

  When multiple signers are being used to sign a zone, it is useful to be able to specify a restricted range of key tags to be used by an operator to sign the zone. The range can be specified with `tag-range` in *dnssec-policy*'s *keys* (for *named* and *dnssec-ksr*) and with the new options *dnssec-keyfromlabel -M* and *dnssec-keygen -M*. **[GL #4830]**

### 14.32.2 Feature Changes

- Exempt prefetches from the *fetches-per-zone* and *fetches-per-server* quotas.

  Fetches generated automatically as a result of *prefetch* are now exempt from the *fetches-per-zone* and *fetches-per-server* quotas. This should help in maintaining the cache from which query responses can be given. **[GL #4219]**

- Improve performance for queries that require an NSEC3 wildcard proof.

  Rather than starting from the longest matching part of the requested name, lookup the shortest partial match. Most of the time this will be the actual closest encloser. **[GL #4460]**

- Follow the number of CPUs set by `taskset`/`cpuset`.

  Administrators may wish to constrain the set of cores that *named* runs on via the `taskset`, `cpuset`, or `numactl` programs (or equivalents on other OSes).

  If the admin has used `taskset`, *named* now automatically uses the given number of CPUs rather than the system-wide count. **[GL #4884]**

### 14.32.3 Bug Fixes

- Delay the release of root privileges until after configuring controls.

  Delay relinquishing root privileges until the control channel has been configured, for the benefit of systems that require root to use privileged port numbers. This mostly affects systems without fine- grained privilege systems (i.e., other than Linux). **[GL #4793]**

- Fix a rare assertion failure when shutting down incoming transfer.

  A very rare assertion failure could be triggered when the incoming transfer was either forcefully shut down, or it finished during the printing of the details about the statistics channel. This has been fixed. **[GL #4860]**

- Fix algorithm rollover bug when there are two keys with the same keytag.

  If there was an algorithm rollover and two keys of different algorithms shared the same keytags, there was the possibility that the check of whether the key matched a specific state could be performed against the wrong key. This has been fixed by not only checking for the matching key tag but also the key algorithm. **[GL #4878]**

- Fix an assertion failure in `validate_dnskey_dsset_done()`.

  Under rare circumstances, *named* could terminate unexpectedly when validating a DNSKEY resource record if the validation had been canceled in the meantime. This has been fixed. **[GL #4911]**

### 14.32.4 Known Issues

- Long-running tasks in offloaded threads (e.g. the loading of RPZ zones or processing zone transfers) may block the resolution of queries during these operations and cause the queries to time out.

  To work around the issue, the `UV_THREADPOOL_SIZE` environment variable can be set to a larger value before starting *named*. The recommended value is the number of RPZ zones (or number of transfers) plus the number of threads BIND should use, which is typically the number of CPUs. **[GL #4898]**

## 14.33 Notes for BIND 9.20.1

### 14.33.1 New Features

- Implement `rndc retransfer -force`.

  A new optional argument `-force` has been added to the command *`rndc retransfer`*. When it is specified, *`named`* aborts the ongoing zone transfer (if there is one) and starts a new transfer. **[GL #2299] [GL !9219]**

- *`dig`* now reports a missing QUESTION section for messages with opcode QUERY.

  Query responses should contain the QUESTION section, with some exceptions. *`dig`* was not reporting this. **[GL #4808] [GL !9269]**

### 14.33.2 Feature Changes

- Tighten *`max-recursion-queries`* and add *`max-query-restarts`* configuration statement.

  There were cases when the *`max-recursion-queries`* quota was ineffective. It was possible to craft zones that would cause a resolver to waste resources by sending excessive queries while attempting to resolve a name. This has been addressed by correcting errors in the implementation of *`max-recursion-queries`* and by reducing the default value from 100 to 32.

  In addition, a new *`max-query-restarts`* configuration statement has been added, which limits the number of times a recursive server will follow CNAME or DNAME records before terminating resolution. This was previously a hard-coded limit of 16 but is now configurable with a default value of 11.

  ISC would like to thank Huayi Duan, Marco Bearzi, Jodok Vieli, and Cagin Tanir from NetSec group, ETH Zurich for discovering and notifying us about the issue. **[GL #4741] [GL !9282]**

- Allow shorter *`resolver-query-timeout`* configuration.

  The minimum allowed value of *`resolver-query-timeout`* was lowered from its previous value of 10 000 milliseconds (which is still the default) to 301 milliseconds. Note however that values of 1 to 300 inclusive are interpreted as seconds before applying the limit. A value of zero is interpreted as the default. **[GL #4320] [GL !9220]**

- Raise the log level of priming failures.

  When a priming query is complete, it was previously logged at level `DEBUG(1)`, regardless of success or failure. It is now logged to `NOTICE` in the case of failure. **[GL #3516] [GL !9250]**

### 14.33.3 Bug Fixes

- Fix a crash caused by valid TSIG signatures with invalid time.

  An assertion failure was triggered when the TSIG had a valid cryptographic signature but the time was invalid. This could happen when the times between the primary and secondary servers were not synchronised. The crash has now been fixed. **[GL #4811] [GL !9245]**

- Return SERVFAIL for a too long CNAME chain.

  When following long CNAME chains, *`named`* was returning NOERROR (along with a partial answer) instead of SERVFAIL, if the chain exceeded the maximum length. This has been fixed. **[GL #4449] [GL !9203]**

- Reconfigure catz member zones during *`named`* reconfiguration.

  During a reconfiguration, *`named`* wasn't reconfiguring catalog zones' member zones. This has been fixed. **[GL #4733]**

- Update key lifetime and metadata after *`dnssec-policy`* reconfiguration.

Adjust key state and timing metadata if *dnssec-policy* key lifetime configuration is updated, so that it also affects existing keys. **[GL #4677] [GL !9191]**

- Fix a crash during zone modification.

  Fix an assertion failure that could happen when an authoritative zone was modified while the server was generating an answer from that zone. **[GL #4691] [GL !9126]**

- Fix assertion failure when executing *named-checkconf -v* to print its version. **[GL #4827] [GL !9246]**

- Fix generation of 6to4-self name expansion from IPv4 address.

  The period between the most significant nibble of the encoded IPv4 address and the 2.0.0.2.IP6.ARPA suffix was missing, resulting in the wrong name being checked. This has been fixed. **[GL #4766] [GL !9217]**

- *dig +yaml* was producing unexpected and/or invalid YAML. output. **[GL #4796] [GL !9213]**

- SVBC ALPN text parsing failed to reject zero-length ALPN. **[GL #4775] [GL !9209]**

- Fix false QNAME minimisation error being reported.

  Remove the false positive `success resolving` log message when QNAME minimisation is in effect and the final result is an NXDOMAIN. **[GL #4784] [GL !9215]**

- Fix `--enable-tracing` build on systems without dtrace.

  A missing `util/dtrace.sh` file prevented builds on systems without the `dtrace` utility. This has been corrected. **[GL #4835] [GL !9272]**

### 14.33.4 Known Issues

- There are no new known issues with this release. See *above* for a list of all known issues affecting this BIND 9 branch.

## 14.34 Notes for BIND 9.20.0

> **Note**
>
> This section only lists changes since BIND 9.18.28, the most recent release on the previous stable branch of BIND at the time of the publication of BIND 9.20.0.

### 14.34.1 New Features

- The *forwarders* statement now supports the *tls* argument, to be used to forward queries to DoT-enabled servers. **[GL #3726]**

- *named* now supports forwarding Dynamic DNS updates through DNS-over-TLS (DoT). **[GL #3512]**

- The *nsupdate* tool now supports DNS-over-TLS (DoT). **[GL !6752]**

- The *tls* block was extended with a new *cipher-suites* option that allows permitted cipher suites for TLSv1.3 to be set. Please consult the documentation for additional details. **[GL #3504]**

- Initial support for the PROXYv2 protocol was added. *named* can now accept PROXYv2 headers over all currently implemented DNS transports and *dig* can insert these headers into the queries it sends. Please consult the related documentation (*allow-proxy*, *allow-proxy-on*, *listen-on*, and *listen-on-v6* for *named*, *dig +proxy* and *dig +proxy-plain* for *dig*) for additional details. **[GL #4388]**

- The client-side support of the EDNS EXPIRE option has been expanded to include IXFR and AXFR query types. This enhancement enables *named* to perform AXFR and IXFR queries while incorporating the EDNS EXPIRE option. **[GL #4170]**
- A new configuration option *require-cookie* has been introduced. It specifies whether there should be a DNS COOKIE in the response for a given prefix; if not, *named* falls back to TCP. This is useful if it is known that a given server supports DNS COOKIE. It can also be used to force all non-DNS COOKIE responses to fall back to TCP. **[GL #2295]**
- The *check-svcb* option has been added to control the checking of additional constraints on SVCB records. This change affects *named*, *named-checkconf*, *named-checkzone*, *named-compilezone*, and *nsupdate*. **[GL #3576]**
- The new *resolver-use-dns64* option enables *named* to apply *dns64* rules to IPv4 server addresses when sending recursive queries, so that resolution can be performed over a NAT64 connection. **[GL #608]**
- A new option to *dnssec-policy* has been added, *cdnskey*, that allows users to enable or disable the publication of CDNSKEY records. **[GL #4050]**
- When using *dnssec-policy*, it is now possible to configure the digest type to use when CDS records need to be published with *cds-digest-types*. Also, publication of specific CDNSKEY/CDS records can now be set with *dnssec-signzone -G*. **[GL #3837]**
- Support for multi-signer model 2 (**RFC 8901**) when using *inline-signing* was added. **[GL #2710]**
- HSM support was added to *dnssec-policy*. Keys can now be configured with a `key-store` that allows users to set the directory where key files are stored and to set a PKCS#11 URI string. The latter requires OpenSSL 3 and a valid PKCS#11 provider to be configured for OpenSSL. **[GL #1129]**
- A new DNSSEC tool *dnssec-ksr* has been added to create Key Signing Request (KSR) and Signed Key Response (SKR) files. **[GL #1128]**
- *dnssec-verify* and *dnssec-signzone* now accept a `-J` option to specify a journal file to read when loading the zone to be verified or signed. **[GL #2486]**
- *dnssec-keygen* now allows the options *-k* and *-f* to be used together. This allows the creation of keys for a given *dnssec-policy* that match only the KSK (`-fK`) or ZSK (`-fZ`) roles. **[GL #1128]**
- The *response-policy* statement was extended with a new argument `ede`. It enables an **RFC 8914** Extended DNS Error (EDE) code of choice to be set for responses which have been modified by a given RPZ. **[GL #3410]**
- A new way of configuring the preferred source address when talking to remote servers, such as *primaries* and *parental-agents*, has been added: setting the `source` and/or `source-v6` arguments for a given statement is now possible. This new approach is intended to eventually replace statements such as *parental-source*, *parental-source-v6*, *transfer-source*, etc. **[GL #3762]**
- The new command-line *delv +ns* option activates name server mode, to more accurately reproduce the behavior of *named* when resolving a query. In this mode, *delv* uses an internal recursive resolver rather than an external server. All messages sent and received during the resolution and validation process are logged. This can be used in place of *dig +trace*. **[GL #3842]**
- The read timeout in *rndc* can now be specified on the command line using the *-t* option, allowing commands that take a long time to complete sufficient time to do so. **[GL #4046]**
- The statistics channel now includes information about incoming zone transfers that are currently in progress. **[GL #3883]**
- Information on incoming zone transfers in the statistics channel now also shows the zones' "first refresh" flag, which indicates that a zone is not fully ready and that its first ever refresh is pending or is in progress. The number of such zones is now also exposed by the *rndc status* command. **[GL #4241]**

- Added a new statistics variable `recursive high-water` that reports the maximum number of simultaneous recursive clients BIND has handled while running. **[GL #4668]**
- A new command, *rndc fetchlimit*, prints a list of name server addresses that are currently rate-limited due to *fetches-per-server* and domain names that are rate-limited due to *fetches-per-zone*. **[GL #665]**
- Queries and responses now emit distinct dnstap entries for DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH), and *dnstap-read* understands these entries. **[GL #4523]**
- *dnstap-read* can now print long timestamps with millisecond precision. **[GL #2360]**
- Support for libsystemd's `sd_notify()` function was added, enabling *named* to report its status to the init system. This allows systemd to wait until *named* is fully ready before starting other services that depend on name resolution. **[GL #1176]**
- Support for User Statically Defined Tracing (USDT) probes has been added. These probes enable fine-grained application tracing and introduce no overhead when they are not enabled. **[GL #4041]**

### 14.34.2 Removed Features

- Support for Red Hat Enterprise Linux version 7 (and clones) has been dropped. A C11-compliant compiler is now required to compile BIND 9. **[GL #3729]**
- Compiling with jemalloc versions older than 4.0.0 is no longer supported; those versions do not provide the features required by current BIND 9 releases. **[GL #4296]**
- The `auto-dnssec` configuration statement has been removed. Please use *dnssec-policy* or manual signing instead. See article how to migrate from `auto-dnssec` to *dnssec-policy*.

  The following statements have become obsolete: *dnskey-sig-validity*, *dnssec-dnskey-kskonly*, *dnssec-update-mode*, *sig-validity-interval*, and *update-check-ksk*. **[GL #3672]**
- Dynamic updates that add and remove DNSKEY and NSEC3PARAM records no longer trigger key rollovers and denial-of-existence operations. This also means that the *dnssec-secure-to-insecure* option has been obsoleted. **[GL #3686]**
- The `glue-cache` *option* has been removed. The glue cache *feature* still works and is now permanently *enabled*. **[GL #2147]**
- Configuring the control channel to use a Unix domain socket has been a fatal error since BIND 9.18. The feature has now been completely removed and *named-checkconf* now reports it as a configuration error. **[GL #4311]**
- The statements setting alternate local addresses for inbound zone transfers (`alt-transfer-source`, `alt-transfer-source-v6`, and `use-alt-transfer-source`) have been removed. **[GL #3714]**
- The `resolver-nonbackoff-tries` and `resolver-retry-interval` statements have been removed. Using them is now a fatal error. **[GL #4405]**
- BIND 9 no longer supports non-zero *stale-answer-client-timeout* values, when the feature is turned on. When using a non-zero value, *named* now generates a warning log message, and treats the value as `0`. **[GL #4447]**
- The Differentiated Services Code Point (DSCP) feature has been removed: configuring DSCP values in `named.conf` is now a configuration error. **[GL #3789]**
- The `keep-response-order` option has been declared obsolete and the functionality has been removed. *named* expects DNS clients to be fully compliant with **RFC 7766**. **[GL #3140]**
- Zone type `delegation-only`, and the `delegation-only` and `root-delegation-only` statements, have been removed. Using them is a configuration error.

  These statements were created to address the SiteFinder controversy, in which certain top-level domains redirected misspelled queries to other sites instead of returning NXDOMAIN responses. Since top-level domains are now DNSSEC-signed, and DNSSEC validation is active by default, the statements are no longer needed. **[GL #3953]**

- The `coresize`, `datasize`, `files`, and `stacksize` options have been removed. The limits these options set should be enforced externally, either by manual configuration (e.g. using `ulimit`) or via the process supervisor (e.g. `systemd`). **[GL #3676]**
- Support for using AES as the DNS COOKIE algorithm (`cookie-algorithm aes;`) has been removed. The only supported DNS COOKIE algorithm is now the current default, SipHash-2-4. **[GL #4421]**
- The TKEY Mode 2 (Diffie-Hellman Exchanged Keying Mode) has been removed and using TKEY Mode 2 is now a fatal error. Users are advised to switch to TKEY Mode 3 (GSS-API). **[GL #3905]**
- Special-case code that was originally added to allow GSS-TSIG to work around bugs in the Windows 2000 version of Active Directory has now been removed, since Windows 2000 is long past end-of-life. The *`-o`* option and the `oldgsstsig` command to *`nsupdate`* have been deprecated, and are now treated as synonyms for *`-g`* and `gsstsig` respectively. **[GL #4012]**
- Support for the `lock-file` statement and the `named -X` command-line option has been removed. An external process supervisor should be used instead. **[GL #4391]**

  Alternatively, the `flock` utility (part of util-linux) can be used on Linux systems to achieve the same effect as `lock-file` or `named -X`:

```
flock -n -x <directory>/named.lock <path>/named <arguments>
```

- The *`named`* command-line option *`-U`*, which specified the number of UDP dispatches, has been removed. Using it now returns a warning. **[GL #1879]**
- The `--with-tuning` option for `configure` has been removed. Each of the compile-time settings that required different values based on the "workload" (which were previously affected by the value of the `--with-tuning` option) has either been removed or changed to a sensible default. **[GL #3664]**
- The functions that were in the `libbind9` shared library have been moved to the `libisc` and `libisccfg` libraries. The now-empty `libbind9` has been removed and is no longer installed. **[GL #3903]**
- The `irs_resconf` module has been moved to the `libdns` shared library. The now-empty `libirs` library has been removed and is no longer installed. **[GL #3904]**

### 14.34.3 Deprecated Features

Features listed in this section still work but are scheduled for eventual removal.

- The use of the *`max-zone-ttl`* option in *`options`* and *`zone`* blocks has been deprecated; it should now be configured as part of *`dnssec-policy`*. A warning is logged if this option is used in *`options`* or *`zone`* blocks. In a future release, it will become nonoperational. **[GL #2918]**
- The *`sortlist`* option has been deprecated and will be removed in a future BIND 9.21.x release. Users should not rely on a specific order of resource records in DNS messages. **[GL #4593]**
- The `fixed` value for the *`rrset-order`* option and the corresponding `configure` script option have been deprecated and will be removed in a future BIND 9.21.x release. Users should not rely on a specific order of resource records in DNS messages. **[GL #4446]**

### 14.34.4 Feature Changes

- BIND now depends on liburcu, Userspace RCU, for lock-free data structures. **[GL #3934]**
- On Linux, libcap is now a required dependency to help *`named`* keep needed privileges. **[GL #3583]**
- Compiling BIND 9 now requires at least libuv version 1.34.0 or higher. libuv should be available on all supported platforms either as a native package or as a backport. **[GL #3567]**

- Outgoing zone transfers are no longer enabled by default. An explicit *allow-transfer* ACL must now be set at the *zone*, *view*, or *options* level to enable outgoing transfers. **[GL #4728]**
- DNS zones signed using *dnssec-policy* now automatically detect their parent servers, and BIND queries them to check the content of the DS RRset. This allows DNSSEC key rollovers to safely and automatically proceed when the parent zone is updated with new DNSSEC keys, i.e. using the CDS/CDNSKEY mechanism. This behavior is facilitated by the new *checkds* feature, which automatically populates *parental-agents* by resolving the parent NS records. These parent name servers are queried to check the DS RRset during a KSK rollover initiated by *dnssec-policy*. **[GL #3901]**
- The responsiveness of *named* was improved, when serving as an authoritative DNS server for a delegation-heavy zone(s) shortly after loading such zone(s). **[GL #4045]**
- To improve query-processing latency under load, the uninterrupted time spent on resolving long chains of cached domain names has been reduced. **[GL #4185]**
- QNAME minimization is now used when looking up the addresses of name servers during the recursive resolution process. **[GL #4209]**
- BIND now returns BADCOOKIE for out-of-date or otherwise bad but well-formed DNS server cookies. **[GL #4194]**
- The DNS name compression algorithm used in BIND 9 has been revised: it now compresses more thoroughly than before, so responses containing names with many labels might have a smaller encoding than before. **[GL #3661]**
- Processing large incremental transfers (IXFR) has been offloaded to a separate work thread so that it does not prevent networking threads from processing regular traffic in the meantime. **[GL #4367]**
- Querying the statistics channel no longer blocks DNS communication on the networking event loop level. **[GL #4680]**
- The *inline-signing* zone option is now ignored if there is no *dnssec-policy* configured for the zone. This means that unsigned zones no longer create redundant signed versions of the zone. **[GL #4349]**
- The *inline-signing* statement can now also be set inside *dnssec-policy*. The default is to use *inline-signing*. This also applies to the built-in policies `default` and ``insecure`. If *inline-signing* is set at the `zone` level, it overrides the value set in *dnssec-policy*. **[GL #3677]**
- Due to the change in default value from `no` to `yes`, DNSSEC-enabled dynamic zones that do not have *inline-signing* explicitly set must now add the option to their configuration with the value `no` if they do not want their zone also to be inline-signed.
- Following **RFC 9276** recommendations, *dnssec-policy* now only allows an NSEC3 iteration count of 0 for the DNSSEC-signed zones using NSEC3 that the policy manages. **[GL #4363]**
- The maximum number of NSEC3 iterations allowed for validation purposes has been lowered from 150 to 50. DNSSEC responses containing NSEC3 records with iteration counts greater than 50 are now treated as insecure. **[GL #4363]**
- The `dnssec-validation yes` option now requires an explicitly configured *trust-anchors* statement. If using manual trust anchors is not operationally required, then please consider using `dnssec-validation auto` instead. **[GL #4373]**
- *named-compilezone* no longer performs zone integrity checks by default; this allows faster conversion of a zone file from one format to another. **[GL #4364]**

  Zone checks can be performed by running *named-checkzone* separately, or the previous default behavior can be restored by using:

```
named-compilezone -i full -k fail -n fail -r warn -m warn -M warn -S warn -T warn␣
↪-W warn -C check-svcb:fail
```

- The red-black tree data structure used in the RBTDB (the default database implementation for cache and zone databases), has been replaced with QP-tries. This is expected to improve performance and scalability, though in the current implementation large zones require roughly 15% more memory than the old red-black tree data structure.

  A side effect of this change is that zone files that are created with *masterfile-style* `relative` - for example, the output of *dnssec-signzone* - will no longer have multiple different *$ORIGIN* statements. There should be no other changes to server behavior.

  The old RBT-based database still exists for now, and can be used by specifying `database rbt` in a `zone` statement in `named.conf`, or by compiling with `configure --with-zonedb=rbt --with-cachedb=rbt`. **[GL #4411] [GL #4614]**

- Multiple RNDC messages are now processed when sent in a single TCP message.

  ISC would like to thank Dominik Thalhammer for reporting the issue and preparing the initial patch. **[GL #4416]**

- The DNSSEC signing data included in zone statistics identified keys only by the key ID; this caused confusion when two keys using different algorithms had the same ID. Zone statistics now identify keys using the algorithm number, followed by "+", followed by the key ID: for example, `8+54274`. **[GL #3525]**

- The TTL of the NSEC3PARAM record for every NSEC3-signed zone was previously set to 0. It is now changed to match the SOA MINIMUM value for the given zone. **[GL #3570]**

- On startup, *named* now sets the limit on the number of open files to the maximum allowed by the operating system, instead of trying to set it to "unlimited". **[GL #3676]**

- When an international domain name is not valid according to IDNA2008, *dig* now tries to convert it according to IDNA2003 rules, or pass it through unchanged, instead of stopping with an error message. The `idna2` utility can be used to check IDNA syntax. **[GL #3527]**

- The memory statistics have been reduced to a single counter, `InUse`; `Malloced` is an alias that holds the same value. The other counters were usable with the old BIND 9 internal memory allocator, but they are unnecessary now that the latter has been removed. **[GL #3718]**

- The log message `resolver priming query complete` has been moved from the INFO log level to the DEBUG(1) log level, to prevent *delv* from emitting that message when setting up its internal resolver. **[GL #3842]**

- Worker threads' event loops are now managed by a new "loop manager" API, significantly changing the architecture of the task, timer, and networking subsystems for improved performance and code flow. **[GL #3508]**

- The code for DNS over TCP and DNS over TLS transports has been replaced with a new, unified transport implementation. **[GL #3374]**

### 14.34.5 Bug Fixes

- When the same *notify-source* address and port number was configured for multiple destinations and zones, an unresponsive server could tie up the relevant network socket until it timed out; in the meantime, NOTIFY messages for other servers silently failed. *named* will now retry sending such NOTIFY messages over TCP. Furthermore, NOTIFY failures are now logged at the INFO level. **[GL #4001] [GL #4002]**

- DNS compression is no longer applied to the root name (`.`) if it is repeatedly used in the same RRset. **[GL #3423]**

- *named* could incorrectly return non-truncated, glueless referrals for responses whose size was close to the UDP packet size limit. This has been fixed. **[GL #1967]**

### 14.34.6 Known Issues

- On some platforms, including FreeBSD, *named* must be run as root to use the *rndc* control channel on a privileged port (i.e., with a port number less than 1024; this includes the default *rndc port*, 953). Currently, using the *named -u* option to switch to an unprivileged user makes *rndc* unusable. This will be fixed in a future release; in the meantime, `mac_portacl` can be used as a workaround, as documented in https://kb.isc.org/docs/aa-00621. **[GL #4793]**
- See *above* for a list of all known issues affecting this BIND 9 branch.

## 14.35 License

BIND 9 is open source software licensed under the terms of the Mozilla Public License, version 2.0 (see the `COPYING` file for the full text).

Those wishing to discuss license compliance may contact ISC at https://www.isc.org/contact/.

## 14.36 End of Life

BIND 9.20 (Extended Support Version) will be supported until at least March, 2028. See https://kb.isc.org/docs/aa-00896 for details of ISC's software support policy.

## 14.37 Thank You

Thank you to everyone who assisted us in making this release possible.

CHAPTER

FIFTEEN

# CHANGELOG

**Note**

The following list contains detailed information about BIND 9 development. Regular users should refer to *Release Notes* for changes relevant to them.

## 15.1 BIND 9.20.29

### 15.1.1 Security Fixes

- [CVE-2026-19668] Prevent excessive CPU use validating crafted DNSSEC responses. `a0a61dba9e`

  A malicious authoritative server could serve a securely delegated zone whose DS and DNSKEY records carry many distinct key tags but no valid match, forcing a validating resolver into excessive key-tag matching and high CPU use for every query. BIND now bounds this work with the per-query validation limit (max-validations-per-fetch). **[GL #5349]**

- [CVE-2026-19033] Require a TSIG on every message of incoming zone transfers. `9404cd2b8c`

  BIND 9 used to accept TSIG-signed zone transfers in which some messages were unsigned, and processed those messages before the next signature could vouch for them. It now requires a TSIG on every message of an incoming AXFR or IXFR; all modern nameserver already sign every message, so no change is expected in practice. **[GL #6062]**

- [CVE-2026-77119] Prevent a DNSSEC downgrade of secure delegations via unrelated NSEC3. `3bed9c8e9e`

  A validating resolver could be tricked into treating a secure delegation as unsigned and accepting forged answers for names beneath it, if an attacker could inject responses to its queries. Such forged proofs are now rejected. **[GL #6234]**

- [CVE-2026-19941] Prevent forged DNSSEC-validated NXDOMAIN responses. `a36bf58daf`

  A validating resolver could accept a signed NSEC record from an unrelated zone as proof that a wildcard did not exist. An on-path attacker or malicious forwarder controlling a signed zone could therefore forge an authenticated NXDOMAIN response for a name that should resolve through a wildcard. BIND now requires the wildcard-denial and name-nonexistence proofs to be signed by the same zone. **[GL #6253]**

- [CVE-2026-19666] DNS64 with break-dnssec could cause an assertion failure. `4cec4965c4`

  When a "dns64" statement is configured with "break-dnssec yes" and its "exclude" list matches some but not all of the addresses in an AAAA RRset, named removes the excluded addresses from the answer instead of synthesizing new ones. If the answer being filtered had been cached together with a proof that the queried name does not exist -- which is what a wildcard match produces -- named terminated with an assertion failure.

Only recursive resolvers are affected, and only when "break-dnssec yes" is in use; the answer has to come from the cache, so a server that is only authoritative cannot reach this. **[GL #6301]**

- [CVE-2026-19667] Reject negative cache records that do not fit in a dns_rdata_t. `dbf08c8581`

  A single crafted response from a server could make a resolver cache a malformed negative entry and then terminate with an assertion failure when reading it back. Only recursive resolvers are affected, on a default configuration. **[GL #6302]**

- [CVE-2026-19662] Prevent resolver crash with cached DNSSEC proofs. `c884cc1ba0`

  Under certain timing conditions, concurrent recursive queries could cause named to crash when cached DNSSEC NOQNAME proof data was replaced while still in use. Cached proof data is now retained until all queries using it have completed. **[GL #6333]**

- [CVE-2026-75029] Discard repeated SOA, CNAME, and DNAME records when parsing DNS messages. `0d630758c2`

  A DNS message could carry the same SOA, CNAME, or DNAME record many times, and named kept every copy while parsing it. With name compression those copies took up far more memory internally than in the message itself, and every later processing step had to handle all of them. named now keeps the first copy of such a record and discards identical repeats. **[GL #6335]**

- [CVE-2026-77692] Fix an unauthenticated crash on HTTPS using SIG(0) `5a24401c5c`

  A specifically crafted HTTPS query using SIG(0) as authentication could crash named if the client closes the connection before named actually verifies the signature. This is now fixed. **[GL #6343]**

- [CVE-2026-81736] Cached HTTPS/SVCB aliases could exhaust resolver CPU. `20bbb1639a`

  A recursive resolver that had cached a large set of interlinked HTTPS or SVCB records in alias form could be driven to do an excessive amount of work assembling a single response, because it followed every cached alias target when building the additional section. A client permitted to use recursion, together with an attacker-controlled zone used to plant the records, could repeat small queries to consume enough CPU to delay or deny service to other clients. The amount of additional processing done for one query is now bounded. **[GL #6347]**

- [CVE-2026-76163] Prevent TKEY queries from terminating named without global options. `7645138538`

  named could terminate unexpectedly when a remote client sent a TKEY query if the configuration did not include a global options statement. This has been fixed.

  ISC thanks Owais Lone (thesecguy) for reporting the issue. **[GL #6357]**

- [CVE-2026-78301] Out-of-zone records in a zone database could be served as authoritative. `72a10c3a0b`

  When a zone database contained records for names outside the zone — such as a delegation above the zone apex, left behind by a secondary that had accepted out-of-zone data from its primary — the server could treat them as authoritative and answer queries for names inside the zone with that out-of-zone data instead of the zone's own. A server that was also a resolver could follow such a delegation and cache the answers of the server it named, affecting names outside the configured zone. Zone database lookups are now confined to names at or below the zone's origin.

  ISC would like to thank Henrique Pereira for reporting the issue. **[GL #6361]**

- [CVE-2026-80274] Crash on wildcard answers carrying both NSEC and NSEC3 proofs. `0e44451b1a`

  When a wildcard answer arrived with both NSEC and NSEC3 records at the name proving that the queried name does not exist, the resolver could pick different records when caching the answer and when retrieving the proof, depending on the order in which the authoritative server sent them. This could terminate named with an assertion failure, fail the query with SERVFAIL, or serve a denial record other than the one that had been verified. The resolver now caches and serves the same denial record it accepted when the answer was received.

  ISC would like to thank hythyt for reporting the issue. **[GL #6369]**

- [CVE-2026-81563] Following HTTPS/SVCB aliases could leak resolver cache memory. `3162df369e`

  When a recursive server answered a query for an HTTPS or SVCB record in alias form and the alias target had more than 13 records, the target records were pinned in the cache permanently instead of being released once the answer was sent. A remote party who could make the server follow such aliases to a steady stream of fresh names could grow the cache beyond the configured max-cache-size until the server was unable to resolve unrelated names. The records are now released correctly.

  ISC would like to thank Samy Medjahed/Ap4sh for reporting the issue. **[GL #6374]**

### 15.1.2 New Features

- Add an agent skill for the isc_job/isc_async/isc_work APIs. `fe32990b06`

  Documents when to use isc_job_run(), isc_async_run() or isc_work_enqueue(), and the contract each one imposes. No functional change. **[GL !12561]**

### 15.1.3 Removed Features

- Remove unused closest encloser proof caching. `abd8b5bfd8`

  BIND used to cache an NSEC3 closest encloser proof alongside positive wildcard answers so that a resolver could re-send it when answering from its cache. That stopped being used in BIND 9.9 (2011), when positive wildcard responses were changed to omit that NSEC3 record — RFC 5155 requires only the next closer name proof — and the closest encloser came to be derived during validation instead. The caching code has been unreachable ever since, so this removes it with no change in behaviour. **[GL #5803] [GL !12660]**

### 15.1.4 Feature Changes

- Reject oversized and malformed DNSKEY records up front. `6c22109924`

  Oversized RSA key material in a DNSKEY record was only rejected after it had been converted, allocating memory proportional to the record size. Such records are now rejected before conversion, as are Ed25519 and Ed448 keys with trailing bytes that were previously silently ignored. **[GL #4537] [GL !12560]**

### 15.1.5 Bug Fixes

- Prevent a crash when using both dns64 and filter-a. `bce5d10d18`

  An assertion failure was possible when using both *dns64* and the *filter-a* plugin simultaneously; this has been fixed. **[GL #5979] [GL !12663]**

- Fix update-policy grant external address passing. `b1e955c326`

  Only TCP client addresses are supposed to be passed to an *external* handler for the associated *update-policy* rule, but UDP client addresses were also being passed. This could have caused the external handler to return a result it otherwise wouldn't. This has been fixed. **[GL #6061] [GL !12555]**

- Missing required NSEC3 for delegation not detected. `e84ed2e9d7`

  A missing required NSEC3 record for an insecure delegation in a non OPTOUT range was not being detected. This has been fixed. **[GL #6063] [GL !12611]**

- Tighten EUI48 and EUI48 text parsing. `ff50f2cdf1`

  Malformed EUI48 and EUI64 records could be accepted. This has been fixed. **[GL #6082] [GL !12521]**

- GeoIP ACL state can be stale or wrong after reload. `63baf425b3`

*named* caches GeoIP information after looking it up, but the cached information was not invalidated when the GeoIP database was reloaded, so it could continue to be used. We now invalidate existing cached GeoIP information as part of the reloading process. **[GL #6083] [GL !12662]**

- Honor DNSSEC policy key tag ranges. `b82e5834b7`

  When a DNSSEC policy configured a non-default tag-range, dnssec-keygen and dnssec-ksr could accept generated keys outside that range. Both tools now honor the configured minimum and maximum key tags. **[GL #6091] [GL !12549]**

- Fix double free in mdig when EDNS options are specified. `af5bd0b0ff`

  When the default_query is cloned the EDNS options need to be cloned rather than the pointer copied. The old behaviour results in a double free of the options. This has been fixed. **[GL #6095] [GL !12661]**

- Fix a crash when an IXFR falls back to AXFR with updates still pending. `e34062bc7e`

  When a secondary zone received an incremental transfer (IXFR) and the primary then caused named to fall back to a full transfer (AXFR) while some of the already-received incremental changes were still waiting to be applied, named could later crash when that transfer finished. The pending changes are now discarded correctly before the AXFR retry. **[GL #6114] [GL !12624]**

- Fix DS requests to parental agents over TLS. `55830d30f6`

  TLS configuration for parental agents was being ignored when sending DS requests. This has been fixed. **[GL #6135] [GL !12613]**

- Fix a crash when resolving names below a cached DNAME. `b94e940f52`

  A recursive resolver could crash when it answered a query for a name beneath a cached DNAME while that same DNAME record was concurrently refreshed or evicted from the cache. **[GL #6182] [GL !12593]**

- Rndc-confgen *-q* (quiet) option is documented but doesn't work. `7e4a7ca1a7`

  The command line parsing in rndc-confgen was broken so *rndc-confgen -q* did not work. This has been fixed. **[GL #6187] [GL !12575]**

- Enforce query ACLs for redirect zones and searched DLZs. `bc69876b2e`

  Queries answered from redirect zones or searched DLZ databases did not consistently honor *allow-query* and *allow-query-on*, potentially exposing restricted DNS data to excluded clients or through excluded listening addresses. These ACLs are now enforced before redirect or DLZ data is returned. **[GL #6251]**, #6252 **[GL !12646]**

- Check "asnum" validity in GeoIP ACLs. `28c2bfdc7b`

  We now check the validity of autonomous system (AS) numbers when parsing GeoIP ACLs that use *asnum* elements at configuration time.

  *asnum* values start with an optional case-insensitive "AS" prefix, followed only by decimal digits, with no spaces or other extraneous characters. The value represented cannot exceed 2^32. **[GL #6255] [GL !12511]**

- Prevent crashes while reporting DNSSEC signing statistics. `c190514f0a`

  Servers with zone-statistics full could terminate while reporting DNSSEC signing statistics for a zone tracking adding more than four signing keys. **[GL #6256] [GL !12674]**

- Fix various nits in the netmgr code. `c28cdad51b`

  The MR consists of couple of small fixes and uncaught errors in the Network Manager. **[GL #6257] [GL !12576]**

- Fix a crash on remote-servers lists that reference themselves. `aaae614f9d`

  Since 9.21.16 and 9.20.17, a remote-servers, primaries, masters, or parental-agents list that referenced itself, directly or through another list, made named crash on startup or reconfiguration. Such references are again skipped and the remaining entries in the list are used, as in earlier versions. **[GL #6287] [GL !12604]**

- A record from outside a response policy zone could stop named. `d135513b37`

  A response policy zone transferred from a primary can contain a record whose name lies outside the zone. Such a record could stop named, both when it arrived and again at every startup afterwards, because a secondary keeps it in its own copy of the zone. Records like this are now rejected and logged; previously one could also silently create a policy entry for an unrelated name. **[GL #6304] [GL !12543]**

- "rndc flushtree ." failed to flush the cache. `96e8b585ed`

  *rndc flushtree* flushes cache data below a specified name. If the name specified is the DNS root, it should fully empty the cache, the same as *rndc flush*. However, there was a bug causing the command, in that case, to have no effect on the cache at all; this has been fixed. **[GL #6308] [GL !12582]**

- Invalid key-store configuration could abort the DNSSEC tools. `1d796ab072`

  Invalid configured key-stores named "key-directory" in configuration files could abort the DNSSEC tools. This has been fixed. **[GL #6313] [GL !12653]**

- NSEC signature set could bypass the secure-delegation check. `c966177f6c`

  When proving that a delegation is insecure, the validator bounded an NSEC record's authority by the signer of whichever RRSIG happened to come first in the record's signature set, rather than the signature that actually verified. A grandparent NSEC padded with an extra, unverifiable signature could therefore pass the check that keeps such proofs from reaching below a signed child zone. The validator now requires every signature on the NSEC to name the same signer and refuses proofs whose signature set is malformed or larger than max-validations-per-fetch allows. **[GL #6321]**

- Fix a possible nsupdate issue when using GSS-TSIG. `4ddcab2d3c`

  The *`nsupdate`* process could terminate unexpectedly when using the GSS-TSIG mode executed with the *`nsupdate -g`* option. This has been fixed. **[GL #6325] [GL !12588]**

- Fix isccc_alist_define error paths. `af1349552a`

  If there is an out of memory error in isccc_alist_define a memory leak (the sexpr holding the key name) or a double free (value) could occur. This has been fixed. **[GL #6329] [GL !12636]**

- Check for empty 'endpoints' list. `23f58af443`

  Configuring an *http* block with *endpoints {};* previously caused a crash in *named*. This is now rejected earlier by the configuration check. **[GL #6330] [GL !12552]**

- Named could crash with a single-element geoip sortlist. `0e996a4d3b`

  If *named* was configured with a single-element sortlist containing a *geoip* ACL element, any matching query triggered an assertion failure. This has been fixed. **[GL #6342] [GL !12583]**

- Prevent out-of-bailiwick CNAMEs from evicting cached records. `cdedd4acd5`

  A recursive resolver could remove valid cached records when a DNS response contained an out-of-bailiwick CNAME with the same owner name. Out-of-bailiwick data is now discarded before it can modify the cache. **[GL #6345] [GL !12651]**

- Restore periodic cleanup of stale resolver address data. `356f4013f8`

  Stale resolver address data could remain cached until memory pressure or an explicit flush. Correct the cleanup interval so it is removed periodically. **[GL #6346] [GL !12589]**

- Fix named-checkconf/named crash with malformed key name. `9f218f6aaf`

  When a primary/remote-server key name was malformed, named-checkconf and named were both crashing (after warning about the invalid key name). This is now fixed. **[GL #6362] [GL !12639]**

- Fix -Wformat-truncation warning in totext_in_wks() `f97c2bea40`

  BIND 9 failed to build with GCC 16 at -O3: rendering a WKS record as text triggered a -Wformat-truncation error, which is fatal in developer builds. The port number is now printed with a 16-bit format specifier, so the compiler can see it always fits the output buffer. **[GL !12542]**

- Fix off-by-one errors caused by magic hardcoded values. `726c6cb795`

  Fix off-by-one comparinson errors: "named -p http=" dropped the first digit of the given port (for example, "http=8080" selected port 80) and now uses the port as given, and "named-rrchecker -C" compared only part of the "CLASS" prefix when filtering generic class names, which was harmless in practice but is now corrected. **[GL !12616]**

- Hmac_verify() now accepts truncated HMACs only when requested. `c81b111496`

  The hmac_verify() function incorrectly compares only up to 'sig->length' bytes, but the signature and its length should not be trusted, e.g. in case if it comes from a user query.

  Don't accept signatures which length isn't equal to the expected calculated HMAC length unless it is explicitly requested by the caller, e.g. for truncated TSIG [1] support.

  [1] https://datatracker.ietf.org/doc/html/rfc8945#name-tsig-truncation -policy **[GL !12629]**

- Prevent resolver crashes while processing DNS over TCP. `81b3b6d89f`

  Recursive resolvers could terminate with an assertion failure while processing DNS responses over TCP under sustained traffic. The failure was observed on resolvers configured globally with forward only; the same transport path is also used by iterative resolution. This has been fixed. **[GL !12537]**

## 15.2 BIND 9.20.28

> **Note**
>
> The BIND 9.20.28 release was withdrawn after the discovery of a regression in it during pre-release testing.

## 15.3 BIND 9.20.27

### 15.3.1 New Features

- Disclose active Negative Trust Anchors with Extended DNS Error 33. `566e7018278`

  A Negative Trust Anchor (RFC 7646) turns off DNSSEC validation for a domain, so a name that would normally fail validation resolves instead. named now marks such answers with Extended DNS Error code 33, "Negative Trust Anchor", so operators can see at a glance when a response came back only because an NTA was in effect. **[GL #6268] [GL !12426]**

- Add development guidance for AI coding agents under .agents/skills/ `ef651c93709`

  This adds a set of skill documents that give AI coding agents the project's established practices up front instead of having them rediscovered (or gotten wrong) in every session: the canonical build and test invocations, the memory-allocator contract, the disciplines for RCU mutation, per-loop sharded structures, struct-layout work, and flight-recorder debugging of concurrency bugs, plus the commit and merge-request conventions. **[GL !12401]**

- Add more unit tests for isc_time API. `9d6855ef735`

  While working on internal 64-bit time for BIND 9, the unit test suite for isc_time API has been extended.

  Backport the unit tests from 64-bit time branch to the main branch. This should make the unit tests for isc_time API (mostly) complete.

Related to #2959 **[GL !12432]**

### 15.3.2 Feature Changes

- Batch qp transaction for RPZ updates. `cb40fb91fa2`

  RPZ was built around fine-grained locking, but that forces the use of many small qp transactions. With this MR, we switch qp transaction to handle the full update to the rpz summary structure. While this serializes the RPZ updates, the reduced overhead from batching qp transactions more than compensates for it and results in improvements for big RPZ zones. **[GL #5787]**, #6270 **[GL !12493]**

- Pass the work callback result to the done callback. `65cdb7c7613`

  The *isc_work* callback now returns *isc_result_t* and the value is handed to the done callback, so the callers no longer need their own result-passing state. **[GL !12391]**

### 15.3.3 Bug Fixes

- Dig +yaml producing invalid YAML when a lookup fails. `65d1776959e`

  When "dig +yaml" was run and no server could be reached, dig printed its plain-text startup banner (the "; <<>> DiG ..." and ";; global options" lines) ahead of the machine-readable output, so the result was not valid YAML and could not be parsed. dig no longer emits that banner in YAML mode. As part of the same change, the banner is now built only after the whole command line has been read, so options given after the query name (such as +nocmd, +short and +yaml) are correctly reflected in it. **[GL #1230] [GL !12430]**

- Ensure NSEC authority does not cross zonecut boundary. `0baaddd096a`

  When using a cached NSEC record to prove that a delegation is insecure, we now check that the signer name in the corresponding RRSIG is not above a known secure delegation point. This prevents a signed namespace from being downgraded to insecure using an NSEC record from the grandparent zone. **[GL #5967] [GL !12394]**

- Treat non canonical RPZ prefixes as any other failure. `17f0828b460`

  RPZ prefixes that were not encoded in canonical form do not work. Treat them as any other encoding error. **[GL #6043] [GL !12482]**

- Properly prevent TSIG generation command line injection attacks. `f777451d261`

  When key names are generated with *rndc-confgen*, *tsig-keygen* and *ddns-confgen*, special characters must be escaped to ensure the configuration is parsed correctly. **[GL #6071] [GL !12396]**

- Dig with IDN output could leak memory on ISC_R_NOSPACE retry. `c4e53b59b2e`

  The IDN to text display call back could leak the memory holding the converted name if it did not fit into the buffer. This has been fixed. **[GL #6073] [GL !12483]**

- Dnssec-signzone had a potential heap bounds overflow write. `994c3bd4323`

  It was possible for *dnssec-signzone* to overflow array bounds while signing. This has been fixed. **[GL #6076] [GL !12503]**

- Restore SMF support on Solaris and illumos. `8bde9318f63`

  SMF support on Solaris and illumos was silently dropped by a build system rewrite in 2018; it is now detected and enabled again. **[GL #6096] [GL !12456]**

- Fix NULL pointer dereference in dnstap-read. `9c459b1854d`

  It was possible to dereference a NULL pointer in dnstap-read causing it to exit on a malformed DNSTAP file. This has been fixed. **[GL #6124] [GL !12494]**

- Change catz coo locking. `78265c8f6fc`

  Catalog zones might need to inspect the change-of-ownership records of other catalog zones, which required to release the lock in the middle of certain operations, leading to possible race conditions.

  Since the operations on change-of-ownership records are limited, we can instead use a design with a second lock protecting the change-of-ownership records on read. We structure the API so that holding two change-of-ownership locks at the same time is impossible. **[GL #6131] [GL !12410]**

- Treat an unusable NSEC3 chain as a verification failure. `879a383ef16`

  When transferring in a mirror zone, DNSSEC verification could incorrectly succeed when the zone had an invalid *NSEC3PARAM* record, leading to subsequent validation failures. This has been fixed. **[GL #6136] [GL !12431]**

- Unterminated OpenSSL private-key *Label:* field can be read past its parser buffer. `932581b74bd`

  Check that the string encoded in the Label: field of the .private file of a key pair is NUL terminated and the correct length. Reject the .private file if it is not. **[GL #6193] [GL !12417]**

- Negative caching stopped working with stale-answer-client-timeout 0. `2687d16ed2f`

  With "stale-answer-client-timeout 0" configured, every client query for a name cached as NXDOMAIN or NODATA was sent on to the authoritative servers, even while the cached negative answer was still within its TTL, so the resolver effectively lost negative caching. Negative answers are now refreshed only once they have actually gone stale. **[GL #6245] [GL !12384]**

- MacOS byte swapping macros already defined. `5c1d3df49c0`

  Don't redefine them if the development environment already defines them. **[GL #6250] [GL !12406]**

- Use memmove in isc_sockaddr_fromin/isc_sockaddr_fromin6. `3f3c82292c9`

  Use memmove instead of direct assignment from the source pointer because the source pointer is not guaranteed to be correctly aligned. **[GL #6260] [GL !12449]**

- Fix compilation on GNU/Hurd. `20f5580e017`

  Fix compilation issues on GNU/Hurd. **[GL #6285] [GL !12496]**

- Restore arc4random() detection dropped in the v9.21.14 merge. `86ad6866eeb`

  Commit 4db9e5d90e2 ("Use arc4random for CSPRNG when available", part of the CVE-2025-40780 fix) guarded the arc4random() code paths in lib/isc/random.h and lib/isc/random.c with HAVE_ARC4RANDOM and added the corresponding function check to meson.build. The manual conflict resolution in merge c2a672bbaef ("Merge tag 'v9.21.14'") kept the code changes but dropped the meson.build hunk, so HAVE_ARC4RANDOM was never defined and platforms with arc4random() (macOS and the BSDs) silently fell back to the internal ChaCha-based CSPRNG. Restore the check.

  Assisted-by: Claude:claude-fable-5 **[GL !12452]**

## 15.4 BIND 9.20.26

### 15.4.1 Security Fixes

- [CVE-2026-11331] Fix handling of rpz CNAME expansion that returns name too long. `b950da5f625`

  Previously, if the expansion of a wildcard CNAME RPZ policy resulted in a name that exceeded the length limit, a self referential CNAME and the original address record were returned, allowing the policy to be bypassed. In branches up to 9.20, this also left query processing in an inconsistent state which could trigger an assertion failure. We now return a YXDOMAIN response, without the address.

  ISC would like to thank Laith Mash'al (0xmshal) for bringing this issue to our attention. **[GL #5856]**

- [CVE-2026-11721] Invalid signed wildcard records were being accepted. `249752cd72d`

  Signed wildcard responses in which the Labels field in the *RRSIG* record was less than the number of labels in the Signer Name field were being incorrectly accepted. This in turn broke *synth-from-dnssec*, which depends on such records being correctly validated. This has been fixed.

  ISC thanks Qifan Zhang of Palo Alto Networks for bringing this issue to our attention. **[GL #5871]**

- [CVE-2026-13321] Fix DNSSEC validation bypass via out-of-zone NSEC Next Field. `6fbc963d4af`

  A malicious zone with out-of-zone NSEC next owner names can cause a DNSSEC validating resolver to cache such record and, if *synth-from-dnssec* is enabled, to generate negative answers for any zone that is covered by the range.

  ISC would like to thank Qifan Zhang of Palo Alto Networks for reporting the issue. **[GL #5873]**

- [CVE-2026-10723] Correct verification of NSEC3 signer name. `df3abfc3184`

  BIND 9 accepted child-zone NSEC3 records where the first label equals the hash of the parent zone as valid parent-zone closest encloser proofs. This has been fixed.

  ISC thanks Qifan Zhang of Palo Alto Networks for reporting the issue. **[GL #5874]**

- [CVE-2026-12617] Do no assert for some specifics CNAME and DNAME queries. `d75d1e93958`

  A bug in the resolver's handling of certain cached DNAME and CNAME responses could cause named to trigger an assertion failure and exit. An attacker controlling a domain name and the authoritative DNS server it is hosted on could exploit this behavior to cause a denial-of-service. This vulnerability has been fixed.

  ISC thanks Qifan Zhang of Palo Alto Networks for bringing this issue to our attention. **[GL #5946]**

- [CVE-2026-10822] Malformed DNSKEY records could trigger an assertion. `a1777ce7ee0`

  Previously, *dns_name_fromwire()* did not honor the record boundary when reading names from the wire, allowing malformed records to be accepted when they should not have been. In particular, malformed DNSKEY records could trigger an assertion failure when being printed. This has been fixed. **[GL #6004]**

- Reclaim memory promptly when DNSSEC validations are canceled. `b8f9d7657ac`

  When a resolver is flooded with queries that require DNSSEC validation — for example during a random-subdomain attack — many of those validations are canceled before they complete. Previously a canceled validation still kept its place in the internal work queue and held the associated response in memory until that queued work eventually ran, so memory could climb sharply under sustained load. The canceled work is now dropped as soon as the validation is canceled, releasing the memory it was holding.

- [CVE-2026-11605] Prevent excessive validation work from crafted negative responses. `5f1bb0aad09`

  A validating resolver could be made to perform a large amount of DNSSEC validation work in response to a single answer, consuming excessive CPU. A malicious authoritative server triggers this by returning a signed negative answer (NXDOMAIN or NODATA) padded with many denial-of-existence proof records, which the resolver continued to verify beyond its per-query validation limit. It now enforces that limit on negative answers and returns SERVFAIL once the limit is reached.

  Closes: https://gitlab.isc.org/isc-projects/bind9/-/work_items/4463

- [CVE-2026-11622] Prevent cache exhaustion under sustained attack. `410aa215eab`

  The cache memory can become exhausted with expired entries whose memory is not released due to a sustained attack on the same DNS name that prevents the cleanup. This has been fixed.

  Closes: https://gitlab.isc.org/isc-projects/bind9/-/work_items/4760

### 15.4.2 New Features

- Print OS platform in "named -V" `5bb30b1f993`

  The "running on" line emitted by *named -V* (as well as the startup log and *rndc status*, which share the same source) now appends the PRETTY_NAME value from /etc/os-release in parentheses after the uname output, e.g.:

  running on Linux x86_64 6.19.14-... (Fedora Linux 42 (Workstation Edition))

  This helps disambiguate environments where the kernel string is not a reliable indicator of the userspace, such as RHEL clones and containers whose kernel does not match the host OS.

  When /etc/os-release is absent, /usr/lib/os-release is tried as a fallback per the systemd os-release(5) specification. When neither is available or no PRETTY_NAME is found, the output is unchanged.

  Assisted-by: Claude:claude-opus-4-7 **[GL #5334] [GL !12319]**

### 15.4.3 Removed Features

- Remove the secondary validator in query.c. `1687a3d0b85`

  Previously, when the additional section of a response was being populated, if cached data was found with pending trust, it would be opportunistically validated. The code implementing this validation was not quite formally correct. Rather than fixing it, the code has been removed: RRsets with pending trust are now omitted from responses. **[GL #5966] [GL #5968] [GL #5972] [GL !12258]**

### 15.4.4 Feature Changes

- Rework isc_work as per-loop, per-lane cancelable worker threads. `51d67c173e9`

  Fold the libuv thread pool and the per-loop isc_helper threads into a single isc_work pool. Each (loop, lane) gets its own SPSC queue and worker, which drops the shared-queue contention, and the FAST/SLOW lanes keep short crypto tasks off the long blocking ones (zone dump/load, xfrin). isc_work jobs are now cancelable: isc_work_cancel() tombstones a still-queued job and its after_cb fires with ISC_R_CANCELED, so abandoned work can be dropped instead of run to completion. **[GL !12267]**

- Sync picohttpparser.c with upstream commit a875a01. `7b249a11e34`

  Synced with the h2o/picohttpparser upstream repository up to commit f4d94b48b31e0abae029ebeafcfd9ca0680ede58.

  This commit is just hygiene and consistency by keeping the vendored copy current. **[GL !12273]**

### 15.4.5 Bug Fixes

- Fix a bug in DNS UPDATE processing with inline-signing enabled. `ad4650c78fa`

  In rare cases the *named* process could terminate unexpectedly when processing authorized DNS UPDATE messages in quick procession which are updating a zone with inline-signing enabled. This has been fixed. **[GL #5816] [GL !12300]**

- Properly detect private records before copying. `5c153b85e54`

  We were triggering an assertion when trying to copy a private record to a buffer for modifying. Extend the private type detection and copy the contents after we have rejected invalid private records. **[GL #5857] [GL !12340]**

- Tighten referral DS acceptance. `ed27fb9524f`

  Named was accepting DS records for sibling zones when it shouldn't have. This has been fixed. **[GL #5870] [GL !12325]**

- Don't synthesize negative responses with pending NSEC. `86b87dfd290`

  If an NSEC record has not yet been validated and is cached with trust pending, don't use it to synthesize negative responses. **[GL #5872] [GL #5887] [GL #5977] [GL !12368]**

- Check that an NSEC signer is at or above the name to be validated. `2afca834141`

  Add a check that an NSEC record being used as a proof of nonexistence for a given name is not signed by a name lower in the DNS hierarchy than the one in question. **[GL #5876] [GL !12324]**

- Don't evict DNSSEC-validated cache data on a CD=1 NXDOMAIN. `3e6cef2ac66`

  When a client sent a query with the checking-disabled (CD) bit set and the answer was NXDOMAIN, the resolver cached that unvalidated negative response and discarded any DNSSEC-validated records it already held for the same name, even though the validated data was more trustworthy. A single such response - including a forged one - could flush validated records from the cache and force the resolver to fetch them again. The resolver now checks the trust level of the existing data first and leaves the cache unchanged when it is already validated. **[GL #5877] [GL !12344]**

- Fix a 'deny-answer-aliases' configuration bypass issue. `47d375b9e63`

  It was possible to use a maliciously crafted authoritative zone to make *named* resolver synthesize a `DNAME` "alias" that should have been rejected by the configured *deny-answer-aliases* option. This has been fixed. **[GL #5930] [GL !12245]**

- Reject external referrals from forwarders. `cdbc5be8b0c`

  Under *forward-first* policy in a forwarding zone BIND could accept NS above the forward zone apex from negative responses. This has been fixed.

  ISC would like to thank Qifan Zhang, of Palo Alto Networks, for the report. **[GL #5937] [GL !12278]**

- Fix a zone transfer over TLS (XoT) issue when using the opportunistic TLS mode. `40b64d13862`

  The *named* process, running as secondary DNS server, configured to transfer a zone from a primary server using an encrypted XoT transport in opportunistic TLS mode (i.e. without peer certificate/hostname validation) could terminate unexpectedly when the TLS ALPN negotiation with primary server was unsuccessful. This has been fixed. **[GL #5957] [GL !12242]**

- Unvalidated opt-out NSEC3 could be accepted in insecurity proof. `c3733e12f9e`

  When determining whether an insecure delegation is legitimate, NSEC3 opt-out records which had not yet passed validation could be used. This has been fixed. **[GL #5970] [GL !12358]**

- Check wildcard signer and NOQNAME signer match. `3621c8169bf`

  A positive wildcard answer, and the NSEC3 proof that the requested name doesn't exist in the zone, must both be from the same zone. Otherwise, an NSEC3 from an ancestor zone could be used to interfere with validation.

  We now retrieve the signer name from a wildcard response's signature. An NSEC3 record cannot be used as a NOQNAME proof for the wildcard unless it exactly matches the name one level above the NSEC3. **[GL #5971] [GL !12279]**

- Require secure trust for covering NSEC in the RBT cache. `b6bfa444e5b`

  The guard against synthesizing negative answers from a pending NSEC (#5977) was only added to the QP cache's find_coveringnsec(); the RBT cache database kept binding any covering NSEC regardless of trust. On builds configured with --with-cachedb=rbt this lets a piggy-backed, unvalidated NSEC drive a synthesized NXDOMAIN, reviving the cache poisoning that was fixed for the default cache.

  query_coveringnsec() only verifies the covering NSEC's trust on its NODATA path; the NXDOMAIN path relies on the database returning a secure record. Require the NSEC and its RRSIG to be dns_trust_secure in find_coveringnsec(), matching the QP cache. **[GL #5977] [GL !12374]**

- Check dns_rdata_fromstruct() return values. `416b0817d1b`

  In some functions implementing RFC 5011 key maintenance, the results of *dns_rdata_fromstruct()* were not checked. This has been fixed. **[GL #5982] [GL !12183]**

- Fix CNAME resolution failure caused by a cached SERVFAIL response. `b1e0a58f4f2`

  Under certain circumstances, a cached SERVFAIL response could incorrectly prevent successful resolution of a CNAME target. This could cause resolution failures to persist until the cached SERVFAIL entry expired, even when the CNAME target itself was otherwise resolvable. This issue has been fixed. **[GL #5983] [GL !12259]**

- [CVE-2026-13204] Prevent crash from malformed NSEC/NSEC3 response. `ad393b09232`

  An assertion could be triggered by an improperly signed NOQNAME proof. This has been fixed.

  ISC thanks Qifan Zhang of Palo Alto Networks for reporting the issue. **[GL #5985]**

- Reject unsupported RSA DNSKEY shapes during DNSSEC validation. `ea91570492c`

  An authoritative server publishing an RSA DNSKEY with an unusually large modulus or an exotic public exponent could make each DNSSEC signature check on a validating recursive resolver noticeably more expensive than for a normally sized key. Such DNSKEYs are now treated as invalid. **[GL #6008] [GL !12207]**

- Fix a bug in GeoIP2 string matching. `f8f5e6a9d77`

  When using GeoIP2 ACLs (see *acl*), *named* could incorrectly match a name using a sub-string instead of the full name match. This has been fixed. **[GL #6019] [GL !12235]**

- Fix DNS-over-HTTPS (DoH) quota configuration issue. `72d538fe269`

  The *http-listener-clients* and *http-streams-per-connection* configuration options could be truncated to smaller values (or to `0`, which means unlimited) when very big configuration values were used, which exceeded `65535`. As a note - it is very unlikely that such big values are used in production, and the default values for the affected options are `300` and `100`, correspondingly. This has been fixed. **[GL #6021] [GL !12231]**

- Fix invalid pointer release in JSON statistics-channel response. `5fe742e3fd1`

  Each response served on a JSON statistics endpoint released the wrong pointer to the JSON library after the response was sent: the response body string instead of the JSON document. With the current responses this does not crash named in practice, but the call is incorrect and can in principle corrupt memory. XML responses are not affected. **[GL #6024] [GL !12274]**

- Truncated reply to a TSIG query no longer stalls the resolver. `c922cabd46d`

  When an upstream server returned a truncated reply to a query that BIND had signed with TSIG, the resolver could keep waiting for a follow-up UDP packet that never arrived, so the query stalled until it hit resolver-query-timeout and the client received no answer. BIND now treats any reply it cannot authenticate as an immediate failure and returns SERVFAIL right away as a defense in depth. **[GL #6028] [GL !12310]**

- Ignore updates removing DNSKEY RRset with class ANY. `6a533ec622c`

  When a Dynamic Update is received that removes the `DNSKEY` (or `CDNSKEY`, or `CDS`) RRset, remove all records except the ones that are in use for signing for the zone. **[GL #6045] [GL !12230]**

- Fix a memory leak when updating a zone with more than 32 DNSSEC keys. `73440816e41`

  Applying changes to a signed zone — via DNS UPDATE or the inline-signing raw-to-secure sync — leaked the surplus keys when the zone's key directory held more than 32, slowly growing named's memory use. **[GL #6051] [GL !12332]**

- Fix the memory ordering in the adaptive read-write lock. `eff9a306d88`

  On hardware with a weak memory model, the internal read-write lock could briefly admit a reader and a writer at the same time, risking sporadic crashes or incorrect data. The reader/writer handshake now uses sequentially consistent ordering so the two can no longer overlap. **[GL #6060] [GL !12205]**

- Print the full OID in PRIVATEOID key comments. `87397c823dc`

  The OID in the "; alg = ..." comment of a PRIVATEOID key was truncated to sixteen characters. **[GL #6092] [GL !12378]**

- Correct locator decoding for NID, L64, and L32 records. `8a8cc34a504`

  NID, L64, and L32 records were decoded incorrectly when converted into their parsed structures, because the preference field was not skipped before the locator. **[GL #6097] [GL !12349]**

- Fix a possible crash when cleaning up a view's caches. `51d7456399d`

  In rare cases named could crash while a view was being removed, for example during reconfiguration or shutdown, as its internal caches were torn down. This has been fixed. **[GL #6119] [GL !12190]**

- Preserve the request message across async SIG(0) processing. `d6a4aa5ce50`

  For SIG(0)-signed requests, view matching is offloaded and the request is finished asynchronously from ns_client_request_continue(), which passes client->inner.buffer to dns_dt_send(). That buffer aliases the network manager's receive buffer, only valid during the read callback, so it may already be freed and reused, producing garbage dnstap frames (e.g. the "upforwd" sig0-over-DoT test fails with UQ=0).

  Copy the request message when entering async mode and reference the copy, freeing it in ns__client_reset_cb().

  Assisted-by: Claude:claude-opus-4-8 **[GL #6139] [GL !12200]**

- Ignore 0-byte reads in the TCP read callback. `8588ffec7de`

  Callbacks for libuv stream reads do not signal zero-length reads as a failure signal but rather as EAGAIN/EWOULDBLOCK. This can trigger an assertion when a zero-length read is pushed onto a PROXYv2 endpoint that has not yet processed the headers as it expects a non-NULL region of positive length. **[GL #6140] [GL !12291]**

- Include the IPv6 address's brackets in the parsed URL/URI host. `9baad48be06`

  The brackets are not included in the host component of the parsed URL/URI. Change the parser to include the brackets. **[GL #6147] [GL !12362]**

- Improve the input validation of the isc_url_parse() function. `f4c8f08042a`

  The isc_url_parse() function failed to check the input buffer's length and assumed that it can't be bigger than UINT16_MAX, because both the 'off' and 'len' fields of the 'isc_url_parser_t' structure are uint16_t.

  Add a check to not accept a buffer longer than 8192 octets. **[GL #6150] [GL !12361]**

- Only print per zone glue stats when query statistics is full. `097e530348f`

  The code printing query statistics was ignoring the zone-statistics option. This has been fixed. **[GL #6164] [GL !12263]**

- CDS/CDNSKEY records were not removed when re-configuring the server. `20db02121f7`

  When on an `rndc reconfig` the DNSSEC policy changes such that it changes the expected `CDNSKEY` and/or `CDS` records in the zone, the RRset should be updated accordingly. This did not happen when removing digests from the configuration, or setting *cdnskey no;*. This has been fixed. **[GL #6166] [GL !12284]**

- Fix a crash when querying an empty non-terminal in a wildcard zone in RBTDB. `b8e5801ec5f`

  A query for an empty non-terminal in a wildcard zone served from the RBT zone database could abort named with an assertion failure. It now returns the correct NODATA answer. **[GL #6170] [GL !12299]**

- Stop reusing outgoing TCP connections the peer has already closed. `fb0ae8a5d6b`

  `named` could hand a new query an idle forwarder/upstream TCP or TLS connection that the peer had already closed, causing the query to fail (and CLOSE-WAIT sockets to pile up). Idle reused connections are now watched, so a

close is noticed and the connection is dropped instead of reused. A new `tcp-reuse-timeout` option controls how long an idle outgoing connection is kept open for reuse (default 5 seconds). **[GL #6171] [GL !12317]**

- Fix DNSSEC validation failures for names under an apex DNAME. `cd006ecf235`

  DNSSEC validation could fail with SERVFAIL for names covered by a DNAME at the apex of a signed zone, unless the zone's keys were already validated in the cache. This regression was introduced by the recent fix for resolver stalls on CNAME responses to DS queries. **[GL #6176] [GL !12356]**

- Avoid needless queries when the DNSSEC validation limit is reached. `45a3e5b8561`

  A validating resolver could send one needless upstream query after reaching its per-fetch DNSSEC validation limit. It now stops immediately.

- Avoid writing through a const pointer in render_xsl() `348e5a9ee54`

  render_xsl() served the static XSL stylesheet by casting away the const qualifier of xslmsg and handing the pointer to isc_buffer_reinit():

  p = UNCONST(xslmsg); isc_buffer_reinit(b, p, strlen(xslmsg));

  isc_buffer_reinit() copies any pre-existing buffer content into the new base with memmove(), so the call would write into xslmsg, which is a 'const char[]' living in read-only memory. This is safe today only because the supplied bodybuffer is always freshly initialized with length 0, so the memmove() never runs -- a fragile, action-at-a-distance invariant that GCC's -fanalyzer flags as a write to a const object (-Wanalyzer-write-to-const).

  Use isc_buffer_constinit(), the primitive intended for pointing a buffer at constant data: it goes through isc_buffer_init() and never writes to the base. This drops the UNCONST cast, keeps xslmsg in read-only memory, and silences the analyzer warning.

  Assisted-by: Claude:claude-opus-4-8 **[GL !12309]**

- Don't serve a stale CNAME or record when fresh data of the other exists. `1cf6838c271`

  When a cached name held both a CNAME and records of another type — one stale, the other still fresh — named with serve-stale could return the expired set instead of the fresh one, in either direction. It now prefers whichever is fresh. **[GL !12305]**

- Fix compatibility with OpenSSL 1.0.2u. `11fcd8206ab`

  BIND failed to build with OpenSSL 1.0.2u. Add the missing header file includes. **[GL !11897]**

- Only update the global tid_count once. `746d9bfd3a8`

  Skip updating the *tid_count* value on repeated calls to prevent ThreadSanitizer 'data race'. **[GL !12188]**

## 15.5 BIND 9.20.25

> **Note**
>
> The BIND 9.20.25 release was withdrawn after the discovery of a regression in a security fix in it during pre-release testing.

## 15.6 BIND 9.20.24

### 15.6.1 Security Fixes

- Fix DNS64 owner case after DNAME restart. `4de2229364`

When BIND 9 is configured to use DNS64 and encounters a DNAME redirect, it could end up using freed memory for the DNS response owner name. This caused the response to contain corrupted data. This fix ensures the correct owner name is used when constructing the synthesized response after a DNAME redirect.

ISC thanks Qifan Zhang of Palo Alto Networks for reporting the issue. **[GL #5934]**

### 15.6.2 New Features

- Enable PR-Agent reviews on merge requests. `46e4c236a3`

  Adds a CI job that runs PR-Agent against each merge request opened from the canonical repository, posting an automated review and code-improvement suggestions as MR comments. The job is gated to same-project source branches so the OpenAI key and personal access token are not exposed to fork pipelines. **[GL !12034]**

### 15.6.3 Removed Features

- Remove ineffective TCP fallback after repeated UDP timeouts. `eb13adcb47`

  When an authoritative server failed to respond to two consecutive UDP queries, named marked the next retry as TCP but still sent it over UDP, producing misleading dnstap records. The ineffective retry path has been removed; a corrected TCP fallback will be restored in future BIND 9 versions. **[GL #5529] [GL !12049]**

- Remove useless PR-Agent jobs. `8851b279d0`

  The experiment was a failure, the PR-Agent doesn't send a full context to the AI Agents and the results are abysmal because of that. **[GL !12120]**

### 15.6.4 Feature Changes

- Fall back to TCP on a UDP response with a mismatched query id. `b2367aaea2`

  BIND used to wait silently for the correct DNS message id on a UDP fetch even after receiving a response from the expected server with the wrong id, leaving room for off-path spoofing attempts to keep guessing within that window. The resolver now retries the fetch over TCP on the first such response, and a new MismatchTCP statistics counter tracks how often the fallback fires. **[GL #5449] [GL !12025]**

- Limit the number of glue records cached from a referral. `eb401f6b92`

  When a delegation response contained many glue addresses per listed nameserver, all of them were cached without a per-nameserver bound, inflating resolver cache memory beyond what resolution could ever use. The cache now keeps at most 20 IPv4 and 20 IPv6 glue addresses per nameserver from a delegation. **[GL #5701] [GL !11972]**

- Fix a resolver stall on a CNAME response to a DS query. `1407f48670`

  A validating resolver could stall for about twelve seconds and then return SERVFAIL when an authoritative server answered a DS query with a CNAME. Such responses are now rejected promptly, so the query fails fast instead of hanging. **[GL #5878] [GL !12147]**

- Named could crash on concurrent TKEY DELETE for the same key. `dd02abde67`

  On a server configured with tkey-gssapi-keytab (or tkey-gssapi-credential), an authenticated peer could crash named by sending two TKEY DELETE requests for the same dynamic key in rapid succession. This has been fixed. **[GL #6001] [GL !12042]**

### 15.6.5 Bug Fixes

- The resolver now removes other RRsets at the same name when caching a CNAME. `1547447491`

  When an RRset is in stale cache, and the authoritative server changes the record type to CNAME, the resolver fails to refresh the stale cache. This has been fixed. **[GL #5302] [GL !12040]**

- Fix nxdomain-redirect combined with dns64. `95274e9455`

  When a resolver was configured with both *nxdomain-redirect* and *dns64* in the same view, an AAAA query for a nonexistent name could abort *named*. The combination failed whenever the redirect zone held A records but no AAAA records. The server now serves the empty AAAA response from the redirect zone as-is, instead of attempting DNS64 synthesis on top of it. **[GL #5789] [GL !12123]**

- Clear REDIRECT flag when it isn't needed. `86bb27060c`

  When *nxdomain-redirect* is in use, and a recursive query is used to get the redirected answer, a flag is set to distinguish it from a normal recursive response. Previously, that flag was left set afterward, which could trigger an assertion if a normal recursive query was sent later on behalf of the same client: for example, because the *filter-aaaa* plugin was in use. This has been fixed. **[GL #5936] [GL !12076]**

- Fix data race during rndc dumpdb or zone load. `947e5c7983`

  'rndc dumpdb' against a server with zones, and async zone load, had a timing window where the operation's completion could fire before the server had finished registering the operation, occasionally leading to a possible crash. The completion is now delivered after the registration is in place. **[GL #5952] [GL !12021]**

- Bound memory use during incoming zone transfers. `5d7f241fdf`

  During an incoming zone transfer, an optimization could let the batch of pending records grow without bound for a large zone, raising memory usage. It gave no measurable performance benefit, so it has been removed. **[GL #5958] [GL !12142]**

- Disable output escaping in bind9.xsl. `b514e663eb`

  The statistics charts where not displaying on some browsers. This has been fixed. **[GL #5990] [GL !12019]**

- Fix crash on badly configured secondary signer. `edc1ef084f`

  A badly configured secondary signer that was missing the 'file' entry caused the server to crash, rather than to reject the configuration. This has been fixed. **[GL #5993] [GL !12112]**

- Avoid named assertion failure during parent-NS lookups when none exist. `5c0c4786dd`

  Configuring the root zone as a signed primary with parental agents (or with notify-on-cds-changes) caused named to exit on an internal assertion as soon as the DS-publication machinery tried to look up the parent NS RRset — the root has no parent. The lookup is now short-circuited cleanly.

  Similar, a zone with no NS records in the parent caused named to exit in the same way. **[GL #5910] [GL #5996] [GL !12053]**

- Reject RRSIG records covering meta-types. `7517e39504`

  A recursive resolver could accept and cache an RRSIG record whose Type-Covered field names a meta-type (ANY, AXFR, IXFR, MAILA, MAILB), even though no real RRset of those types ever exists. Such records are now rejected by the DNS message parser. **[GL #6002] [GL !12051]**

- Validate nsec3hash arguments instead of relying on atoi() `a59080c053`

  The nsec3hash tool parsed its algorithm, flags, and iterations arguments with atoi(), then range-checked the result. For values that overflow int during digit-by-digit accumulation, atoi() is undefined; in practice on musl libc the modular wrap leaves n == 0, which silently passes the "iterations > 0xffffU" check. On Alpine Linux this made nsec3hash succeed with iterations treated as 0 for inputs like 4294967296 (2^32).

  The latent bug only surfaced when the recent image rebuild pulled in Hypothesis 6.152.9 (2026-05-19), which unified the distribution used for bounded and unbounded integers() strategies. The new smoother distribution explores the 2^32 boundary on unbounded ranges like integers(min_value=65536); earlier versions did not reach there, so test_nsec3hash_too_many_iterations only started failing on Alpine after the image refresh.

Replace the three atoi() calls with isc_parse_uint8 / isc_parse_uint16, which uniformly reject overflow, trailing garbage, leading sign, and non-numeric input across libc implementations. As a side effect, error messages now include the offending argument and a specific reason ("out of range" vs "not a valid number").

Assisted-by: Claude:claude-opus-4-7 **[GL #6013] [GL !12074]**

- Fix wrong variable in named_server_sync() log message. `09f9889ef6`

  named_server_sync() logged isc_result_totext(result) but returns tresult. The loop accumulates errors into tresult, so result only holds the last iteration's value. If the last view succeeded but an earlier one failed, the log would incorrectly say "success". **[GL !12156]**

- Refine resolver fetch loop detection. `787faa02a7`

  The resolver's fetch loop detection now triggers only when a new fetch would join an already in-flight fetch that is also one of its own ancestors, which is the actual loop condition. Previously the check ran against the original request before the fetch was set up. **[GL !12146]**

## 15.7 BIND 9.20.23

### 15.7.1 Security Fixes

- Fix outgoing zone transfers' quota issue. `1006b044b7`

  Unauthorized clients could consume outgoing zone transfers quota and block authorized zone transfer clients. This has been fixed. **[GL #3589]**

- [CVE-2026-3592] Limit resolver server list size. `c3f3879560`

  When resolving a domain with many nameservers that share overlapping IP addresses (e.g., 10 NS records all pointing at the same set of addresses), BIND could previously waste time querying duplicate addresses and build up excessively large server lists. Deduplicate addresses in the resolver's server list so that each unique IP is only queried once per resolution attempt, regardless of how many NS records point to it and cap the number of addresses stored per nameserver name to 6 (combined A and AAAA), preventing memory and CPU overhead from domains with unusually large NS/glue sets. **[GL #5641]**

- [CVE-2026-3039] Fix GSS-API resource leak. `92d5c60855`

  Fixed a memory leak where each GSS-API TKEY negotiation leaked a security context inside the GSS library. An unauthenticated attacker could exhaust server memory by sending repeated TKEY queries to a server with tkey-gssapi-keytab configured. The leaked memory was allocated by the GSS library, bypassing BIND's memory accounting.

  Multi-round GSS-API negotiation (GSS_S_CONTINUE_NEEDED) is now rejected, as BIND never supported it correctly and Kerberos/SPNEGO completes in a single round.

  Also implemented missing RFC 3645 requirement: the client now verifies that mutual authentication and integrity flags are granted by the GSS-API mechanism (Section 3.1.1). **[GL #5752]**

- [CVE-2026-5950] Avoid unbounded recursion loop. `568be408bc`

  A bug during bad server handling could cause the resolver to enter an infinite loop, continuously sending queries to an upstream server with no exit condition, until the resolver query timeout was hit. This has been fixed.

  ISC would like to thank Billy Baraja (BielraX) for bringing this issue to our attention. **[GL #5804]**

- [CVE-2026-5947] Fix crash in resolver when SIG(0)-signed responses are received under load. `9831f41894`

  A resolver could crash when handling a SIG(0)-signed response if the matching client query was cancelled while signature verification was still in progress — for example, when the recursive-clients quota was exhausted. This has been fixed. **[GL #5819]**

- [CVE-2026-3593] Add system test for HTTP/2 SETTINGS frame flood. `3be272e26d`

  A use-after-free vulnerability in the DNS-over-HTTPS implementation could cause named to crash when a client sends a flood of HTTP/2 SETTINGS frames while a DoH response is being written. This affects servers with DoH (DNS-over-HTTPS) enabled.

  ISC would like to thank Naresh Kandula Parmar (Nottiboy) for reporting this.

  For: #5755

- [CVE-2026-5946] Disable recursion, UPDATE, and NOTIFY for non-IN views. `014be8be87`

  Recursion, dynamic updates (UPDATE), and zone change notifications (NOTIFY) are now disabled for views with a class other than IN (such as CHAOS or HESIOD); authoritative service for non-IN zones (e.g. version.bind in class CHAOS) continues to work as before. Servers configured with recursion yes in a non-IN view will log a warning at startup, and named-checkconf flags the same condition. UPDATE and NOTIFY messages that specify the meta-classes ANY or NONE in the question section are now rejected with FORMERR.

  This addresses a set of closely related security issues collectively identified as CVE-2026-5946. ISC would like to thank Mcsky23 for bringing these issues to our attention.

### 15.7.2 Removed Features

- Remove obsolete KEY record EXTENDED flag deprecated by RFC 3445. `99c226576a`

  KEY resource records originally defined EXTENDED flag that was removed by RFC 3445 back in 2002. BIND still carried code to parse and emit it, including the additional two-octet flags field that followed when the EXTENDED bit was set. That handling has been removed and the affected bit positions are now reserved.

  Dropping the extended-flags handling also eliminates a possible crash that could be reached when signing a zone containing an invalid key. **[GL #5900]**

  Partial backport of MR !11961 **[GL !11962]**

### 15.7.3 Feature Changes

- Revert isdelegation() to return boolean value again. `6d89bfdf03`

  **[GL #5838] [GL !11802]**

- Fix CPU spikes and slow queries when cache approaches memory limit. `e21ae6358a`

  When the cache grew close to the configured max-cache-size, every subsequent entry triggered all worker threads to run cache cleanup at once, causing CPU spikes and a drop in query throughput. Cleanup is now spread probabilistically across inserts as memory approaches the limit, so the work is distributed evenly instead of piling up at the threshold.

- Fix off by one error in dnssec-ksr sign. `819df0d19e`

  If the inception time of the signature is exactly equal to the inactive time of the key, add the signature. **[GL !11795]**

### 15.7.4 Bug Fixes

- Check validator name when adding EDE text. `b6c3390aea`

  When a validator is being shut down, the associated name *val->name* is set to NULL. This could cause a crash if a worker thread subsequently added an EDE code with *val->name* in the extra text.

  *validator_addede()* now checks whether the name is NULL before trying to add it to the extra text. **[GL #5613] [GL !11977]**

- Use the zone file's basename as origin in DNSSEC tools. `097c14da45`

  In *dnssec-signzone* and *dnssec-verify*, when the zone origin is not specified using the *-o* parameter, the default behavior is to try to sign using the zone's file name as the origin. So, for example, *dnssec-signzone -S example.com* will work, so long as the file name matches the zone name.

  This now also works if the zone is in a different directory. For example, *dnssec-signzone -S zones/example.com* will set the origin value to *example.com*. **[GL #5678] [GL !11784]**

- Fix a possible race condition during zone transfers. `a48b287d9f`

  The *named* process could terminate unexpectedly when processing an IXFR message during a zone transfer. This has been fixed. **[GL #5767] [GL !11799]**

- Make BIND9 compatible with OpenSSL 4. `8242105d5d`

  OPENSSL_cleanup() in OpenSSL 4 doesn't free the memory, and that is not compatible with BIND 9's memory leak detection code. Don't use custom allocation/deallocation functions for OpenSSL's internal memory management.

  See https://github.com/openssl/openssl/pull/29721 **[GL #5808] [GL !11896]**

- Fix named crash when processing SIG records in dynamic updates. `9e34ef0f7e`

  Previously, *named* could abort if a client sent a dynamic update containing a SIG record (the legacy signature type) to a zone configured with an update-policy. The function *dns_db_findrdataset* had an incorrect requirements prerequisite that prevented SIG records being looked up, which was triggered as part of processing an UPDATE request and could be triggered remotely by any client permitted to send updates. This has been fixed by ensuring that SIG records are handled consistently with RRSIG records during update processing. **[GL #5818] [GL !11876]**

- Fix crash in resolver when SIG(0)-signed responses are received under load. `bbe0b9b8f6`

  A resolver could crash when handling a SIG(0)-signed response if the matching client query was cancelled while signature verification was still in progress — for example, when the recursive-clients quota was exhausted. This has been fixed. **[GL #5819]**

- Fix zone verification of NSEC3 signed zones. `de4a9b4fa6`

  Previously, when computing the compressed bitmap during verification of an NSEC3-signed zone, an undersized buffer was used that resulted in an out-of-bounds write if there were too many active windows in the bitmap. This impacted mirror zones which are NSEC3-signed, *dnssec-signzone* and *dnssec-verifyzone*. This has been fixed. **[GL #5834] [GL !11833]**

- Prevent a crash when using both dns64 and filter-aaaa. `ddcacbc5a8`

  An assertion failure could be triggered if both *dns64* and the *filter-aaaa* plugin were in use simultaneously. This happened if the plugin triggered a second recursion process, which then attempted to store DNS64 state information in a pointer that had already been set by the original recursion process. This has been fixed. **[GL #5854] [GL !11967]**

- Remove unnecessary dns_name_free call. `35d94fffb0`

  When processing a catalog zone member's primaries definition and there is a TXT record containing an invalid name TSIG key name, dns_name_free was incorrectly called triggering an assertion. This has been fixed. **[GL #5858] [GL !11848]**

- Tidy up the cleanup path in check_signer() `cf517f73d5`

  When check_signer() processed a DNSKEY whose public-key data could not be parsed, the early return on the parse error skipped the cleanup of the cloned signature rdataset. In every code path that currently reaches this function the cloned rdataset holds no resources, so no memory was actually leaked, but the cleanup is restructured so the parse and the iteration cannot diverge again. **[GL #5869] [GL !11957]**

- Prevent malicious DNSSEC zones from exhausting validator CPU. `c425827743`

  A DNSSEC-signed zone could publish a DNSKEY with an unusually large RSA public exponent and force any validator resolving names in that zone to spend disproportionate CPU verifying signatures. The validator now rejects such DNSKEYs, matching the limit already applied to keys read from files or HSMs. **[GL #5881] [GL !11923]**

- Fix inverted gethostname() check in rndc status. `5ede4a87eb`

  The replacement of named_os_gethostname() with raw gethostname() inverted the success check: the "localhost" fallback runs on success, and on failure the uninitialized hostname buffer is read by snprintf(), leaking stack memory via the rndc status reply. **[GL #5889] [GL !11881]**

- Fix rndc-confgen aborting on HMAC-SHA-384/512 keys above 512 bits. `7e1eace6cd`

  *rndc-confgen -A hmac-sha384* and *-A hmac-sha512* documented a *-b* range of 1..1024, but any value above 512 aborted on hardened builds instead of producing a key. The full advertised range now works. **[GL #5903] [GL !11910]**

- Prevent crafted queries from degrading RRL performance. `bf4cdca7e9`

  With response rate limiting enabled, an attacker sending queries from many spoofed source addresses could steer entries into the same slot of the internal rate-limit table and slow down query processing on the affected server. The table now uses a per-process keyed hash so the placement of entries cannot be predicted or influenced from the network. **[GL #5906] [GL !11952]**

- Fix swapped arguments in redirect2() single-label branch. `3728b405ea`

  On a recursive resolver with nxdomain-redirect configured, an NXDOMAIN result for a query whose qname is the root could corrupt the view's nxdomain-redirect target, after which the redirect feature stopped working for every subsequent query in that view until named was restarted. **[GL #5908] [GL !11913]**

- Free per-command rndc state when response serialisation fails. `070b394f53`

  When isccc_cc_towire failed while building an rndc reply, control_respond returned without releasing the per-command request, response, HMAC secret copy, and text buffer. They were eventually freed when the connection closed, but until then the HMAC key copy stayed in named's memory. The failure path now goes through the same cleanup label as every other error. **[GL #5913] [GL !11919]**

- Prevent rare named crash when notifies are cancelled. `49509dcbae`

  Under heavy load, named could occasionally crash when a queued outbound notify or zone refresh was cancelled at the moment it was being sent — for example, while a zone was being reloaded or removed. The race that caused the crash is now prevented. **[GL #5915] [GL !11922]**

- Stop delv from aborting on a malformed query name. `ca8315bb4d`

  delv aborts with SIGABRT instead of exiting cleanly when given a query name that fails wire-format conversion (e.g. a label longer than 63 octets). After this change delv prints the parse error and exits with a normal failure code. **[GL #5916] [GL !11927]**

- Fix a crash when reconfiguring while an NTA is being rechecked. `971ca4df1a`

  When named was reconfigured or shut down while a negative trust anchor was being rechecked against authoritative servers, the in-flight recheck could outlive the view that owned it and cause *named* to crash. This has been fixed. **[GL #5938] [GL !11966]**

- Fix a bug in allow-query/allow-transfer catalog zone custom properties. `e962fd459e`

  The *named* process could terminate unexpectedly when processing a catalog zone with an invalid `allow-query` or `allow-transfer` custom property (i.e. having a non-APL type) coexisting with the valid property. This has been fixed. **[GL #5941] [GL !11975]**

- Fix a stack use-after-free in qpzone. `ddea991c07`

  In previous_closest_nsec(), a new qpreader was opened to search the NSEC tree. It was possible for that to be used to update a QP iterator object owned by the caller, and then be destroyed when the function returned.

  This has been addressed by having the caller open the NSEC qpreader instead. **[GL #5942] [GL !11956]**

- Fix a memory leak issue in the catalog zones. `5fcb6d8809`

  The *named* process could leak small amounts of memory when processing a catalog zone entry which had defined custom primary servers with TSIG keys using both the regular `primaries` custom property syntax and the legacy alternative syntax (`masters`) at the same time. This has been fixed. **[GL #5943] [GL !11973]**

- Fix suppressed missing-glue check in named-checkzone. `dc5eb3fe25`

  named-checkzone and named-checkconf -z silently skipped the missing-glue check for any NS name that had already triggered an extra-AAAA-glue warning, so zones missing required A glue could pass validation and be deployed with broken delegations. **[GL !11905]**

- Implement seamless outgoing TCP connection reuse. `eb117e16b9`

  The resolver can and will reuse outgoing TCP connections to the same host, as recommended by RFC 7766. This prevents a whole class of attacks that abuse the fact that establishing a TCP connection is expensive and it is fairly easy to deplete the outgoing TCP ports by putting them into TIME_WAIT state.

  The number of pipelined queries per connection is capped at 256 to limit the impact of a connection drop. **[GL !11846]**

- Pass empty string instead of NULL to ns_client_dumpmessage() `24cdf8c096`

  Pass "" instead of NULL to ns_client_dumpmessage() to get the log message printed.

- Reject record sets too large to serve in DNS. `933a8de056`

  When BIND was asked to store a record set whose total size exceeds what fits in a DNS message, it would allocate memory and build the structure, then fail later at response time. Such oversized record sets are now rejected at the time of storage with an error, avoiding wasted work on data that can never be served. **[GL !11964]**

## 15.8 BIND 9.20.22

### 15.8.1 Security Fixes

- Fix crash when reconfiguring zone update policy during active updates. `ee7832ae583`

  Fixed a crash that could occur when running rndc reconfig to change a zone's update policy (e.g., from allow-update to update-policy) while DNS UPDATE requests were being processed for that zone.

  ISC would like to thank Vitaly Simonovich for bringing this issue to our attention. **[GL #5817] [GL !11738]**

### 15.8.2 New Features

- Add MOVE_OWNERSHIP() macro for transferring pointer ownership. `13a656f79aa`

  A helper macro that returns the current value of a pointer and sets it to NULL in one expression, useful for transferring ownership in designated initializers. **[GL !11736]**

### 15.8.3 Feature Changes

- Exclude named.args.j2 and system test README files from license header checks. `d65e3922bbb`

  Exclude named.args.j2 files from license header checks so named.args can be generated from Jinja templates. Also exclude system test README files from the license header checks. **[GL !11696]**

- Skip cache flush ordering on NTA expiry. `5f97f5b0501`

  dns_view_flushnode() was called in the delete_expired() async callback, which runs after the query that detected the NTA expiry. This created a race: the query would proceed with stale cached data from the NTA period before the flush had a chance to run, resulting in transient SERVFAIL with EDE 22 (No Reachable Authority).

  Skip dns_view_flushnode() in the older branches as the solutions for older branches are too complicated and this was not a critical bug.

  Also simplify the expiry comparison in delete_expired() to a direct pointer comparison (nta == pval) instead of comparing expiry timestamps. **[GL !11730]**

- Use underscore for system test names. `d270709b499`

  Change the convention for system test directory names to always use an underscore rather than a hyphen. Names using underscore are valid python package names and can be used with standard *import* facilities in python, which allows easier code reuse. **[GL !11711]**

### 15.8.4 Bug Fixes

- Fix intermittent named crashes during asynchronous zone operations. `ac042af5766`

  Asynchronous zone loading and dumping operations occasionally dispatched tasks to the wrong internal event loop. This threading violation triggered internal safety assertions that abruptly terminated named. Strict loop affinity is now enforced for these tasks, ensuring they execute on their designated threads and preventing the crashes. **[GL #4882] [GL !11684]**

- Count temporal problems with DNSSEC validation as attempts. `e4399fc6b26`

  After KeyTrap, the temporal DNSSEC were originally hard errors that caused validation failures even if the records had another valid signature. This has been changed and the RRSIGs outside of the inception and expiration time are not counted as hard errors. However, these errors are not even counted as validation attempts, so excessive number of expired RRSIGs would cause some non-cryptograhic extra work for the validator. This has been fixed and the temporal errors are correctly counted as validation attempts. **[GL #5760] [GL !11763]**

- Clear errno correctly. `0de8a660117`

  Zero errno before calling strtol. **[GL #5773] [GL !11703]**

- Fix a possible deadlock in RPZ processing. `a2bedda1321`

  The *named* process could hang when processing a maliciously crafted update for a response policy zone (RPZ). This has been fixed. **[GL #5775] [GL !11687]**

- Fix use-after-free in xfrin_recv_done. `46099d2d9af`

  Move the LIBDNS_XFRIN_RECV_DONE probe execution before dns_xfrin_detach in xfrin_recv_done.

  Previously, dns_xfrin_detach was called before the trace probe, which could free the xfr object. Because the accessed member xfr->info is an embedded array, the expression evaluates via pointer arithmetic rather than a direct memory dereference. Although this prevents a reliable crash in practice, it technically remains a use-after-free issue. Reorder the statements to ensure the transfer context is fully valid when the probe executes. **[GL #5786] [GL !11694]**

- Backport test for update-policy per-type max quota bypass via crafted UPDATE messages. `545ce3ae224`

  An authenticated DDNS client could bypass update-policy per-type record limits (e.g. TXT(3)) by including padding records in the UPDATE message that are silently skipped during processing in the main branch.

  As BIND 9.20 is not affected, only backport the test. **[GL #5799] [GL !11760]**

- Fix a crash triggered by rndc modzone on zone from configuration file. `6d9482bd6b8`

  Calling *rndc modzone* on a zone that was configured in the configuration file caused a crash. This has been fixed.

ISC would like to thank Nathan Reilly for reporting this. **[GL #5800] [GL !11698]**

- Fix the processing of empty catalog zone ACLs. `ce365083d9d`

  The *named* process could terminate unexpectedly when processing a catalog zone ACL in an APL resource record that was completely empty. This has been fixed. **[GL #5801] [GL !11759]**

- Fix OpenSSL 4 compatibility issue when calling X509_get_subject_name() `1d43bf8263f`

  Starting from OpenSSL 4 the the X509_get_subject_name() function returns a 'const' pointer to a name instead of a regular pointer. Duplicate the name before operating on it, then free it. **[GL #5807] [GL !11692]**

- Take dns_dtenv_t reference before an async function call. `be7b811fffc`

  A 'dns_dtenv_t' pointer is passed to an async function without taking a reference first, which can potentially cause a use-after-free error. Take a reference, then detach in the async function. **[GL #5820] [GL !11714]**

- Fix a crash triggered by rndc modzone on zone that already existed in NZF file. `46dbcd7c9a5`

  Calling *rndc modzone* didn't work properly for a zone hat was configured in the configuration file. It could crash if BIND 9 was built without LMDB or if there was already an NZF file for the zone. In addition, *rndc modzone* failed in subsequent attempts. These problems are now fixed. **[GL #5826] [GL !11743]**

- Fix couple of reference counting bugs. `fc5e26cfc9f`

  Fix missing detach/free on error paths. **[GL !11691]**

- Fix data race in server round-trip time tracking. `31cbfc9fb36`

  The SRTT (Smoothed Round-Trip Time) update for remote servers was not atomic — concurrent callers could each read the same value and one update would be silently lost. Additionally, the aging decay applied once per second could run multiple times if several threads entered the function simultaneously.

  Use compare-and-swap loops for the SRTT update and for the aging timestamp to ensure no updates are lost. **[GL !11723]**

- Fix data race on fctx->vresult in validated() `996c66aee7a`

  Move the write to fctx->vresult after LOCK(&fctx->lock). The field was being set before acquiring the lock, but dns_resolver_logfetch() reads it under the same lock from another thread. **[GL !11721]**

- Fix isc_buffer_init capacity mismatch in DoH data chunk callback. `f0a2b07359c`

  isc_buffer_init() is given MAX_DNS_MESSAGE_SIZE (65535) as capacity but only h2->content_length bytes are allocated. This makes the buffer believe it has more space than actually allocated. A secondary bounds check (new_bufsize <= h2->content_length) prevents actual overflow, but the buffer invariant is violated.

  Pass h2->content_length as the capacity to match the allocation. **[GL !11709]**

- Fix memory leak in dns_catz_options_setdefault() for zonedir. `1844afec7ba`

  When defaults->zonedir is set, opts->zonedir is unconditionally overwritten without freeing the previous value. This leaks memory on every catalog zone update when zonedir defaults are configured.

  Free the existing opts->zonedir before replacing it. **[GL !11685]**

- Fix potential resource during resolver error handling. `6a32c1acdc7`

  Under specific error conditions during query processing, resources were not being properly released, which could eventually lead to unnecessary memory consumption for the server. The a potential resource leak in the resolver has been fixed. **[GL !11706]**

- Fix resquery reference imbalance on TCP connect failure. `b69bbf2e0ed`

  In fctx_query(), resquery_ref(query) is called before dns_dispatch_connect() in anticipation of the resquery_connected() callback consuming the reference. When dns_dispatch_connect() fails synchronously on TCP

(e.g. from dns_transport_get_tlsctx() failing in tcp_dispatch_connect()), the connect callback is never scheduled, so the extra reference is never consumed. This has been fixed. **[GL !11656]**

## 15.9 BIND 9.20.21

### 15.9.1 Security Fixes

- [CVE-2026-1519] Fix unbounded NSEC3 iterations when validating referrals to unsigned delegations. `5af03a06066`

  DNSSEC-signed zones may contain high iteration-count NSEC3 records, which prove that certain delegations are insecure. Previously, a validating resolver encountering such a delegation processed these iterations up to the number given, which could be a maximum of 65,535. This has been addressed by introducing a processing limit, set at 50. Now, if such an NSEC3 record is encountered, the delegation will be treated as insecure.

  ISC would like to thank Samy Medjahed/Ap4sh for bringing this vulnerability to our attention. **[GL #5708]**

- [CVE-2026-3104] Fix memory leaks in code preparing DNSSEC proofs of non-existence. `13215b9cbbf`

  An attacker controlling a DNSSEC-signed zone could trigger a memory leak in the logic preparing DNSSEC proofs of non-existence, by creating more than *`max-records-per-type`* RRSIGs for NSEC records. These memory leaks have been fixed.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5742]**

- [CVE-2026-3119] Prevent a crash in code processing queries containing a TKEY record. `308baa89105`

  The *`named`* process could terminate unexpectedly when processing a correctly signed query containing a TKEY record. This has been fixed.

  ISC would like to thank Vitaly Simonovich for bringing this vulnerability to our attention. **[GL #5748]**

- [CVE-2026-3591] Fix a stack use-after-return flaw in SIG(0) handling code. `aaaae0fd97e`

  A stack use-after-return flaw in SIG(0) handling code could enable ACL bypass and/or assertion failures in certain circumstances. This flaw has been fixed.

  ISC would like to thank Mcsky23 for bringing this vulnerability to our attention. **[GL #5754]**

### 15.9.2 Bug Fixes

- Resolve "key defined in view is not found" `819fe452745`

  Commit *2956e4fc* hardened the *key* name check when used in *primaries* to reject the configuration if the key was not defined, rather than simply checking whether the key name was correctly formed.

  However, the key name check didn't include the view configuration, causing keys not to be recognized if they were defined inside the view and not at the global level. This regression is now fixed. **[GL #5761] [GL !11613]**

## 15.10 BIND 9.20.20

### 15.10.1 Feature Changes

- Record query time for all dnstap responses. `f4fdcee03f1`

  Not all DNS responses had the query time set in their corresponding dnstap messages. This has been fixed. **[GL #3695] [GL !11534]**

- Implement Fisher-Yates shuffle for nameserver selection. `dd453590a0e`

Replace the two-pass "random start index and wrap around" logic in fctx_getaddresses_nameservers() with a statistically sound partial Fisher-Yates shuffle.

The previous implementation picked a random starting node and did two passes over the linked list to find query candidates. The new logic introduces fctx_getaddresses_nsorder() to perform an in-place randomization of indices into a bounded, stack-allocated lookup array (nsorder) representing the "winning" fetch slots.

The nameserver dataset is now traversed in exactly one sequential pass: 1. Every nameserver is evaluated for local cached data. 2. If the current nameserver's sequential index exists in the randomized nsorder array, it is permitted to launch an outgoing network fetch. 3. If not, it is restricted to local lookups via DNS_ADBFIND_NOFETCH.

This guarantees a fair random distribution for outbound queries while maximizing local cache hits, entirely within O(1) memory and without the overhead of linked-list pointer shuffling or dynamic allocation. **[GL #5695] [GL !11606]**

- Invalid NSEC3 can cause OOB read of the isdelegation() stack. `e6f234169e2`

  When .next_length is longer than NSEC3_MAX_HASH_LENGTH, it causes a harmless out-of-bound read of the isdelegation() stack. This has been fixed. **[GL #5749] [GL !11594]**

- Optimize the TCP source port selection on Linux. `d4426f85b36`

  Enable a socket option on the outgoing TCP sockets to allow faster selection of the source <address,port> tuple for different destination <address,port> tuples when nearing over 70-80% of the source port utilization. **[GL !11573]**

### 15.10.2 Bug Fixes

- Fix errors when retrying over TCP in notify_send_toaddr. `a1232333196`

  If the source address is not available do not attempt to retry over TCP otherwise clear the TSIG key from the message prior to retrying. **[GL #5457] [GL !11567]**

- Fetch loop detection improvements. `892c3e78926`

  Fixes a case where an in-domain NS with an expired glue would fail to resolve.

  Let's consider the following parent-side delegation (both for *foo.example.* and *dnshost.example.*

  ` foo.example. 3600 NS ns.dnshost.example. dnshost.example. 3600 NS ns.dnshost.example. ns.dnshost.example. 3600 A 1.2.3.4 ` Then the child-side of *dnshost.example.*:

  ` dnshost.example. 300 NS ns.dnshost.example. ns.dnshost.example. 300 A 1.2.3.4 ` And then the child-side of *foo.example.*:

  ` foo.example 3600 NS ns.dnshost.example. a.foo.example 300 A 5.6.7.8 `

  While there is a zone misconfiguration (the TTL of the delegation and glue doesn't match in the parent and the child), it is possible to resolve *a.foo.example* on a cold-cache resolver. However, after the *ns.dnshost.example.* glue expires, the resolution would have failed with a "fetch loop detected" error. This is now fixed. **[GL #5588] [GL !11547]**

- Remove deterministic selection of nameserver. `c6c6e490fd8`

  When selecting nameserver addresses to be looked up we where always selecting them in dnssec name order from the start of the nameserver rrset. This could lead to resolution failure despite there being address that could be resolved for the other names. Use a random starting point when selecting which names to lookup. **[GL #5695] [GL #5745] [GL !11600]**

- DNSTAP wasn't logging forwarded queries correctly. `0a5922bcf7a`

  **[GL #5724] [GL !11555]**

- Fix read UAF in BIND9 dns_client_resolve() via DNAME Response. `c0c4bf526a1`

  An attacker controlling a malicious DNS server returns a DNAME record, and the we stores a pointer to resp->foundname, frees the response structure, then uses the dangling pointer in dns_name_fullcompare() possibly causing invalid match. Only the *delv`is affected. This has been fixed. :gl:`#5728* **[GL !11571]**

- Clear serve-stale flags when following the CNAME chains. `68fb2312948`

  A stale answer could have been served in case of multiple upstream failures when following the CNAME chains. This has been fixed. **[GL #5751] [GL !11583]**

- Fail DNSKEY validation when supported but invalid DS is found. `2e1971873a1`

  A regression was introduced when adding the EDE code for unsupported DNSKEY and DS algorithms. When the parent has both supported and unsupported algorithm in the DS record, the validator would treat the supported DS algorithm as insecure when validating DNSKEY records instead of BOGUS. This has not security impact as the rest of the child zone correctly ends with BOGUS status, but it is incorrect and thus the regression has been fixed. **[GL #5757] [GL !11590]**

- Importing invalid SKR file might corrupt stack memory. `9869a14ce3a`

  If an BIND 9 administrator imports an invalid SKR file, local stack in the import function might overflow. This could lead to a memory corruption on the stack and ultimately server crash. This has been fixed.

  ISC would like to thank mcsky23 for bringing this bug to our attention. **[GL #5758] [GL !11598]**

- Do not update the case on unchanged rdatasets. `8931f82dc8b`

  Fix assertion failure on unchanged rdataset during IXFR. **[GL #5759] [GL !11587]**

- Return FORMERR for ECS family 0. `8ac316bf0f6`

  RFC 7871 only defines family 1 (IPv4) and 2 (IPv6). Additionally it requires FORMERR to be returned for all unknown families. **[GL !11565]**

## 15.11 BIND 9.20.19

### 15.11.1 Feature Changes

- Update requirements for system test suite. `a2a9b1b878`

  Python 3.10 or newer is now required for running the system test suite. The required python packages and their version requirements are now tracked in *bin/tests/system/requirements.txt*.

  Support for pytest 9.0.0 has been added its minimum supported version has been raised to 7.0.0. The minimum supported dnspython version has been raised to 2.3.0. **[GL #5690] [GL #5614] [GL !11469]**

- Use enum rather than numbers for isc_base64_tobuffer and isc_hex_tobuffer. `47b8ca5ac7`

  Use isc_one_or_more and isc_zero_or_more rather than (-2) and (-1) when calling isc_base64_tobuffer. Similarly for isc_hex_tobuffer. This should help reduce the probability that the wrong number is used and it makes the intent clearer. **[GL #5713] [GL !11498]**

### 15.11.2 Bug Fixes

- Fix inbound IXFR performance regression. `318a7535d2`

  Very large inbound IXFR transfers were much slower compared to BIND 9.18. The performance was improved by adding specialized logic to handle IXFR transfers. **[GL #5442] [GL !11355]**

- Make catalog zone names and member zones' entry names case-insensitive. `cd23f0250a`

Previously, the catalog zone names and their member zones' entry names were unintentionally case-sensitive. This has been fixed. **[GL #5693] [GL !11450]**

- Use const pointer with strchr of const pointer. `736b84ad46`

  **[GL #5694] [GL !11463]**

- Fix brid and hhit implementation. `f73ef3b24f`

  Fix bugs in BRID and HHIT implementation and enable the unit tests. **[GL #5710] [GL !11492]**

- DSYNC record incorrectly used two octets for the Scheme Field. `bd9f73c705`

  When creating the *DSYNC* record from a structure, *uint16_tobuffer* was used instead of *uint8_tobuffer* when adding the scheme, causing a *DSYNC* record that was one octet too long. This has been fixed. **[GL #5711] [GL !11483]**

- Fix a possible issue with reponse policy zones and catalog zones. `3d0823ee68`

  If a response policy zone (RPZ) or a catalog zone contained an *$INCLUDE* directive, then manually reloading that zone could fail to process the changes in the response policy or in the catalog, respectively. This has been fixed. **[GL #5714] [GL !11496]**

## 15.12 BIND 9.20.18

### 15.12.1 Security Fixes

- [CVE-2025-13878] Fix incorrect length checks for BRID and HHIT records. `d4c0d61701`

  Malformed BRID and HHIT records could trigger an assertion failure. This has been fixed.

  ISC would like to thank Vlatko Kosturjak from Marlink Cyber for bringing this vulnerability to our attention. **[GL #5616]**

### 15.12.2 Feature Changes

- Support compilation with cmocka 2.0.0+ `bb9234c6ce`

  The *assert_in_range()* function was deprecated in favor of *assert_int_in_range()* and *assert_uint_in_range()*. Add compatibility shims for cmocka<2.0.0 and use the new functions. **[GL #5699] [GL !11437]**

- Add more information to the rndc recursing output about fetches. `9766feb4df`

  This adds more information about the active fetches for debugging and diagnostic purposes. **[GL !11358]**

### 15.12.3 Bug Fixes

- Make key rollovers more robust. `7a70d05b5d`

  A manual rollover when the zone is in an invalid DNSSEC state causes predecessor keys to be removed too quickly. Additional safeguards to prevent this have been added. DNSSEC records will not be removed from the zone until the underlying state machine has moved back into a valid DNSSEC state. **[GL #5458] [GL !11329]**

- Fix a catalog zones issue when a member zone could fail to load. `95cbc2c327`

  A catalog zone's member zone could fail to load in some rare cases, when the internally generated zone configuration string was exceeding 512 bytes. That condition only was not enough for the issue to arise, but it was a necessary condition. This could happen, for example, if the catalog zone's default primary servers list contained a large number of items. This has been fixed. **[GL #5658] [GL !11349]**

- Allow glue in delegations with QTYPE=ANY. `441158ac18`

  When a query for type ANY triggered a delegation response, all additional data was omitted from the response, including mandatory glue. This has been corrected. **[GL #5659] [GL !11283]**

- Adding NSEC3 opt-out records could leave invalid records in chain. `1b90296e1f`

  When creating an NSEC3 opt-out chain, a node in the chain could be removed too soon, causing the previous NSEC3 being unable to be found, resulting in invalid NSEC3 records to be left in the zone. This has been fixed. **[GL #5671] [GL !11340]**

- Fix slow speed of NSEC3 optout large delegation zone signing. `88f915b77b`

  BIND 9.20 takes much more time signing a large delegation zone with NSEC3 optout compared to version 9.18. This has been restored. **[GL #5672] [GL !11362]**

- Reconfigure NSEC3 opt-out zone to NSEC causes zone to be invalid. `1d0e19c612`

  A zone that is signed with NSEC3, opt-out enabled, and then reconfigured to use NSEC, causes the zone to be published with missing NSEC records. This has been fixed. **[GL #5679] [GL !11401]**

- Fix a possible catalog zone issue during reconfiguration. `911b45b2b3`

  The *named* process could terminate unexpectedly during reconfiguration when a catalog zone update was taking place at the same time. This has been fixed. **[GL !11386]**

- Fix the charts in the statistics channel. `7c7b01dd65`

  The charts in the statistics channel could sometimes fail to render in the browser, and were completely disabled for Mozilla-based browsers for historical reasons. This has been fixed. **[GL !11364]**

## 15.13 BIND 9.20.17

### 15.13.1 New Features

- Add spatch to detect implicit bool/int/result cast. `02be363d1f`

  Detection of implicit cast from a boolean into an int, or an isc_result_t into a boolean (either in an assignement or return position).

  If such pattern is found, a warning comment is added into the code (and the CI will fails) so the error can be spotted and manually fixed. **[GL !11237]**

### 15.13.2 Feature Changes

- Use atomics for CMM_{LOAD,STORE}_SHARED with ThreadSanitizer. `94fa721705`

  Upstream has removed the atomics implementation of CMM_LOAD_SHARED and CMM_STORE_SHARED as these can be used also with non-stdatomics types. As we only use the CMM api with stdatomics types, we can restore the previous behaviour to prevent ThreadSanitizer warnings. **[GL #5660] [GL !11290]**

- Provide more information when the memory allocation fails. `6749725610`

  Provide more information about the failure when the memory allocation fails. **[GL !11304]**

- Reduce the number of outgoing queries. `457b470e96`

  Reduces the number of outgoing queries when resolving the nameservers for delegation points. This helps the DNS resolver with cold cache resolve client queries with complex delegation chains and redirections. **[GL !11258]**

### 15.13.3 Bug Fixes

- Fix the spurious timeouts while resolving names. `d96cf874fb`

  Sometimes the loops in the resolving (e.g. to resolve or validate ns1.example.com we need to resolve ns1.example.com) were not properly detected leading to spurious 10 seconds delay. This has been fixed and such loops are properly detected. **[GL #3033]**, #5578 **[GL !11298]**

- Fix bug where zone switches from NSEC3 to NSEC after retransfer. `3b40ffbf83`

  When a zone is re-transferred, but the zone journal on an inline-signing secondary is out of sync, the zone could fall back to using NSEC records instead of NSEC3. This has been fixed. **[GL #5527] [GL !11274]**

- Attach socket before async streamdns_resume_processing. `bb9451c73f`

  Call to *streamdns_resume_processing* is asynchronous but the socket passed as argument is not attached when scheduling the call.

  While there is no reproducible way (so far) to make the socket reference number down to 0 before *streamdns_resume_processing* is called, attach the socket before scheduling the call. This guard against an hypothetic case where, for some reasons, the socket refcount would reach 0, and be freed from memory when *streamdns_resume_processing* is called. **[GL #5620] [GL !11260]**

- AMTRELAY type 0 presentation format handling was wrong. `adf104a063`

  RFC 8777 specifies a placeholder value of "." for the gateway field when the gateway type is 0 (no gateway). This was not being checked for nor emitted when displaying the record. This has been corrected.

  Instances of this record will need the placeholder period added to them when upgrading. **[GL #5639] [GL !11255]**

- Fix parsing bug in remote-servers with key or tls. `d9400c5967`

  The *`remote-servers`* clause enable the following pattern using a named `server-list`:

  remote-servers a { 1.2.3.4; ... }; remote-servers b { a key foo; };

  However, such configuration was wrongly rejected, with an "unexpected token 'foo'" error. Such configuration is now accepted. **[GL #5646] [GL !11300]**

- Fix TLS contexts cache object usage bug in the resolver. `13adf94006`

  *`named`* could terminate unexpectedly when reconfiguring or reloading, and if client-side TLS transport was in use (for example, when forwarding queries to a DoT server). This has been fixed. **[GL #5653] [GL !11299]**

- Fix unitiailized pointer check on getipandkeylist. `5ed0cf091b`

  Function *named_config_getipandkeylist* could, in case of error in the early code attempting to get the *port* or *tls-port*, make a pointer check on a non-initialized value. This is now fixed. **[GL !11306]**

- Standardize CHECK and RETERR macros. `ef714e91ac`

  previously, there were over 40 separate definitions of CHECK macros, of which most used "goto cleanup", and the rest "goto failure" or "goto out". there were another 10 definitions of RETERR, of which most were identical to CHECK, but some simply returned a result code instead of jumping to a cleanup label.

  this has now been standardized throughout the code base: RETERR is for returning an error code in the case of an error, and CHECK is for jumping to a cleanup tag, which is now always called "cleanup". both macros are defined in isc/util.h. **[GL !11069]**

- Adding NSEC3 opt-out records could leave invalid records in chain. `1d83a8ad46`

  When creating an NSEC3 opt-out chain, a node in the chain could be removed too soon, causing the previous NSEC3 being unable to be found, resulting in invalid NSEC3 records to be left in the zone. This has been fixed.

  Closes [#5671](#5671)

## 15.14 BIND 9.20.16

### 15.14.1 Feature Changes

- Fix assertion failure from arc4random_uniform with invalid limit. `1040282de7e`

When the arc4random_uniform() is called on NetBSD with upper_bound that makes no sense statistically (0 or 1), the call crashes the calling program. Fix this by returning 0 when upper bound is < 2 as does Linux, FreeBSD and NetBSD. (Hint: System CSPRNG should never crash.) **[GL #5596] [GL !11151]**

### 15.14.2 Bug Fixes

- Fix dnssec-keygen key collision checking for KEY rrtype keys. `ac8b23b80bf`

  The *dnssec-keygen* utility program failed to detect possible Key ID collisions with the existing keys generated using the non-default `-T KEY` option (e.g. for `SIG(0)`). This has been fixed. **[GL #5506] [GL !11128]**

- Fix shutdown INSIST in dns_dispatchmgr_getblackhole. `f0aaaef166c`

  Previously, *named* could trigger an assertion in *dns_dispatchmgr_getblackhole* while shutting down. This has been fixed. **[GL #5525] [GL !11162]**

- Dnssec-verify now uses exit code 1 when failing due to illegal options. `6ead0aa4a2b`

  Previously, dnssec-verify exited with code 0 if the options could not be parsed. This has been fixed. **[GL #5574] [GL !11129]**

- Prevent assertion failures of dig when server is specified before the -b option. `deada63e2b2`

  Previously, *dig* could exit with an assertion failure when the server was specified before the *dig -b* option. This has been fixed. **[GL #5609] [GL !11204]**

- Skip unsupported algorithms when looking for signing key. `c346fe88a1b`

  A mix of supported and unsupported DNSSEC algorithms in the same zone could have caused validation failures. Ignore the DNSSEC keys with unsupported algorithm when looking for the signing keys. **[GL #5622] [GL !11210]**

- Fix configuration bugs involving global defaults. `a85d6fb581c`

  The configuration code for the *max-cache-size*, *dnssec-validation*, and *response-padding* options were unnecessarily complicated, and in the case of *max-cache-size*, buggy. These have been fixed. The *optionmaps* variable in *configure_view()* is no longer needed and has been removed. **[GL !11172]**

- Skip buffer allocations if not logging. `4f601175bd0`

  Currently, during IXFR we allocate a 2KB buffer for IXFR change logging regardless of the log level. This commit introduces an early check on the log level in dns_diff_print to avoid this.

  Results in a speedup from 28% in the test case from issue #5442. **[GL !11192]**

## 15.15 BIND 9.20.15

### 15.15.1 Security Fixes

- [CVE-2025-8677] DNSSEC validation fails if matching but invalid DNSKEY is found. `0d676bf9f23`

  Previously, if a matching but cryptographically invalid key was encountered during DNSSEC validation, the key was skipped and not counted towards validation failures. *named* now treats such DNSSEC keys as hard failures and the DNSSEC validation fails immediately, instead of continuing with the next DNSKEYs in the RRset.

  ISC would like to thank Zuyao Xu and Xiang Li from the All-in-One Security and Privacy Laboratory at Nankai University for bringing this vulnerability to our attention. **[GL #5343]**

- [CVE-2025-40778] Address various spoofing attacks. `23de94fd236`

  Previously, several issues could be exploited to poison a DNS cache with spoofed records for zones which were not DNSSEC-signed or if the resolver was configured to not do DNSSEC validation. These issues were assigned CVE-2025-40778 and have now been fixed.

As an additional layer of protection, *named* no longer accepts DNAME records or extraneous NS records in the AUTHORITY section unless these are received via spoofing-resistant transport (TCP, UDP with DNS cookies, TSIG, or SIG(0)).

ISC would like to thank Yuxiao Wu, Yunyi Zhang, Baojun Liu, and Haixin Duan from Tsinghua University for bringing this vulnerability to our attention. **[GL #5414]**

- [CVE-2025-40780] Cache-poisoning due to weak pseudo-random number generator. `34af35c2df8`

  It was discovered during research for an upcoming academic paper that a xoshiro128** internal state can be recovered by an external 3rd party, allowing the prediction of UDP ports and DNS IDs in outgoing queries. This could lead to an attacker spoofing the DNS answers with great efficiency and poisoning the DNS cache.

  The internal random generator has been changed to a cryptographically secure pseudo-random generator.

  ISC would like to thank Prof. Amit Klein and Omer Ben Simhon from Hebrew University of Jerusalem for bringing this vulnerability to our attention. **[GL #5484]**

### 15.15.2 New Features

- Add dnssec-policy keys configuration check to named-checkconf. `1f5a0405f72`

  A new option *-k* is added to *named-checkconf* that allows checking the *dnssec-policy keys* configuration against the configured key stores. If the found key files are not in sync with the given *dnssec-policy*, the check will fail.

  This is useful to run before migrating to *dnssec-policy*. **[GL #5486] [GL !11011]**

### 15.15.3 Feature Changes

- Minor refactor of dst code. `c6acbaa020b`

  Convert the defines to enums. Initialize the tags more explicitly and less ugly. **[GL !11038]**

### 15.15.4 Bug Fixes

- Use signer name when disabling DNSSEC algorithms. `986816baa74`

  `disable-algorithms` could cause DNSSEC validation failures when the parent zone was signed with the algorithms that were being disabled for the child zone. This has been fixed; *disable-algorithms* now works on a whole-of-zone basis.

  If the zone's name is at or below the `disable-algorithms` name the algorithm is disabled for that zone, using deepest match when there are multiple `disable-algorithms` clauses. **[GL #5165] [GL !11014]**

- Rndc sign during ZSK rollover will now replace signatures. `d2f551140cd`

  When performing a ZSK rollover, if the new DNSKEY is omnipresent, the *rndc sign* command now signs the zone completely with the successor key, replacing all zone signatures from the predecessor key with new ones. **[GL #5483] [GL !11017]**

- Missing DNSSEC information when CD bit is set in query. `968a6be41fb`

  The RRSIGs for glue records were not being cached correctly for CD=1 queries. This has been fixed. **[GL #5502] [GL !10956]**

- Preserve cache when reload fails and reload the server again. `975aeda10b4`

  Fixes an issue where failing to reconfigure/reload the server would prevent to preserved the views caches on the subsequent server reconfiguration/reload. **[GL #5523] [GL !10988]**

- Check plugin config before registering. `e2260b80702`

In *named_config_parsefile()*, when checking the validity of *named.conf*, the checking of plugin correctness was deliberately postponed until the plugin is loaded and registered. However, the checking was never actually done: the *plugin_register()* implementation was called, but *plugin_check()* was not.

*ns_plugin_register()* (used by *named*) now calls the check function before the register function, and aborts if either one fails. *ns_plugin_check()* (used by *named-checkconf*) calls only the check function. **[GL !11032]**

## 15.16 BIND 9.20.14

**Note**

The BIND 9.20.14 release was withdrawn after the discovery of a regression in a security fix in it during pre-release testing.

## 15.17 BIND 9.20.13

### 15.17.1 New Features

- Add manual mode configuration option to dnsec-policy. `1e435b107f`

  Add a new option `manual-mode` to *dnssec-policy*. The intended use is that if it is enabled, it will not automatically move to the next state transition, but instead the transition is logged. Only after manual confirmation with `rndc dnssec -step` the transition is made. **[GL #4606] [GL !10880]**

- Add a new 'servfail-until-ready' configuration option for RPZ. `925af17d21`

  By default, when *named* is started it may start answering to queries before the response policy zones are completely loaded and processed. This new feature gives an option to the users to tell *named* that incoming requests should result in SERVFAIL answer until all the response policy zones are processed and ready. Note that if one or more response policy zones fail to load, *named* starts responding to queries according to those zones that did load.

  Note, that enabling this option has no effect when a DNS Response Policy Service (DNSRPS) interface is used. **[GL #5222] [GL !10889]**

- Support for parsing HHIT and BRID records has been added. `1f051af24d`

  **[GL #5444] [GL !10932]**

### 15.17.2 Removed Features

- Deprecate the "tkey-gssapi-credential" statement. `b239a70cac`

  The *tkey-gssapi-keytab* statement allows GSS-TSIG to be set up in a simpler and more reliable way than using the *tkey-gssapi-credential* statement and setting environment variables (e.g. `KRB5_KTNAME`). Therefore, the *tkey-gssapi-credential* statement has been deprecated; *tkey-gssapi-keytab* should be used instead.

  For configurations currently using a combination of both *tkey-gssapi-keytab* *and* *tkey-gssapi-credential*, the latter should be dropped and the keytab pointed to by *tkey-gssapi-keytab* should now only contain the credential previously specified by *tkey-gssapi-credential*. **[GL #4204] [GL !10924]**

- Obsolete the "tkey-domain" statement. `9352ae65d7`

  Mark the `tkey-domain` statement as obsolete, since it has not had any effect on server behavior since support for TKEY Mode 2 (Diffie-Hellman) was removed (in BIND 9.20.0). **[GL #4204] [GL !10926]**

### 15.17.3 Feature Changes

- Update clang-format style with options added in newer versions. `0c2c477c31`

  Add and apply InsertBraces statement to add missing curly braces around one-line statements and use ControlStatementsExceptControlMacros for SpaceBeforeParens to remove space between foreach macro and the brace, e.g. *FOREACH (x) {* becomes *FOREACH(x) {*. **[GL !10864]**

### 15.17.4 Bug Fixes

- Ensure file descriptors 0-2 are in use. `35dee6eb90`

  libuv expect file descriptors <= STDERR_FILENO are in use. otherwise, it may abort when closing a file descriptor it opened. **[GL #5226] [GL !10908]**

- Prevent spurious SERVFAILs for certain 0-TTL resource records. `6b266b222c`

  Under certain circumstances, BIND 9 can return SERVFAIL when updating existing entries in the cache with new NS, A, AAAA, or DS records with 0-TTL. **[GL #5294] [GL !10898]**

- Use DNS_RDATACOMMON_INIT to hide branch differences. `a64df9729b`

  Initialization of the common members of rdata type structures varies across branches. Standardize it by using the *DNS_RDATACOMMON_INIT* macro for all types, so that new types are more likely to use it, and hence backport more cleanly. **[GL #5467] [GL !10834]**

- RPZ canonical warning displays zone entry incorrectly. `d833676515`

  When an IPv6 rpz prefix entry is entered incorrectly the log message was just displaying the prefix rather than the full entry. This has been corrected. **[GL #5491] [GL !10930]**

- Fix a catalog zone issue when having an unset 'default-primaries' configuration clause. `293e75af28`

  A catalog zone with an unset `default-primaries` clause could cause an unexpected termination of the *named* process after two reloading or reconfiguration commands. This has been fixed. **[GL #5494] [GL !10905]**

- Add and use __attribute__((nonnull)) in dnssec-signzone.c. `a8eed36d3e`

  Clang 20 was spuriously warning about the possibility of passing a NULL file pointer to *fprintf()*, which uses the 'nonnull' attribute. To silence the warning, the functions calling *fprintf()* have been marked with the same attribute to assure that NULL can't be passed to them in the first place.

  Close #5487 **[GL !10913]**

- RPZ 'servfail-until-ready': skip updating SERVFAIL cache. `af2fb26325`

  In order to not pollute the SERVFAIL cache with the configured SERVFAIL answers while RPZ is loading, set the NS_CLIENTATTR_NOSETFC attribute for the client. **[GL !10940]**

## 15.18 BIND 9.20.12

### 15.18.1 New Features

- Support for parsing the DSYNC record has been added. `f440fe712d`

  **[GL #5440] [GL !10820]**

### 15.18.2 Feature Changes

- Adaptive memory allocation strategy for qp-tries. `9a046cbed5`

  qp-tries allocate their nodes (twigs) in chunks to reduce allocator pressure and improve memory locality. The choice of chunk size presents a tradeoff: larger chunks benefit qp-tries with many values (as seen in large zones and resolvers) but waste memory in smaller use cases.

  Previously, our fixed chunk size of 2^10 twigs meant that even an empty qp-trie would consume 12KB of memory, while reducing this size would negatively impact resolver performance.

  This MR implements an adaptive chunking strategy that tracks the size of the most recently allocated chunk and doubles the chunk size for each new allocation until reaching a predefined maximum.

  This approach effectively balances memory efficiency for small tries while maintaining the performance benefits of larger chunk sizes for bigger data structures. **[GL #5445] [GL !10804]**

- Add deprecation warnings for RSASHA1, RSASHA1-NSEC3SHA1 and DS digest type 1. `5aefaa4b97`

  RSASHA1 and RSASHA1-NSEC-SHA1 DNSKEY algorithms have been deprecated by the IETF and should no longer be used for DNSSEC. DS digest type 1 (SHA1) has also been deprecated. Validators are now expected to treat these algorithms and digest as unknown, resulting in some zones being treated as insecure when they were previously treated as secure. Warnings have been added to named and tools when these algorithms and this digest are being used for signing.

  Zones signed with RSASHA1 or RSASHA1-NSEC-SHA1 should be migrated to a different DNSKEY algorithm.

  Zones with DS or CDS records with digest type 1 (SHA1) should be updated to use a different digest type (e.g. SHA256) and the digest type 1 records should be removed.

  Related to #5358 **[GL !10738]**

### 15.18.3 Bug Fixes

- Stale RRsets in a CNAME chain were not always refreshed. `ed37c7825e`

  With serve-stale enabled, a CNAME chain that contains a stale RRset, the refresh query doesn't always properly refresh the stale RRsets. This has been fixed. **[GL #5243] [GL !10767]**

- Add RPZ extended DNS error for zones with a CNAME override policy configured. `39ad2016c1`

  When the zone is configured with a CNAME override policy, or the response policy zone contains a wildcard CNAME, the extended DNS error code was not added. This has been fixed. **[GL #5342] [GL !10819]**

- Fix a possible crash when adding a zone while recursing. `7a3ec8dd94`

  A query for a zone that was not yet loaded may yield an unexpected result such as a CNAME or DNAME, triggering an assertion failure. This has been fixed. **[GL #5357] [GL !10718]**

- Fix dig issues. `8c50819aa8`

  When used with the `+keepopen` option with a TCP connection, iscman:*dig* could terminate unexpectedly in rare situations. Additionally, iscman:*dig* could hang and fail to shutdown properly when interrupted during a query. These have been fixed. **[GL #5381] [GL !10727]**

- Log dropped or slipped responses in the query-errors category. `47470b586d`

  Responses which were dropped or slipped because of RRL (Response Rate Limiting) were logged in the `rate-limit` category instead of the `query-errors` category, as documented in ARM. This has been fixed. **[GL #5388] [GL !10725]**

- Separate out adbname type flags. `fc689c6525`

There are three adbname flags that are used to identify different types of adbname lookups when hashing rather than using multiple hash tables. Separate these to their own structure element as these need to be able to be read without locking the adbname structure. **[GL #5404] [GL !10695]**

- Synth-from-dnssec was not working in some scenarios. `bc54f059e0`

  Aggressive use of DNSSEC-Validated cache with NSEC was not working in scenarios when no parent NSEC was not in cache. This has been fixed. **[GL #5422] [GL !10754]**

- Clean enough memory when adding new ADB names/entries under memory pressure. `b7e7923daa`

  The ADB memory cleaning is opportunistic even when we are under memory pressure (in the overmem condition). Split the opportunistic LRU cleaning and overmem cleaning and make the overmem cleaning always cleanup double of the newly allocated adbname/adbentry to ensure we never allocate more memory than the assigned limit. **[GL !10707]**

- Prevent spurious validation failures. `3b98c7cc9d`

  Under rare circumstances, validation could fail if multiple clients simultaneously iterated the same set of signatures.

  References #3014 **[GL !10815]**

- Rename variable called 'free' to prevent the clash with free() `7f25d92c5d`

  **[GL !10757]**

## 15.19 BIND 9.20.11

### 15.19.1 Security Fixes

- [CVE-2025-40777] Fix a possible assertion failure when using the 'stale-answer-client-timeout 0' option. `055a592fd97`

  In specific circumstances the *named* resolver process could terminate unexpectedly when stale answers were enabled and the `stale-answer-client-timeout 0` configuration option was used. This has been fixed. **[GL #5372]**

### 15.19.2 New Features

- Add support for the CO flag to dig. `47108af9f2e`

  Add support to display the CO (Compact Answers OK flag) when displaying messages.

  Add support to set the CO flag when making queries in dig (+coflag). **[GL #5319] [GL !10578]**

### 15.19.3 Bug Fixes

- Fix the default interface-interval from 60s to 60m. `e8ffe3a15ca`

  When the interface-interval parser was changed from uint32 parser to duration parser, the default value stayed at plain number *60* which now means 60 seconds instead of 60 minutes. The documentation also incorrectly states that the value is in minutes. That has been fixed. **[GL #5246] [GL !10679]**

- Fix purge-keys bug when using views. `35efa742b03`

  Previously, when a DNSSEC key was purged by one zone view, other zone views would return an error about missing key files. This has been fixed. **[GL #5315] [GL !10598]**

- Use IPv6 queries in delv +ns. `4916fe0c6bd`

  *delv +ns* invokes the same code to perform name resolution as *named*, but it neglected to set up an IPv6 dispatch object first. Consequently, it was behaving more like *named -4*. It now sets up dispatch objects for both address

families, and performs resolver queries to both v4 and v6 addresses, except when one of the address families has been suppressed by using *delv -4* or *delv -6*. **[GL #5352] [GL !10573]**

# 15.20 BIND 9.20.10

## 15.20.1 New Features

- Implement a new 'notify-defer' configuration option. `a24db6433e6`

  This new option sets a delay (in seconds) to wait before sending a set of NOTIFY messages for a zone. Whenever a NOTIFY message is ready to be sent, sending will be deferred for this duration. This option is not to be confused with the *notify-delay* option. The default is 0 seconds. **[GL #5259] [GL !10465]**

## 15.20.2 Removed Features

- Implement the systemd notification protocol manually to remove dependency on libsystemd. `4f7e806a12b`

  libsystemd, despite being useful, adds a huge surface area for just using the sd_notify API. libsystemd's surface has been exploited in the past [1].

  Implement the systemd notification protocol by hand since it is just sending newline-delimited datagrams to a UNIX socket. The code shouldn't need more attention in the future since the notification protocol is covered under systemd's stability promise [2].

  We don't need to support VSOCK-backed service notifications since they are only intended for virtual machine inits.

  [1]: https://www.openwall.com/lists/oss-security/2024/03/29/4 [2]: https://systemd.io/PORTABILITY_AND_STABILITY/ **[GL !10454]**

## 15.20.3 Bug Fixes

- Fix zone deletion issue. `66fc4ee86e0`

  A secondary zone could initiate a new zone transfer from the primary server after it had been already deleted from the secondary server, and before the internal garbage collection was activated to clean it up completely. This has been fixed. **[GL #5291] [GL !10496]**

- Fix a zone refresh bug. `f09bb8b88c6`

  A secondary zone could fail to further refresh with new versions of the zone from a primary server if named was reconfigured during the SOA request step of an ongoing zone transfer. This has been fixed. **[GL #5307] [GL !10495]**

- Allow keystore.c to compile on Solaris. `108adab25a0`

  keystore.c failed to compile on Solaris because NAME_MAX was undefined. Include 'isc/dir.h' which defines NAME_MAX for platforms that don't define it. **[GL #5327] [GL !10523]**

- Set name for all the isc_mem contexts. `bdcd698edf7`

  **[GL !10498]**

# 15.21 BIND 9.20.9

## 15.21.1 Security Fixes

- [CVE-2025-40775] Prevent assertion when processing TSIG algorithm. `b8c198ac5ca`

DNS messages that included a Transaction Signature (TSIG) containing an invalid value in the algorithm field caused *named* to crash with an assertion failure. This has been fixed. **(CVE-2025-40775) [GL #5300]**

### 15.21.2 Feature Changes

- Use jinja2 templates in system tests. `8f545784ff0`

  *python-jinja2* is now required to run system tests. **[GL #4938] [GL !10396]**

### 15.21.3 Bug Fixes

- Fix EDNS yaml output. `8c3b226d89b`

  *dig* was producing invalid YAML when displaying some EDNS options. This has been corrected.

  Several other improvements have been made to the display of EDNS option data: - We now use the correct name for the UPDATE-LEASE option, which was previously displayed as "UL", and split it into separate LEASE and LEASE-KEY components in YAML mode. - Human-readable durations are now displayed as comments in YAML mode so as not to interfere with machine parsing. - KEY-TAG options are now displayed as an array of integers in YAML mode. - EDNS COOKIE options are displayed as separate CLIENT and SERVER components, and cookie STATUS is a retrievable variable in YAML mode. **[GL #5014] [GL !10414]**

- Return DNS COOKIE and NSID with BADVERS. `34b7323bad6`

  This change allows the client to identify the server that returns the BADVERS and to provide a DNS SERVER COOKIE to be included in the resend of the request. **[GL #5235] [GL !10392]**

- Disable own memory context for libxml2 on macOS. `51e51d5ea8f`

  Apple broke custom memory allocation functions in the system-wide libxml2 starting with macOS Sequoia 15.4. Usage of the custom memory allocation functions has been disabled on macOS. **[GL #5268] [GL !10411]**

- *check_private* failed to account for the length byte before the OID. `2b827380e75`

  In PRIVATEOID keys, the key data begins with a length byte followed by an ASN.1 object identifier that indicates the cryptographic algorithm to use. Previously, the length byte was not accounted for when checking the contents of keys and signatures, which could have led to interoperability problems with any zones signed using PRIVATEOID. This has been fixed. **[GL #5270] [GL !10376]**

- Fix a serve-stale issue with a delegated zone. `d839d11bf62`

  When `stale-answer-client-timeout 0` option was enabled, it could be ignored when resolving a zone which is a delegation of an authoritative zone belonging to the resolver. This has been fixed. **[GL #5275] [GL !10420]**

- Fix the ksr two-tone test. `3e2b255b5b7`

  The two-tone ksr subtest (test_ksr_twotone) depended on the dnssec-policy keys algorithm values in named.conf being entered in numerical order. As the algorithms used in the test can be selected randomly this does not always happen. Sort the dnssec-policy keys by algorithm when adding them to the key list from named.conf. **[GL #5286] [GL !10435]**

- Revert NSEC3 closest encloser lookup improvements. `ac41f158fad`

  The performance improvements for NSEC3 closest encloser lookups that were restored in BIND 9.20.8 turned out to cause incorrect NSEC3 records to be returned in nonexistence proofs and were therefore reverted again. **[GL #5292] [GL !10443]**

## 15.22 BIND 9.20.8

### 15.22.1 New Features

- Add support for EDE 20 (Not Authoritative) `f8a293aa11`

  Support was added for EDE codes 20 (Not Authoritative) when client requests recursion (RD) but the server has recursion disabled.

  RFC 8914 mention EDE 20 should also be returned if the client doesn't have the RD bit set (and recursion is needed) but it doesn't apply for BIND as BIND would try to resolve from the "deepest" referral in AUTHORITY section. For example, if the client asks for "www.isc.org/A" but the server only knows the root domain, it will return NOERROR but no answer for "www.isc.og/A", just the list of other servers to ask. **[GL #1836] [GL !10243]**

- Add support for EDE 7 and EDE 8. `27442c3104`

  Support was added for EDE codes 7 (Signature Expired) and 8 (Signature Not Yet Valid) which might occur during DNSSEC validation. **[GL #2715] [GL !10242]**

- Dig can now display the received BADVERS message during negotiation. `a763080a87`

  Dig +showbadvers now displays the received BADVERS message and continues the EDNS version negotiation. Previously to see the BADVERS message +noednsneg had to be specified which terminated the EDNS negotiation. Additionally the specified EDNS value (+edns=value) is now used when making all the initial queries with +trace. i.e EDNS version negotiation will be performed with each server when performing the trace. **[GL #5234] [GL !10247]**

- Add an rndc command to reset some statistics counters. `7c4603fc4a`

  The new `reset-stats` command for `rndc` allows some statistics counters to be reset during runtime. At the moment only two "high-water" counters are supported, so the ability to reset them after the initial peaks during the server's "warm-up" phase may be useful for some operators. **[GL #5251] [GL !10335]**

- Implement -T cookiealwaysvalid. `1da738ffbb`

  When *-T cookiealwaysvalid* is passed to *named*, DNS cookie checks for the incoming queries always pass, given they are structurally correct. **[GL !10264]**

### 15.22.2 Removed Features

- Remove dns_qpmulti_lockedread declaration. `42ab4fce4a`

  This function was removed in 6217e434b57bd5d60ed69f792ae9a1a65a008f57 but not from the header file. **[GL !10314]**

### 15.22.3 Feature Changes

- Carefully check if the server name used for SNI is a hostname. `92eb6416e5`

  Previously the code would not check if the string intended to be used for SNI is a hostname.

  See also: !9923 **[GL #5225] [GL !10347]**

### 15.22.4 Bug Fixes

- Restore NSEC3 closest encloser lookup improvements. `ab6fb7b8f2`

  A performance improvement for finding the closest encloser when generating authoritative responses from NSEC3 zones was previously reverted after a bug was found that could trigger an assertion failure. ( **[GL #4460]**, #4950, and #5108 for details.) The bug has now been fixed, and the performance improvement has been restored. **[GL #5204] [GL !10034]**

- Add missing locks when returning addresses. `2c7594709c`

  Add missing locks in dns_zone_getxfrsource4 et al. Addresses CID 468706, 468708, 468741, 468742, 468785, and 468778.

  Cleanup dns_zone_setxfrsource4 et al to now return void.

  Remove double copies with dns_zone_getprimaryaddr and dns_zone_getsourceaddr. **[GL #4933] [GL !10259]**

- Stop caching lack of EDNS support. `96bb3a1952`

  *named* could falsely learn that a server doesn't support EDNS when a spoofed response was received; that subsequently prevented DNSSEC lookups from being made. This has been fixed. **[GL #3949] [GL #5066]**

- Step() could ignore rollbacks. `2334b7a371`

  The *step()* function (used for stepping to the prececessor or successor of a database node) could overlook a node if there was an rdataset that was marked IGNORE because it had been rolled back, covering an active rdataset under it. **[GL #5170] [GL !10257]**

- Fix resolver statistics counters for timed out responses. `1d8334a62a`

  When query responses timed out, the resolver could incorrectly increase the regular responses counters, even if no response was received. This has been fixed. **[GL #5193] [GL !10287]**

- Validating ADB fetches could cause a crash in import_rdataset() `b0c6af6ad7`

  Previously, in some cases, the resolver could return rdatasets of type CNAME or DNAME without the result code being set to *DNS_R_CNAME* or *DNS_R_DNAME*. This could trigger an assertion failure in the ADB. The resolver error has been fixed. **[GL #5201] [GL !10329]**

- Nested DNS validation could cause assertion failure. `6bf4390f25`

  When multiple nested DNS validations were destroyed out of order, the EDE context could be freed before all EDE codes were copied, which could cause an assertion failure. This has been fixed. **[GL #5213] [GL !10366]**

- [9.20] Wait for memory reclamation to finish in named-checkconf. `9986dad2dc`

  Previously, when named-checkzone loaded the zone to the QP database, the delayed memory reclamation could cause an assertion check on exit. This has been fixed. **[GL #5220] [GL !10307]**

- Ensure max-clients-per-query is at least clients-per-query. `8f78219cc1`

  If the *max-clients-per-query* option is set to a lower value than *clients-per-query*, the value is adjusted to match *clients-per-query*. **[GL #5224] [GL !10244]**

- Fix handling of revoked keys. `a347273b9c`

  When a key is revoked, its key ID changes due to the inclusion of the "revoked" flag. A collision between this changed key ID and an unrelated public-only key could cause a crash in *dnssec-signzone*. **[GL #5231] [GL !10256]**

- Fix adbname reference. `7ebcc54d3b`

  Call *dns_adbname_ref* before calling *dns_resolver_createfetch* to ensure *adbname->name* remains stable for the life of the fetch. **[GL #5239] [GL !10303]**

- Fix write after free in validator code. `5de1b3ba3c`

  Raw integer pointers were being used for the validator's nvalidations and nfails values but the memory holding them could be freed before they ceased to be used. Use reference counted counters instead. **[GL #5239] [GL !10300]**

- Don't enforce NOAUTH/NOCONF flags in DNSKEYs. `5d126d8081`

  All DNSKEY keys are able to authenticate. The *DNS_KEYTYPE_NOAUTH* (and *DNS_KEYTYPE_NOCONF*) flags were defined for the KEY rdata type, and are not applicable to DNSKEY. Previously, however, because the

DNSKEY implementation was built on top of KEY, the *_NOAUTH* flag prevented authentication in DNSKEYs as well. This has been corrected. **[GL #5240] [GL !10315]**

- Fix several small DNSSEC timing issues. `3a78a4c288`

  The following small issues related to *dnssec-policy* have been fixed: - In some cases the key manager inside BIND 9 could run every hour, while it could have run less often. - While *CDS* and *CDNSKEY* records will be removed correctly from the zone when the corresponding *DS* record needs to be updated, the expected timing metadata when this will happen was never set. - There were a couple of cases where the safety intervals are added inappropriately, delaying key rollovers longer than necessary. - If you have identical *keys* in your *dnssec-policy*, they may be retired inappropriately. Note that having keys with identical properties is discouraged in all cases. **[GL #5242] [GL !10301]**

- Copy __FILE__ when allocating memory. `59635e33d0`

  When allocating memory under -m trace|record, the __FILE__ pointer is stored, so it can be printed out later in order to figure out in which file an allocation leaked. (among others, like the line number).

  However named crashes when called with -m record and using a plugin leaking memory. The reason is that plugins are unloaded earlier than when the leaked allocations are dumped (obviously, as it's done as late as possible). In such circumstances, *__FILE__* is dangling because the dynamically loaded library (the plugin) is not in memory anymore.

  Fix the crash by systematically copying the *__FILE__* string instead of copying the pointer. Of course, this make each allocation to consume a bit more memory (and longer, as it needs to calculate the length of *__FILE__*) but this occurs only under -m trace|record debugging flags. **[GL !10336]**

- Fix invalid cache-line padding for qpcache buckets. `4297ae4795`

  The isc_queue_t was missing in the calculation of the required padding size inside the qpcache bucket structure. **[GL !10317]**

## 15.23 BIND 9.20.7

### 15.23.1 New Features

- Implement the min-transfer-rate-in configuration option. `4a5a9c8256`

  A new option 'min-transfer-rate-in <bytes> <minutes>' has been added to the view and zone configurations. It can abort incoming zone transfers which run very slowly due to network related issues, for example. The default value is set to 10240 bytes in 5 minutes. **[GL #3914] [GL !10137]**

- Add digest methods for SIG and RRSIG. `6d8c513986`

  ZONEMD digests RRSIG records and potentially digests SIG record. Add digests methods for both record types. **[GL #5219] [GL !10218]**

- Add HTTPS record query to host command line tool. `2ddfb57b45`

  The host command was extended to also query for the HTTPS RR type by default. **[GL !10123]**

### 15.23.2 Bug Fixes

- Prevent a reference leak when using plugins. `0201e3eacb`

  The *NS_QUERY_DONE_BEGIN* and *NS_QUERY_DONE_SEND* plugin hooks could cause a reference leak if they returned *NS_HOOK_RETURN* without cleaning up the query context properly. **[GL #2094] [GL !10170]**

- Fix isc_quota bug. `dbc635c148`

Running jobs which were entered into the isc_quota queue is the responsibility of the isc_quota_release() function, which, when releasing a previously acquired quota, checks whether the queue is empty, and if it's not, it runs a job from the queue without touching the 'quota->used' counter. This mechanism is susceptible to a possible hangup of a newly queued job in case when between the time a decision has been made to queue it (because used >= max) and the time it was actually queued, the last quota was released. Since there is no more quotas to be released (unless arriving in the future), the newly entered job will be stuck in the queue.

Fix the issue by adding checks in both isc_quota_release() and isc_quota_acquire_cb() to make sure that the described hangup does not happen. Also see code comments. **[GL #4965] [GL !10139]**

- Fix dual-stack-servers configuration option. `a47dab2c5e`

  The dual-stack-servers configuration option was not working as expected; the specified servers were not being used when they should have been, leading to resolution failures. This has been fixed. **[GL #5019] [GL !10174]**

- Implement sig0key-checks-limit and sig0message-checks-limit. `95af81b674`

  Previously a hard-coded limitation of maximum two key or message verification checks were introduced when checking the message's SIG(0) signature. It was done in order to protect against possible DoS attacks. The logic behind choosing the number 2 was that more than a single key should only be required during key rotations, and in that case two keys are enough. But later it became apparent that there are other use cases too where even more keys are required, see issue number #5050 in GitLab.

  This change introduces two new configuration options for the views, *sig0key-checks-limit* and *sig0message-checks-limit*, which define how many keys are allowed to be checked to find a matching key, and how many message verifications are allowed to take place once a matching key has been found. The latter protects against expensive cryptographic operations when there are keys with colliding tags and algorithm numbers, with default being 2, and the former protects against a bit less expensive key parsing operations and defaults to 16. **[GL #5050] [GL !10141]**

- Fix the data race causing a permanent active client increase. `20cf51dfc5`

  Previously, a data race could cause a newly created fetch context for a new client to be used before it had been fully initialized, which would cause the query to become stuck; queries for the same data would be either paused indefinitely or dropped because of the *clients-per-query* limit. This has been fixed. **[GL #5053] [GL !10147]**

- Fix deferred validation of unsigned DS and DNSKEY records. `ba5fe2dd12`

  When processing a query with the "checking disabled" bit set (CD=1), *named* stores the unvalidated result in the cache, marked "pending". When the same query is sent with CD=0, the cached data is validated, and either accepted as an answer, or ejected from the cache as invalid. This deferred validation was not attempted for DS and DNSKEY records if they had no cached signatures, causing spurious validation failures. We now complete the deferred validation in this scenario.

  Also, if deferred validation fails, we now re-query the data to find out whether the zone has been corrected since the invalid data was cached. **[GL #5066] [GL !10105]**

- When recording an rr trace, use libtool. `17ca2fbbdc`

  When a system test is run with the *USE_RR* environment variable set to 1, an *rr* trace is now correctly generated for each instance of *named*. **[GL #5079] [GL !10207]**

- Do not cache signatures for rejected data. `9b3e1facf6`

  The cache has been updated so that if new data is rejected - for example, because there was already existing data at a higher trust level - then its covering RRSIG will also be rejected. **[GL #5132] [GL !10134]**

- Fix RPZ race condition during a reconfiguration. `eca9a3279e`

  With RPZ in use, *named* could terminate unexpectedly because of a race condition when a reconfiguration command was received using *rndc*. This has been fixed. **[GL #5146] [GL !10144]**

- "CNAME and other data check" not applied to all types. `a68f5dd74b`

An incorrect optimization caused "CNAME and other data" errors not to be detected if certain types were at the same node as a CNAME. This has been fixed. **[GL #5150] [GL !10100]**

- Relax private DNSKEY and RRSIG constraints. `455080866c`

  DNSKEY, KEY, RRSIG and SIG constraints have been relaxed to allow empty key and signature material after the algorithm identifier for PRIVATEOID and PRIVATEDNS. It is arguable whether this falls within the expected use of these types as no key material is shared and the signatures are ineffective but these are private algorithms and they can be totally insecure. **[GL #5167] [GL !10173]**

- Delete dead nodes when committing a new version. `0682684028`

  In the qpzone implementation of *dns_db_closeversion()*, if there are changed nodes that have no remaining data, delete them. **[GL #5169] [GL !10124]**

- Revert "Delete dead nodes when committing a new version" `d2ec6d1db4`

  This reverts commit 67255da4b376f65138b299dcd5eb6a3b7f9735a9, reversing changes made to 74c9ff384e695d1b27fa365d1fee84576f869d4c. **[GL #5169] [GL !10226]**

- Fix dns_qp_insert() checks in qpzone. `11cc40ebf6`

  Remove code in the QP zone database to handle failures of *dns_qp_insert()* which can't actually happen. **[GL #5171] [GL !10114]**

- Remove NSEC/DS/NSEC3 RRSIG check from dns_message_parse. `b752db0c3f`

  Previously, when parsing responses, named incorrectly rejected responses without matching RRSIG records for NSEC/DS/NSEC3 records in the authority section. This rejection, if appropriate, should have been left for the validator to determine and has been fixed. **[GL #5185] [GL !10142]**

- Fix TTL issue with ANY queries processed through RPZ "passthru" `b1bf17096a`

  Answers to an "ANY" query which were processed by the RPZ "passthru" policy had the response-policy's *max-policy-ttl* value unexpectedly applied. This has been fixed. **[GL #5187] [GL !10180]**

- Dnssec-signzone needs to check for a NULL key when setting offline. `2d4b4fe15e`

  dnssec-signzone could dereference a NULL key pointer when resigning a zone. This has been fixed. **[GL #5192] [GL !10169]**

- Acquire the database reference before possibly last node release. `2b5b4e9dd1`

  Acquire the database reference in the detachnode() to prevent the last reference to be release while the NODE_LOCK being locked. The NODE_LOCK is locked/unlocked inside the RCU critical section, thus it is most probably this should not pose a problem as the database uses call_rcu memory reclamation, but this it is still safer to acquire the reference before releasing the node. **[GL #5194] [GL !10156]**

- Fix a logic error in cache_name() `b8bd65763c`

  A change in 6aba56ae8 (checking whether a rejected RRset was identical to the data it would have replaced, so that we could still cache a signature) inadvertently introduced cases where processing of a response would continue when previously it would have been skipped. **[GL #5197] [GL !10158]**

- Fix a bug in the statistics channel when querying zone transfers information. `b50d9b601d`

  When querying zone transfers information from the statistics channel there was a rare possibility that *named* could terminate unexpectedly if a zone transfer was in a state when transferring from all the available primary servers had failed earlier. This has been fixed. **[GL #5198] [GL !10194]**

- Fix assertion failure when dumping recursing clients. `5d913c3383`

  Previously, if a new counter was added to the hashtable while dumping recursing clients via the *rndc recursing* command, and *fetches-per-zone* was enabled, an assertion failure could occur. This has been fixed. **[GL #5200] [GL !10168]**

- Call isc__iterated_hash_initialize in isc__work_cb. `693a1d41ed`

  isc_iterated_hash didn't work in offloaded threads as the per thread initialisation has not been done. This has been fixed. **[GL #5214] [GL !10210]**

- Fix a bug in get_request_transport_type() `aa3c6584c6`

  When *dns_remote_done()* is true, calling *dns_remote_curraddr()* asserts. Add a *dns_remote_curraddr()* check before calling *dns_remote_curraddr()*. **[GL #5215] [GL !10223]**

- Dump the active resolver fetches from dns_resolver_dumpfetches() `b2033b7e4c`

  Previously, active resolver fetches were only dumped when the *fetches-per-zone* configuration option was enabled. Now, active resolver fetches are dumped along with the number of *clients-per-server* counters per resolver fetch. **[GL !10148]**

- Fix wrong logging severity in do_nsfetch() `fd623c6ecc`

  **[GL !10118]**

- Post [CVE-2024-12705] Performance Drop Fixes, Part 2. `8cc425a5bb`

  This merge request addresses several key performance bottlenecks in the DoH (DNS over HTTPS) implementation by introducing significant optimizations and improvements.

  ### Key Improvements

  1. **Simplification and Optimisation of `http_do_bio()` Function**: - The code flow in the *http_do_bio()* function has been significantly simplified. 2. **Flushing HTTP Write Buffer on Outgoing DNS Messages**: - The buffer is flushed and a send operation is performed when there is an outgoing DNS message. 3. **Bumping Active Streams Processing Limit**: - The total number of active streams has been increased to 60% of the total streams limit.

  These changes collectively enhance the performance and reliability of the DoH implementation, making it more efficient and robust for handling high-load scenarios, particularly noticeable in long runs (>= 1h) of *stress:long:rpz:doh+udp:linux:** tests. It improves perf. for tests for BIND 9.18, but it likely will have a positive but less pronounced effect on newer versions as well.

  In essence, the merge request fixes three bottlenecks stacked upon each other.

  *It is a logical continuation of the merge requests !10109.* !10109, unfortunately, did not completely [address the performance drop in 9.18](https://gitlab.isc.org/isc-projects/bind9/-/pipelines/221545) for longer runs of the stress test. This merge request [addresses that](https://gitlab.isc.org/isc-projects/bind9/-/pipelines/223661).

  **P.S.**

  The origin of the fixes is, in fact, the branch in !10193. So this MR is a ... *forward port* of them. **[GL !10199]**

- Post [CVE-2024-12705] Performance Drop Fixes. `9d4aa15c1f`

  This merge request fixes a [performance drop](https://gitlab.isc.org/isc-projects/bind9/-/pipelines/216728) after merging the fixes for #4795, in particular in 9.18.

  The MR [fixes the problem](https://gitlab.isc.org/isc-projects/bind9/-/pipelines/219825) without affecting performance for the newer versions, in particular for [the development version](https://gitlab.isc.org/isc-projects/bind9/-/pipelines/220619). **[GL !10129]**

- Sync the TSAN CC, CFLAGS and LDFLAGS in the respdiff:tsan job. `ff58e0ed2b`

  **[GL !10212]**

## 15.24 BIND 9.20.6

### 15.24.1 New Features

- Adds support for EDE code 1 and 2. `b3eab79bc18`

  Add support for EDE codes 1 & 2 which might occurs during DNSSEC validation in case of unsupported RRSIG algorithm or DNSKEY digest. **[GL #2715] [GL !9996]**

- Add a rndc command to toggle jemalloc profiling. `38c51c84014`

  The new command is *rndc memprof*. The memory profiling status is also reported inside *rndc status*. The status also shows whether named can toggle memory profiling or not and if the server is built with jemalloc. **[GL #4759] [GL !10000]**

- Add support for multiple extended DNS errors. `4d945128dc1`

  Extended DNS error mechanism (EDE) may have several errors raised during a DNS resolution. *named* is now able to add up to three EDE codes in a DNS response. In the case of duplicate error codes, only the first one will be part of the DNS response. **[GL #5085] [GL !9978]**

- Print the expiration time of the stale records. `b5cce0f5972`

  Print the expiration time of the stale RRsets in the cache dump. **[GL !10061]**

### 15.24.2 Feature Changes

- Refactor reference counting in both QPDB and RBTDB. `3244f7848fd`

  Clean up the pattern in the newref() and decref() functions in QP and RBTDB databases. Replace the *db_nodelock_t* structure with plain reference counting for every active database node in QPDB.

  Related to #5134 **[GL !10035]**

- Shutdown the fetch context after canceling the last fetch. `55b7cc9596e`

  Shutdown the fetch context immediately after the last fetch has been canceled from that particular fetch context. **[GL !9977]**

### 15.24.3 Bug Fixes

- Fix possible truncation in dns_keymgr_status() `1333dac316c`

  If the generated status output exceeds 4096 it was silently truncated, now we output that the status was truncated. **[GL #4180] [GL !9981]**

- Recently expired records could be returned with timestamp in future. `9a4df4caac0`

  Under rare circumstances, the RRSet that expired at the time of the query could be returned with TTL far in the future. This has been fixed.

  As a side-effect, the expiration time of expired RRSets are no longer printed out in the cache dump. **[GL #5094] [GL !10059]**

- Yaml string not terminated in negative response in delv. `74640b3613c`

  **[GL #5098] [GL !9979]**

- Fix a bug in dnssec-signzone related to keys being offline. `ddda6cb59e5`

  In the case when *dnssec-signzone* is called on an already signed zone, and the private key file is unavailable, a signature that needs to be refreshed may be dropped without being able to generate a replacement. This has been fixed. **[GL #5126] [GL !9982]**

- Apply the memory limit only to ADB database items. `0ab22458f51`

  Resolver under heavy-load could exhaust the memory available for storing the information in the Address Database (ADB) effectively evicting already stored information in the ADB. The memory used to retrieve and provide information from the ADB is now not a subject of the same memory limits that are applied for storing the information in the Address Database. **[GL #5127] [GL !9975]**

- Avoid unnecessary locking in the zone/cache database. `60b81239de1`

  Prevent lock contention among many worker threads referring to the same database node at the same time. This would improve zone and cache database performance for the heavily contended database nodes. **[GL #5130] [GL !9964]**

- Fix EDE 22 time out detection. `8662424442c`

  Extended DNS error 22 (No reachable authority) was previously detected when *fctx_expired* fired. It turns out this function is used as a "safety net" and the timeout detection should be caught earlier.

  It was working though, because of another issue fixed by !9927. But then, the recursive request timed out detection occurs before *fctx_expired* making impossible to raise the EDE 22 error.

  This fixes the problem by triggering the EDE 22 in the part of the code detecting the (TCP or UDP) time out and taking the decision to cancel the whole fetch (i.e. There is no other server to attempt to contact).

  Note this is not targeting users (no release note) because there is no release versions of BIND between !9927 and this changes. Thus a release note would be confusing. **[GL #5137] [GL !10001]**

- Split and simplify the use of EDE list implementation. `23a9bed310b`

  Instead of mixing the dns_resolver and dns_validator units directly with the EDE code, split-out the dns_ede functionality into own separate compilation unit and hide the implementation details behind abstraction.

  Additionally, the new dns_edelist_t doesn't have to be copied into all responses as those are attached to the fetch context, but it could be only passed by reference.

  This makes the dns_ede implementation simpler to use, although sligtly more complicated on the inside. **[GL #5141] [GL !10030]**

- Fix the cache findzonecut() implementation. `619f163e680`

  The search for the deepest known zone cut in the cache could improperly reject a node if it contained any stale data, regardless of whether it was the NS RRset that was stale. **[GL #5155] [GL !10050]**

- DNSSEC EDE system tests on FIPS platform. `917181b4e27`

  Changes introducing the support of extended DNS error code 1 and 2 uses SHA-1 digest for some tests which break FIPS platform. The digest itself was irrelevant, another digest is used. **[GL !10031]**

- Reduce the false sharing the dns_qpcache and dns_qpzone. `5c27e9cdda6`

  Instead of having many node_lock_count * sizeof(<member>) arrays, pack all the members into a qpcache_bucket_t that is cacheline aligned to prevent false sharing between RWLocks. **[GL !10074]**

## 15.25 BIND 9.20.5

### 15.25.1 Security Fixes

- [CVE-2024-12705] DNS-over-HTTP(s) flooding fixes. `51900adf29c`

  Fix DNS-over-HTTP(S) implementation issues that arise under heavy query load. Optimize resource usage for *named* instances that accept queries over DNS-over-HTTP(S).

Previously, *named* would process all incoming HTTP/2 data at once, which could overwhelm the server, especially when dealing with clients that send requests but don't wait for responses. That has been fixed. Now, *named* handles HTTP/2 data in smaller chunks and throttles reading until the remote side reads the response data. It also throttles clients that send too many requests at once.

Additionally, *named* now carefully processes data sent by some clients, which can be considered "flooding." It logs these clients and drops connections from them. **[GL #4795]**

In some cases, *named* could leave DNS-over-HTTP(S) connections in the *CLOSE_WAIT* state indefinitely. That also has been fixed. ISC would like to thank JF Billaud for thoroughly investigating the issue and verifying the fix. **[GL #5083] [GL #4795] [GL #5083]**

- [CVE-2024-11187] Limit the additional processing for large RDATA sets. `4d3d17c344f`

  When answering queries, don't add data to the additional section if the answer has more than 13 names in the RDATA. This limits the number of lookups into the database(s) during a single client query, reducing query processing load. **[GL #5034]**

### 15.25.2 New Features

- Add Extended DNS Error Code 22 - No Reachable Authority. `ee77a192091`

  When the resolver is trying to query an authority server and eventually timed out, a SERVFAIL answer is given to the client. Add the Extended DNS Error Code 22 - No Reachable Authority to the response. **[GL #2268] [GL !9814]**

- Add a new option to configure the maximum number of outgoing queries per client request. `844a5310532`

  The configuration option 'max-query-count' sets how many outgoing queries per client request is allowed. The existing 'max-recursion-queries' is the number of permissible queries for a single name and is reset on every CNAME redirection. This new option is a global limit on the client request. The default is 200.

  This allows us to send a bit more queries while looking up a single name. The default for 'max-recursion-queries' is changed from 32 to 50. **[GL #4980] [GL #4921] [GL !9832]**

### 15.25.3 Removed Features

- Drop single-use RETERR macro. `87f70696c87`

  If the RETERR define is only used once in a file, just drop the macro. **[GL !9885]**

### 15.25.4 Feature Changes

- Update picohttpparser.{c,h} with upstream repository. `3c9657a3f48`

  **[GL #4485] [GL !9863]**

- The configuration clauses parental-agents and primaries are renamed to remote-servers. `b483cd4638c`

  The top blocks 'primaries' and 'parental-agents' are no longer preferred and should be renamed to 'remote-servers'. The zone statements 'parental-agents' and 'primaries' are still used, and may refer to any 'remote-servers' top block. **[GL #4544] [GL !9911]**

- Add none parameter to query-source and query-source-v6 to disable IPv4 or IPv6 upstream queries. `e260eb39c56`

  Add a none parameter to named configuration option *query-source* (respectively *query-source-v6*) which forbid usage of IPv4 (respectively IPv6) addresses when named is doing an upstream query. **[GL #4981]** Turning-off upstream IPv6 queries while still listening to downstream queries on IPv6. **[GL !9727] [GL !9775]**

- Optimize memory layout of core structs. `67fa22a7746`

Reduce memory footprint by: - Reordering struct fields to minimize padding. - Using exact-sized atomic types instead of *_least/*_fast variants - Downsizing integer fields where possible

Affected structs: - dns_name_t - dns_slabheader_t - dns_rdata_t - qpcnode_t - qpznode_t **[GL #5022] [GL !9793]**

- Revert "Fix NSEC3 closest encloser lookup for names with empty non-terminals" `993cb761489`

  Revert the fix for #4950 for 9.20.

  This reverts MR !9438.

  History: A performance improvement for NSEC3 closest encloser lookups (#4460) was introduced (in MR !9436) and backported to 9.20 (MR !9438) and to 9.18 in (MR !9439). It was released in 9.18.30 (and 9.20.2 and 9.21.1).

  There was a bug in the code (#4950), so we reverted the change in !9611, !9613 and !9614.

  Then a new attempt was merged in main (MR !9610) and backported to 9.20 (MR !9631) and 9.18 (MR !9632). The latter should not have been backported and was reverted in !9689.

  We now also revert the fix for 9.20 **[GL #5108] [GL !9947]**

- Add TLS SNI extension to all outgoing TLS connections. `b14148ac897`

  **[GL !9933]**

- Remove unused maxquerycount. `d61bfeb91e0`

  Related to #4980 **[GL !9853]**

- Use query counters in validator code. `d91835160a2`

  Commit af7db8951364a89c468eda1535efb3f53adc2c1f as part of #4141 was supposed to apply the 'max-recursion-queries' quota to validator queries, but the counter was never actually passed on to 'dns_resolver_createfetch()'. This has been fixed, and the global query counter ('max-query-count', per client request) is now also added.

  Related to #4980 **[GL !9866]**

### 15.25.5 Bug Fixes

- Fix nsupdate hang when processing a large update. `4ca7a5d6011`

  To mitigate DNS flood attacks over a single TCP connection, we throttle the connection when the other side does not read the data. Throttling should only occur on server-side sockets, but erroneously also happened for nsupdate, which acts as a client. When nsupdate started throttling the connection, it never attempts to read again. This has been fixed. **[GL #4910] [GL !9834]**

- Lock and attach when returning zone stats. `79e6519168e`

  When returning zone statistics counters, the statistics sets are now attached while the zone is locked. This addresses Coverity warnings CID 468720, 468728 and 468729. **[GL #4934] [GL !9843]**

- Fix possible assertion failure when reloading server while processing updates. `41af766cd08`

  **[GL #5006] [GL !9820]**

- Preserve cache across reconfig when using attach-cache. `826dfa006e2`

  When the *attach-cache* option is used in the *options* block with an arbitrary name, it causes all views to use the same cache. Previously, this configuration caused the cache to be deleted and a new cache created every time the server was reconfigured. This has been fixed. **[GL #5061] [GL !9862]**

- Resolve the spurious drops in performance due GLUE cache. `eb3c66304f3`

  For performance reasons, the returned GLUE records are cached on the first use. The current implementation could randomly cause a performance drop and increased memory use. This has been fixed. **[GL #5064] [GL !9918]**

- Fix dnssec-signzone signing non-DNSKEY RRsets with revoked keys. `c577c3b544d`

  *dnssec-signzone* was using revoked keys for signing RRsets other than DNSKEY. This has been corrected. **[GL #5070] [GL !9840]**

- Revert "Lock and attach when returning zone stats" `d954d9c20b9`

  **[GL #5082] [GL !9860]**

- Unknown directive in resolv.conf not handled properly. `7738fd28c91`

  The line after an unknown directive in resolv.conf could accidentally be skipped, potentially affecting dig, host, nslookup, nsupdate, or delv. This has been fixed. **[GL #5084] [GL !9877]**

- Fix response policy zones and catalog zones with an $INCLUDE statement defined. `cc0cbbe697c`

  Response policy zones (RPZ) and catalog zones were not working correctly if they had an $INCLUDE statement defined. This has been fixed. **[GL #5111] [GL !9941]**

- Finalize removal of memory debug flags size and mctx. `31918336e8a`

  Commit 4b3d0c66009d30f5c0bc12ee128fc59f1d853f44 has removed them, but did not remove few traces in documentation and help. Remove them from remaining places. **[GL !9842]**

- Fix m4 macro in configure.ac. `ae739c80ccb`

  **[GL !9813]**

- Mark loop as shuttingdown earlier in shutdown_cb. `fed5e55e339`

  **[GL !9891]**

- Use CMM_{STORE,LOAD}_SHARED to store/load glue in gluelist. `fa7443d3fd2`

  ThreadSanitizer has trouble understanding that gluelist->glue is constant after it is assigned to the slab-header with cmpxchg. Help ThreadSanitizer to understand the code by using CMM_STORE_SHARED and CMM_LOAD_SHARED on gluelist->glue. **[GL !9936]**

## 15.26 BIND 9.20.4

### 15.26.1 New Features

- Update bind.keys with the new 2025 IANA root key. `1f988e2cc7`

  Add an 'initial-ds' entry to bind.keys for the new root key, ID 38696, which is scheduled for publication in January 2025. **[GL #4896] [GL !9746]**

- Support jinja2 templates in pytest runner. `4a9380835f`

  Configuration files in system tests which require some variables (e.g. port numbers) filled in during test setup, can now use jinja2 templates when *jinja2* python package is available.

  Any **.j2* file found within the system test directory will be automatically rendered with the environment variables into a file without the *.j2* extension by the pytest runner. E.g. *ns1/named.conf.j2* will become *ns1/named.conf* during test setup. To avoid automatic rendering, use *.j2.manual* extension and render the files manually at test time.

  New *templates* pytest fixture has been added. Its *render()* function can be used to render a template with custom test variables. This can be useful to fill in different config options during the test. With advanced jinja2 template syntax, it can also be used to include/omit entire sections of the config file rather than using *named1.conf.in*, *named2.conf.in* etc. **[GL #4938] [GL !9699]**

### 15.26.2 Removed Features

- Move contributed DLZ modules into a separate repository. `a1cd30cd25`

  The DLZ modules are poorly maintained as we only ensure they can still be compiled, the DLZ interface is blocking, so anything that blocks the query to the database blocks the whole server and they should not be used except in testing. The DLZ interface itself is going to be scheduled for removal.

  The DLZ modules now live in https://gitlab.isc.org/isc-projects/dlz-modules repository. **[GL #4865] [GL !9777]**

### 15.26.3 Feature Changes

- Use lists of expected artifacts in system tests. `e5fa109599`

  `clean.sh` scripts have been replaced by lists of expected artifacts for each system test module. The list is defined using the custom `pytest.mark.extra_artifacts` mark, which can use both filenames and globs. **[GL #4261] [GL !9734]**

- Dnssec-ksr now supports KSK rollovers. `834c04fc77`

  The tool 'dnssec-ksr' now allows for KSK generation, as well as planned KSK rollovers. When signing a bundle from a Key Signing Request (KSR), only the key that is active in that time frame is being used for signing. Also, the CDS and CDNSKEY records are now added and removed at the correct time. **[GL #4697] [GL #4705] [GL !9711]**

- Incrementally apply AXFR transfer. `4509b92e21`

  Reintroduce logic to apply diffs when the number of pending tuples is above 128. The previous strategy of accumulating all the tuples and pushing them at the end leads to excessive memory consumption during transfer.

  This effectively reverts half of e3892805d6 **[GL #4986] [GL !9761]**

- Print expire option in transfer summary. `6cd001a68b`

  The zone transfer summary will now print the expire option value in the zone transfer summary. **[GL #5013] [GL !9714]**

- Add two new clang-format options that help with code formatting. `4230b2b514`

  - Add new clang-format option to remove redundant semicolons
  - Add new clang-format option to remove redundant parentheses

  **[GL !9750]**

- Emit more helpful log for exceeding max-records-per-type. `74e7e229f2`

  The new log message is emitted when adding or updating an RRset fails due to exceeding the max-records-per-type limit. The log includes the owner name and type, corresponding zone name, and the limit value. It will be emitted on loading a zone file, inbound zone transfer (both AXFR and IXFR), handling a DDNS update, or updating a cache DB. It's especially helpful in the case of zone transfer, since the secondary side doesn't have direct access to the offending zone data.

  It could also be used for max-types-per-name, but this change doesn't implement it yet as it's much less likely to happen in practice. **[GL !9771]**

- Harden key management when key files have become unavailabe. `11b0f41f80`

  Prior to doing key management, BIND 9 will check if the key files on disk match the expected keys. If key files for previously observed keys have become unavailable, this will prevent the internal key manager from running. **[GL !9622]**

### 15.26.4 Bug Fixes

- Use TLS for notifies if configured to do so. `c1b82c1fb8`

  Notifies configured to use TLS will now be sent over TLS, instead of plaintext UDP or TCP. Also, failing to load the TLS configuration for notify now also results in an error. **[GL #4821] [GL !9684]**

- '{&dns}' is as valid as '{?dns}' in a SVCB's dohpath. `b27cb14616`

  *dig* fails to parse a valid (as far as I can tell, and accepted by *kdig* and *Wireshark*) *SVCB* record with a *dohpath* URI template containing a *{&dns}*, like *dohpath=/some/path?key=value{&dns}"*. If the URI template contains a *{?dns}* instead *dig* is happy, but my understanding of rfc9461 and section 1.2. "Levels and Expression Types" of rfc6570 is that *{&dns}* is valid. See for example section 1.2. "Levels and Expression Types" of rfc6570.

  Note that Peter van Dijk suggested that *{dns}* and *{dns,someothervar}* might be valid forms as well, so my patch might be too restrictive, although it's anyone's guess how DoH clients would handle complex templates. **[GL #4922] [GL !9769]**

- Make dns_validator_cancel() respect the data ownership. `8002fda38c`

  There was a data race dns_validator_cancel() was called when the offloaded operations were in progress. Make dns_validator_cancel() respect the data ownership and only set new .canceling variable when the offloaded operations are in progress. The cancel operation would then finish when the offloaded work passes the ownership back to the respective thread. **[GL #4926] [GL !9790]**

- Fix NSEC3 closest encloser lookup for names with empty non-terminals. `76dc8accd3`

  The performance improvement for finding the NSEC3 closest encloser when generating authoritative responses could cause servers to return incorrect NSEC3 records in some cases. This has been fixed. **[GL #4950] [GL !9631]**

- Revert "Improve performance when looking for the closest encloser" `29c460a4e5`

  Revert "fix: chg: Improve performance when looking for the closest encloser when returning NSEC3 proofs"

  This reverts merge request !9436 **[GL #4950] [GL !9613]**

- Fix a data race in dns_zone_getxfrintime() `dd72a5eb8d`

  The dns_zone_getxfrintime() function fails to lock the zone before accessing its 'xfrintime' structure member, which can cause a data race between soa_query() and the statistics channel. Add the missing locking/unlocking pair, like it's done in numerous other similar functions. **[GL #4976] [GL !9601]**

- 'Recursive-clients 0;' triggers an assertion. `747a19bc00`

  BIND 9.20.0 broke *recursive-clients 0;*. This has now been fixed. **[GL #4987] [GL !9654]**

- Transport needs to be a selector when looking for an existing dispatch. `09fb8e354a`

  This allows for dispatch to use existing TCP/HTTPS/TLS etc. streams without accidentally using an unexpected transport. **[GL #4989] [GL !9682]**

- Parsing of hostnames in rndc.conf was broken. `b46f2376d0`

  When DSCP support was removed, parsing of hostnames in rndc.conf was accidentally broken, resulting in an assertion failure. This has been fixed. **[GL #4991] [GL !9676]**

- Restore values when dig prints command line. `f604c31ad2`

  Options of the form *[+-]option=<value>* failed to display the value on the printed command line. This has been fixed. **[GL #4993] [GL !9666]**

- Provide more visibility into configuration errors. `41fd5e9955`

  by logging SSL_CTX_use_certificate_chain_file and SSL_CTX_use_PrivateKey_file errors individually. **[GL #5008] [GL !9767]**

- Fix a data race between dns_zone_getxfr() and dns_xfrin_create() `2cb91e0631`

  There is a data race between the statistics channel, which uses *dns_zone_getxfr()* to get a reference to *zone->xfr*, and the creation of *zone->xfr*, because the latter happens outside of a zone lock.

  Split the *dns_xfrin_create()* function into two parts to separate the zone transfer starting part from the zone transfer object creation part. This allows us to attach the new object to a local variable first, then attach it to *zone->xfr* under a lock, and only then start the transfer. **[GL #5011] [GL !9728]**

- Fix race condition when canceling ADB find. `668ea24467`

  When canceling the ADB find, the lock on the find gets released for a brief period of time to be locked again inside adbname lock. During the brief period that the ADB find is unlocked, it can get canceled by other means removing it from the adbname list which in turn causes assertion failure due to a double removal from the adbname list. This has been fixed. **[GL #5024] [GL !9744]**

- Improve the memory cleaning in the SERVFAIL cache. `fa5d270f95`

  The SERVFAIL cache doesn't have a memory bound and the cleaning of the old SERVFAIL cache entries was implemented only in opportunistic manner. Improve the memory cleaning of the SERVFAIL cache to be more aggressive, so it doesn't consume a lot of memory in the case the server encounters many SERVFAILs at once. **[GL #5025] [GL !9794]**

- Fix trying the next primary server when the preivous one was marked as unreachable. `ab138bb717`

  In some cases (there is evidence only when XoT was used) *named* failed to try the next primary server in the list when the previous one was marked as unreachable. This has been fixed. **[GL #5038] [GL !9788]**

- Clean up 'nodetach' in ns_client. `47a77a3b12`

  The 'nodetach' member is a leftover from the times when non-zero 'stale-answer-client-timeout' values were supported, and currently is always 'false'. Clean up the member and its usage. **[GL !9600]**

- Fix error path bugs in the manager's "recursing-clients" list management. `d2ea42e237`

  In two places, after linking the client to the manager's "recursing-clients" list using the check_recursionquota() function, the query.c module fails to unlink it on error paths. Fix the bugs by unlinking the client from the list. **[GL !9604]**

- Remove unused <openssl/{hmac,engine}.h> headers from OpenSSL shims. `6d717e88c0`

  The <openssl/{hmac,engine}.h> headers were unused and including the <openssl/engine.h> header might cause build failure when OpenSSL doesn't have Engines support enabled.

  See https://fedoraproject.org/wiki/Changes/OpensslDeprecateEngine **[GL !9593]**

- Use attach()/detach() functions instead of touching .references. `1e9c3af75a`

  In rbtdb.c, there were places where the code touched .references directly instead of using the helper functions. Use the helper functions instead.

  Forward port from https://gitlab.isc.org/isc-private/bind9/-/merge_requests/753 **[GL !9795]**

## 15.27 BIND 9.20.3

### 15.27.1 New Features

- Log query response status to the query log. `cee11c8610f`

  Log a query response summary using the new category *responses*. Logging can be controlled by the option *responselog* and *rndc responselog*. **[GL #459] [GL !9526]**

- Added WALLET type. `dad3fafe9eb`

  Add the new record type WALLET (262). This provides a mapping from a domain name to a cryptographic currency wallet. Multiple mappings can exist if multiple records exist. **[GL #4947] [GL !9554]**

### 15.27.2 Feature Changes

- Set logging category for notify/xfer-in related messages. `1f553c61f76`

  Some 'notify' and 'xfer-in' related log messages were logged at the 'general' category instead of their own category. This has been fixed. **[GL #2730] [GL !9514]**

- Restore the number of threadpool threads back to original value. `a0eada53883`

  The issue of long-running operations potentially blocking query resolution has been fixed. Revert this temporary workaround and restore the number of threadpool threads. **[GL #4898] [GL !9532]**

- Allow IXFR-to-AXFR fallback on DNS_R_TOOMANYRECORDS. `30c4cbd4035`

  This change allows fallback from an IXFR failure to AXFR when the reason is *DNS_R_TOOMANYRECORDS*. This is because this error condition could be temporary only in an intermediate version of IXFR transactions and it's possible that the latest version of the zone doesn't have that condition. In such a case, the secondary would never be able to update the zone (even if it could) without this fallback.

  This fallback behavior is particularly useful with the recently introduced *max-records-per-type* and *max-types-per-name* options: the primary may not have these limitations and may temporarily introduce "too many" records, breaking IXFR. If the primary side subsequently deletes these records, this fallback will help recover the zone transfer failure automatically; without it, the secondary side would first need to increase the limit, which requires more operational overhead and has its own adverse effect. **[GL #4928] [GL !9471]**

- Remove statslock from dnssec-signzone. `12eb16186ff`

  Silence Coverity CID 468757 and 468767 (DATA RACE read not locked) by converting dnssec-signzone to use atomics for statistics counters rather than using a lock. **[GL #4939] [GL !9500]**

- Use release memory ordering when incrementing reference counter. `19e3cd0cd2c`

  As the relaxed memory ordering doesn't ensure any memory synchronization, it is possible that the increment will succeed even in the case when it should not - there is a race between atomic_fetch_sub(..., acq_rel) and atomic_fetch_add(..., relaxed). Only the result is consistent, but the previous value for both calls could be same when both calls are executed at the same time. **[GL !9567]**

### 15.27.3 Bug Fixes

- Fix a statistics channel counter bug when 'forward only' zones are used. `2287dc0ac0d`

  When resolving a zone with a 'forward only' policy, and finding out that all the forwarders are marked as "bad", the 'ServerQuota' counter of the statistics channel was incorrectly increased. This has been fixed. **[GL #1793] [GL !9502]**

- Fix a bug in the static-stub implementation. `72626cf9405`

  Static-stub addresses and addresses from other sources were being mixed together, resulting in static-stub queries going to addresses not specified in the configuration, or alternatively, static-stub addresses being used instead of the correct server addresses. **[GL #4850] [GL !9571]**

- Don't allow statistics-channel if libxml2 and libjson-c are unsupported. `02822b70eee`

  When the libxml2 and libjson-c libraries are not supported, the statistics channel can't return anything useful, so it is now disabled. Use of *statistics-channel* in *named.conf* is a fatal error. **[GL #4895] [GL !9486]**

- Separate DNSSEC validation from the long-running tasks. `c0022f68025`

  As part of the KeyTrap [CVE-2023-50387] mitigation, the DNSSEC CPU- intensive operations were offloaded to a separate threadpool that we use to run other tasks that could affect the networking latency.

  If that threadpool is running some long-running tasks like RPZ, catalog zone processing, or zone file operations, it would delay DNSSEC validations to a point where the resolving signed DNS records would fail.

  Split the CPU-intensive and long-running tasks into separate threadpools in a way that the long-running tasks don't block the CPU- intensive operations. **[GL #4898] [GL !9495]**

- Fix assertion failure when processing access control lists. `a15d975dbe2`

  The named process could terminate unexpectedly when processing access control lists (ACLs). This has been fixed. **[GL #4908] [GL !9466]**

- Fix bug in Offline KSK that is using ZSK with unlimited lifetime. `3f115d3cdae`

  If the ZSK has unlimited lifetime, the timing metadata "Inactive" and "Delete" cannot be found and is treated as an error, preventing the zone to be signed. This has been fixed. **[GL #4914] [GL !9453]**

- Fix data race in offloaded dns_message_checksig() `3b5c4f94d70`

  When verifying a message in an offloaded thread there is a race with the worker thread which writes to the same buffer. Clone the message buffer before offloading. **[GL #4929] [GL !9490]**

- Limit the outgoing UDP send queue size. `251b90c25e0`

  If the operating system UDP queue gets full and the outgoing UDP sending starts to be delayed, BIND 9 could exhibit memory spikes as it tries to enqueue all the outgoing UDP messages. Try a bit harder to deliver the outgoing UDP messages synchronously and if that fails, drop the outgoing DNS message that would get queued up and then timeout on the client side. **[GL #4930] [GL !9511]**

- Do not set SO_INCOMING_CPU. `6c9f3d0d1ed`

  We currently set SO_INCOMING_CPU incorrectly, and testing by Ondrej shows that fixing the issue by setting affinities is worse than letting the kernel schedule threads without constraints. So we should not set SO_INCOMING_CPU anymore. **[GL #4936] [GL !9504]**

- Fix the 'rndc dumpdb' command's error reporting. `d35f654d674`

  The 'rndc dumpdb' command wasn't reporting errors which occurred when starting up the database dump process by named, like, for example, a permission denied error for the 'dump-file' file. This has been fixed. Note, however, that 'rndc dumpdb' performs asynchronous writes, so errors can also occur during the dumping process, which will not be reported back to 'rndc', but which will still be logged by named. **[GL #4944] [GL !9553]**

- Fix long-running incoming transfers. `c5cadd29d87`

  Incoming transfers that took longer than 30 seconds would stop reading from the TCP stream and the incoming transfer would be indefinitely stuck causing BIND 9 to hang during shutdown.

  This has been fixed and the *max-transfer-time-in* and *max-transfer- idle-in* timeouts are now honoured. **[GL #4949] [GL !9536]**

- Fix assertion failure when receiving DNS responses over TCP. `e2058ab4619`

  When matching the received Query ID in the TCP connection, an invalid received Query ID can very rarely cause assertion failure. **[GL #4952] [GL !9582]**

- Don't ignore the local port number in dns_dispatch_add() for TCP. `97fad455d73`

  The dns_dispatch_add() function registers the 'resp' entry in 'disp->mgr->qids' hash table with 'resp->port' being 0, but in tcp_recv_success(), when looking up an entry in the hash table after a successfully received data the port is used, so if the local port was set (i.e. it was not 0) it fails to find the entry and results in an unexpected error.

Set the 'resp->port' to the given local port value extracted from 'disp->local'. **[GL #4969] [GL !9581]**

- Add a missing rcu_read_unlock() call on exit path. `5db2ec07395`

  An exit path in the dns_dispatch_add() function fails to get out of the RCU critical section when returning early. Add the missing rcu_read_unlock() call. **[GL !9564]**

- Don't enable REUSEADDR on outgoing UDP sockets. `a6692e793c3`

  The outgoing UDP sockets enabled *SO_REUSEADDR* that allows sharing of the UDP sockets, but with one big caveat - the socket that was opened the last would get all traffic. The dispatch code would ignore the invalid responses in the dns_dispatch, but this could lead to unexpected results. **[GL !9583]**

## 15.28 BIND 9.20.2

### 15.28.1 New Features

- Support for Offline KSK implemented. `3555094a686`

  Add a new configuration option *offline-ksk* to enable Offline KSK key management. Signed Key Response (SKR) files created with *dnssec-ksr* (or other program) can now be imported into *named* with the new *rndc skr -import* command. Rather than creating new DNSKEY, CDS and CDNSKEY records and generating signatures covering these types, these records are loaded from the currently active bundle from the imported SKR.

  The implementation is loosely based on: https://web.archive.org/web/20250121040252/https://www.iana.org/dnssec/archive/files/draft-icann-dnssec-keymgmt-01.txt **[GL #1128] [GL !9119]**

- Print the full path of the working directory in startup log messages. `1c8eeafffb0`

  named now prints its initial working directory during startup and the changed working directory when loading or reloading its configuration file if it has a valid 'directory' option defined. **[GL #4731] [GL !9372]**

- Support restricted key tag range when generating new keys. `d0899632635`

  It is useful when multiple signers are being used to sign a zone to able to specify a restricted range of range of key tags that will be used by an operator to sign the zone. This adds controls to named (dnssec-policy), dnssec-signzone, dnssec-keyfromlabel and dnssec-ksr (dnssec-policy) to specify such ranges. **[GL #4830] [GL !9396]**

### 15.28.2 Feature Changes

- Exempt prefetches from the fetches-per-zone and fetches-per-server quotas. `5e78cade523`

  Fetches generated automatically as a result of 'prefetch' are now exempt from the 'fetches-per-zone' and 'fetches-per-server' quotas. This should help in maintaining the cache from which query responses can be given. **[GL #4219] [GL !9420]**

- Restore the ability to select individual unit tests. `cfac05cc966`

  This adds the command line arguments: *-d* (debug), *-l* (list tests) and *-t test* (run this test) to the unit tests, e.g.:

```
% ./rdata_test -t zonemd
[==========] selected:
Running 1 test(s).
[ RUN      ] zonemd
[       OK ] zonemd
[==========] selected: 1 test(s) run.
[  PASSED  ] 1 test(s).
%
```

  **[GL #4579] [GL !9385]**

- Process also the ISC_R_CANCELED result code in rpz_rewrite() `eb2e0991e1a`

  Log canceled resolver queries (e.g. when shutting down a hung fetch) in DEBUG3 level instead of DEBUG1 which is used for the "unrecognized" result codes. **[GL #4797] [GL !9347]**

- Remove code to read and parse /proc/net/if_inet6 on Linux. `e3cc5034ab0`

  The getifaddr() works fine for years, so we don't have to keep the callback to parse /proc/net/if_inet6 anymore. **[GL #4852] [GL !9341]**

- Use seteuid()/setegid() instead of setreseuid()/setresgid() `1127b2b3d16`

  It looks like that all supported systems now have support for _POSIX_SAVED_IDS, so it's safe to use setegid() and setegid() because those will not change saved used/group IDs. **[GL #4862] [GL !9371]**

- Follow the number of CPU set by taskset/cpuset. `ce3209b1dcf`

  Administrators may wish to constrain the set of cores that BIND 9 runs on via the 'taskset', 'cpuset' or 'numactl' programs (or equivalent on other O/S).

  If the admin has used taskset, the *named* will now follow to automatically use the given number of CPUs rather than the system wide count. **[GL #4884] [GL !9442]**

- Double the number of threadpool threads. `cfdded46676`

  Introduce this temporary workaround to reduce the impact of long- running tasks in offload threads which can block the resolution of queries. **[GL #4898]**

### 15.28.3 Bug Fixes

- Delay release of root privileges until after configuring controls. `0b7eb9d7a90`

  Delay relinquishing root privileges until the control channel has been configured, for the benefit of systems that require root to use privileged port numbers. This mostly affects systems without fine- grained privilege systems (i.e., other than Linux). **[GL #4793] [GL !9444]**

- Fix the assertion failure in the isc_hashmap iterator. `92e54fa9b7f`

  When the round robin hashing reorders the map entries on deletion, we were adjusting the iterator table size only when the reordering was happening at the internal table boundary. The iterator table size had to be reduced by one to prevent seeing the entry that resized on position [0] twice because it migrated to [iter->size - 1] position.

  However, the same thing could happen when the same entry migrates a second time from [iter->size - 1] to [iter->size - 2] position (and so on) because the check that we are manipulating the entry just in the [0] position was insufficient. Instead of checking the position [pos == 0], we now check that the [pos % iter->size == 0], thus ignoring all the entries that might have moved back to the end of the internal table. **[GL #4838] [GL !9310]**

- Add -Wno-psabi to CFLAGS for x86 (32-bit) builds. `9f2061e31eb`

  GCC 11.1+ emits a note during compilation when there are 64-bit atomic fields in a structure, because it fixed a compiler bug by changing the alignment of such fields, which caused ABI change.

  Add -Wno-psabi to CFLAGS for such builds in order to silence the warning. That shouldn't be a problem since we don't expose our structures to the outside. **[GL #4841] [GL !9322]**

- Check if logconfig is NULL before using it in isc_log_doit() `11cb3767256`

  Check if 'lctx->logconfig' is NULL before using it in isc_log_doit(), because it's possible that isc_log_destroy() was already called, e.g. when a 'call_rcu' function wants to log a message during shutdown. **[GL #4842] [GL !9323]**

- Change the NS_PER_SEC (and friends) from enum to static const. `91cecebf4c6`

New version of clang (19) has introduced a stricter checks when mixing integer (and float types) with enums. In this case, we used enum {} as C17 doesn't have constexpr yet. Change the time conversion constants to be static const unsigned int instead of enum values. **[GL #4845] [GL !9339]**

- Check the result of dirfd() before calling unlinkat() `335796f32a1`

  Instead of directly using the result of dirfd() in the unlinkat() call, check whether the returned file descriptor is actually valid. That doesn't really change the logic as the unlinkat() would fail with invalid descriptor anyway, but this is cleaner and will report the right error returned directly by dirfd() instead of EBADF from unlinkat(). **[GL #4853] [GL !9343]**

- Fix rare assertion failure when shutting down incoming transfer. `02d4755cc31`

  A very rare assertion failure can be triggered when the incoming transfer is either forcefully shut down or it is finished during printing the details about the statistics channel. This has been fixed. **[GL #4860] [GL !9377]**

- Fix the resesuid() shim implementation for NetBSD. `d959c035e89`

  The shim implementation of setresuid() was wrong - there was a copy and paste error and it was calling setresgid() instead. This only affects NetBSD because Linux, FreeBSD and OpenBSD have setresuid() and setresgid() implementation available from the system library. **[GL #4862] [GL !9361]**

- Fix algoritm rollover bug when there are two keys with the same keytag. `2f2003c55d4`

  If there is an algorithm rollover and two keys of different algorithm share the same keytags, then there is a possibility that if we check that a key matches a specific state, we are checking against the wrong key. This has been fixed by not only checking for matching key tag but also key algorithm. **[GL #4878] [GL !9393]**

- Stop using malloc_usable_size and malloc_size. `1b7fa52d8ff`

  The *malloc_usable_size()* can return size larger than originally allocated and when these sizes disagree the fortifier enabled by *_FORTIFY_SOURCE=3* detects overflow and stops the *named* execution abruptly. Stop using these convenience functions as they are primary used for introspection-only. **[GL #4880] [GL !9418]**

- Preserve statement tag order in documentation. `57a9e3da00c`

  This supports bit-for-bit reproducibility of built documentation. **[GL #4886] [GL !9408]**

- Fix an assertion failure in validate_dnskey_dsset_done() `870f0be27eb`

  Under rare circumstances, named could terminate unexpectedly when validating a DNSKEY resource record if the validation was canceled in the meantime. This has been fixed. **[GL #4911]**

- Silence all warnings that stem from the default config. `dde38470476`

  As we now setup the logging very early, parsing the default config would always print warnings about experimental (and possibly deprecated) options in the default config. This would even mess with commands like *named -V* and it is also wrong to warn users about using experimental options in the default config, because they can't do anything about this. Add CFG_PCTX_NODEPRECATED and CFG_PCTX_NOEXPERIMENTAL options that we can pass to cfg parser and silence the early warnings caused by using experimental options in the default config. **[GL !9305]**

## 15.29 BIND 9.20.1

### 15.29.1 New Features

- Tighten 'max-recursion-queries' and add 'max-query-restarts' option. `42e70b0f0e`

  There were cases in resolver.c when the *max-recursion-queries* quota was ineffective. It was possible to craft zones that would cause a resolver to waste resources by sending excessive queries while attempting to resolve a name. This has been addressed by correcting errors in the implementation of *max-recursion-queries*, and by reducing the default value from 100 to 32.

In addition, a new *max-query-restarts* option has been added which limits the number of times a recursive server will follow CNAME or DNAME records before terminating resolution. This was previously a hard-coded limit of 16, and now defaults to 11. **[GL #4741] [GL !9282]**

- Implement rndc retransfer -force. `008bfb6249`

  A new optional argument '-force' has been added to the command channel command 'rndc retransfer'. When it is specified, named aborts the ongoing zone transfer (if there is one), and starts a new transfer. **[GL #2299] [GL !9219]**

- Generate changelog from git log. `cf60eb2738`

  Use a single source of truth, the git log, to generate the list of CHANGES. Use the .rst format and include it in the ARM for a quick reference with proper gitlab links to issues and merge requests. **[GL #75] [GL !9180]**

### 15.29.2 Feature Changes

- Call rcu_barrier() in the isc_mem_destroy() just once. `e00b13ac6e`

  The previous work in this area was led by the belief that we might be calling call_rcu() from within call_rcu() callbacks. After carefully checking all the current callback, it became evident that this is not the case and the problem isn't enough rcu_barrier() calls, but something entirely else.

  Call the rcu_barrier() just once as that's enough and the multiple rcu_barrier() calls will not hide the real problem anymore, so we can find it. **[GL !9247]**

- Don't open route socket if we don't need it. `4f369af51e`

  When automatic-interface-scan is disabled, the route socket was still being opened. Add new API to connect / disconnect from the route socket only as needed.

  Additionally, move the block that disables periodic interface rescans to a place where it actually have access to the configuration values. Previously, the values were being checked before the configuration was loaded. **[GL !9239]**

- Allow shorter resolver-query-timeout configuration. `840e56a979`

  The minimum allowed value of 'resolver-query-timeout' was lowered to 301 milliseconds instead of the earlier 10000 milliseconds (which is the default). As earlier, values less than or equal to 300 are converted to seconds before applying the limit. **[GL #4320] [GL !9220]**

- Replace *#define DNS_GETDB_* with struct of bools. `6d1fdb8505`

  Replace *#define DNS_GETDB_* with struct of bools to make it easier to pretty-print the attributes in a debugger. **[GL #4559] [GL !9205]**

- Fix data race in clean_finds_at_name. `be1e649974`

  Stop updating *find.result_v4* and *find.result_v4* in *clean_finds_at_name*. The values are supposed to be static. **[GL #4118] [GL !9197]**

### 15.29.3 Bug Fixes

- Reconfigure catz member zones during named reconfiguration. `9a0c59c89a`

  During a reconfiguration named wasn't reconfiguring catalog zones' member zones. This has been fixed. **[GL #4733]**

- Disassociate the SSL object from the cached SSL_SESSION. `54b24fb015`

  When the SSL object was destroyed, it would invalidate all SSL_SESSION objects including the cached, but not yet used, TLS session objects.

  Properly disassociate the SSL object from the SSL_SESSION before we store it in the TLS session cache, so we can later destroy it without invalidating the cached TLS sessions. **[GL #4834] [GL !9274]**

- Attach/detach to the listening child socket when accepting TLS. `24ac7a7cd2`

  When TLS connection (TLSstream) connection was accepted, the children listening socket was not attached to sock->server and thus it could have been freed before all the accepted connections were actually closed.

  In turn, this would cause us to call isc_tls_free() too soon - causing cascade errors in pending SSL_read_ex() in the accepted connections.

  Properly attach and detach the children listening socket when accepting and closing the server connections. **[GL #4833] [GL !9273]**

- Fix --enable-tracing build on systems without dtrace. `d8d49c9340`

  Missing file util/dtrace.sh prevented builds on system without dtrace utility. This has been corrected.

- Make hypothesis optional for system tests. `c5f1cb8a04`

  Ensure that system tests can be executed without Python hypothesis package. **[GL #4831] [GL !9267]**

- Dig now reports missing query section for opcode QUERY. `b277a6f1f0`

  Query responses should contain the question section with some exceptions. Dig was not reporting this. **[GL #4808] [GL !9269]**

- Fix assertion failure in the glue cache. `f8a0c0bed6`

  Fix an assertion failure that could happen as a result of data race between free_gluetable() and addglue() on the same headers. **[GL #4691] [GL !9256]**

- Don't use 'create' flag unnecessarily in findnode() `4281aaab45`

  when searching the cache for a node so that we can delete an rdataset, it isn't necessary to set the 'create' flag. if the node doesn't exist yet, we won't be able to delete anything from it anyway. **[GL !9253]**

- Raise the log level of priming failures. `074c7cc12c`

  When a priming query is complete, it's currently logged at level ISC_LOG_DEBUG(1), regardless of success or failure. We are now raising it to ISC_LOG_NOTICE in the case of failure. [GL #3516] **[GL #3516] [GL !9250]**

- Fix assertion failure when checking named-checkconf version. `42e84e4b97`

  Checking the version of *named-checkconf* would end with assertion failure. This has been fixed. **[GL #4827] [GL !9246]**

- Valid TSIG signatures with invalid time cause crash. `2438db2eae`

  An assertion failure triggers when the TSIG has valid cryptographic signature, but the time is invalid. This can happen when the times between the primary and secondary servers are not synchronised. **[GL #4811] [GL !9245]**

- Don't skip the counting if fcount_incr() is called with force==true. `9cd2880a82`

  The fcount_incr() was incorrectly skipping the accounting for the fetches-per-zone if the force argument was set to true. We want to skip the accounting only when the fetches-per-zone is completely disabled, but for individual names we need to do the accounting even if we are forcing the result to be success. **[GL #4786] [GL !9241]**

- Don't skip the counting if fcount_incr() is called with force==true (v2) `1db5c6a0d3`

  The fcount_incr() was not increasing counter->count when force was set to true, but fcount_decr() would try to decrease the counter leading to underflow and assertion failure. Swap the order of the arguments in the condition, so the !force is evaluated after incrementing the .count. **[GL #4846] [GL !9299]**

- Fix PTHREAD_MUTEX_ADAPTIVE_NP and PTHREAD_MUTEX_ERRORCHECK_NP usage. `46caf5f4a4`

  The PTHREAD_MUTEX_ADAPTIVE_NP and PTHREAD_MUTEX_ERRORCHECK_NP are usually not defines, but enum values, so simple preprocessor check doesn't work.

Check for PTHREAD_MUTEX_ADAPTIVE_NP from the autoconf AS_COMPILE_IFELSE block and define HAVE_PTHREAD_MUTEX_ADAPTIVE_NP. This should enable adaptive mutex on Linux and FreeBSD.

As PTHREAD_MUTEX_ERRORCHECK actually comes from POSIX and Linux glibc does define it when compatibility macros are being set, we can just use PTHREAD_MUTEX_ERRORCHECK instead of PTHREAD_MUTEX_ERRORCHECK_NP. **[GL !9240]**

- Remove extra newline from yaml output. `53738634c3`

  I split this into two commits, one for the actual newline removal, and one for issues I found, ruining the yaml output when some errors were outputted.

- CID 498025 and CID 498031: Overflowed constant INTEGER_OVERFLOW. `b6298b394e`

  Add INSIST to fail if the multiplication would cause the variables to overflow. **[GL #4798] [GL !9229]**

- Remove unnecessary operations. `067f87f158`

  Decrementing optlen immediately before calling continue is unneccesary and inconsistent with the rest of dns_message_pseudosectiontoyaml and dns_message_pseudosectiontotext. Coverity was also reporting an impossible false positive overflow of optlen (CID 499061). **[GL !9223]**

- Fix generation of 6to4-self name expansion from IPv4 address. `00ce93a69c`

  The period between the most significant nibble of the encoded IPv4 address and the 2.0.0.2.IP6.ARPA suffix was missing resulting in the wrong name being checked. Add system test for 6to4-self implementation. **[GL #4766] [GL !9217]**

- Fix false QNAME minimisation error being reported. `fb07c38697`

  Remove the false positive "success resolving" log message when QNAME minimisation is in effect and the final result is NXDOMAIN. **[GL #4784] [GL !9215]**

- Dig +yaml was producing unexpected and/or invalid YAML output. `a42afbce2e`

  **[GL #4796] [GL !9213]**

- SVBC alpn text parsing failed to reject zero length alpn. `1a1413ff59`

  **[GL #4775] [GL !9209]**

- Return SERVFAIL for a too long CNAME chain. `d7e5f7903d`

  When cutting a long CNAME chain, named was returning NOERROR instead of SERVFAIL (alongside with a partial answer). This has been fixed. **[GL #4449] [GL !9203]**

- Properly calculate the amount of system memory. `c63b7fad49`

  On 32 bit machines isc_meminfo_totalphys could return an incorrect value. **[GL #4799] [GL !9199]**

- Update key lifetime and metadata after dnssec-policy reconfig. `a5f554959e`

  Adjust key state and timing metadata if dnssec-policy key lifetime configuration is updated, so that it also affects existing keys. **[GL #4677] [GL !9191]**

## 15.30 Changes prior to 9.20.1

```
        --- 9.20.0 released ---

6404.   [placeholder]

6403.   [security]      qctx-zversion was not being cleared when it should have
                        been leading to an assertion failure if it needed to be
```

(continues on next page)

(continued from previous page)

```
                        reused. (CVE-2024-4076) [GL #4507]

6402.   [security]      A malicious DNS client that sends many queries with a
                        SIG(0)-signed message can cause the server to respond
                        slowly or not respond at all to other clients. Use the
                        offload threadpool for SIG(0) signature verifications,
                        add the 'sig0checks-quota' configuration option to
                        introduce a quota for SIG(0)-signed queries running in
                        parallel and add the 'sig0checks-quota-exempt' option to
                        exempt certain clients by their IP/network addresses.
                        (CVE-2024-1975) [GL #4480]

6401.   [security]      An excessively large number of rrtypes per owner can
                        slow down database query processing, so a limit has been
                        placed on the number of rrtypes that can be stored per
                        owner (node) in a cache or zone database. This is
                        configured with the new "max-rrtypes-per-name" option,
                        and defaults to 100. (CVE-2024-1737)
                        [GL #3403] [GL #4548]

6400.   [security]      Excessively large rdatasets can slow down database
                        query processing, so a limit has been placed on the
                        number of records that can be stored per rdataset
                        in a cache or zone database. This is configured
                        with the new "max-records-per-type" option, and
                        defaults to 100. (CVE-2024-1737)
                        [GL #497] [GL #3405]

6399.   [security]      Malicious DNS client that sends many queries over
                        TCP but never reads responses can cause server to
                        respond slowly or not respond at all for other
                        clients. (CVE-2024-0760) [GL #4481]

6398.   [bug]           Fix potential data races in our DoH implementation
                        related to HTTP/2 session object management and
                        endpoints set object management after reconfiguration.
                        We would like to thank Dzintars and Ivo from nic.lv
                        for bringing this to our attention. [GL #4473]

6397.   [placeholder]

6396.   [func]          Outgoing zone transfers are no longer enabled by
                        default. To enable them, an "allow-transfer" ACL
                        must be specified. [GL #4728]

6395.   [bug]           Handle ISC_R_HOSTDOWN and ISC_R_NETDOWN in resolver.c.
                        [GL #4736]

6394.   [bug]           Named's -4 and -6 options now apply to zone primaries,
                        also-notify and parental-agents.  Report when a zone
                        has these options configured but does not have an IPv4
                        or IPv6 address listed respectively. [GL #3472]
```

(continues on next page)

(continued from previous page)

```
6393.   [func]          Deal with uv_tcp_close_reset() error return codes
                        more gracefully. [GL #4708]

6392.   [bug]           Use a completely new memory context when flushing the
                        cache. [GL #2744]

6391.   [placeholder]

6390.   [placeholder]

6389.   [bug]           dnssec-verify and dnssec-signzone could fail if there
                        was an obscured DNSKEY RRset at a delegatation.
                        [GL #4517]

6388.   [placeholder]

6387.   [func]          Added a new statistics variable "recursive high-water"
                        that reports the maximum number of simultaneous
                        recursive clients BIND has handled while running.
                        [GL #4668]

6386.   [bug]           When shutting down catzs->view could point to freed
                        memory. Obtain a reference to the view to prevent this.
                        [GL #4502]

6385.   [func]          Relax SVCB alias mode checks to allow parameters.
                        [GL #4704]

6384.   [bug]           Remove infinite loop when including a directory in a
                        zone file. [GL #4357]

6383.   [bug]           Address an infinite loop in $GENERATE when a negative
                        value was converted in nibble mode. [GL #4353]

6382.   [bug]           Fix RPZ response's SOA record TTL, which was incorrectly
                        set to 1 if 'add-soa' is used. [GL #3323]
```

```
        --- 9.19.24 released ---

6381.   [bug]           dns_qp_lookup() could position the iterator at the
                        wrong predecessor when searching for names with
                        uncommon characters, which are encoded as two-octet
                        sequences in QP trie keys. [GL #4702]

6380.   [func]          Queries and responses now emit distinct dnstap entries
                        for DoT and DoH. [GL #4523]

6379.   [bug]           A QP iterator bug could result in DNSSEC validation
                        failing because the wrong NSEC was returned. [GL #4659]

6378.   [func]          The option to specify the number of UDP dispatches was
```

(continues on next page)

(continued from previous page)

```
                        previously removed. An attempt to use the option now
                        prints a warning. [GL #1879]

6377.   [func]          Introduce 'dnssec-ksr', a DNSSEC tool to create
                        Key Signing Requests (KSRs) and Signed Key Responses
                        (SKRs). [GL #1128]

6376.   [func]          Allow 'dnssec-keygen' options '-f' and '-k' to be used
                        together to create a subset of keys from the DNSSEC
                        policy. [GL !8188]

6375.   [func]          Allow multiple RNDC message to be processed from
                        a single TCP read. [GL #4416]

6374.   [func]          Don't count expired / future RRSIGs in verification
                        failure quota. [GL #4586]

6373.   [func]          Offload the isc_http response processing to worker
                        thread. [GL #4680]

6372.   [func]          Implement signature jitter for dnssec-policy. [GL #4554]

6371.   [bug]           Access to the trust bytes in the ncache data needed to
                        be made thread safe. [GL #4475]

6370.   [bug]           Wrong source address used for IPv6 notify messages.
                        [GL #4669]
```

```
        --- 9.19.23 released ---

6369.   [func]          The 'fixed' value for the 'rrset-order' option has
                        been marked and documented as deprecated. [GL #4446]

6368.   [func]          The 'sortlist' option has been marked and documented
                        as deprecated. [GL #4593]

6367.   [bug]           Since the dns_validator_destroy() function doesn't
                        guarantee that it destroys the validator, rename it to
                        dns_validator_shutdown() and require explicit
                        dns_validator_detach() to follow. Implement an expected
                        behavior of the function to release a name associated
                        with the validator. [GL #4654]

6366.   [bug]           An assertion could be triggered in the QPDB cache when
                        encountering a delegation below a DNAME. [GL #4652]

6365.   [placeholder]

6364.   [protocol]      Add RESOLVER.ARPA to the built in empty zones.
                        [GL #4580]

6363.   [bug]           dig/mdig +ednsflags=<non-zero-value> did not re-enable
```

(continues on next page)

(continued from previous page)

```
                        EDNS if it had been disabled. [GL #4641]

6362.   [bug]           Reduce memory consumption of QP-trie based databases
                        by dynamically allocating the nodenames. [GL #4614]

6361.   [bug]           Some invalid ISO 8601 durations were accepted
                        erroneously. [GL #4624]

6360.   [bug]           Don't return static-stub synthesised NS RRset.
                        [GL #4608]

6359.   [bug]           Fix bug in Depends (keymgr_dep) function. [GL #4552]
```

```
        --- 9.19.22 released ---

6358.   [bug]           Fix validate_dnskey_dsset when KSK is not signing,
                        do not skip remainder of DS RRset. [GL #4625]

6357.   [func]          The QP zone database implementation introduced in
                        change #6355 has now been replaced with a version
                        based on the multithreaded dns_qpmulti API, which
                        is based on RCU and reduces the need for locking.
                        The new implementation is called "qpzone". The
                        previous "qp" implementation has been renamed
                        "qpcache", and can only be used for the cache.
                        [GL #4348]

6356.   [bug]           Attach the loop also in the dns_cache_flush(), so
                        the cache pruning still works after the flush.
                        [GL #4621]

6355.   [func]          The red-black tree data structure underlying the
                        RBTDB has been replaced with QP-tries.  This is
                        expected to improve scalability and reduce
                        CPU consumption under load. It is currently known to
                        have higher memory consumption than the traditional
                        RBTDB; this will be addressed in future releases.

                        Nodes in a QP-trie contain the full domain name,
                        while nodes in a red-black tree only contain names
                        relative to a parent.  Because of this difference,
                        zone files dumped with masterfile-style "relative"
                        will no longer have multiple different $ORIGIN
                        statements throughout the file.

                        This version is a minimal adaptation, keeping RBTDB
                        code largely unchanged, except as needed to replace
                        the underlying data structure. It uses the
                        single-thread "dns_qp" interface with locks for
                        synchronization. A future version will use the
                        multithreaded "dns_qpmulti" interface instead,
                        and will be renamed to QPDB.
```

(continues on next page)

(continued from previous page)

```
                        The RBT-based version of RBTDB is still in place
                        for now, and can be used by specifying "database rbt"
                        in a "zone" statement, or by compiling with
                        "configure --with-zonedb=rbt --with-cachedb=rbt".
                        [GL #4411]

6354.   [bug]           Change 6035 introduced a regression when chasing DS
                        records resulting in an assertion failure. [GL #4612]

6353.   [bug]           Improve the TTL-based cleaning by removing the expired
                        headers from the heap, so they don't block the next
                        cleaning round and clean more than a single item for
                        each new addition to the RBTDB. [GL #4591]

6352.   [bug]           Revert change 6319 and decrease lock contention during
                        RBTDB tree pruning by not cleaning up nodes recursively
                        within a single prune_tree() call. [GL #4596]

6351.   [protocol]      Support for the RESINFO record type has been added.
                        [GL #4413]

6350.   [bug]           Address use after free in expire_lru_headers. [GL #4495]

6349.   [placeholder]

6348.   [bug]           BIND could previously abort when trying to
                        establish a connection to a remote server using an
                        incorrect 'tls' configuration. That has been
                        fixed. Thanks to Tobias Wolter for bringing
                        the issue to our attention. [GL #4572]

6347.   [func]          Disallow stale-answer-client-timeout non-zero values.
                        [GL #4447]

6346.   [bug]           Cleaned up several minor bugs in the RBTDB dbiterator
                        implementation. [GL !8741]

6345.   [bug]           Added missing dns_rdataset_disassociate calls in
                        validator.c:findnsec3proofs. [GL #4571]

6344.   [bug]           Fix case insensitive setting for isc_ht hashtable.
                        [GL #4568]

6343.   [bug]           Fix case insensitive setting for isc_ht hashtable.
                        [GL #4568]

6342.   [placeholder]

6341.   [bug]           Address use after free in ccmsg_senddone. [GL #4549]

6340.   [test]          Fix incorrectly reported errors when running tests
```

(continues on next page)

(continued from previous page)

```
                        with `make test` on platforms with older pytest.
                        [GL #4560]

6339.   [bug]           The alignas() can't be used on types larger than
                        max_align_t; instead add padding into the structures
                        where we want avoid false memory sharing. [GL #4187]

6338.   [func]          Optimize slabheader placement, so the infrastructure
                        records are put in the beginning of the slabheader
                        linked list. [GL !8675]

6337.   [bug]           Nsupdate could assert while shutting down. [GL #4529]

6336.   [func]          Expose the zones with the 'first refresh' flag set in
                        statistics channel's "Incoming Zone Transfers" section
                        to indicate the zones that are not yet fully ready, and
                        their first refresh is pending or is in-progress. Also
                        expose the number of such zones in the output of the
                        'rndc status' command. [GL #4241]

6335.   [func]          The 'dnssec-validation yes' option now requires an
                        explicitly configured 'trust-anchors' statement (or
                        'managed-keys' or 'trusted-keys' statements, both
                        deprecated). [GL #4373]

6334.   [doc]           Improve ARM parental-agents definition. [GL #4531]

6333.   [bug]           Fix the DNS_GETDB_STALEFIRST flag, which was defined
                        incorrectly in lib/ns/query.c. [GL !8683]

6332.   [bug]           Range-check the arguments to fetch-quota-param.
                        [GL #362]

6331.   [func]          Add HSM support for dnssec-policy. You can now
                        configure keys with a key-store that allows you to
                        set the directory to store key files and to set a
                        PKCS #11 URI string. [GL #1129]

6330.   [doc]           Update ZSK minimum lifetime documentation in ARM, also
                        depends on signing delay. [GL #4510]

6329.   [func]          Nsupdate can now set the UL EDNS option when sending
                        UPDATE requests. [GL #4419]

6328.   [func]          Add workaround to enforce dynamic linker to pull
                        jemalloc earlier than libc to ensure all memory
                        allocations are done via jemalloc. [GL #4404]

6327.   [func]          Expose the TCP client count in statistics channel.
                        [GL #4425]

6326.   [bug]           Changes to "listen-on" statements were ignored on
```

(continues on next page)

(continued from previous page)

```
                        reconfiguration unless the port or interface address was
                        changed, making it impossible to change a related
                        listener transport type. Thanks to Thomas Amgarten.
                        [GL #4518] [GL #4528]

6325.   [func]          The 'tls' block was extended with a new
                        'cipher-suites' option that allows setting
                        allowed cipher suites for TLSv1.3.
                        [GL #3504]

6324.   [bug]           Fix a possible crash in 'dig +nssearch +nofail' and
                        'host -C' commands when one of the name servers returns
                        SERVFAIL. [GL #4508]
```

```
        --- 9.19.21 released ---

6323.   [placeholder]

6322.   [security]      Specific DNS answers could cause a denial-of-service
                        condition due to DNS validation taking a long time.
                        (CVE-2023-50387) [GL #4424]

                        The same code change also addresses another problem:
                        preparing NSEC3 closest encloser proofs could exhaust
                        available CPU resources. (CVE-2023-50868) [GL #4459]

6321.   [security]      Change 6315 inadvertently introduced regressions that
                        could cause named to crash. [GL #4234]

6320.   [placeholder]
```

```
        --- 9.19.20 released ---

6319.   [func]          Limit isc_async_run() overhead for RBTDB tree pruning.
                        [GL #4383]

6318.   [placeholder]

6317.   [security]      Restore DNS64 state when handling a serve-stale timeout.
                        (CVE-2023-5679) [GL #4334]

6316.   [security]      Specific queries could trigger an assertion check with
                        nxdomain-redirect enabled. (CVE-2023-5517) [GL #4281]

6315.   [security]      Speed up parsing of DNS messages with many different
                        names. (CVE-2023-4408) [GL #4234]

6314.   [bug]           Address race conditions in dns_tsigkey_find().
                        [GL #4182]

6313.   [bug]           When dnssec-policy is in effect the DNSKEY's TTLs in
```

(continues on next page)

(continued from previous page)

```
                        the zone where not being updated to match the policy.
                        This lead to failures when DNSKEYs where updated as the
                        TTLs mismatched. [GL #4466]

6312.   [bug]           Conversion from NSEC3 signed to NSEC signed could
                        temporarily put the zone into a state where it was
                        treated as unsigned until the NSEC chain was built.
                        Additionally conversion from one set of NSEC3 parameters
                        to another could also temporarily put the zone into a
                        state where it was treated as unsigned until the new
                        NSEC3 chain was built. [GL #1794] [GL #4495]

6311.   [func]          Zone content checks are now disabled by default
                        when running named-compilezone. named-checkzone
                        can still be used for checking zone integrity,
                        or the former checks in named-compilezone can be
                        re-enabled by using "named-compilezone -i full
                        -k fail -n fail -r warn -m warn -M warn -S warn
                        -T warn -W warn -C check-svcb:fail". [GL #4364]

6310.   [bug]           Memory leak in zone.c:sign_zone. When named signed a
                        zone it could leak dst_keys due to a misplaced
                        'continue'. [GL #4488]

6309.   [bug]           Changing a zone's primaries while a refresh was in
                        progress could trigger an assertion. [GL #4310]

6308.   [bug]           Prevent crashes caused by the zone journal getting
                        destroyed before all changes from an incoming IXFR are
                        written to it. [GL #4496]

6307.   [bug]           Obtain a client->handle reference when calling
                        async_restart. [GL #4439]

6306.   [func]          Log more details about the cause of "not exact" errors.
                        [GL #4500]

6305.   [placeholder]

6304.   [bug]           The wrong time was being used to determine what RRSIGs
                        where to be generated when dnssec-policy was in use.
                        [GL #4494]

6303.   [bug]           Dig failed to correctly process a SIGINT received while
                        waiting for a TCP connection to complete. [GL #4138]

6302.   [func]          The "trust-anchor-telemetry" statement is no longer
                        marked as experimental. This silences a relevant log
                        message that was emitted even when the feature was
                        explicitly disabled. [GL #4497]

6301.   [bug]           Fix data races with atomic members of the xfrin
```

(continues on next page)

(continued from previous page)

```
                        structure in xfrin_start() and xfrin_send_request()
                        functions. [GL #4493]

6300.   [bug]           Fix statistics export to use full 64 bit signed numbers
                        instead of truncating values to unsigned 32 bits.
                        [GL #4467]

6299.   [port]          NetBSD has added 'hmac' to libc which collides with our
                        use of 'hmac'. [GL #4478]

6298.   [bug]           Fix dns_qp_lookup bugs related to the iterator.
                        [GL !8558]
```

```
        --- 9.19.19 released ---

6297.   [bug]           Improve LRU cleaning behaviour. [GL #4448]

6296.   [func]          The "resolver-nonbackoff-tries" and
                        "resolver-retry-interval" options have been removed;
                        Using them is now a fatal error. [GL #4405]

6295.   [bug]           Fix an assertion failure which could occur during
                        shutdown when DNSSEC validation was running. [GL #4462]

6294.   [bug]           BIND might sometimes crash after startup or
                        re-configuration when one 'tls' entry is used multiple
                        times to connect to remote servers due to initialisation
                        attempts from contexts of multiple threads. That has
                        been fixed. [GL #4464]

6293.   [func]          Initial support for accepting the PROXYv2 protocol in
                        all currently implemented DNS transports in BIND and
                        complementary support for sending it in dig are included
                        into this release. [GL #4388]

6292.   [func]          Lower the maximum number of allowed NSEC3 iterations,
                        from 150 to 50. DNSSEC responses with a higher
                        iteration count are treated as insecure. For signing
                        with dnssec-policy, iterations must be set to zero.
                        [GL #4363]

6291.   [bug]           SIGTERM failed to properly stop multiple outstanding
                        lookup in dig. [GL #4457]

6290.   [bug]           Dig +yaml will now report "no servers could be reached"
                        also for UDP setup failure when no other servers or
                        tries are left. [GL #1229]

6289.   [test]          Remove legacy system test runner in favor of pytest.
                        [GL #4251]

6288.   [func]          Refactor the isc_mem overmem handling to always use
```

(continues on next page)

(continued from previous page)

```
                        isc_mem_isovermem and remove the water callback.
                        [GL #4451]

6287.   [bug]           Recognize escapes when reading the public key from file.
                        [GL !8502]

6286.   [bug]           Dig +yaml will now report "no servers could be reached"
                        on TCP connection failure as well as for UDP timeouts.
                        [GL #4396]

6285.   [func]          Remove AES-based DNS cookies. [GL #4421]

6284.   [bug]           Fix a catz db update notification callback registration
                        logic error, which could cause an assertion failure when
                        receiving an AXFR update for a catalog zone while the
                        previous update process of the catalog zone was already
                        running. [GL #4418]

6283.   [bug]           Fix a data race in isc_hashmap by using atomics for the
                        iterators number. [GL !8474]

6282.   [func]          Deprecate AES-based DNS cookies. [GL #4421]

6281.   [bug]           Fix a data race in dns_tsigkeyring_dump(). [GL #4328]
```

```
        --- 9.19.18 released ---

6280.   [bug]           Fix missing newlines in the output of "rndc nta -dump".
                        [GL !8454]

6279.   [func]          Use QNAME minimization when fetching nameserver
                        addresses. [GL #4209]

6278.   [bug]           The call to isc_mem_setwater() was incorrectly
                        removed from dns_cache_setcachesize(), causing
                        cache overmem conditions not to be detected. [GL #4340]

6277.   [bug]           Take into account local authoritative zones when
                        falling back to serve-stale. [GL #4355]

6276.   [cleanup]       Remove both lock-file configuration option and the
                        -X argument to named. [GL #4391]

6275.   [bug]           Fix assertion failure when using lock-file configuration
                        option together -X argument to named. [GL #4386]

6274.   [bug]           The 'lock-file' file was being removed when it
                        shouldn't have been making it ineffective if named was
                        started 3 or more times. [GL #4387]

6273.   [bug]           Don't reuse the existing TCP streams in dns_xfrin, so
                        parallel TCP transfers works again. [GL #4379]
```

(continues on next page)

(continued from previous page)

```
6272.   [func]          Enable systemd units support with the 'notify-reload'
                        service type by settng the MONOTONIC_USEC field when
                        sending an sd_notify() message to the service manager
                        to notify it about reloading the service. Note that the
                        'NotifyAccess=all' option is required in the systemd
                        unit file's '[Service]' section. [GL #4377]

6271.   [bug]           Fix a shutdown race in dns__catz_update_cb(). [GL #4381]

6270.   [bug]           Handle an assertion when the primary server returned
                        NOTIMP to IXFR or FORMERR to EDNS to SOA/IXFR/AXFR
                        request when transfering a zone. [GL #4372]

6269.   [maint]         B.ROOT-SERVERS.NET addresses are now 170.247.170.2 and
                        2801:1b8:10::b. [GL #4101]

6268.   [func]          Offload the IXFR and AXFR processing to unblock
                        the networking threads. [GL #4367]

6267.   [func]          The timeouts for resending zone refresh queries over UDP
                        were lowered to enable named to more quickly determine
                        that a primary is down. [GL #4260]

6266.   [func]          The zone option 'inline-signing' is ignored from now
                        on iff there is no 'dnssec-policy' configured for the
                        corresponding zone. [GL #4349]

6265.   [bug]           Don't schedule resign operations on the raw version
                        of an inline-signing zone. [GL #4350]

6264.   [func]          Use atomics to handle some ADB entry members
                        to reduce ADB locking contention. [GL #4326]

6263.   [func]          Convert the RPZ summary database to use a QP trie
                        instead of an RBT. [GL !8352]

6262.   [bug]           Duplicate control sockets didn't generate a
                        configuration failure leading to hard to diagnose
                        rndc connection errors.  These are now caught by
                        named-checkconf and named. [GL #4253]

6261.   [bug]           Fix a possible assertion failure on an error path in
                        resolver.c:fctx_query(), when using an uninitialized
                        link. [GL #4331]

6260.   [func]          Added options to the QP trie that will be needed
                        when it is used as a zone or cache database: backward
                        iteration, and retrieval of DNSSEC predecessor
                        nodes and node chains. [GL !8338]

6259.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
6258.   [func]          Use explictly created external memory pools for
                        dns_message in the ns_client and dns_resolver.
                        [GL #4325]

6257.   [func]          Expose the "Refresh SOA" query state (before the XFR)
                        in the incoming zone transfers section of the
                        statistics channel and show the local and remote
                        addresses for that query. Also Improve the
                        "Duration (s)" field to show the duration of the
                        "Pending" and "Refresh SOA" states too, before the
                        actual transfer starts. [GL !8305]

6256.   [func]          Expose the SOA query transport type (used before/during
                        XFR) in the incoming zone transfers section of the
                        statistics channel. [GL !8240]

6255.   [func]          Expose data about incoming zone transfers in progress
                        using statistics channel. [GL #3883]

6254.   [cleanup]       Add semantic patch to do an explicit cast from char
                        to unsigned char in ctype.h class of functions.
                        [GL #4327]

6253.   [cleanup]       Remove the support for control channel over Unix
                        Domain Sockets. [GL #4311]

6252.   [test]          Python system tests have to be executed by invoking
                        pytest directly. Executing them with the legacy test
                        runner is no longer supported. [GL #4250]

6251.   [bug]           Interating a hashmap could return the same element
                        twice. [GL #3422]

6250.   [bug]           The wrong covered value was being set by
                        dns_ncache_current for RRSIG records in the returned
                        rdataset structure. This resulted in TYPE0 being
                        reported as the covered value of the RRSIG when dumping
                        the cache contents. [GL #4314]

6249.   [cleanup]       Reduce the number of reserved UDP dispatches
                        to the number of loops, replace the round-robin
                        mechanism in dns_dispatchset_t with dispatches
                        pinned to loops, and use lock-free hash tables
                        for looking up query IDs and active TCP
                        connections. [GL !8304]

6248.   [func]          Add an option "resolver-use-dns64", which enables
                        application of DNS64 rules to server addresses
                        when sending recursive queries. This allows
                        resolution to be performed via NAT64. [GL #608]
```

(continues on next page)

(continued from previous page)

```
6247.   [func]          Implement incremental hashing in both isc_siphash
                        and isc_hash units. [GL #4306]
```

```
        --- 9.19.17 released ---

6246.   [placeholder]

6245.   [security]      Limit the amount of recursion that can be performed
                        by isccc_cc_fromwire. (CVE-2023-3341) [GL #4152]

6244.   [bug]           Adjust log levels on malformed messages to NOTICE when
                        transferring in a zone. [GL #4290]

6243.   [bug]           Restore the call order of dns_validator_destroy and
                        fetchctx_detach to prevent use after free. [GL #4214]

6242.   [func]          Ignore jemalloc versions before 4.0.0 as we now
                        need explicit memory arenas and tcache support.
                        [GL #4296]

6241.   [placeholder]

6240.   [bug]           Use dedicated per-worker thread jemalloc memory
                        arenas for send buffers allocation to reduce memory
                        consumption and avoid lock contention. [GL #4038]

6239.   [func]          Deprecate the 'dnssec-must-be-secure' option.
                        [GL #3700]

6238.   [cleanup]       Refactor several objects relying on dns_rbt trees
                        to instead of dns_nametree, a wrapper around dns_qp.
                        [GL !8213]

6237.   [bug]           Address memory leaks due to not clearing OpenSSL error
                        stack. [GL #4159]

6236.   [func]          Add isc_mem_cget() and isc_mem_cput() calloc-like
                        functions that take nmemb and size, do checked
                        multiplication and zero the memory before returning
                        it to the user.  Replace isc_mem_getx(..., ISC_MEM_ZERO)
                        with isc_mem_cget(...) usage. [GL !8237]

6235.   [doc]           Clarify BIND 9 time formats. [GL #4266]

6234.   [bug]           Restore stale-refresh-time value after flushing the
                        cache. [GL #4278]

6233.   [func]          Extend client side support for the EDNS EXPIRE option
                        to IXFR and AXFR query types. [GL #4170]

6232.   [bug]           Following the introduction of krb5-subdomain-self-rhs
                        and ms-subdomain-self-rhs update rules, removal of
```

(continues on next page)

(continued from previous page)

```
                        nonexistent PTR and SRV records via UPDATE could fail.
                        [GL #4280]

6231.   [func]          Make nsupdate honor -v for SOA requests only if the
                        server is specified. [GL #1181]

6230.   [bug]           Prevent an unnecessary query restart if a synthesized
                        CNAME target points to the CNAME owner. [GL #3835]

6229.   [func]          Add basic USDT framework for adding static
                        tracing points. [GL #4041]

6228.   [func]          Limit the number of inactive network manager handles
                        and uvreq objects that we keep around for reusing
                        later. [GL #4265]

6227.   [bug]           Check the statistics-channel HTTP Content-length
                        to prevent negative or overflowing values from
                        causing a crash. [GL #4125]

6226.   [bug]           Attach dispatchmgr in the dns_view object to prevent
                        use-after-free when shutting down. [GL #4228]

6225.   [func]          Convert dns_nta, dns_forward and dns_keytable units
                        to use QP trie instead of an RBT. [GL !7811]

6224.   [bug]           Check the If-Modified-Since value length to prevent
                        out-of-bounds write. [GL #4124]

6223.   [func]          Make -E engine option for OpenSSL Engine API use only.
                        OpenSSL Provider API will now require engine to not be
                        set. [GL #8153]

6222.   [func]          Fixes to provider/engine based ECDSA key handling.
                        [GL !8152]
```

```
        --- 9.19.16 released ---

6221.   [cleanup]       Refactor dns_rdataset internals, move rdatasetheader
                        declarations out of rbtdb.c so they can be used by other
                        databases in the future, and split the zone and cache
                        functions from rbtdb.c into separate modules. [GL !7873]

6220.   [func]          Deprecate the 'dialup' and 'heartbeat-interval'
                        options. [GL #3700]

6219.   [bug]           Ignore 'max-zone-ttl' on 'dnssec-policy insecure'.
                        [GL #4032]

6218.   [func]          Add inline-signing to dnssec-policy. [GL #3677]

6217.   [func]          The dns_badcache unit was refactored to use cds_lfht
```

(continues on next page)

(continued from previous page)

```
                        instead of hand-crafted locked hashtable. [GL #4223]

6216.   [bug]           Pin dns_request events to the originating loop
                        to serialize access to the data. [GL #4086]

6215.   [protocol]      Return REFUSED to GSS-API TKEY requests if GSS-API
                        support is not configured. [GL #4225]

6214.   [bug]           Fix the memory leak in for struct stub_glue_request
                        allocated in stub_request_nameserver_address() but not
                        freed in stub_glue_response(). [GL #4227]

6213.   [bug]           Mark a primary server as temporarily unreachable if the
                        TCP connection attempt times out. [GL #4215]

6212.   [placeholder]

6211.   [func]          Remove 'auto-dnssec'. This obsoletes the configuration
                        options 'dnskey-sig-validity', 'dnssec-dnskey-kskonly',
                        'dnssec-update-mode', 'sig-validity-interval', and
                        'update-check-ksk'. [GL #3672]

6210.   [func]          Don't add signing records for DNSKEY added with dynamic
                        update. The dynamic update DNSSEC management feature was
                        removed with GL #3686. [GL !8070]

6209.   [func]          Reduce query-response latency by making recursive
                        queries (CNAME, DNAME, NSEC) asynchronous instead
                        of directly calling the respective functions. [GL #4185]

6208.   [func]          Return BADCOOKIE for out-of-date or otherwise bad, well
                        formed DNS SERVER COOKIES. [GL #4194]
```

```
        --- 9.19.15 released ---

6207.   [cleanup]       The code implementing TSIG/TKEY support has been cleaned
                        up and refactored for improved robustness, readability,
                        and consistency with other code modules. [GL !7828]

6206.   [bug]           Add shutdown checks in dns_catz_dbupdate_callback() to
                        avoid a race with dns_catz_shutdown_catzs(). [GL #4171]

6205.   [bug]           Restore support to read legacy HMAC-MD5 K file pairs.
                        [GL #4154]

6204.   [bug]           Use NS records for relaxed QNAME-minimization mode.
                        This reduces the number of queries named makes when
                        resolving, as it allows the non-existence of NS RRsets
                        at non-referral nodes to be cached in addition to the
                        referrals that are normally cached. [GL #3325]

6203.   [cleanup]       Ensure that the size calculation does not overflow
```

(continues on next page)

(continued from previous page)

```
                        when allocating memory for an array.
                        [GL #4120] [GL #4121] [GL #4122]

6202.   [func]          Use per-loop memory contexts for dns_resolver
                        objects. [GL !8015]

6201.   [bug]           The free_all_cpu_call_rcu_data() call at the end
                        of isc_loopmgr_run() was causing ~200 ms extra
                        latency. [GL #4163]

6200.   [placeholder]

6199.   [bug]           Improve HTTP Connection: header protocol conformance
                        in the statistics channel. [GL #4126]

6198.   [func]          Remove the holes in the isc_result_t enum to compact
                        the isc_result tables. [GL #4149]

6197.   [bug]           Fix a data race between the dns_zone and dns_catz
                        modules when registering/unregistering a database
                        update notification callback for a catalog zone.
                        [GL #4132]

6196.   [cleanup]       Report "permission denied" instead of "unexpected error"
                        when trying to update a zone file on a read-only file
                        system. Thanks to Midnight Veil. [GL #4134]

6195.   [bug]           Use rcu to reference view->adb. [GL #4021]

6194.   [func]          Change function 'find_zone_keys()' to look for signing
                        keys by looking for key files instead of a DNSKEY
                        RRset lookup. [GL #4141]

6193.   [bug]           Fix a catz db update notification callback registration
                        logic error, which could crash named when receiving an
                        AXFR update for a catalog zone while the previous update
                        process of the catalog zone was already running.
                        [GL #4136]
```

```
       --- 9.19.14 released ---

6192.   [placeholder]

6191.   [placeholder]

6190.   [security]      Improve the overmem cleaning process to prevent the
                        cache going over the configured limit. (CVE-2023-2828)
                        [GL #4055]

6189.   [bug]           Fix an extra dns_validator deatch when encountering
                        deadling which would lead to assertion failure.
                        [GL #4115]
```

(continues on next page)

(continued from previous page)

```
6188.   [performance]   Reduce memory consumption by allocating properly
                        sized send buffers for stream-based transports.
                        [GL #4038]

6187.   [bug]           Address view shutdown INSIST when accessing the
                        zonetable. [GL #4093]

6186.   [bug]           Fix a 'clients-per-query' miscalculation bug. When the
                        'stale-answer-enable' options was enabled and the
                        'stale-answer-client-timeout' option was enabled and
                        larger than 0, named was taking two places from the
                        'clients-per-query' limit for each client and was
                        failing to gradually auto-tune its value, as configured.
                        [GL #4074]

6185.   [func]          Add "ClientQuota" statistics channel counter, which
                        indicates the number of the resolver's spilled queries
                        due to reaching the clients per query quota. [GL !7978]

6184.   [func]          Special-case code that was added to allow GSS-TSIG
                        to work around bugs in the Windows 2000 version of
                        Active Directory has been removed. The 'nsupdate -o'
                        option and 'oldgsstsig' command have been
                        deprecated, and are now treated as synonyms for
                        'nsupdate -g' and 'gsstsig' respectively. [GL #4012]

6183.   [bug]           Fix a serve-stale bug where a delegation from cache
                        could be returned to the client. [GL #3950]

6182.   [cleanup]       Remove configure checks for epoll, kqueue and
                        /dev/poll. [GL #4098]

6181.   [placeholder]

6180.   [bug]           The session key object could be incorrectly added
                        to multiple different views' keyrings. [GL #4079]

6179.   [bug]           Fix an interfacemgr use-after-free error in
                        zoneconf.c:isself(). [GL #3765]

6178.   [func]          Add support for the multi-signer model 2 (RFC 8901) when
                        using inline-signing. [GL #2710]

6177.   [placeholder]

6176.   [test]          Add support for using pytest & pytest-xdist to
                        execute the system test suite. [GL #3978]

6175.   [test]          Fix the `upforwd` system test to be more reliable,

6174.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
6173.   [bug]           Properly process extra "nameserver" lines in
                        resolv.conf otherwise the next line is not properly
                        processed. [GL #4066]

6172.   [cleanup]       Refactor the loop manager and qp-trie code to remove
                        isc_qsbr and use liburcu instead. [GL #3936]

6171.   [cleanup]       Remove the stack implementation added in change 6108:
                        we are using the liburcu concurrent data structures
                        instead. [GL !7920]

6170.   [func]          The 'rndc -t' option allows a timeout to be set in
                        seconds, so that commands that take a long time to
                        complete (e.g., reloading a very large configuration)
                        can be given time to do so. The default is 60
                        seconds. [GL #4046]

6169.   [bug]           named could crash when deleting inline-signing zones
                        with "rndc delzone". [GL #4054]

6168.   [func]          Refactor the glue cache to store list of the GLUE
                        directly in the rdatasetheader instead of keeping
                        it in the hashtable indexed by the node pointer.
                        [GL #4045]

6167.   [func]          Add 'cdnskey' configuration option. [GL #4050]

6166.   [func]          Retry without DNS COOKIE on FORMERR if it appears that
                        the FORMERR was due to the presence of a DNS COOKIE
                        option. [GL #4049]

6165.   [bug]           Fix a logic error in dighost.c which could call the
                        dighost_shutdown() callback twice and cause problems
                        if the callback function was not idempotent. [GL #4039]
```

```
        --- 9.19.13 released ---

6164.   [bug]           Set the rndc idle read timeout back to 60 seconds,
                        from the netmgr default of 30 seconds, in order to
                        match the behavior of 9.16 and earlier. [GL #4046]

6163.   [func]          Add option to dnstap-read to use timestamps in
                        milliseconds (thanks to Oliver Ford). [GL #2360]

6162.   [placeholder]

6161.   [bug]           Fix log file rotation when using absolute path as
                        file. [GL #3991]

6160.   [bug]           'delv +ns' could print duplicate output. [GL #4020]
```

(continues on next page)

(continued from previous page)

```
6159.   [bug]           Fix use-after-free bug in TCP accept connection
                        failure. [GL #4018]

6158.   [func]          Add ISC_LIST_FOREACH() and ISC_LIST_FOREACH_SAFE()
                        to walk the ISC_LIST() in a unified manner and use
                        the safe macro to fix the potential UAF when shutting
                        down the isc_httpd. [GL #4031]

6157.   [bug]           When removing delegations in an OPTOUT range
                        empty-non-terminal NSEC3 records generated by
                        those delegations were not removed. [GL #4027]

6156.   [bug]           Reimplement the maximum and idle timeouts for incoming
                        zone tranfers. [GL #4004]

6155.   [bug]           Treat ISC_R_INVALIDPROTO as a networking error
                        in the dispatch code to avoid retrying with the
                        same server. [GL #4005]

6154.   [func]          Add spinlock implementation.  The spinlock is much
                        smaller (8 bytes) than pthread_mutex (40 bytes), so
                        it can be easily embedded into objects for more
                        fine-grained locking (per-object vs per-bucket).

                        On the other hand, the spinlock is unsuitable for
                        situations where the lock might be held for a long
                        time as it keeps the waiting threads in a spinning
                        busy loop. [GL #3977]

6153.   [bug]           Fix the streaming protocols (TCP, TLS) shutdown
                        sequence. [GL #4011]

6152.   [bug]           In dispatch, honour the configured source-port
                        selection when UDP connection fails with address
                        in use error.

                        Also treat ISC_R_NOPERM same as ISC_R_ADDRINUSE.
                        [GL #3986]

6151.   [bug]           When the same ``notify-source`` address and port number
                        was configured for multiple destinations and zones, an
                        unresponsive server could tie up the socket until it
                        timed out; in the meantime, NOTIFY messages for other
                        servers silently failed.``named`` will now retry these
                        failing messages over TCP.  NOTIFY failures are now
                        logged at level INFO. [GL #4001] [GL #4002]

6150.   [bug]           If the zones have active upstream forwards, the
                        shutting down the server might cause assertion
                        failures as the forward were all canceled from
                        the main loop instead from the loops associated
                        with the zone. [GL #4015]
```

(continues on next page)

(continued from previous page)

```
6149.   [test]          As a workaround, include an OpenSSL header file before
                        including cmocka.h in the unit tests, because OpenSSL
                        3.1.0 uses __attribute__(malloc), conflicting with a
                        redefined malloc in cmocka.h. [GL #4000]

6148.   [bug]           Fix a use-after-free bug in dns_xfrin_create().
                        [GL !7832]

6147.   [performance]   Fix the TCP server parent quota use. [GL #3985]
```

```
        --- 9.19.12 released ---

6146.   [performance]   Replace the zone table red-black tree and associated
                        locking with a lock-free qp-trie. [GL !7582]

6145.   [bug]           Fix a possible use-after-free bug in the
                        dns__catz_done_cb() function. [GL #3997]

6144.   [bug]           A reference counting problem (double detach) might
                        occur when shutting down zone transfer early after
                        switching the dns_xfrin to use dns_dispatch API.
                        [GL #3984]

6143.   [bug]           A reference counting problem on the error path in
                        the xfrin_connect_done() might cause an assertion
                        failure on shutdown.  [GL #3989]

6142.   [bug]           Reduce the number of dns_dnssec_verify calls made
                        determining if revoked keys needs to be removed from
                        the trust anchors. [GL #3981]

6141.   [bug]           Fix several issues in nsupdate timeout handling and
                        update the -t option's documentation. [GL #3674]

6140.   [func]          Implement automatic parental-agents ('checkds yes').
                        [GL #3901]

6139.   [func]          Add isc_histo_t general-purpose log-linear histograms,
                        and use them for message size statistics. [GL !7696]

6138.   [doc]           Fix the DF-flag documentation on the outgoing
                        UDP packets. [GL #3710]

6137.   [cleanup]       Remove the trampoline jump when spawning threads.
                        [GL !7293]

6136.   [cleanup]       Remove the isc_fsaccess API in favor of creating
                        temporary file first and atomically replace the key
                        with non-truncated content. [GL #3982]

6135.   [cleanup]       Change isc_stdtime_get(&t) to t = isc_stdtime_now().
```

(continues on next page)

(continued from previous page)

```
                        [GL !7757]

6134.   [bug]           Fix a crash when dig or host receive a signal.
                        [GL #3970]

6133.   [cleanup]       Refactor the isc_job_run() to not make any allocations
                        by embedding isc_job_t into callback argument, and
                        running it directly.  As a side-effect, isc_async_run
                        and isc_job_run now executes jobs in the natural order.

                        Use the new improved API to execute connect, read and
                        send callbacks from netmgr in more straightforward
                        manner, speeding up the networking. [GL #3961]

6132.   [doc]           Remove a dead link in the DNSSEC guide. [GL #3967]

6131.   [test]          Add a minimal test-only library to allow testing
                        of the DNSRPS API without FastRPZ installed.
                        Thanks to Farsight Securty. [GL !7693]

6130.   [func]          The new "delv +ns" option activates name server mode,
                        in which delv sets up an internal recursive
                        resolver and uses that, rather than an external
                        server, to look up the requested data. All messages
                        sent and received during the resolution and
                        validation process are logged. This can be used in
                        place of "dig +trace"; it more accurately
                        replicates the behavior of named when resolving
                        a query. [GL #3842]

6129.   [cleanup]       Value stored to 'source' during its initialization is
                        never read. [GL #3965]

6128.   [bug]           Fix an omission in an earlier commit to avoid a race
                        between the 'dns__catz_update_cb()' and
                        'dns_catz_dbupdate_callback()' functions. [GL #3968]

6127.   [cleanup]       Refactor network manager netievent callbacks to
                        use isc_job_run()/isc_async_run(). [GL #3964]

6126.   [func]          Remove zone type "delegation-only" and the
                        "delegation-only" and "root-delegation-only"
                        options. [GL #3953]

6125.   [bug]           Hold a catz reference while the update process is
                        running, so that the catalog zone is not destroyed
                        during shutdown until the update process is finished or
                        properly canceled by the activated 'shuttingdown' flag.
                        [GL #3955]

6124.   [bug]           When changing from a NSEC3 capable DNSSEC algorithm to
                        an NSEC3 incapable DNSSEC algorithm using KASP the zone
```

(continues on next page)

(continued from previous page)

```
                        could sometimes be incompletely signed. [GL #3937]

6123.   [placeholder]

6122.   [func]          BIND now requires liburcu for lock-free data structures
                        and concurrent safe memory reclamation. It replaces the
                        home-grown lock-free linked list and QSBR machinery
                        added in changes 6108 and 6109.  [GL #3935]

6121.   [cleanup]       Remove support for TKEY Mode 2 (Diffie-Hellman Exchanged
                        Keying). [GL #3905]
```

```
        --- 9.19.11 released ---

6120.   [bug]           Use two pairs of dns_db_t and dns_dbversion_t in a
                        catalog zone structure to avoid a race between the
                        dns__catz_update_cb() and dns_catz_dbupdate_callback()
                        functions. [GL #3907]

6119.   [bug]           Make sure to revert the reconfigured zones to the
                        previous version of the view, when the new view
                        reconfiguration fails during the configuration of
                        one of the configured zones. [GL #3911]

6118.   [func]          Add 'cds-digest-types' configuration option. Also allow
                        dnssec-signzone to create multple CDS records.
                        [GL #3837]

6117.   [func]          Add a qp-trie data structure. This is a foundation for
                        our plan to replace, in stages, BIND's red-black tree.
                        The qp-trie has lock-free multithreaded reads, using
                        QSBR for safe memory reclamation. [GL !7130]

6116.   [placeholder]

6115.   [bug]           Unregister db update notify callback before detaching
                        from the previous db inside the catz update notify
                        callback. [GL #3777]

6114.   [func]          Run the catalog zone update process on the offload
                        threads. [GL #3881]

6113.   [func]          Add shutdown signaling for catalog zones. [GL !7571]

6112.   [func]          Add reference count tracing for dns_catz_zone_t and
                        dns_catz_zones_t. [GL !7570]

6111.   [cleanup]       Move irs_resconf into libdns, and remove the
                        now empty libirs. [GL !7463]

6110.   [cleanup]       Refactor the dns_xfrin module to use dns_dispatch
                        to set up TCP connections and send and receive
```

(continues on next page)

(continued from previous page)

```
                        messages. [GL #3886]

6109.   [func]          Infrastructure for QSBR, asynchronous safe memory
                        reclamation for lock-free data structures. [GL !7471]

6108.   [func]          Support for simple lock-free singly-linked stacks.
                        [GL !7470]

6107.   [cleanup]       Remove the dns_sdb API and rewrite the named
                        builtin databases to implement dns_db directly.
                        [GL #3882]

6106.   [cleanup]       Move bind9_getaddresses() to isc_getaddresses()
                        and remove the now empty libbind9. [GL !7462]

6105.   [bug]           Detach 'rpzs' and 'catzs' from the previous view in
                        configure_rpz() and configure_catz(), respectively,
                        just after attaching it to the new view. [GL #3880]

6104.   [cleanup]       Move libbind9's configuration checking code into
                        libisccfg alongside the other configuration code.
                        [GL !7461]

6103.   [func]          All uses of the isc_task and isc_event APIs have
                        been refactored to use isc_loop instead, and the
                        original APIs have been removed. [GL #3797]

6102.   [cleanup]       Several nugatory headers have been removed from libisc.
                        [GL !7464]

6101.   [port]          Clarify the portability dodge needed for `strerror_r()`
                        [GL !7465]

6100.   [cleanup]       Deprecate <isc/deprecated.h>, because obsolete
                        functions are now deleted instead of marked with
                        an attribute. [GL !7466]

6099.   [performance]   Change the internal read-write lock to modified C-RW-WP
                        algorithm that is more reader-writer fair and has better
                        performance for our workloads. [GL #1609]

6098.   [test]          Don't test HMAC-MD5 when not supported by libcrypto.
                        [GL #3871]

6097.   [port]          Improve support for yield / pause instructions in spin
                        loops on AArch64 platforms. [GL !7469]

6096.   [bug]           Fix RPZ reference counting error on shutdown in
                        dns__rpz_timer_cb(). [GL #3866]

6095.   [test]          Test various 'islands of trust' configurations when
                        using managed keys. [GL #3662]
```

(continues on next page)

(continued from previous page)

```
6094.   [bug]           Building against (or running with) libuv versions
                        1.35.0 and 1.36.0 is now a fatal error.  The rules for
                        mixing and matching compile-time and run-time libuv
                        versions have been tightened for libuv versions between
                        1.35.0 and 1.40.0. [GL #3840]

6093.   [performance]   Reduce the size of each rdataset header object
                        by 16 bytes. [GL !7505]

6092.   [bug]           dnssec-cds failed to cleanup properly. [GL #3831]

6091.   [cleanup]       Drop RHEL 7 and clones support. [GL #3729]

6090.   [bug]           Fix a bug in resolver's resume_dslookup() function by
                        making sure that dns_resolver_createfetch() is called
                        with valid parameters, as required by the function.
                        [GL #3839]

6089.   [bug]           Source ports configured for query-source,
                        transfer-source, etc, were being ignored. (This
                        feature is deprecated, but it is not yet removed,
                        so the bug still needed fixing.) [GL #3790]

6088.   [cleanup]       /etc/bind.keys is no longer needed and has been
                        removed from the distribution. named and delv can
                        still load keys from a file for testing purposes,
                        but they no longer do so by default. [GL #3850]

6087.   [cleanup]       Remove support for the `DNS_NAME_DOWNCASE` option to
                        the various dns_*_fromwire() functions. It has long
                        been unused and is unsupported since change 6022.
                        [GL !7467]

6086.   [cleanup]       Remove some remnants of bitstring labels. [GL !7196]

6085.   [func]          Add isc_time_monotonic() to simplify time measurements.
                        [GL !7468]

6084.   [bug]           When BIND was built without jemalloc, the allocator flag
                        ISC_MEM_ZERO could return non-zero memory. [GL #3845]
```

```
        --- 9.19.10 released ---

6083.   [bug]           Fix DNSRPS-enabled builds as they were inadvertently
                        broken by changes 5949 and 6042. [GL #3827]

6082.   [test]          fuzz/dns_message_checksig leaked memory when shutting
                        down. [GL #3828]

6081.   [bug]           Handle primary server address lookup failures in
                        nsupdate more gracefully. [GL #3830]
```

(continues on next page)

(continued from previous page)

```
6080.   [bug]           'named -V' leaked memory. [GL #3829]

6079.   [bug]           Force set the DS state after a 'rdnc dnssec -checkds'
                        command. [GL #3822]

6078.   [func]          Cleanup the memory statistic counters to a bare
                        minumum - InUse with Malloced as alias. [GL #3718]

6077.   [func]          Implement query forwarding to DoT-enabled upstream
                        servers. [GL #3726]

6076.   [bug]           Handle OS errors when creating UDP and TCP sockets
                        more gracefully. [GL #3800]

6075.   [bug]           Add missing node lock when setting node->wild in
                        add_wildcard_magic. [GL #3799]

6074.   [func]          Refactor the isc_nm_xfr_allowed() function to return
                        isc_result_t instead of boolean. [GL #3808]

6073.   [bug]           Set RD=1 on DS requests to parental-agents. [GL #3783]

6072.   [bug]           Avoid the OpenSSL lock contention when initializing
                        Message Digest Contexts by using explicit algorithm
                        fetching, initializing static contexts for every
                        supported algorithms, and initializing the new context
                        by copying the static copy. [GL #3795]

6071.   [func]          The use of "port" when configuring query-source,
                        transfer-source, notify-source and parental-source
                        addresses has been deprecated, along with the
                        use-v[46]-udp-ports and avoid-v[46]-udp-ports
                        options. A warning will be logged when these
                        options are used. In a future release, they
                        will be removed. [GL #3781]

6070.   [func]          DSCP parsing has now been fully removed, and
                        configuration of DSCP values in named.conf is a
                        configuration error. [GL #3789]

6069.   [bug]           Detach from the view in zone_shutdown() to
                        release the memory held by the dead view
                        early. [GL #3801]

6068.   [bug]           Downloading a zone via TLS from a server which does
                        not negotiate "dot" ALPN token could crash BIND
                        on shutdown. That has been fixed. [GL #3767]
```

```
        --- 9.19.9 released ---

6067.   [security]      Fix serve-stale crash when recursive clients soft quota
```

(continues on next page)

(continued from previous page)

```
                        is reached. (CVE-2022-3924) [GL #3619]

6066.   [security]      Handle RRSIG lookups when serve-stale is active.
                        (CVE-2022-3736) [GL #3622]

6065.   [placeholder]

6064.   [security]      An UPDATE message flood could cause named to exhaust all
                        available memory. This flaw was addressed by adding a
                        new "update-quota" statement that controls the number of
                        simultaneous UPDATE messages that can be processed or
                        forwarded. The default is 100. A stats counter has been
                        added to record events when the update quota is
                        exceeded, and the XML and JSON statistics version
                        numbers have been updated. (CVE-2022-3094) [GL #3523]

6063.   [cleanup]       The RSA and ECDSA parts of the DNSSEC has been
                        refactored for a better OpenSSL 3.x integration and
                        preliminary PKCS#11 support via for OpenSSL Providers
                        has been added. [GL #3785]

6062.   [func]          The DSCP implementation, which has been
                        nonfunctional for some time, is now marked as
                        obsolete and the implementation has been removed.
                        Configuring DSCP values in named.conf has no
                        effect, and a warning will be logged that
                        the feature should no longer be used. [GL #3773]

6061.   [bug]           Fix unexpected "Prohibited" extended DNS error
                        on allow-recursion. [GL #3743]

6060.   [bug]           Fix a use-after-free bug in dns_zonemgr_releasezone()
                        by detaching from the zone manager outside of the write
                        lock. [GL #3768]

6059.   [bug]           In some serve stale scenarios, like when following an
                        expired CNAME record, named could return SERVFAIL if the
                        previous request wasn't successful. Consider non-stale
                        data when in serve-stale mode. [GL #3678]

6058.   [bug]           Prevent named from crashing when "rndc delzone"
                        attempts to delete a zone added by a catalog zone.
                        [GL #3745]

6057.   [bug]           Fix shutdown and error path bugs in the rpz unit.
                        [GL #3735]

6056.   [bug]           Fix a race in adb.c:clean_namehooks(), so that an ADB
                        entry does not expire without holding the entries lock.
                        [GL #3754]

6055.   [cleanup]       Remove setting alternate transfer sources, make options
```

(continues on next page)

(continued from previous page)

```
                        alt-transfer-source, alt-transfer-transfer-source-v6,
                        and use-alt-transfer-source ancient. [GL #3714]

6054.   [func]          Refactor remote servers (primaries, parental-agents)
                        in zone.c. Store common code in new source files
                        remote.c and remote.h. Introduce a new way to set the
                        source address and port. [GL !7110]

6053.   [bug]           Fix an ADB quota management bug in resolver. [GL #3752]

6052.   [func]          Replace DNS over TCP and DNS over TLS transports
                        code with a new, unified transport implementation.
                        [GL #3374]

6051.   [bug]           Improve thread safety in the dns_dispatch unit.
                        [GL #3178] [GL #3636]

6050.   [bug]           Changes to the RPZ response-policy min-update-interval
                        and add-soa options now take effect as expected when
                        named is reconfigured. [GL #3740]

6049.   [bug]           Exclude ABD hashtables from the ADB memory
                        overmem checks and don't clean ADB names
                        and ADB entries used in the last 10 seconds
                        (ADB_CACHE_MINIMUM). [GL #3739]

6048.   [bug]           Fix a log message error in dns_catz_update_from_db(),
                        where serials with values of 2^31 or larger were logged
                        incorrectly as negative numbers. [GL #3742]

6047.   [bug]           Try the next server instead of trying the same
                        server again on an outgoing query timeout.
                        [GL #3637]

6046.   [bug]           TLS session resumption might lead to handshake
                        failures when client certificates are used for
                        authentication (Mutual TLS).  This has been fixed.
                        [GL #3725]

6045.   [cleanup]       The list of supported DNSSEC algorithms changed log
                        level from "warning" to "notice" to match named's other
                        startup messages. [GL !7217]

6044.   [bug]           There was an "RSASHA236" typo in a log message.
                        [GL !7206]
```

```
        --- 9.19.8 released ---

6043.   [bug]           The key file IO locks objects would never get
                        deleted from the hashtable due to off-by-one error.
                        [GL #3727]
```

(continues on next page)

(continued from previous page)

```
6042.   [bug]           ANY responses could sometimes have the wrong TTL.
                        [GL #3613]

6041.   [func]          Set the RLIMIT_NOFILE to rlim_max returned from
                        getrlimit() instead of trying to guess the maximum
                        allowed value. [GL #3676]

6040.   [bug]           Speed up the named shutdown time by explicitly
                        canceling all recursing ns_client objects for
                        each ns_clientmgr. [GL #3183]

6039.   [bug]           Removing a catalog zone from catalog-zones without
                        also removing the referenced zone could leave a
                        dangling pointer. [GL #3683]

6038.   [placeholder]

6037.   [func]          Reject zones which have DS records not at delegation
                        points. [GL #3697]

6036.   [bug]           nslookup and host were not honoring the selected port
                        in TCP mode. [GL #3721]

6035.   [bug]           Refactor the dns_resolver unit to store the fetch
                        contexts and zone counter directly in the hash
                        tables without buckets and implement effective
                        cleaning of both objects. [GL #3709]

6034.   [func]          Deprecate alt-transfer-source, alt-transfer-source-v6
                        and use-alt-transfer-source. [GL #3694]

6033.   [func]          Log messages related to serve-stale now include the RR
                        type involved. [GL !7145]

6032.   [bug]           After change 5995, zone transfers were using a small
                        compression context that only had space for the first
                        few dozen names in each message. They now use a large
                        compression context with enough space for every name.
                        [GL #3706]

6031.   [bug]           Move the "final reference detached" log message
                        from dns_zone unit to the DEBUG(1) log level.
                        [GL #3707]

6030.   [bug]           Refactor the ADB to use a global LRU queue, store
                        the ADB names and ADB entries directly in the hash
                        tables instead of buckets, and properly clean the
                        ADB names and entries when not in use. [GL #3239]
                        [GL #3238] [GL #2615] [GL #2078] [GL #2437]
                        [GL #3312] [GL #2441]

6029.   [cleanup]       Remove the unused external cache cleaning mechanism
```

(continues on next page)

(continued from previous page)

```
                        as RBTDB has its own internal cache cleaning
                        mechanism and we don't support any other database
                        implementations. [GL #3639]

6028.   [performance]   Build-time code generation of DNS RRtype switches
                        is now much faster. [GL !7121]

6027.   [bug]           Fix assertion failure in isc_http API used by
                        statschannel if the read callback would be called
                        on HTTP request that has been already closed.
                        [GL #3693]

6026.   [cleanup]       Deduplicate time unit conversion factors.
                        [GL !7033]

6025.   [bug]           Copy TLS identifier when setting up primaries for
                        catalog member zones. [GL #3638]

6024.   [func]          Deprecate 'auto-dnssec'. [GL #3667]

6023.   [func]          Remove dynamic update DNSSEC management feature.
                        [GL #3686]

6022.   [performance]   The decompression implementation in dns_name_fromwire()
                        is now smaller and faster. [GL #3655]

6021.   [bug]           Use the current domain name when checking answers from
                        a dual-stack-server. [GL #3607]

6020.   [bug]           Ensure 'named-checkconf -z' respects the check-wildcard
                        option when loading a zone.  [GL #1905]

6019.   [func]          Deprecate `coresize`, `datasize`, `files`, and
                        `stacksize` named.conf options. [GL #3676]

6018.   [cleanup]       Remove the --with-tuning configure option.
                        [GL #3664]

6017.   [bug]           The view's zone table was not locked when it should
                        have been leading to race conditions when external
                        extensions that manipulate the zone table where in
                        use. [GL #3468]

6016.   [func]          Change NSEC3PARAM TTL to match the SOA MINIMUM.
                        [GL #3570]

6015.   [bug]           Some browsers (Firefox) send more than 10 HTTP
                        headers.  Bump the number of allowed HTTP headers
                        to 100. [GL #3670]

6014.   [func]          Add isc_hashmap API implementation that implements
                        Robin Hood hashing.  The API requires the keys to
```

(continues on next page)

(continued from previous page)

```
                        be stored with the stored value.  [GL !6790]
```

```
        --- 9.19.7 released ---

6013.   [bug]           Fix a crash that could happen when you change
                        a dnssec-policy zone with NSEC3 to start using
                        inline-signing. [GL #3591]

6012.   [placeholder]

6011.   [func]          Refactor the privilege setting part of named_os unit
                        to make libcap on Linux mandatory and use setreuid
                        and setregid if available. [GL #3583]

6010.   [func]          Make the initial interface scan happen before
                        dropping the privileges.  This requires exiting
                        exclusive mode before scanning the interfaces
                        and re-entering it again when we are done.  This
                        is because starting the listening on interfaces
                        requires the loopmgr to be running and not paused.
                        [GL #3583]

6009.   [bug]           Don't trust a placeholder KEYDATA from the managed-keys
                        zone by adding it into secroots. [GL #2895]

6008.   [bug]           Fixed a race condition that could cause a crash
                        in dns_zone_synckeyzone(). [GL #3617]

6007.   [cleanup]       Don't enforce the jemalloc use on NetBSD. [GL #3634]

6006.   [cleanup]       The zone dumping was using isc_task API to launch
                        the zonedump on the offloaded threadpool.  Remove
                        the task and launch the offloaded work directly.
                        [GL #3628]

6005.   [func]          The zone loading has been moved to the offload
                        threadpool instead of doing incremental repeated
                        tasks, so zone loading scheduling is now driven
                        by the operating system scheduler rather than fixed
                        (100) quantum. [GL #3625]

6004.   [func]          Add check-svcb to control the checking of additional
                        constraints on SVBC records.  This change impacts on
                        named, named-checkconf, named-checkzone,
                        named-compilezone and nsupdate. [GL #3576]

6003.   [bug]           Fix an inheritance bug when setting the port on
                        remote servers in configuration. [GL #3627]

6002.   [bug]           Fix a resolver prefetch bug when the record's TTL value
                        is equal to the configured prefetch eligibility value,
                        but the record was erroneously not treated as eligible
```

(continues on next page)

(continued from previous page)

```
                        for prefetching. [GL #3603]

6001.   [bug]           Always call dns_adb_endudpfetch() after calling
                        dns_adb_beginudpfetch() for UDP queries in resolver.c,
                        in order to adjust back the quota. [GL #3598]

6000.   [bug]           Fix a startup issue on Solaris systems with many
                        (reportedly > 510) CPUs. Thanks to Stacey Marshall from
                        Oracle for deep investigation of the problem. [GL #3563]

5999.   [bug]           rpz-ip rules could be ineffective in some scenarios
                        with CD=1 queries. [GL #3247]

5998.   [placeholder]

5997.   [cleanup]       Less ceremonial UNEXPECTED_ERROR() and FATAL_ERROR()
                        reporting macros. [GL !6914]

5996.   [bug]           Fix a couple of bugs in cfg_print_duration(), which
                        could result in generating incomplete duration values
                        when printing the configuration using named-checkconf.
                        [GL !6880]

5995.   [performance]   A new algorithm for DNS name compression based on a
                        hash set of message offsets. Name compression is now
                        more complete as well as being generally faster, and
                        the implementation is less complicated and requires
                        much less memory. [GL !6517]

5994.   [func]          Refactor the isc_httpd implementation used in the
                        statistics channel. [GL !6879]

5993.   [cleanup]       Store dns_name_t attributes as boolean members of
                        the structure. Remove DNS_NAMEATTR_* macros.
                        Fix latent attribute handling bug in RBT. [GL !6902]
```

```
        --- 9.19.6 released ---

5992.   [func]          Introduce the new isc_mem_*x() APIs that takes extra
                        flags as the last argument.  Currently ISC_MEM_ZERO
                        and ISC_MEM_ALIGN(n) flags have been implemented that
                        clears the memory to avoid the isc_mem_get()/memset()
                        pattern and make aligned allocation which replaces the
                        previous isc_mem_*_aligned() calls. [GL !6398]

5991.   [protocol]      Add support for parsing and validating "dohpath" to
                        SVCB. [GL #3544]

5990.   [test]          fuzz/dns_message_checksig now creates the key directory
                        it uses when testing in /tmp at run time. [GL #3569]

5989.   [func]          Implement support for DDNS update forwarding using DoT
```

(continues on next page)

(continued from previous page)

```
                        to TLS-enabled primary servers. [GL #3512]

5988.   [bug]           Some out of memory conditions in opensslrsa_link.c
                        could lead to memory leaks. [GL #3551]

5987.   [func]          Provide custom isc_mem based allocators for libuv,
                        OpenSSL and libxml2 libraries that support replacing
                        the internal allocators. [GL #3559]

5986.   [func]          Make the memory context debugging options local to
                        the memory context and make it immutable for the memory
                        context lifetime. [GL #3559]

5985.   [func]          Bump the minimal libuv version to 1.34.0. [GL #3567]

5984.   [func]          'named -V' now reports the list of supported
                        DNSSEC/DS/HMAC algorithms and the supported TKEY modes.
                        [GL #3541]

5983.   [bug]           Changing just the TSIG key names for primaries in
                        catalog zones' member zones was not effective.
                        [GL #3557]

5982.   [func]          Extend dig to allow requests to be signed using SIG(0)
                        as well as providing a mechanism to specify the signing
                        time. [GL !5923]

5981.   [test]          Add dns_message_checksig fuzzer to check messages
                        signed using TSIG or SIG(0). [GL !5923]

5980.   [func]          The internal isc_entropy API provider has been
                        changed from OpenSSL RAND_bytes() to uv_random()
                        to use system provided entropy. [GL !6803]

5979.   [func]          Implement DoT support for nsupdate. [GL #1781]

5978.   [port]          The ability to use pkcs11 via engine_pkcs11 has been
                        restored, by only using deprecated APIs in
                        OpenSSL 3.0.0. BIND needs to be compiled with
                        '-DOPENSSL_API_COMPAT=10100' specified in the CFLAGS
                        at compile time. [GL !6711]

5977.   [bug]           named could incorrectly return non-truncated, glueless
                        referrals for responses whose size was close to the UDP
                        packet size limit. [GL #1967]

5976.   [cleanup]       isc_timer_t objects are now created, started and
                        destroyed in a particular loop, and timer callbacks
                        run in that loop. isc_timer_stop() can still be called
                        from any loop; when run from a different loop than
                        the one associated with the timer, the request will
                        be recorded in atomic variable and the timer will
```

(continues on next page)

(continued from previous page)

```
                        be stopped on the next callback call. [GL #3202]

5975.   [func]          Implement TLS transport support for dns_request and
                        dns_dispatch. [GL #3529]

5974.   [bug]           Fix an assertion failure in dispatch caused by
                        extra read callback call. [GL #3545]

5973.   [bug]           Fixed a possible invalid detach in UPDATE
                        processing. [GL #3522]

5972.   [bug]           Gracefully handle when the statschannel HTTP connection
                        gets cancelled during sending data back to the client.
                        [GL #3542]

5971.   [func]          Add libsystemd sd_notify() support. [GL #1176]

5970.   [func]          Log the reason why a query was refused. [GL !6669]

5969.   [bug]           DNSSEC signing statistics failed to identify the
                        algorithm involved.  The key names have been changed
                        to be the algorithm number followed by "+" followed
                        by the key id (e.g. "8+54274"). [GL #3525]

5968.   [cleanup]       Remove 'resolve' binary from tests. [GL !6733]

5967.   [cleanup]       Flagged the obsolete "random-device" option as
                        ancient; it is now an error to configure it. [GL #3399]

5966.   [func]          You can now specify if a server must return a DNS
                        COOKIE before accepting the response over UDP.
                        [GL #2295]

                        server <prefix> { require-cookie <yes_or_no>; };

5965.   [cleanup]       Move the duplicated ASCII case conversion tables to
                        isc_ascii where they can be shared, and replace the
                        various hot-path tolower() loops with calls to new
                        isc_ascii implementations. [GL !6516]

5964.   [func]          When an international domain name is not valid, DiG will
                        now pass it through unchanged, instead of stopping with
                        an error message. [GL #3527]

5963.   [bug]           Ensure struct named_server is properly initialized.
                        [GL #6531]
```

```
        --- 9.19.5 released ---

5962.   [security]      Fix memory leak in EdDSA verify processing.
                        (CVE-2022-38178) [GL #3487]
```

(continues on next page)

(continued from previous page)

```
5961.   [placeholder]

5960.   [security]      Fix serve-stale crash that could happen when
                       stale-answer-client-timeout was set to 0 and there was
                       a stale CNAME in the cache for an incoming query.
                       (CVE-2022-3080) [GL #3517]

5959.   [security]      Fix memory leaks in the DH code when using OpenSSL 3.0.0
                       and later versions. The openssldh_compare(),
                       openssldh_paramcompare(), and openssldh_todns()
                       functions were affected. (CVE-2022-2906) [GL #3491]

5958.   [security]      When an HTTP connection was reused to get
                       statistics from the stats channel, and zlib
                       compression was in use, each successive
                       response sent larger and larger blocks of memory,
                       potentially reading past the end of the allocated
                       buffer. (CVE-2022-2881) [GL #3493]

5957.   [security]      Prevent excessive resource use while processing large
                       delegations. (CVE-2022-2795) [GL #3394]

5956.   [func]          Make RRL code treat all QNAMEs that are subject to
                       wildcard processing within a given zone as the same
                       name. [GL #3459]

5955.   [port]          The libxml2 library has deprecated the usage of
                       xmlInitThreads() and xmlCleanupThreads() functions. Use
                       xmlInitParser() and xmlCleanupParser() instead.
                       [GL #3518]

5954.   [func]          Fallback to IDNA2003 processing in dig when IDNA2008
                       conversion fails. [GL #3485]

5953.   [bug]           Fix a crash on shutdown in delete_trace_entry(). Add
                       mctx attach/detach pair to make sure that the memory
                       context used by a memory pool is not destroyed before
                       the memory pool itself. [GL #3515]

5952.   [bug]           Use quotes around address strings in YAML output.
                       [GL #3511]

5951.   [bug]           In some cases, the dnstap query_message field was
                       erroneously set when logging response messages.
                       [GL #3501]

5950.   [func]          Implement a feature to set an Extended DNS Error (EDE)
                       code on responses modified by RPZ. [GL #3410]

5949.   [func]          Add new isc_loopmgr API that runs the application
                       event loops and completely replaces the isc_app
                       API. Refactor the isc_taskmgr, isc_timermgr and
```

(continues on next page)

(continued from previous page)

```
                        isc_netmgr to use the isc_loopmgr event loops.
                        [GL #3508]

5948.   [bug]           Fix nsec3.c:dns_nsec3_activex() function, add a missing
                        dns_db_detachnode() call. [GL #3500]

5947.   [func]          Change dnssec-policy to allow graceful transition from
                        an NSEC only zone to NSEC3. [GL #3486]

5946.   [bug]           Fix statistics channel's handling of multiple HTTP
                        requests in a single connection which have non-empty
                        request bodies. [GL #3463]

5945.   [bug]           If parsing /etc/bind.key failed, delv could assert
                        when trying to parse the built in trust anchors as
                        the parser hadn't been reset. [GL !6468]

5944.   [bug]           Fix +http-plain-get and +http-plain-post options
                        support in dig. Thanks to Marco Davids at SIDN for
                        reporting the problem. [GL !6672]

5943.   [placeholder]

5942.   [bug]           Fix tkey.c:buildquery() function's error handling by
                        adding the missing cleanup code. [GL #3492]

5941.   [func]          Zones with dnssec-policy now require dynamic DNS or
                        inline-siging to be configured explicitly. [GL #3381]

5940.   [placeholder]

5939.   [placeholder]

5938.   [bug]           An integer type overflow could cause an assertion
                        failure when freeing memory. [GL #3483]

5937.   [cleanup]       The dns_rdatalist_tordataset() and
                        dns_rdatalist_fromrdataset() functions can no
                        longer fail. Clean up their prototypes and error
                        handling, and that of other calling functions that
                        subsequently cannot fail, including
                        dns_message_setquerytsig(). [GL #3467]

5936.   [bug]           Don't enable serve-stale for lookups that error because
                        it is a duplicate query or a query that would be
                        dropped. [GL #2982]

5935.   [bug]           Fix DiG lookup reference counting bug, which could
                        be observed in NSSEARCH mode. [GL #3478]
```

```
        --- 9.19.4 released ---
```

(continues on next page)

(continued from previous page)

```
5934.   [func]          Improve fetches-per-zone fetch limit logging to log
                        the final allowed and spilled values of the fetch
                        counters before the counter object gets destroyed.
                        [GL #3461]

5933.   [port]          Automatically disable RSASHA1 and NSEC3RSASHA1 in
                        named on Fedorda 33, Oracle Linux 9 and RHEL9 when
                        they are disabled by the security policy. [GL #3469]

5932.   [bug]           Fix rndc dumpdb -expired and always include expired
                        RRsets, not just for RBTDB_VIRTUAL time window.
                        [GL #3462]

5931.   [bug]           Fix DiG query error handling robustness in NSSEARCH
                        mode by making sure that udp_ready(), tcp_connected(),
                        and send_done() callbacks start the next query in chain
                        even if there is some kind of error with the previous
                        query. [GL #3419]

5930.   [bug]           Fix DiG query retry and fail-over bug in UDP mode.
                        Also simplify the overall retry and fail-over logic to
                        make it behave predictably, and always respect the
                        documented +retry/+tries count set by a command-line
                        option (or use the default values of 2 or 3
                        respectively). [GL #3407]

5929.   [func]          The use of the "max-zone-ttl" option in "zone" and
                        "options" blocks is now deprecated; this should
                        now be configured as part of "dnssec-policy"
                        instead. The old option still works in zones
                        with no "dnssec-policy" configured, but a warning
                        will be logged when loading configuration. Its
                        functionality will be removed in a future release.
                        Using "max-zone-ttl" and "dnssec-policy" in the
                        same zone is now a fatal error. [GL #2918]

5928.   [placeholder]

5927.   [bug]           A race was possible in dns_dispatch_connect()
                        that could trigger an assertion failure if two
                        threads called it near-simultaneously. [GL #3456]

5926.   [func]          Handle transient TCP connect() EADDRINUSE failures
                        on FreeBSD (and possibly other BSDs) by trying three
                        times before giving up. [GL #3451]

5925.   [bug]           With a forwarder configured for all queries, resolution
                        failures encountered during DS chasing could trigger
                        assertion failures due to a logic bug in
                        resume_dslookup() that caused it to call
                        dns_resolver_createfetch() with an invalid name.
                        [GL #3439]
```

(continues on next page)

(continued from previous page)

```
5924.   [func]          When it's necessary to use AXFR to respond to an
                        IXFR request, a message explaining the reason
                        is now logged at level info. [GL #2683]

5923.   [bug]           Fix inheritance for dnssec-policy when checking for
                        inline-signing. [GL #3438]

5922.   [bug]           Forwarding of UPDATE message could fail with the
                        introduction of netmgr. This has been fixed. [GL #3389]

5921.   [test]          Convert system tests to use a default DNSKEY algorithm
                        where the test is not DNSKEY algorithm specific.
                        [GL #3440]

5920.   [bug]           Don't pass back the current name offset when the
                        compression is disabled in the non-improving case.
                        [GL #3423]
```

```
        --- 9.19.3 released ---

5919.   [func]          The "rndc fetchlimit" command lists name servers
                        and domain names that are being rate-limited by
                        "fetches-per-server" or "fetches-per-zone" limits.
                        [GL #665]

5918.   [test]          Convert system tests to use a default HMAC algorithm
                        where the test is not HMAC specific. [GL #3433]

5917.   [bug]           Update ifconfig.sh script as is miscomputed interface
                        identifiers when destroying interfaces. [GL #3061]

5916.   [bug]           When resolving a name, don't give up immediately if an
                        authoritative server returns FORMERR; try the other
                        servers first. [GL #3152]

5915.   [bug]           Detect missing closing brace (}) and computational
                        overflows in $GENERATE directives. [GL #3429]

5914.   [bug]           When synth-from-dnssec generated a response using
                        records from a higher zone, it could unexpectedly prove
                        non-existance of records in a subordinate grafted-on
                        namespace. [GL #3402]

5913.   [placeholder]

5912.   [cleanup]       The "glue-cache" option has been removed. The glue cache
                        feature still works and is now permanently enabled.
                        [GL #2147]

5911.   [bug]           Update HTTP listener settings on reconfiguration.
                        [GL #3415]
```

(continues on next page)

(continued from previous page)

```
5910.   [cleanup]       Move built-in dnssec-policies into the defaultconf.
                        These are now printed with 'named -C'. [GL !6467]

5909.   [bug]           The server-side destination port was missing from dnstap
                        captures of client traffic. [GL #3309]

5908.   [bug]           Fix race conditions in route_connected(). [GL #3401]

5907.   [bug]           Fix a crash in dig NS search mode when one of the NS
                        server queries fail. [GL #3207]

5906.   [cleanup]       Various features (e.g. prefetch, RPZ) no longer share
                        common pointers when initiating recursion. This
                        rationalizes recursion quota handling and makes the
                        value of the RecursClients statistics counter more
                        accurate. [GL #3168]

5905.   [bug]           When the TCP connection would be closed/reset between
                        the connect/accept and the read, the uv_read_start()
                        return value would be unexpected and cause an assertion
                        failure. [GL #3400]

5904.   [func]          Changed dnssec-signzone -H default to 0 additional
                        NSEC3 iterations. [GL #3395]

5903.   [bug]           When named checks that the OPCODE in a response matches
                        that of the request, if there is a mismatch named logs
                        an error.  Some of those error messages incorrectly
                        used RCODE instead of OPCODE to lookup the nemonic.
                        This has been corrected. [GL !6420]

5902.   [func]          NXDOMAIN cache records are no longer retained in
                        the cache after expiry, even when serve-stale is
                        in use. [GL #3386]

5901.   [bug]           When processing a catalog zone member zone make sure
                        that there is no configured pre-existing forward-only
                        forward zone with that name. [GL #2506]

5900.   [placeholder]
```

```
        --- 9.19.2 released ---

5899.   [func]          Don't try to process DNSSEC-related and ZONEMD records
                        in catz. [GL #3380]

5898.   [cleanup]       Simplify BIND's internal DNS name compression API. As
                        RFC 6891 explains, it isn't practical to deploy new
                        label types or compression methods, so it isn't
                        necessary to have an API designed to support them.
                        Remove compression terminology that refers to Internet
```

(continues on next page)

(continued from previous page)

```
                        Drafts that expired in the 1990s. [GL !6270]

5897.   [bug]           Views that weren't configured to use RFC 5011 key
                        management would still set up an empty managed-keys
                        zone. This has been fixed. [GL #3349]

5896.   [func]          Add some more dnssec-policy checks to detect weird
                        policies. [GL #1611]

5895.   [test]          Add new set of unit test macros and move the unit
                        tests under single namespace in /tests/. [GL !6243]

5894.   [func]          Avoid periodic interface re-scans on Linux by
                        default, where a reliable event-based mechanism for
                        detecting interface state changes is available.
                        [GL #3064]

5893.   [func]          Add TLS session resumption support to the client-side
                        TLS code. [GL !6274]

5892.   [cleanup]       Refactored the the hash tables in resolver.c to
                        use the isc_ht API. [GL !6271]

5891.   [func]          Key timing options for `dnssec-settime` and related
                        utilities now accept "UNSET" times as printed by
                        `dnssec-settime -p`. [GL #3361]

5890.   [bug]           When the fetches-per-server quota was adjusted
                        because of an authoritative server timing out more
                        or less frequently, it was incorrectly set to 1
                        rather than the intended value.  This has been
                        fixed. [GL #3327]

5889.   [cleanup]       Refactored and simplified the shutdown processes in
                        dns_view, dns_resolver, dns_requestmgr, and dns_adb
                        by reducing interdependencies between the objects.
                        [GL !6278]

5888.   [bug]           Only write key files if the dnssec-policy keymgr has
                        changed the metadata. [GL #3302]

5887.   [cleanup]       Remove the on-shutdown mechanics from isc_task API.
                        Replace it by isc_task_send() when we are shutting
                        down. [GL !6275]
```

```
       --- 9.19.1 released ---

5886.   [security]      Fix a crash in DNS-over-HTTPS (DoH) code caused by
                        premature TLS stream socket object deletion.
                        (CVE-2022-1183) [GL #3216]

5885.   [bug]           RPZ NSIP and NSDNAME rule processing didn't handle stub
```

(continues on next page)

(continued from previous page)

```
                        and static-stub zones at or above the query name.  This
                        has now been addressed. [GL #3232]

5884.   [cleanup]       Reduce struct padding in ADB address entries, and use a
                        binary hash function to find addresses. [GL !6219]

5883.   [cleanup]       Move netmgr/uv-compat.{c,h} to <isc/uv.h>, so
                        the compatibility libuv shims could be used outside
                        the network manager. [GL !6199]

5882.   [contrib]       Avoid name space collision in dlz modules by prefixing
                        functions with 'dlz_'. [GL !5778]

5881.   [placeholder]

5880.   [func]          Add new named command-line option -C to print built-in
                        defaults. [GL #1326]

5879.   [contrib]       dlz: Add FALLTHROUGH and UNREACHABLE macros. [GL #3306]

5878.   [func]          Check the algorithm name or OID embedded at the start
                        of the signature field for PRIVATEDNS and PRIVATEOID
                        SIG and RRSIG records are well formed. [GL #3296]

5877.   [func]          Introduce the concept of broken catalog zones described
                        in the DNS catalog zones draft version 5 document.
                        [GL #3224]

5876.   [func]          Add DNS Extended Errors when stale answers are returned
                        from cache. [GL #2267]

5875.   [bug]           Fixed a deadlock that could occur if an rndc
                        connection arrived during the shutdown of network
                        interfaces. [GL #3272]

5874.   [placeholder]

5873.   [bug]           Refactor the fctx_done() function to set fctx to
                        NULL after detaching, so that reference counting
                        errors will be easier to avoid. [GL #2969]

5872.   [bug]           udp_recv() in dispatch could trigger an INSIST when the
                        callback's result indicated success but the response
                        was canceled in the meantime. [GL #3300]

5871.   [bug]           Fix dig hanging on TLS context creation errors.
                        [GL #3285]

5870.   [cleanup]       Remove redundant macros in the RBT implementation.
                        [GL !6158]

5869.   [func]          Enable use of IP(V6)_RECVERR on Linux that allows
```

(continues on next page)

(continued from previous page)

```
                        the kernel to report destination host/network
                        unreachable errors to the userspace application.
                        [GL #4251]

5868.   [cleanup]       Use Daniel Lemire's "nearly divisionless" algorithm
                        for unbiased bounded random numbers, and move
                        re-seeding out of the hot path. [GL !6161]

5867.   [bug]           Fix assertion failure triggered by attaching to dns_adb
                        in dns_adb_createfind() that has been triggered to shut
                        down in different thread between the check for shutting
                        down condition and the attach to dns_adb. [GL #3298]

5866.   [bug]           Work around a jemalloc quirk which could trigger an
                        out-of-memory condition in named over time. [GL #3287]

5865.   [func]          Make statistics channel and control channel listen
                        on a single network manager thread. [GL !6032]

5864.   [func]          The OID embedded at the start of a PRIVATEOID public
                        key in a KEY, DNSKEY, CDNSKEY, or RKEY RR is now
                        checked for validity when reading from wire or from
                        zone files, and the OID is printed when
                        'dig +rrcomments' is used. Similarly, the name
                        embedded at the start of a PRIVATEDNS public key
                        is also checked for validity. [GL #3234]

5863.   [bug]           If there was a pending negative cache DS entry,
                        validations depending upon it could fail. [GL #3279]

5862.   [bug]           dig returned a 0 exit status on UDP connection failure.
                        [GL #3235]

5861.   [func]          Implement support for catalog zones change of ownership
                        (coo) mechanism described in the DNS catalog zones draft
                        version 5 document. [GL #3223]

5860.   [func]          Implement support for catalog zones options new syntax
                        based on catalog zones custom properties with "ext"
                        suffix described in the DNS catalog zones draft version
                        5 document. [GL #3222]

5859.   [bug]           Fix an assertion failure when using dig with +nssearch
                        and +tcp options by starting the next query in the
                        send_done() callback (like in the UDP mode) instead
                        of doing that recursively in start_tcp(). Also
                        ensure that queries interrupted while connecting
                        are detached properly. [GL #3144]

5858.   [bug]           Don't remove CDS/CDNSKEY DELETE records on zone sign
                        when using 'auto-dnssec maintain;'. [GL #2931]
```

(continues on next page)

(continued from previous page)

```
5857.   [bug]           Fixed a possible crash during shutdown due to ADB
                        entries being unlinked from the hash table too
                        soon. [GL #3256]
```

```
        --- 9.19.0 released ---

5856.   [bug]           The "starting maxtime timer" message related to outgoing
                        zone transfers was incorrectly logged at the ERROR level
                        instead of DEBUG(1). [GL #3208]

5855.   [bug]           Ensure that zone maintenance queries have a retry limit.
                        [GL #3242]

5854.   [func]          Implement reference counting for TLS contexts and
                        allow reloading of TLS certificates on reconfiguration
                        without destroying the underlying TCP listener sockets
                        for TLS-based DNS transports. [GL #3122]

5853.   [bug]           When using both the `+qr` and `+y` options `dig` could
                        crash if the connection to the first server was not
                        successful. [GL #3244]

5852.   [func]          Add new "reuseport" option to enable/disable load
                        balancing of sockets. [GL #3249]

5851.   [placeholder]

5850.   [func]          Run the RPZ update process on the offload threads.
                        [GL #3190]

5849.   [cleanup]       Remove use of exclusive mode in ns_interfacemgr in
                        favor of rwlocked access to localhost and localnets
                        members of dns_aclenv_t structure. [GL #3229]

5848.   [bug]           dig could hang in some cases involving multiple servers
                        in a lookup, when a request fails and the next one
                        refuses to start for some reason, for example if it was
                        an IPv4 mapped IPv6 address. [GL #3248]

5847.   [cleanup]       Remove task privileged mode in favor of processing
                        all events in the loadzone task in a single run
                        by setting the quantum to UINT_MAX. [GL #3253]

5846.   [func]          In dns_zonemgr, create per-thread task, zonetask, and
                        loadtask and pin the zones to individual threads,
                        instead of having "many", spreading the zones among
                        them and hoping for the best.  This also removes any
                        need to dynamically reallocate the pools with memory
                        contexts and tasks. [GL #3226]

5845.   [bug]           Refactor the timer to keep track of posted events
                        as to use isc_task_purgeevent() instead of using
```

(continues on next page)

(continued from previous page)

```
                        isc_task_purgerange().  The isc_task_purgeevent()
                        has been refactored to purge a single event instead
                        of walking through the list of posted events.
                        [GL #3252]

5844.   [bug]           dig +nssearch was hanging until manually interrupted.
                        [GL #3145]

5843.   [bug]           When an UPDATE targets a zone that is not configured,
                        the requested zone name is now logged in the "not
                        authoritative" error message, so that it is easier to
                        track down problematic update clients. [GL #3209]

5842.   [cleanup]       Remove the task exclusive mode use in ns_clientmgr.
                        [GL #3230]

5841.   [bug]           Refactor the address database:
                        - Use self-resizing hash tables, eliminating the
                          need to go into task-exclusive mode when resizing.
                        - Simplify reference counting of ADB objects
                          and the process for shutting down. [GL #3213]

5840.   [cleanup]       Remove multiple application context use in dns_client
                        unit. [GL !6041]

5839.   [func]          Add support for remote TLS certificates
                        verification, both to BIND and dig, making it possible
                        to implement Strict and Mutual TLS authentication,
                        as described in RFC 9103, Section 9.3. [GL #3163]

5838.   [cleanup]       When modifying a member zone in a catalog zone, and it
                        is detected that the zone exists and was not created by
                        the current catalog zone, distinguish the two cases when
                        the zone was not added by a catalog zone at all, and
                        when the zone was added by a different catalog zone,
                        and log a warning message accordingly. [GL #3221]

5837.   [func]          Key timing options for `dnssec-keygen` and
                        `dnssec-settime` now accept times as printed by
                        `dnssec-settime -p`. [GL !2947]

5836.   [bug]           Quote the dns64 prefix in error messages that complain
                        about problems with it, to avoid confusion with the
                        following dns64 ACLs. [GL #3210]

5835.   [cleanup]       Remove extrahandlesize from the netmgr, the callers
                        now have to allocate the object before calling
                        isc_nm_setdata() and deallocate the memory in the close
                        callback passed to isc_nm_setdata(). [GL #3227]

5834.   [cleanup]       C99 variable-length arrays are difficult to use safely,
                        so avoid them except in test code. [GL #3201]
```

(continues on next page)

(continued from previous page)

```
5833.   [bug]           When encountering socket error while trying to initiate
                        a TCP connection to a server, dig could hang
                        indefinitely, when there were more servers to try.
                        [GL #3205]

5832.   [bug]           When timing-out or having other types of socket errors
                        during a query, dig wasn't trying to perform the lookup
                        using other servers, in case they exist. [GL #3128]

5831.   [bug]           When resending a UDP request in the result of a timeout,
                        the recv_done() function in dighost.c was prepending
                        the new query into the loookup's queries list instead
                        of inserting, which could cause an assertion failure
                        when the resent query's result was SERVFAIL. [GL #3020]

5830.   [func]          Implement incremental resizing of isc_ht hash tables to
                        perform the rehashing gradually. The catalog zone
                        implementation has been optimized to work with hundreds
                        of thousands of member zones. [GL #3212] [GL #3744]

5829.   [func]          Refactor and simplify isc_timer API in preparation
                        for further refactoring on top of network manager
                        loops. [GL #3202]

5828.   [bug]           Replace single TCP write timer with per-TCP write
                        timers. [GL #3200]

5827.   [cleanup]       The command-line utilities printed their version numbers
                        inconsistently; they all now print to stdout. (They are
                        still inconsistent abotut whether you use `-v` or `-V`
                        to request the version). [GL #3189]

5826.   [cleanup]       Stop dig from complaining about lack of IDN support when
                        the user asks for no IDN translation. [GL #3188]

5825.   [func]          Set the minimum MTU on UDPv6 and TCPv6 sockets and
                        limit TCP maximum segment size (TCP_MAXSEG) to (1220)
                        for both TCPv4 and TCPv6 sockets. [GL #2201]

5824.   [bug]           Invalid dnssec-policy definitions were being accepted
                        where the defined keys did not cover both KSK and ZSK
                        roles for a given algorithm.  This is now checked for
                        and the dnssec-policy is rejected if both roles are
                        not present for all algorithms in use. [GL #3142]

5823.   [func]          Replace hazard pointers based lock-free list with
                        locked-list based queue that's simpler and has no or
                        little performance impact. [GL #3180]

5822.   [bug]           When calling dns_dispatch_send(), attach/detach
                        dns_request_t object as the read callback could
```

(continues on next page)

(continued from previous page)

```
                        be called before send callback dereferencing
                        dns_request_t object too early. [GL #3105]

5821.   [bug]           Fix query context management issues in the TCP part
                        of dig. [GL #3184]

5820.   [security]      An assertion could occur in resume_dslookup() if the
                        fetch had been shut down earlier. (CVE-2022-0667)
                        [GL #3129]

5819.   [security]      Lookups involving a DNAME could trigger an INSIST when
                        "synth-from-dnssec" was enabled. (CVE-2022-0635)
                        [GL #3158]

5818.   [security]      A synchronous call to closehandle_cb() caused
                        isc__nm_process_sock_buffer() to be called recursively,
                        which in turn left TCP connections hanging in the
                        CLOSE_WAIT state blocking indefinitely when
                        out-of-order processing was disabled. (CVE-2022-0396)
                        [GL #3112]

5817.   [security]      The rules for acceptance of records into the cache
                        have been tightened to prevent the possibility of
                        poisoning if forwarders send records outside
                        the configured bailiwick. (CVE-2021-25220) [GL #2950]

5816.   [bug]           Make BIND compile with LibreSSL 3.5.0, as it was using
                        not very accurate pre-processor checks for using shims.
                        [GL #3172]

5815.   [bug]           If an oversized key name of a specific length was used
                        in the text form of an HTTP or SVBC record, an INSIST
                        could be triggered when parsing it. [GL #3175]

5814.   [bug]           The RecursClients statistics counter could underflow
                        in certain resolution scenarios. [GL #3147]

5813.   [func]          The "keep-response-order" ACL has been declared
                        obsolete, and is now non-operational. [GL #3140]

5812.   [func]          Drop the artificial limit on the number of queries
                        processed in a single TCP read callback. [GL #3141]

5811.   [bug]           Reimplement the maximum and idle timeouts for outgoing
                        zone tranfers. [GL #1897]

5810.   [func]          New option '-J' for dnssec-signzone and dnssec-verify
                        allows loading journal files. [GL #2486]

5809.   [bug]           Reset client TCP connection when data received cannot
                        be parsed as a valid DNS request. [GL #3149]
```

(continues on next page)

(continued from previous page)

```
5808.   [bug]           Certain TCP failures were not caught and handled
                        correctly by the dispatch manager, causing
                        connections to time out rather than returning
                        SERVFAIL. [GL #3133]

5807.   [bug]           Add a TCP "write" timer, and time out writing
                        connections after the "tcp-idle-timeout" period
                        has elapsed. [GL #3132]

5806.   [bug]           An error in checking the "blackhole" ACL could cause
                        DNS requests sent by named to fail if the
                        destination address or prefix was specifically
                        excluded from the ACL. [GL #3157]

5805.   [func]          The result of each resolver priming attempt is now
                        included in the "resolver priming query complete" log
                        message. [GL #3139]

5804.   [func]          Add a debug log message when starting and ending
                        the task exclusive mode. [GL #3137]

5803.   [func]          Use compile-time paths in the documentation.
                        [GL #2717]

5802.   [test]          Add system test to test engine_pkcs11. [GL !5727]

5801.   [bug]           Log "quota reached" message when hard quota
                        is reached when accepting a connection. [GL #3125]

5800.   [func]          Add ECS support to the DLZ interface. [GL #3082]

5799.   [bug]           Use L1 cache-line size detected at runtime. [GL #3108]

5798.   [test]          Add system test to test dnssec-keyfromlabel. [GL #3092]

5797.   [bug]           A failed view configuration during a named
                        reconfiguration procedure could cause inconsistencies
                        in BIND internal structures, causing a crash or other
                        unexpected errors. [GL #3060]

5796.   [bug]           Ignore the invalid (<= 0) values returned
                        by the sysconf() check for the L1 cache line
                        size.  [GL #3108]

5795.   [bug]           rndc could crash when interrupted by a signal
                        before receiving a response. [GL #3080]

5794.   [func]          Set the IPV6_V6ONLY on all IPv6 sockets to
                        restrict the IPv6 sockets to sending and
                        receiving IPv6 packets only. [GL #3093]

5793.   [bug]           Correctly detect and enable UDP recvmmsg support
```

(continues on next page)

(continued from previous page)

```
                        in all versions of libuv that support it. [GL #3095]

5792.   [bug]           Don't schedule zone events on ISC_R_SHUTTINGDOWN
                        event failures. [GL #3084]

5791.   [func]          Remove workaround for servers returning FORMERR
                        when receiving NOTIFY query with SOA record in
                        ANSWER section. [GL #3086]

5790.   [bug]           The control channel was incorrectly looking for
                        ISC_R_CANCELED as a signal that the named is
                        shutting down.  In the dispatch refactoring,
                        the result code returned from network manager
                        is now ISC_R_SHUTTINGDOWN.  Change the control
                        channel code to use ISC_R_SHUTTINGDOWN result
                        code to detect named being shut down. [GL #3079]
```

```
        --- 9.17.22 released ---

5789.   [bug]           Allow replacing expired zone signatures with
                        signatures created by the KSK. [GL #3049]

5788.   [bug]           An assertion could occur if a catalog zone event was
                        scheduled while the task manager was being shut
                        down. [GL #3074]

5787.   [doc]           Update 'auto-dnssec' documentation, it may only be
                        activated at zone level. [GL #3023]

5786.   [bug]           Defer detaching from zone->raw in zone_shutdown() if
                        the zone is in the process of being dumped to disk, to
                        ensure that the unsigned serial number information is
                        always written in the raw-format header of the signed
                        version on an inline-signed zone. [GL #3071]

5785.   [bug]           named could leak memory when two dnssec-policy clauses
                        had the same name. named failed to log this error.
                        [GL #3085]

5784.   [func]          Implement TLS-contexts reuse. Reusing the
                        previously created TLS context objects can reduce
                        initialisation time for some configurations and enables
                        TLS session resumption for incoming zone transfers over
                        TLS (XoT). [GL #3067]

5783.   [func]          named is now able to log TLS pre-master secrets for
                        debugging purposes. This requires setting the
                        SSLKEYLOGFILE environment variable appropriately.
                        [GL #2723]

5782.   [func]          Use ECDSA P-256 instead of a 4096-bit RSA when
                        generating ephemeral key and certificate for the
```

(continues on next page)

(continued from previous page)

```
                        'tls ephemeral' configuration. [GL #2264]

5781.   [bug]           Make BIND work with OpenSSL 3.0.1 as it is now
                        enforcing minimum buffer lengths in EVP_MAC_final and
                        hence EVP_DigestSignFinal.  rndc and TSIG at a minimum
                        were broken by this change. [GL #3057]

5780.   [bug]           The Linux kernel may send netlink messages
                        indicating that network interfaces have changed
                        when they have not. This caused frequent unnecessary
                        re-scans of the interfaces.  Netlink messages now
                        only trigger re-scanning if a new address is seen
                        or an existing address is removed. [GL #3055]

5779.   [test]          Drop cppcheck suppressions and workarounds. [GL #2886]

5778.   [bug]           Destroyed TLS contexts could have been used after a
                        reconfiguration, making BIND unable to serve queries
                        over TLS and HTTPS. [GL #3053]

5777.   [bug]           TCP connections could hang after receiving
                        non-matching responses. [GL #3042]

5776.   [bug]           Add a missing isc_condition_destroy() for nmsocket
                        condition variable and add missing isc_mutex_destroy()
                        for nmworker lock. [GL #3051]
```

```
        --- 9.17.21 released ---

5775.   [bug]           Added a timer in the resolver to kill fetches that
                        have deadlocked as a result of dependency loops
                        with the ADB or the validator. This condition is
                        now logged with the message "shut down hung fetch
                        while resolving '<name>/<type>'". [GL #3040]

5774.   [func]          Restore NSEC Aggressive Cache ("synth-from-dnssec")
                        as active by default. It is limited to NSEC only
                        and by default ignores NSEC records with next name
                        in form \000.domain. [GL #1265]

5773.   [func]          Change the message when accepting TCP connection has
                        failed to say "Accepting TCP connection failed" and
                        change the log level for ISC_R_NOTCONNECTED, ISC_R_QUOTA
                        and ISC_R_SOFTQUOTA results codes from ERROR to INFO.
                        [GL #2700]

5772.   [bug]           The resolver could hang on shutdown due to dispatch
                        resources not being cleaned up when a TCP connection
                        was reset. [GL #3026]

5771.   [bug]           Use idn2 UseSTD3ASCIIRules=false to disable additional
                        unicode validity checks because enabling the additional
```

(continues on next page)

(continued from previous page)

```
                      checks would break valid domain names that contains
                      non-alphanumerical characters such as underscore
                      character (_) or wildcard (*).  This reverts change
                      [GL !5738] from the previous release. [GL #1610]

5770.   [func]        BIND could abort on startup on systems using old
                      OpenSSL versions when 'protocols' option is used inside
                      a 'tls' statement. [GL !5602]

5769.   [func]        Added support for client-side 'tls' parameters when
                      doing incoming zone transfers via XoT. [GL !5602]

5768.   [bug]         dnssec-dsfromkey failed to omit revoked keys. [GL #853]

5767.   [func]        Extend allow-transfer option with 'port' and
                      'transport' options to restrict zone transfers to
                      a specific port and DNS transport protocol.
                      [GL #2776]

5766.   [func]        Unused 'tls' clause options 'ca-file' and 'hostname'
                      were disabled. [GL !5600]

5765.   [bug]         Fix a bug in DoH implementation making 'dig'
                      abort when ALPN negotiation fails. [GL #3022]

5764.   [bug]         dns_sdlz_putrr failed to process some valid resource
                      records. [GL #3021]

5763.   [bug]         Fix a bug in DoT code leading to an abort when
                      a zone transfer ends with an unexpected DNS message.
                      [GL #3004]

5762.   [bug]         Fix a "named" crash related to removing and restoring a
                      `catalog-zone` entry in the configuration file and
                      running `rndc reconfig`. [GL #1608]

5761.   [bug]         OpenSSL 3.0.0 support could fail to correctly read
                      ECDSA private keys leading to incorrect signatures
                      being generated. [GL #3014]

5760.   [bug]         Prevent a possible use-after-free error in resolver.
                      [GL #3018]

5759.   [func]        Set Extended DNS Error Code 18 - Prohibited if query
                      access is denied to the specific client. [GL #1836]

5758.   [bug]         mdig now honors the operating system's preferred
                      ephemeral port range. [GL #2374]

5757.   [test]        Replace sed in nsupdate system test with awk to
                      construct the nsupdate command.  The sed expression
                      was not reliably changing the ttl. [GL #3003]
```

(continues on next page)

(continued from previous page)

```
5756.   [func]          Assign HTTP freshness lifetime to responses sent
                        via DNS-over-HTTPS, according to the recommendations
                        given in RFC 8484. [GL #2854]
```

```
        --- 9.17.20 released ---

5755.   [bug]           The statistics channel wasn't correctly handling
                        multiple HTTP requests, or pipelined or truncated
                        requests. [GL #2973]

5754.   [bug]           "tls" statements may omit "key-file" and "cert-file",
                        but if either one is specified, then both must be.
                        [GL #2986]

5753.   [placeholder]

5752.   [bug]           Fix an assertion failure caused by missing member zones
                        during a reload of a catalog zone. [GL #2308]

5751.   [port]          Add support for OpenSSL 3.0.0.  OpenSSL 3.0.0
                        deprecated 'engine' support.  If OpenSSL 3.0.0 has
                        been built without support for deprecated functionality
                        pkcs11 via engine_pkcs11 is no longer available.
                        [GL #2843]

5750.   [bug]           Fix a bug when comparing two RSA keys. There was a typo
                        which caused the "p" prime factors to not being
                        compared. [GL #2972]

5749.   [bug]           Handle duplicate references to the same catalog
                        zone gracefully. [GL #2916]

5748.   [func]          Update "nsec3param" defaults to iterations 0, salt
                        length 0. [GL #2956]

5747.   [func]          Update rndc serve-stale status output to be less
                        confusing. [GL #2742]

5746.   [bug]           A lame server delegation could lead to a loop in which
                        a resolver fetch depends on an ADB find which depends
                        on the same resolver fetch. Previously, this would
                        cause the fetch to hang until timing out, but after
                        change #5730 it would hang forever. The condition is
                        now detected and avoided. [GL #2927]

5745.   [bug]           Fetch context objects now use attach/detach
                        semantics to make it easier to find and debug
                        reference-counting errors, and several such errors
                        have been fixed. [GL #2953]

5744.   [func]          The network manager is now used for netlink sockets
```

(continues on next page)

(continued from previous page)

```
                        to monitor network interface changes. This was the
                        last remaining use of the old isc_socket and
                        isc_socketmgr APIs, so they have now been removed.
                        The "named -S" argument and the "reserved-sockets"
                        option in named.conf have no function now, and are
                        deprecated. "socketmgr" statistics are no longer
                        reported in the statistics channel. [GL #2926]

5743.   [func]          Add finer-grained "update-policy" rules,
                        "krb5-subdomain-self-rhs" and "ms-subdomain-self-rhs",
                        which restrict SRV and PTR record changes, allowing
                        only records whose content matches the machine name
                        embedded in the Kerberos principal making the change.
                        [GL #481]

5742.   [func]          ISC_LIKELY() and ISC_UNLIKELY() macros have been
                        removed. [GL #2952]

5741.   [bug]           Log files with "timestamp" suffixes could be left in
                        place after rolling, even if the number of preserved
                        log files exceeded the configured "versions" limit.
                        [GL #828]

5740.   [func]          Implement incremental resizing of RBT hash table to
                        perform the rehashing gradually. [GL #2941]

5739.   [func]          Change default of 'dnssec-dnskey-kskonly' to 'yes'.
                        [GL #1316]

5738.   [bug]           Enable idn2 UseSTD3ASCIIRules=true to implement
                        additional unicode validity checks. [GL #1610]

5737.   [bug]           Address Coverity warning in lib/dns/dnssec.c.
                        [GL #2935]
```

```
        --- 9.17.19 released ---

5736.   [security]      The "lame-ttl" option is now forcibly set to 0. This
                        effectively disables the lame server cache, as it could
                        previously be abused by an attacker to significantly
                        degrade resolver performance. (CVE-2021-25219)
                        [GL #2899]

5735.   [cleanup]       The result codes which BIND 9 uses internally are now
                        all defined as a single list of enum values rather than
                        as multiple sets of integers scattered around shared
                        libraries. This prevents the need for locking in some
                        functions operating on result codes, and makes result
                        codes more debugger-friendly. [GL #719]

5734.   [bug]           Fix intermittent assertion failures in dig which were
                        triggered during zone transfers. [GL #2884]
```

(continues on next page)

(continued from previous page)

```
5733.   [func]          Require the "dot" Application-Layer Protocol Negotiation
                        (ALPN) token to be selected in the TLS handshake for
                        zone transfers over TLS (XoT), as required by RFC 9103
                        section 7.1. [GL #2794]

5732.   [cleanup]       Remove the dns_lib_init(), dns_lib_shutdown(),
                        ns_lib_init(), and ns_lib_shutdown() functions, as they
                        no longer served any useful purpose. [GL #88]

5731.   [bug]           Disallow defining "http" configuration clauses called
                        "default" as they were silently ignored. [GL #2925]

5730.   [func]          The resolver and the request and dispatch managers have
                        been substantially refactored, and are now based on the
                        network manager instead of the old isc_socket API. All
                        outgoing DNS queries and requests now use the new API;
                        isc_socket is only used to monitor for network interface
                        changes. [GL #2401]

5729.   [func]          Allow finer control over TLS protocol configuration by
                        implementing new options for "tls" configuration clauses
                        ("dhparam-file", "ciphers", "prefer-server-ciphers",
                        "session-tickets"). These options make achieving perfect
                        forward secrecy (PFS) possible for DNS-over-TLS (DoT)
                        and DNS-over-HTTPS (DoH). [GL #2796]

5728.   [func]          Allow specifying supported TLS protocol versions for
                        each "tls" configuration clause. [GL #2795]

5727.   [placeholder]

5726.   [bug]           Fix a use-after-free bug which was triggered while
                        checking for duplicate "http" configuration clauses.
                        [GL #2924]

5725.   [bug]           Fix an assertion failure triggered by passing an invalid
                        HTTP path to dig. [GL #2923]

5724.   [bug]           Address a potential deadlock when checking zone content
                        consistency. [GL #2908]

5723.   [bug]           Change 5709 broke backward compatibility for the
                        "check-names master ..." and "check-names slave ..."
                        options. This has been fixed. [GL #2911]

5722.   [bug]           Preserve the contents of the receive buffer for TCPDNS
                        and TLSDNS when growing its size. [GL #2917]

5721.   [func]          A new realloc()-like function, isc_mem_reget(), was
                        added to the libisc API for resizing memory chunks
                        allocated using isc_mem_get(). Memory (re)allocation
```

(continues on next page)

(continued from previous page)

```
                        functions are now guaranteed to return non-NULL pointers
                        for zero-sized allocation requests. [GL !5440]

5720.   [contrib]       Remove old-style DLZ drivers that had to be enabled at
                        build time. [GL #2814]

5719.   [func]          Remove support for the "map" zone file format.
                        [GL #2882]

5718.   [bug]           The "sig-signing-type" zone configuration option was
                        processed incorrectly, causing valid configurations to
                        be rejected. This has been fixed. [GL #2906]

5717.   [func]          The "cache-file" option, which was documented as "for
                        testing purposes only" and not to be used, has been
                        removed. [GL #2903]

5716.   [placeholder]

5715.   [func]          Add a check for ports specified in "*-source(-v6)"
                        options clashing with a global listening port. Such a
                        configuration was already unsupported, but it failed
                        silently; it is now treated as an error. [GL #2888]

5714.   [bug]           Remove the "adjust interface" mechanism which was
                        responsible for setting up listeners on interfaces when
                        the "*-source(-v6)" address and port were the same as
                        the "listen-on(-v6)" address and port. Such a
                        configuration is no longer supported; under certain
                        timing conditions, that mechanism could prevent named
                        from listening on some TCP ports. This has been fixed.
                        [GL #2852]

5713.   [func]          Add "primaries" as a synonym for "masters" and
                        "default-primaries" as a synonym for "default-masters"
                        in catalog zone configuration options. [GL #2818]

5712.   [func]          Remove native PKCS#11 support in favor of engine_pkcs11
                        from the OpenSC project. [GL #2691]
```

```
        --- 9.17.18 released ---

5711.   [bug]           "map" files exceeding 2GB in size failed to load due to
                        a size comparison that incorrectly treated the file size
                        as a signed integer. [GL #2878]

5710.   [placeholder]

5709.   [func]          When reporting zone types in the statistics channel, the
                        terms "primary" and "secondary" are now used instead of
                        "master" and "slave", respectively. Enum values
                        throughout the code have been updated to use this
```

(continues on next page)

(continued from previous page)

```
                        terminology as well. [GL #1944]

5708.   [placeholder]

5707.   [bug]           A bug was fixed which prevented dig from querying
                        DNS-over-HTTPS (DoH) servers via IPv6. [GL #2860]

5706.   [cleanup]       Support for external applications to register with
                        libisc and use it has been removed. Export versions of
                        BIND 9 libraries have not been supported for some time,
                        but the isc_lib_register() function was still available;
                        it has now been removed. [GL !2420]

5705.   [bug]           Change #5686 altered the internal memory structure of
                        zone databases, but neglected to update the MAPAPI value
                        for zone files in "map" format. This caused named to
                        attempt to load incompatible map files, triggering an
                        assertion failure on startup. The MAPAPI value has now
                        been updated, so named rejects outdated files when
                        encountering them. [GL #2872]

5704.   [bug]           Change #5317 caused the EDNS TCP Keepalive option to be
                        ignored inadvertently in client requests. It has now
                        been fixed and this option is handled properly again.
                        [GL #1927]

5703.   [bug]           Fix a crash in dig caused by closing an HTTP/2 socket
                        associated with an unused HTTP/2 session. [GL #2858]

5702.   [bug]           Improve compatibility with DNS-over-HTTPS (DoH) clients
                        by allowing HTTP/2 request headers in any order.
                        [GL #2875]

5701.   [bug]           named-checkconf failed to detect syntactically invalid
                        values of the "key" and "tls" parameters used to define
                        members of remote server lists. [GL #2461]

5700.   [bug]           When a member zone was removed from a catalog zone,
                        journal files for the former were not deleted.
                        [GL #2842]

5699.   [func]          Data structures holding DNSSEC signing statistics are
                        now grown and shrunk as necessary upon key rollover
                        events. [GL #1721]

5698.   [bug]           When a DNSSEC-signed zone which only has a single
                        signing key available is migrated to use KASP, that key
                        is now treated as a Combined Signing Key (CSK).
                        [GL #2857]

5697.   [func]          dnssec-cds now only generates SHA-2 DS records by
                        default and avoids copying deprecated SHA-1 records from
```

(continues on next page)

(continued from previous page)

```
                        a child zone to its delegation in the parent. If the
                        child zone does not publish SHA-2 CDS records,
                        dnssec-cds will generate them from the CDNSKEY records.
                        The "-a algorithm" option now affects the process of
                        generating DS digest records from both CDS and CDNSKEY
                        records. Thanks to Tony Finch. [GL #2871]

5696.   [protocol]      Support for HTTPS and SVCB record types has been added.
                        [GL #1132]

5695.   [func]          Add a new dig command-line option, "+showbadcookie",
                        which causes a BADCOOKIE response message to be
                        displayed when it is received from the server.
                        [GL #2319]

5694.   [bug]           Stale data in the cache could cause named to send
                        non-minimized queries despite QNAME minimization being
                        enabled. [GL #2665]

5693.   [func]          Restore support for reading "timeout" and "attempts"
                        options from /etc/resolv.conf, and use their values in
                        dig, host, and nslookup. (This was previously supported
                        by liblwres, and was still mentioned in the man pages,
                        but had stopped working after liblwres was deprecated in
                        favor of libirs.) [GL #2785]

5692.   [bug]           Fix a rare crash in DNS-over-HTTPS (DoH) code caused by
                        detaching from an HTTP/2 session handle too early when
                        sending data. [GL #2851]

5691.   [bug]           When a dynamic zone was made available in another view
                        using the "in-view" statement, running "rndc freeze"
                        always reported an "already frozen" error even though
                        the zone was successfully frozen. [GL #2844]

5690.   [func]          dnssec-signzone now honors Predecessor and Successor
                        metadata found in private key files: if a signature for
                        an RRset generated by the inactive predecessor exists
                        and does not need to be replaced, no additional
                        signature is now created for that RRset using the
                        successor key. This enables dnssec-signzone to gradually
                        replace RRSIGs during a ZSK rollover. [GL #1551]
```

```
        --- 9.17.17 released ---

5689.   [security]      An assertion failure occurred when named attempted to
                        send a UDP packet that exceeded the MTU size, if
                        Response Rate Limiting (RRL) was enabled.
                        (CVE-2021-25218) [GL #2856]

5688.   [bug]           Zones using KASP and inline-signed zones failed to apply
                        changes from the unsigned zone to the signed zone under
```

(continues on next page)

(continued from previous page)

```
                        certain circumstances. This has been fixed. [GL #2735]

5687.   [bug]           "rndc reload <zonename>" could trigger a redundant
                        reload for an inline-signed zone whose zone file was not
                        modified since the last "rndc reload". This has been
                        fixed. [GL #2855]

5686.   [func]          The number of internal data structures allocated for
                        each zone was reduced. [GL #2829]

5685.   [bug]           named failed to check the opcode of responses when
                        performing zone refreshes, stub zone updates, and UPDATE
                        forwarding. This has been fixed. [GL #2762]

5684.   [func]          The DNS-over-HTTP (DoH) configuration syntax was
                        extended:
                        - The maximum number of active DoH connections can now
                          be set using the "http-listener-clients" option. The
                          default is 300.
                        - The maximum number of concurrent HTTP/2 streams per
                          connection can now be set using the
                          "http-streams-per-connection" option. The default is
                          100.
                        - Both of these values can also be set on a per-listener
                          basis using the "listener-clients" and
                          "streams-per-connection" parameters in an "http"
                          statement.
                        [GL #2809]

5683.   [bug]           The configuration-checking code now verifies HTTP paths.
                        [GL !5231]

5682.   [bug]           Some changes to "zone-statistics" settings were not
                        properly processed by "rndc reconfig". This has been
                        fixed. [GL #2820]

5681.   [func]          Relax the checks in the dns_zone_cdscheck() function to
                        allow CDS and CDNSKEY records in the zone that do not
                        match an existing DNSKEY record, as long as the
                        algorithm matches. This allows a clean rollover from one
                        provider to another in a multi-signer DNSSEC
                        configuration. [GL #2710]

5680.   [bug]           HTTP GET requests without query strings caused a crash
                        in DoH code. This has been fixed. [GL !5268]

5679.   [func]          Thread affinity is no longer set. [GL #2822]

5678.   [bug]           The "check DS" code failed to release all resources upon
                        named shutdown when a refresh was in progress. This has
                        been fixed. [GL #2811]
```

(continues on next page)

(continued from previous page)

```
5677.   [func]          Previously, named accepted FORMERR responses both with
                        and without an OPT record, as an indication that a given
                        server did not support EDNS. To implement full
                        compliance with RFC 6891, only FORMERR responses without
                        an OPT record are now accepted. This intentionally
                        breaks communication with servers that do not support
                        EDNS and that incorrectly echo back the query message
                        with the RCODE field set to FORMERR and the QR bit set
                        to 1. [GL #2249]

5676.   [func]          Memory allocation has been substantially refactored; it
                        is now based on the memory allocation API provided by
                        the jemalloc library, which is a new optional build
                        dependency for BIND 9. [GL #2433]

5675.   [bug]           Compatibility with DoH clients has been improved by
                        ignoring the value of the "Accept" HTTP header.
                        [GL !5246]

5674.   [bug]           A shutdown hang was triggered by DoH clients prematurely
                        aborting HTTP/2 streams. This has been fixed. [GL !5245]

5673.   [func]          Add a new build-time option, --disable-doh, to allow
                        building BIND 9 without the libnghttp2 library.
                        [GL #2478]

5672.   [bug]           Authentication of rndc messages could fail if a
                        "controls" statement was configured with multiple key
                        algorithms for the same listener. This has been fixed.
                        [GL #2756]
```

```
        --- 9.17.16 released ---

5671.   [bug]           A race condition could occur where two threads were
                        competing for the same set of key file locks, leading to
                        a deadlock. This has been fixed. [GL #2786]

5670.   [bug]           create_keydata() created an invalid placeholder keydata
                        record upon a refresh failure, which prevented the
                        database of managed keys from subsequently being read
                        back. This has been fixed. [GL #2686]

5669.   [func]          KASP support was extended with the "check DS" feature.
                        Zones with "dnssec-policy" and "parental-agents"
                        configured now check for DS presence and can perform
                        automatic KSK rollovers. [GL #1126]

5668.   [bug]           Rescheduling a setnsec3param() task when a zone failed
                        to load on startup caused a hang on shutdown. This has
                        been fixed. [GL #2791]

5667.   [bug]           The configuration-checking code failed to account for
```

(continues on next page)

(continued from previous page)

```
                        the inheritance rules of the "dnssec-policy" option.
                        This has been fixed. [GL #2780]

5666.   [doc]           The safe "edns-udp-size" value was tweaked to match the
                        probing value from BIND 9.16 for better compatibility.
                        [GL #2183]

5665.   [bug]           If nsupdate sends an SOA request and receives a REFUSED
                        response, it now fails over to the next available
                        server. [GL #2758]

5664.   [func]          For UDP messages larger than the path MTU, named now
                        sends an empty response with the TC (TrunCated) bit set.
                        In addition, setting the DF (Don't Fragment) flag on
                        outgoing UDP sockets was re-enabled. [GL #2790]

5663.   [bug]           Non-zero OPCODEs are now properly handled when receiving
                        queries over DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH)
                        channels. [GL #2787]

5662.   [bug]           Views with recursion disabled are now configured with a
                        default cache size of 2 MB unless "max-cache-size" is
                        explicitly set. This prevents cache RBT hash tables from
                        being needlessly preallocated for such views. [GL #2777]

5661.   [bug]           Change 5644 inadvertently introduced a deadlock: when
                        locking the key file mutex for each zone structure in a
                        different view, the "in-view" logic was not considered.
                        This has been fixed. [GL #2783]

5660.   [bug]           The configuration-checking code failed to account for
                        the inheritance rules of the "key-directory" option.
                        [GL #2778]

                        This change was included in BIND 9.17.15.

5659.   [bug]           When preparing DNS responses, named could replace the
                        letters 'W' (uppercase) and 'w' (lowercase) with '\000'.
                        This has been fixed. [GL #2779]

                        This change was included in BIND 9.17.15.

5658.   [bug]           Increasing "max-cache-size" for a running named instance
                        (using "rndc reconfig") did not cause the hash tables
                        used by cache databases to be grown accordingly. This
                        has been fixed. [GL #2770]

5657.   [cleanup]       Support was removed for both built-in atomics in old
                        versions of Clang (< 3.6.0) and GCC (< 4.7.0), and
                        atomics emulated with a mutex. [GL #2606]

5656.   [bug]           Named now ensures that large responses work correctly
```

(continues on next page)

(continued from previous page)

```
                        over DNS-over-HTTPS (DoH), and that zone transfer
                        requests over DoH are explicitly rejected. [GL !5148]

5655.   [bug]           Signed, insecure delegation responses prepared by named
                        either lacked the necessary NSEC records or contained
                        duplicate NSEC records when both wildcard expansion and
                        CNAME chaining were required to prepare the response.
                        This has been fixed. [GL #2759]

5654.   [port]          Windows support has been removed. [GL #2690]

5653.   [bug]           A bug that caused the NSEC3 salt to be changed on every
                        restart for zones using KASP has been fixed. [GL #2725]
```

```
        --- 9.17.14 released ---

5652.   [bug]           A copy-and-paste error in change 5584 caused the
                        IP_DONTFRAG socket option to be enabled instead of
                        disabled. This has been fixed. [GL #2746]

5651.   [func]          Refactor zone dumping to be processed asynchronously via
                        the uv_work_t thread pool API. [GL #2732]

5650.   [bug]           Prevent a crash that could occur if serve-stale was
                        enabled and a prefetch was triggered during a query
                        restart. [GL #2733]

5649.   [bug]           If a query was answered with stale data on a server with
                        DNS64 enabled, an assertion could occur if a non-stale
                        answer arrived afterward. [GL #2731]

5648.   [bug]           The calculation of the estimated IXFR transaction size
                        in dns_journal_iter_init() was invalid. [GL #2685]

5647.   [func]          The interface manager has been refactored to use fewer
                        client manager objects, which in turn use fewer memory
                        contexts and tasks. This should result in less
                        fragmented memory and better startup performance.
                        [GL #2433]

5646.   [bug]           The default TCP timeout for rndc has been increased to
                        60 seconds. This was its original value, but it had been
                        inadvertently lowered to 10 when rndc was updated to use
                        the network manager. [GL #2643]

5645.   [cleanup]       Remove the rarely-used dns_name_copy() function and
                        rename dns_name_copynf() to dns_name_copy(). [GL !5081]

5644.   [bug]           Fix a race condition in reading and writing key files
                        for zones using KASP and configured in multiple views.
                        [GL #1875]
```

(continues on next page)

(continued from previous page)

```
5643.   [placeholder]

5642.   [bug]           Zones which are configured in multiple views with
                        different values set for "dnssec-policy" and with
                        identical values set for "key-directory" are now
                        detected and treated as a configuration error.
                        [GL #2463]

5641.   [bug]           Address a potential memory leak in
                        dst_key_fromnamedfile(). [GL #2689]

5640.   [func]          Add new configuration options for setting the size of
                        receive and send buffers in the operating system:
                        "tcp-receive-buffer", "tcp-send-buffer",
                        "udp-receive-buffer", and "udp-send-buffer". [GL #2313]

5639.   [bug]           Check that the first and last SOA record of an AXFR are
                        consistent. [GL #2528]
```

```
        --- 9.17.13 released ---

5638.   [bug]           Improvements related to network manager/task manager
                        integration:
                        - isc_managers_create() and isc_managers_destroy()
                          functions were added to handle setup and teardown of
                          netmgr, taskmgr, timermgr, and socketmgr, since these
                          require a precise order of operations now.
                        - Event queue processing is now quantized to prevent
                          infinite looping.
                        - The netmgr can now be paused from within a netmgr
                          thread.
                        - Deadlocks due to a conflict between netmgr's
                          pause/resume and listen/stoplistening operations were
                          fixed.
                        [GL #2654]

5637.   [placeholder]

5636.   [bug]           named and named-checkconf did not report an error when
                        multiple zones with the "dnssec-policy" option set were
                        using the same zone file. This has been fixed.
                        [GL #2603]

5635.   [bug]           Journal compaction could fail when a journal with
                        invalid transaction headers was not detected at startup.
                        This has been fixed. [GL #2670]

5634.   [bug]           If "dnssec-policy" was active and a private key file was
                        temporarily offline during a rekey event, named could
                        incorrectly introduce replacement keys and break a
                        signed zone. This has been fixed. [GL #2596]
```

(continues on next page)

(continued from previous page)

```
5633.   [doc]           The "inline-signing" option was incorrectly described as
                        being inherited from the "options"/"view" levels and was
                        incorrectly accepted at those levels without effect.
                        This has been fixed. [GL #2536]

5632.   [func]          Add a new built-in KASP, "insecure", which is used to
                        transition a zone from a signed to an unsigned state.
                        The existing built-in KASP "none" should no longer be
                        used to unsign a zone. [GL #2645]

5631.   [protocol]      Update the implementation of the ZONEMD RR type to match
                        RFC 8976. [GL #2658]

5630.   [func]          Treat DNSSEC responses containing NSEC3 records with
                        iteration counts greater than 150 as insecure.
                        [GL #2445]

5629.   [func]          Reduce the maximum supported number of NSEC3 iterations
                        that can be configured for a zone to 150. [GL #2642]

5628.   [bug]           Host and nslookup could crash upon receiving a SERVFAIL
                        response. This has been fixed. [GL #2564]

5627.   [bug]           RRSIG(SOA) RRsets placed anywhere other than at the zone
                        apex were triggering infinite resigning loops. This has
                        been fixed. [GL #2650]

5626.   [bug]           When generating zone signing keys, KASP now also checks
                        for key ID conflicts among newly created keys, rather
                        than just between new and existing ones. [GL #2628]

5625.   [bug]           A deadlock could occur when multiple "rndc addzone",
                        "rndc delzone", and/or "rndc modzone" commands were
                        invoked simultaneously for different zones. This has
                        been fixed. [GL #2626]

5624.   [func]          Task manager events are now processed inside network
                        manager loops. The task manager no longer needs its own
                        set of worker threads, which improves resolver
                        performance. [GL #2638]

5623.   [bug]           When named was shut down during an ongoing zone
                        transfer, xfrin_fail() could incorrectly be called
                        twice. This has been fixed. [GL #2630]

5622.   [cleanup]       The lib/samples/ directory has been removed, as export
                        versions of libraries are no longer maintained.
                        [GL !4835]

5621.   [placeholder]

5620.   [bug]           If zone journal files written by BIND 9.16.11 or earlier
```

(continues on next page)

(continued from previous page)

```
                        were present when BIND was upgraded, the zone file for
                        that zone could have been inadvertently rewritten with
                        the current zone contents. This caused the original zone
                        file structure (e.g. comments, $INCLUDE directives) to
                        be lost, although the zone data itself was preserved.
                        This has been fixed. [GL #2623]

5619.   [protocol]      Implement draft-vandijk-dnsop-nsec-ttl, updating the
                        protocol such that NSEC(3) TTL values are set to the
                        minimum of the SOA MINIMUM value or the SOA TTL.
                        [GL #2347]

5618.   [bug]           Change 5149 introduced some inconsistencies in the way
                        record TTLs were presented in cache dumps. These
                        inconsistencies have been eliminated. [GL #389]
                        [GL #2289]
```

```
        --- 9.17.12 released ---

5617.   [placeholder]

5616.   [security]      named crashed when a DNAME record placed in the ANSWER
                        section during DNAME chasing turned out to be the final
                        answer to a client query. (CVE-2021-25215) [GL #2540]

5615.   [security]      Insufficient IXFR checks could result in named serving a
                        zone without an SOA record at the apex, leading to a
                        RUNTIME_CHECK assertion failure when the zone was
                        subsequently refreshed. This has been fixed by adding an
                        owner name check for all SOA records which are included
                        in a zone transfer. (CVE-2021-25214) [GL #2467]

5614.   [bug]           Ensure all resources are properly cleaned up when a call
                        to gss_accept_sec_context() fails. [GL #2620]

5613.   [bug]           It was possible to write an invalid transaction header
                        in the journal file for a managed-keys database after
                        upgrading. This has been fixed. Invalid headers in
                        existing journal files are detected and named is able
                        to recover from them. [GL #2600]

5612.   [bug]           Continued refactoring of the network manager:
                        - allow recovery from read and connect timeout events,
                        - ensure that calls to isc_nm_*connect() always
                          return the connection status via a callback
                          function.
                        [GL #2401]

5611.   [func]          Set "stale-answer-client-timeout" to "off" by default.
                        [GL #2608]

5610.   [bug]           Prevent a crash which could happen when a lookup
```

(continues on next page)

(continued from previous page)

```
                        triggered by "stale-answer-client-timeout" was attempted
                        right after recursion for a client query finished.
                        [GL #2594]

5609.   [func]          The ISC implementation of SPNEGO was removed from BIND 9
                        source code. It was no longer necessary as all major
                        contemporary Kerberos/GSSAPI libraries include support
                        for SPNEGO. [GL #2607]

5608.   [bug]           When sending queries over TCP, dig now properly handles
                        "+tries=1 +retry=0" by not retrying the connection when
                        the remote server closes the connection prematurely.
                        [GL #2490]

5607.   [bug]           As "rndc dnssec -checkds" and "rndc dnssec -rollover"
                        commands may affect the next scheduled key event,
                        reconfiguration of zone keys is now triggered after
                        receiving either of these commands to prevent
                        unnecessary key rollover delays. [GL #2488]

5606.   [bug]           CDS/CDNSKEY DELETE records are now removed when a zone
                        transitions from a secure to an insecure state.
                        named-checkzone also no longer reports an error when
                        such records are found in an unsigned zone. [GL #2517]

5605.   [bug]           "dig -u" now uses the CLOCK_REALTIME clock source for
                        more accurate time reporting. [GL #2592]

5604.   [experimental]  A "filter-a.so" plugin, which is similar to the
                        "filter-aaaa.so" plugin but which omits A records
                        instead of AAAA records, has been added. Thanks to
                        GitLab user @treysis. [GL #2585]

5603.   [placeholder]

5602.   [bug]           Fix TCPDNS and TLSDNS timers in Network Manager. This
                        makes the "tcp-initial-timeout" and "tcp-idle-timeout"
                        options work correctly again. [GL #2583]

5601.   [bug]           Zones using KASP could not be thawed after they were
                        frozen using "rndc freeze". This has been fixed.
                        [GL #2523]

5600.   [bug]           Send a full certificate chain instead of just the leaf
                        certificate to DNS-over-TLS (DoT) and DNS-over-HTTPS
                        (DoH) clients. This makes BIND 9 DoT/DoH servers
                        compatible with a broader set of clients. [GL #2514]

5599.   [bug]           Fix a named crash which occurred after skipping a
                        primary server while transferring a zone over TLS.
                        [GL #2562]
```

(continues on next page)

(continued from previous page)

```
5598.   [port]          Silence -Wchar-subscripts compiler warnings triggered on
                        some platforms due to calling character classification
                        functions declared in the <ctype.h> header with
                        arguments of type char. [GL #2567]
```

```
        --- 9.17.11 released ---

5597.   [bug]           When serve-stale was enabled and starting the recursive
                        resolution process for a query failed, a named instance
                        could crash if it was configured as both a recursive and
                        authoritative server. This problem was introduced by
                        change 5573 and has now been fixed. [GL #2565]

5596.   [func]          Client-side support for DNS-over-HTTPS (DoH) has been
                        added to dig. "dig +https" can now query a server via
                        HTTP/2. [GL #1641]

5595.   [cleanup]       Public header files for BIND 9 libraries no longer
                        directly include third-party library headers. This
                        prevents the need to include paths to third-party header
                        files in CFLAGS whenever BIND 9 public header files are
                        used, which could cause build-time issues on hosts with
                        older versions of BIND 9 installed. [GL #2357]

5594.   [bug]           Building with --enable-dnsrps --enable-dnsrps-dl failed.
                        [GL #2298]

5593.   [bug]           Journal files written by older versions of named can now
                        be read when loading zones, so that journal
                        incompatibility does not cause problems on upgrade.
                        Outdated journals are updated to the new format after
                        loading. [GL #2505]

5592.   [bug]           Prevent hazard pointer table overflows on machines with
                        many cores, by allowing the thread IDs (serving as
                        indices into hazard pointer tables) of finished threads
                        to be reused by those created later. [GL #2396]

5591.   [bug]           Fix a crash that occurred when
                        "stale-answer-client-timeout" was triggered without any
                        (stale) data available in the cache to answer the query.
                        [GL #2503]

5590.   [bug]           NSEC3 records were not immediately created for dynamic
                        zones using NSEC3 with "dnssec-policy", resulting in
                        such zones going bogus. Add code to process the
                        NSEC3PARAM queue at zone load time so that NSEC3 records
                        for such zones are created immediately. [GL #2498]

5589.   [placeholder]

5588.   [func]          Add a new "purge-keys" option for "dnssec-policy". This
```

(continues on next page)

(continued from previous page)

```
                        option determines the period of time for which key files
                        are retained after they become obsolete. [GL #2408]

5587.   [bug]           A standalone libtool script no longer needs to be
                        present in PATH to build BIND 9 from a source tarball
                        prepared using "make dist". [GL #2504]

5586.   [bug]           An invalid direction field in a LOC record resulted in
                        an INSIST failure when a zone file containing such a
                        record was loaded. [GL #2499]

5585.   [func]          Memory contexts and memory pool implementations were
                        refactored to reduce lock contention for shared memory
                        contexts by replacing mutexes with atomic operations.
                        The internal memory allocator was simplified so that it
                        is only a thin wrapper around the system allocator. This
                        change made the "-M external" named option redundant and
                        it was therefore removed. [GL #2433]

5584.   [bug]           No longer set the IP_DONTFRAG option on UDP sockets, to
                        prevent dropping outgoing packets exceeding
                        "max-udp-size". [GL #2466]

5583.   [func]          Changes to DNS-over-HTTPS (DoH) configuration syntax:
                        - When "http" is specified in "listen-on" or
                          "listen-on-v6" statements, "tls" must also now be
                          specified. If an unencrypted connection is desired
                          (for example, when running behind a reverse proxy),
                          use "tls none".
                        - "http default" can now be specified in "listen-on" and
                          "listen-on-v6" statements to use the default HTTP
                          endpoint of "/dns-query". It is no longer necessary to
                          include an "http" statement in named.conf unless
                          overriding this value.
                        [GL #2472]

5582.   [bug]           BIND 9 failed to build when static OpenSSL libraries
                        were used and the pkg-config files for libssl and/or
                        libcrypto were unavailable. This has been fixed by
                        ensuring that the correct linking order for libssl and
                        libcrypto is always used. [GL #2402]

5581.   [bug]           Fix a memory leak that occurred when inline-signed zones
                        were added to the configuration, followed by a
                        reconfiguration of named. [GL #2041]

5580.   [test]          The system test framework no longer differentiates
                        between SKIPPED and UNTESTED system test results. Any
                        system test which is not run is now marked as SKIPPED.
                        [GL !4517]

5579.   [bug]           If an invalid key name (e.g. "a..b") was specified in a
```

(continues on next page)

(continued from previous page)

```
                        primaries list in named.conf, the wrong size was passed
                        to isc_mem_put(), resulting in the returned memory being
                        put on the wrong free list. This prevented named from
                        starting up. [GL #2460]
```

```
        --- 9.17.10 released ---

5578.   [protocol]      Make "check-names" accept A records below "_spf",
                        "_spf_rate", and "_spf_verify" labels in order to cater
                        for the "exists" SPF mechanism specified in RFC 7208
                        section 5.7 and appendix D.1. [GL #2377]

5577.   [bug]           Fix the "three is a crowd" key rollover bug in KASP by
                        correctly implementing Equation (2) of the "Flexible and
                        Robust Key Rollover" paper. [GL #2375]

5576.   [experimental]  Initial server-side implementation of DNS-over-HTTPS
                        (DoH). Support for both TLS-encrypted and unencrypted
                        HTTP/2 connections has been added to the network manager
                        and integrated into named. (Note: there is currently no
                        client-side support for DNS-over-HTTPS; this will be
                        added to dig in a future release.) [GL #1144]

5575.   [bug]           When migrating to KASP, BIND 9 considered keys with the
                        "Inactive" and/or "Delete" timing metadata to be
                        possible active keys. This has been fixed. [GL #2406]

5574.   [func]          Incoming zone transfers can now use TLS. Addresses in a
                        "primaries" list take an optional "tls" argument,
                        specifying either a previously configured "tls" block or
                        "ephemeral"; SOA queries and zone transfer requests are
                        then sent via TLS. [GL #2392]

5573.   [func]          When serve-stale is enabled and stale data is available,
                        named now returns stale answers upon encountering any
                        unexpected error in the query resolution process.
                        However, the "stale-refresh-time" window is still only
                        started upon a timeout. [GL #2434]

5572.   [bug]           Address potential double free in generatexml().
                        [GL #2420]

5571.   [bug]           named failed to start when its configuration included a
                        zone with a non-builtin "allow-update" ACL attached.
                        [GL #2413]

5570.   [bug]           Improve performance of the DNSSEC verification code by
                        reducing the number of repeated calls to
                        dns_dnssec_keyfromrdata(). [GL #2073]

5569.   [bug]           Emit useful error message when "rndc retransfer" is
                        applied to a zone of inappropriate type. [GL #2342]
```

(continues on next page)

(continued from previous page)

```
5568.   [bug]           Fixed a crash in "dnssec-keyfromlabel" when using ECDSA
                        keys. [GL #2178]

5567.   [bug]           Dig now reports unknown dash options while pre-parsing
                        the options. This prevents "-multi" instead of "+multi"
                        from reporting memory usage before ending option parsing
                        with "Invalid option: -lti". [GL #2403]

5566.   [func]          Add "stale-answer-client-timeout" option, which is the
                        amount of time a recursive resolver waits before
                        attempting to answer the query using stale data from
                        cache. [GL #2247]

5565.   [func]          The SONAMEs for BIND 9 libraries now include the current
                        BIND 9 version number, in an effort to tightly couple
                        internal libraries with a specific release. [GL #2387]

5564.   [cleanup]       Network manager's TLSDNS module was refactored to use
                        libuv and libssl directly instead of a stack of TCP/TLS
                        sockets. [GL #2335]

5563.   [cleanup]       Changed several obsolete configuration options to
                        ancient, making them fatal errors. Also cleaned up the
                        number of clause flags in the configuration parser.
                        [GL #1086]

5562.   [placeholder]

5561.   [bug]           KASP incorrectly set signature validity to the value of
                        the DNSKEY signature validity. This is now fixed.
                        [GL #2383]

5560.   [func]          The default value of "max-stale-ttl" has been changed
                        from 12 hours to 1 day and the default value of
                        "stale-answer-ttl" has been changed from 1 second to 30
                        seconds, following RFC 8767 recommendations. [GL #2248]
```

```
        --- 9.17.9 released ---

5559.   [bug]           The --with-maxminddb=PATH form of the build-time option
                        enabling support for libmaxminddb was not working
                        correctly. This has been fixed. [GL #2366]

5558.   [bug]           Asynchronous hook modules could trigger an assertion
                        failure when the fetch handle was detached too late.
                        Thanks to Jinmei Tatuya at Infoblox. [GL #2379]

5557.   [bug]           Prevent RBTDB instances from being destroyed by multiple
                        threads at the same time. [GL #2317]

5556.   [bug]           Further tweak newline printing in dnssec-signzone and
```

(continues on next page)

(continued from previous page)

```
                        dnssec-verify. [GL #2359]

5555.   [placeholder]

5554.   [bug]           dnssec-signzone and dnssec-verify were missing newlines
                        between log messages. [GL #2359]

5553.   [bug]           When reconfiguring named, removing "auto-dnssec" did not
                        turn off DNSSEC maintenance. [GL #2341]

5552.   [func]          When switching to "dnssec-policy none;", named now
                        permits a safe transition to insecure mode and publishes
                        the CDS and CDNSKEY DELETE records, as described in RFC
                        8078. [GL #1750]

5551.   [bug]           named no longer attempts to assign threads to CPUs
                        outside the CPU affinity set. Thanks to Ole Bjørn
                        Hessen. [GL #2245]

5550.   [func]          dnssec-signzone and named now log a warning when falling
                        back to the "increment" SOA serial method. [GL #2058]

5549.   [protocol]      ipv4only.arpa is now served when DNS64 is configured.
                        [GL #385]

5548.   [placeholder]

5547.   [placeholder]
```

```
        --- 9.17.8 released ---

5546.   [placeholder]

5545.   [func]          OS support for load-balanced sockets is no longer
                        required to receive incoming queries in multiple netmgr
                        threads. [GL #2137]

5544.   [func]          Restore the default value of "nocookie-udp-size" to 4096
                        bytes. [GL #2250]

5543.   [bug]           Fix UDP performance issues caused by making netmgr
                        callbacks asynchronous-only. [GL #2320]

5542.   [bug]           Refactor netmgr. [GL #1920] [GL #2034] [GL #2061]
                        [GL #2194] [GL #2221] [GL #2266] [GL #2283] [GL #2318]
                        [GL #2321]

5541.   [func]          Adjust the "max-recursion-queries" default from 75 to
                        100. [GL #2305]

5540.   [port]          Fix building with native PKCS#11 support for AEP Keyper.
                        [GL #2315]
```

(continues on next page)

(continued from previous page)

```
5539.   [bug]           Tighten handling of missing DNS COOKIE responses over
                        UDP by falling back to TCP. [GL #2275]

5538.   [func]          Add NSEC3 support to KASP. A new option for
                        "dnssec-policy", "nsec3param", can be used to set the
                        desired NSEC3 parameters. NSEC3 salt collisions are
                        automatically prevented during resalting. Salt
                        generation is now logged with zone context. [GL #1620]

5537.   [func]          The query plugin mechanism has been extended
                        to support asynchronous operations. For example, a
                        plugin can now trigger recursion and resume
                        processing when it is complete. Thanks to Jinmei
                        Tatuya at Infoblox. [GL #2141]

5536.   [func]          Dig can now report the DNS64 prefixes in use
                        (+dns64prefix). [GL #1154]

5535.   [bug]           dig/nslookup/host could crash on shutdown after an
                        interrupt. [GL #2287] [GL #2288]

5534.   [bug]           The CNAME synthesized from a DNAME was incorrectly
                        followed when the QTYPE was CNAME or ANY. [GL #2280]
```

```
        --- 9.17.7 released ---

5533.   [func]          Add the "stale-refresh-time" option, a time window that
                        starts after a failed lookup, during which a stale RRset
                        is served directly from cache before a new attempt to
                        refresh it is made. [GL #2066]

5532.   [cleanup]       Unused header files were removed:
                        bin/rndc/include/rndc/os.h, lib/isc/timer_p.h,
                        lib/isccfg/include/isccfg/dnsconf.h and code related
                        to those files. [GL #1913]

5531.   [func]          Add support for DNS over TLS (DoT) to dig and named.
                        dig output now includes the transport protocol used.
                        [GL #1816] [GL #1840]

5530.   [bug]           dnstap did not capture responses to forwarded UPDATE
                        requests. [GL #2252]

5529.   [func]          The network manager API is now used by named to send
                        zone transfer requests. [GL #2016]

5528.   [func]          Convert dig, host, and nslookup to use the network
                        manager API. As a side effect of this change, "dig
                        +unexpected" no longer works, and has been disabled.
                        [GL #2140]
```

(continues on next page)

(continued from previous page)

```
5527.   [bug]           A NULL pointer dereference occurred when creating an NTA
                        recheck query failed. [GL #2244]

5526.   [bug]           Fix a race/NULL dereference in TCPDNS read. [GL #2227]

5525.   [placeholder]

5524.   [func]          Added functionality to the network manager to support
                        outgoing DNS queries in addition to incoming ones.
                        [GL #2235]

5523.   [bug]           The initial lookup in a zone transitioning to/from a
                        signed state could fail if the DNSKEY RRset was not
                        found. [GL #2236]

5522.   [bug]           Fixed a race/NULL dereference in TCPDNS send. [GL #2227]

5521.   [func]          All use of libltdl was dropped. libuv's shared library
                        handling interface is now used instead. [GL !4278]

5520.   [bug]           Fixed a number of shutdown races, reference counting
                        errors, and spurious log messages that could occur
                        in the network manager. [GL #2221]

5519.   [cleanup]       Unused source code was removed: lib/dns/dbtable.c,
                        lib/dns/portlist.c, lib/isc/bufferlist.c, and code
                        related to those files. [GL #2060]

5518.   [bug]           Stub zones now work correctly with primary servers using
                        "minimal-responses yes". [GL #1736]

5517.   [bug]           Do not treat UV_EOF as a TCP4RecvErr or a TCP6RecvErr.
                        [GL #2208]
```

```
        --- 9.17.6 released ---

5516.   [func]          The default EDNS buffer size has been changed from 4096
                        to 1232 bytes, the EDNS buffer size probing has been
                        removed, and named now sets the DF (Don't Fragment) flag
                        on outgoing UDP packets. [GL #2183]

5515.   [func]          Add 'rndc dnssec -rollover' command to trigger a manual
                        rollover for a specific key. [GL #1749]

5514.   [bug]           Fix KASP expected key size for Ed25519 and Ed448.
                        [GL #2171]

5513.   [doc]           The ARM section describing the "rrset-order" statement
                        was rewritten to make it unambiguous and up-to-date with
                        the source code. [GL #2139]

5512.   [bug]           "rrset-order" rules using "order none" were causing
```

(continues on next page)

(continued from previous page)

```
                        named to crash despite named-checkconf treating them as
                        valid. [GL #2139]

5511.   [bug]           'dig -u +yaml' failed to display timestamps to the
                        microsecond. [GL #2190]

5510.   [bug]           Implement the attach/detach semantics for dns_message_t
                        to fix a data race in accessing an already-destroyed
                        fctx->rmessage. [GL #2124]

5509.   [bug]           filter-aaaa: named crashed upon shutdown if it was in
                        the process of recursing for A RRsets. [GL #1040]

5508.   [func]          Added new parameter "-expired" for "rndc dumpdb" that
                        also prints expired RRsets (awaiting cleanup) to the
                        dump file. [GL #1870]

5507.   [bug]           Named could compute incorrect SIG(0) responses.
                        [GL #2109]

5506.   [bug]           Properly handle failed sysconf() calls, so we don't
                        report invalid memory size. [GL #2166]

5505.   [bug]           Updating contents of a mixed-case RPZ could cause some
                        rules to be ignored. [GL #2169]

5504.   [func]          The "glue-cache" option has been marked as deprecated.
                        The glue cache feature will be permanently enabled in a
                        future release. [GL #2146]

5503.   [bug]           Cleaned up reference counting of network manager
                        handles, now using isc_nmhandle_attach() and _detach()
                        instead of _ref() and _unref(). [GL #2122]
```

```
        --- 9.17.5 released ---

5502.   [func]          'dig +bufsize=0' no longer disables EDNS. [GL #2054]

5501.   [func]          Log CDS/CDNSKEY publication. [GL #1748]

5500.   [bug]           Fix (non-)publication of CDS and CDNSKEY records.
                        [GL #2103]

5499.   [func]          Add '-P ds' and '-D ds' arguments to dnssec-settime.
                        [GL #1748]

5498.   [test]          The --with-gperftools-profiler configure option was
                        removed. [GL !4045]

5497.   [placeholder]

5496.   [bug]           Address a TSAN report by ensuring each rate limiter
```

(continues on next page)

(continued from previous page)

```
                        object holds a reference to its task. [GL #2081]

5495.   [bug]           With query minimization enabled, named failed to
                        resolve ip6.arpa. names that had extra labels to the
                        left of the IPv6 part. [GL #1847]

5494.   [bug]           Silence the EPROTO syslog message on older systems.
                        [GL #1928]

5493.   [bug]           Fix off-by-one error when calculating new hash table
                        size. [GL #2104]

5492.   [bug]           Tighten LOC parsing to reject a period (".") and/or "m"
                        as a value. Fix handling of negative altitudes which are
                        not whole meters. [GL #2074]

5491.   [bug]           rbtversion->glue_table_size could be read without the
                        appropriate lock being held. [GL #2080]

5490.   [func]          Refactor readline support to use pkg-config and add
                        support for the editline library. [GL !3942]

5489.   [bug]           Named erroneously accepted certain invalid resource
                        records that were incorrectly processed after
                        subsequently being written to disk and loaded back, as
                        the wire format differed. Such records include: CERT,
                        IPSECKEY, NSEC3, NSEC3PARAM, NXT, SIG, TLSA, WKS, and
                        X25. [GL !3953]

5488.   [bug]           NTA code needed to have a weak reference on its
                        associated view to prevent the latter from being deleted
                        while NTA tests were being performed. [GL #2067]

5487.   [cleanup]       Update managed keys log messages to be less confusing.
                        [GL #2027]

5486.   [func]          Add 'rndc dnssec -checkds' command, which signals to
                        named that the DS record for a given zone or key has
                        been updated in the parent zone. [GL #1613]
```

```
        --- 9.17.4 released ---

5485.   [placeholder]

5484.   [func]          Expire zero TTL records quickly rather than using them
                        for stale answers. [GL #1829]

5483.   [func]          Keeping "stale" answers in cache has been disabled by
                        default and can be re-enabled with a new configuration
                        option "stale-cache-enable". [GL #1712]

5482.   [bug]           If the Duplicate Address Detection (DAD) mechanism had
```

(continues on next page)

(continued from previous page)

```
                        not yet finished after adding a new IPv6 address to the
                        system, BIND 9 would fail to bind to IPv6 addresses in a
                        tentative state. [GL #2038]

5481.   [security]      "update-policy" rules of type "subdomain" were
                        incorrectly treated as "zonesub" rules, which allowed
                        keys used in "subdomain" rules to update names outside
                        of the specified subdomains. The problem was fixed by
                        making sure "subdomain" rules are again processed as
                        described in the ARM. (CVE-2020-8624) [GL #2055]

5480.   [security]      When BIND 9 was compiled with native PKCS#11 support, it
                        was possible to trigger an assertion failure in code
                        determining the number of bits in the PKCS#11 RSA public
                        key with a specially crafted packet. (CVE-2020-8623)
                        [GL #2037]

5479.   [security]      named could crash in certain query resolution scenarios
                        where QNAME minimization and forwarding were both
                        enabled. (CVE-2020-8621) [GL #1997]

5478.   [security]      It was possible to trigger an assertion failure by
                        sending a specially crafted large TCP DNS message.
                        (CVE-2020-8620) [GL #1996]

5477.   [bug]           The idle timeout for connected TCP sockets, which was
                        previously set to a high fixed value, is now derived
                        from the client query processing timeout configured for
                        a resolver. [GL #2024]

5476.   [security]      It was possible to trigger an assertion failure when
                        verifying the response to a TSIG-signed request.
                        (CVE-2020-8622) [GL #2028]

5475.   [bug]           Wildcard RPZ passthru rules could incorrectly be
                        overridden by other rules that were loaded from RPZ
                        zones which appeared later in the "response-policy"
                        statement. This has been fixed. [GL #1619]

5474.   [bug]           dns_rdata_hip_next() failed to return ISC_R_NOMORE
                        when it should have. [GL !3880]

5473.   [func]          The RBT hash table implementation has been changed
                        to use a faster hash function (HalfSipHash2-4) and
                        Fibonacci hashing for better distribution. Setting
                        "max-cache-size" now preallocates a fixed-size hash
                        table so that rehashing does not cause resolution
                        brownouts while the hash table is grown. [GL #1775]

5472.   [func]          The statistics channel has been updated to use the
                        new network manager. [GL #2022]
```

(continues on next page)

(continued from previous page)

```
5471.   [bug]           The introduction of KASP support inadvertently caused
                        the second field of "sig-validity-interval" to always be
                        calculated in hours, even in cases when it should have
                        been calculated in days. This has been fixed. (Thanks to
                        Tony Finch.) [GL !3735]

5470.   [port]          gsskrb5_register_acceptor_identity() is now only called
                        if gssapi_krb5.h is present. [GL #1995]

5469.   [port]          On illumos, a constant called SEC is already defined in
                        <sys/time.h>, which conflicts with an identically named
                        constant in libbind9. This conflict has been resolved.
                        [GL #1993]

5468.   [bug]           Addressed potential double unlock in process_fd().
                        [GL #2005]

5467.   [func]          The control channel and the rndc utility have been
                        updated to use the new network manager. To support
                        this, the network manager was updated to enable
                        the initiation of client TCP connections. Its
                        internal reference counting has been refactored.

                        Note: As a side effect of this change, rndc cannot
                        currently be used with UNIX-domain sockets, and its
                        default timeout has changed from 60 seconds to 30.
                        These will be addressed in a future release.
                        [GL #1759]

5466.   [bug]           Addressed an error in recursive clients stats reporting.
                        [GL #1719]

5465.   [func]          Added fallback to built-in trust-anchors, managed-keys,
                        or trusted-keys if the bindkeys-file (bind.keys) cannot
                        be parsed. [GL #1235]

5464.   [bug]           Requesting more than 128 files to be saved when rolling
                        dnstap log files caused a buffer overflow. This has been
                        fixed. [GL #1989]

5463.   [placeholder]

5462.   [bug]           Move LMDB locking from LMDB itself to named. [GL #1976]

5461.   [bug]           The STALE rdataset header attribute was updated while
                        the write lock was not being held, leading to incorrect
                        statistics. The header attributes are now converted to
                        use atomic operations. [GL #1475]

5460.   [cleanup]       tsig-keygen was previously an alias for
                        ddns-confgen and was documented in the ddns-confgen
                        man page. This has been reversed; tsig-keygen is
```

(continues on next page)

(continued from previous page)

```
                        now the primary name. [GL #1998]

5459.   [bug]           Fixed bad isc_mem_put() size when an invalid type was
                        specified in an "update-policy" rule. [GL #1990]
```

```
        --- 9.17.3 released ---

5458.   [bug]           Prevent a theoretically possible NULL dereference caused
                        by a data race between zone_maintenance() and
                        dns_zone_setview_helper(). [GL #1627]

5457.   [placeholder]

5456.   [func]          Added "primaries" as a synonym for "masters" in
                        named.conf, and "primary-only" as a synonym for
                        "master-only" in the parameters to "notify", to bring
                        terminology up-to-date with RFC 8499. [GL #1948]

5455.   [bug]           named could crash when cleaning dead nodes in
                        lib/dns/rbtdb.c that were being reused. [GL #1968]

5454.   [bug]           Address a startup crash that occurred when the server
                        was under load and the root zone had not yet been
                        loaded. [GL #1862]

5453.   [bug]           named crashed on shutdown when a new rndc connection was
                        received during shutdown. [GL #1747]

5452.   [bug]           The "blackhole" ACL was accidentally disabled for client
                        queries. [GL #1936]

5451.   [func]          Add 'rndc dnssec -status' command. [GL #1612]

5450.   [placeholder]

5449.   [bug]           Fix a socket shutdown race in netmgr udp. [GL #1938]

5448.   [bug]           Fix a race condition in isc__nm_tcpdns_send().
                        [GL #1937]

5447.   [bug]           IPv6 addresses ending in "::" could break YAML
                        parsing. A "0" is now appended to such addresses
                        in YAML output from dig, mdig, delv, and dnstap-read.
                        [GL #1952]

5446.   [bug]           The validator could fail to accept a properly signed
                        RRset if an unsupported algorithm appeared earlier in
                        the DNSKEY RRset than a supported algorithm. It could
                        also stop if it detected a malformed public key.
                        [GL #1689]

5445.   [cleanup]       Disable and disallow static linking. [GL #1933]
```

(continues on next page)

(continued from previous page)

```
5444.   [bug]           'rndc dnstap -roll <value>' did not limit the number of
                        saved files to <value>. [GL !3728]

5443.   [bug]           The "primary" and "secondary" keywords, when used
                        as parameters for "check-names", were not
                        processed correctly and were being ignored. [GL #1949]

5442.   [func]          Add support for outgoing TCP connections in netmgr.
                        [GL #1958]

5441.   [placeholder]

5440.   [placeholder]

5439.   [bug]           The DS RRset returned by dns_keynode_dsset() was used in
                        a non-thread-safe manner. [GL #1926]
```

```
        --- 9.17.2 released ---

5438.   [bug]           Fix a race in TCP accepting code. [GL #1930]

5437.   [bug]           Fix a data race in lib/dns/resolver.c:log_formerr().
                        [GL #1808]

5436.   [security]      It was possible to trigger an INSIST when determining
                        whether a record would fit into a TCP message buffer.
                        (CVE-2020-8618) [GL #1850]

5435.   [tests]         Add RFC 4592 responses examples to the wildcard system
                        test. [GL #1718]

5434.   [security]      It was possible to trigger an INSIST in
                        lib/dns/rbtdb.c:new_reference() with a particular zone
                        content and query patterns. (CVE-2020-8619) [GL #1111]
                        [GL #1718]

5433.   [placeholder]

5432.   [bug]           Check the question section when processing AXFR, IXFR,
                        and SOA replies when transferring a zone in. [GL #1683]

5431.   [func]          Reject DS records at the zone apex when loading
                        master files. Log but otherwise ignore attempts to
                        add DS records at the zone apex via UPDATE. [GL #1798]

5430.   [doc]           Update docs - with netmgr, a separate listening socket
                        is created for each IPv6 interface (just as with IPv4).
                        [GL #1782]

5429.   [cleanup]       Move BIND binaries which are neither daemons nor
                        administrative programs to $bindir. [GL #1724]
```

(continues on next page)

(continued from previous page)

```
5428.   [bug]           Clean up GSSAPI resources in nsupdate only after taskmgr
                        has been destroyed. Thanks to Petr Menšík. [GL !3316]

5427.   [placeholder]

5426.   [bug]           Don't abort() when setting SO_INCOMING_CPU on the socket
                        fails. [GL #1911]

5425.   [func]          The default value of "max-stale-ttl" has been changed
                        from 1 week to 12 hours. [GL #1877]

5424.   [bug]           With KASP, when creating a successor key, the "goal"
                        state of the current active key (predecessor) was not
                        changed and thus never removed from the zone. [GL #1846]

5423.   [bug]           Fix a bug in keymgr_key_has_successor(): it incorrectly
                        returned true if any other key in the keyring had a
                        successor. [GL #1845]

5422.   [bug]           When using dnssec-policy, print correct key timing
                        metadata. [GL #1843]

5421.   [bug]           Fix a race that could cause named to crash when looking
                        up the nodename of an RBT node if the tree was modified.
                        [GL #1857]

5420.   [bug]           Add missing isc_{mutex,conditional}_destroy() calls
                        that caused a memory leak on FreeBSD. [GL #1893]

5419.   [func]          Add new dig command line option, "+qid=<num>", which
                        allows the query ID to be set to an arbitrary value.
                        Add a new ./configure option, --enable-singletrace,
                        which allows trace logging of a single query when QID is
                        set to 0. [GL #1851]

5418.   [bug]           delv failed to parse deprecated trusted-keys-style
                        trust anchors. [GL #1860]

5417.   [cleanup]       The code determining the advertised UDP buffer size in
                        outgoing EDNS queries has been refactored to improve its
                        clarity. [GL #1868]

5416.   [bug]           Fix a lock order inversion in lib/isc/unix/socket.c.
                        [GL #1859]

5415.   [test]          Address race in dnssec system test that led to
                        test failures. [GL #1852]

5414.   [test]          Adjust time allowed for journal truncation to occur
                        in nsupdate system test to avoid test failure.
                        [GL #1855]
```

(continues on next page)

(continued from previous page)

```
5413.   [test]          Address race in autosign system test that led to
                        test failures. [GL #1852]

5412.   [bug]           'provide-ixfr no;' failed to return up-to-date responses
                        when the serial was greater than or equal to the
                        current serial. [GL #1714]

5411.   [cleanup]       TCP accept code has been refactored to use a single
                        accept() and pass the accepted socket to child threads
                        for processing. [GL !3320]

5410.   [func]          Add the ability to specify per-type record count limits,
                        which are enforced when adding records via UPDATE, in an
                        "update-policy" statement. [GL #1657]

5409.   [performance]   When looking up NSEC3 data in a zone database, skip the
                        check for empty non-terminal nodes; the NSEC3 tree does
                        not have any. [GL #1834]

5408.   [protocol]      Print Extended DNS Errors if present in OPT record.
                        [GL #1835]

5407.   [func]          Zone timers are now exported via statistics channel.
                        Thanks to Paul Frieden, Verizon Media. [GL #1232]

5406.   [func]          Add a new logging category, "rpz-passthru", which allows
                        RPZ passthru actions to be logged in a separate channel.
                        [GL #54]

5405.   [bug]           'named-checkconf -p' could include spurious text in
                        server-addresses statements due to an uninitialized DSCP
                        value. [GL #1812]

5404.   [bug]           'named-checkconf -z' could incorrectly indicate
                        success if errors were found in one view but not in a
                        subsequent one. [GL #1807]

5403.   [func]          Do not set UDP receive/send buffer sizes - use system
                        defaults. [GL #1713]

5402.   [bug]           On FreeBSD, use SO_REUSEPORT_LB instead of SO_REUSEPORT.
                        Enable use of SO_REUSEADDR on all platforms which
                        support it. [GL !3365]

5401.   [bug]           The number of input queues allocated during dnstap
                        initialization was too low, which could prevent some
                        dnstap data from being logged. [GL #1795]

5400.   [func]          Add engine support to OpenSSL EdDSA implementation.
                        [GL #1763]
```

(continues on next page)

(continued from previous page)

```
5399.   [func]          Add engine support to OpenSSL ECDSA implementation.
                        [GL #1534]

5398.   [bug]           Named could fail to restart if a zone with a double
                        quote (") in its name was added with 'rndc addzone'.
                        [GL #1695]

5397.   [func]          Update PKCS#11 EdDSA implementation to PKCS#11 v3.0.
                        Thanks to Aaron Thompson. [GL !3326]

5396.   [func]          When necessary (i.e. in libuv >= 1.37), use the
                        UV_UDP_RECVMMSG flag to enable recvmmsg() support in
                        libuv. [GL #1797]

5395.   [security]      Further limit the number of queries that can be
                        triggered from a request.  Root and TLD servers
                        are no longer exempt from max-recursion-queries.
                        Fetches for missing name server address records
                        are limited to 4 for any domain. (CVE-2020-8616)
                        [GL #1388]

5394.   [cleanup]       Named formerly attempted to change the effective UID and
                        GID in named_os_openfile(), which could trigger a
                        spurious log message if they were already set to the
                        desired values. This has been fixed. [GL #1042]
                        [GL #1090]

5393.   [cleanup]       Unused and/or redundant APIs were removed from libirs.
                        [GL #1758]

5392.   [bug]           It was possible for named to crash during shutdown
                        or reconfiguration if an RPZ zone was still being
                        updated. [GL #1779]

5391.   [func]          The BIND 9 build system has been changed to use a
                        typical autoconf+automake+libtool stack. When building
                        from the Git repository, run "autoreconf -fi" first.
                        [GL #4]

5390.   [security]      Replaying a TSIG BADTIME response as a request could
                        trigger an assertion failure. (CVE-2020-8617)
                        [GL #1703]

5389.   [bug]           Finish PKCS#11 code cleanup, fix a couple of smaller
                        bugs and use PKCS#11 v3.0 EdDSA macros and constants.
                        Thanks to Aaron Thompson. [GL !3391]

5388.   [func]          Reject AXFR streams where the message ID is not
                        consistent. [GL #1674]

5387.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
5386.   [cleanup]       Address Coverity warnings in lib/dns/keymgr.c.
                        [GL #1737]

5385.   [func]          Make ISC rwlock implementation the default again.
                        [GL #1753]

5384.   [bug]           With "dnssec-policy" in effect, "inline-signing" was
                        implicitly set to "yes". Now "inline-signing" is only
                        set to "yes" if the zone is not dynamic. [GL #1709]
```

```
        --- 9.17.1 released ---

5383.   [func]          Add a quota attach function with a callback and clean up
                        the isc_quota API. [GL !3280]

5382.   [bug]           Use clock_gettime() instead of gettimeofday() for
                        isc_stdtime() function. [GL #1679]

5381.   [bug]           Fix logging API data race by adding rwlock and caching
                        logging levels in stdatomic variables to restore
                        performance to original levels. [GL #1675] [GL #1717]

5380.   [contrib]       Fix building MySQL DLZ modules against MySQL 8
                        libraries. [GL #1678]

5379.   [placeholder]

5378.   [bug]           Receiving invalid DNS data was triggering an assertion
                        failure in nslookup. [GL #1652]

5377.   [placeholder]

5376.   [bug]           Fix ineffective DNS rebinding protection when BIND is
                        configured as a forwarding DNS server. Thanks to Tobias
                        Klein. [GL #1574]

5375.   [test]          Fix timing issues in the "kasp" system test. [GL #1669]

5374.   [bug]           Statistics counters tracking recursive clients and
                        active connections could underflow. [GL #1087]

5373.   [bug]           Collecting statistics for DNSSEC signing operations
                        (change 5254) caused an array of significant size (over
                        100 kB) to be allocated for each configured zone. Each
                        of these arrays is tracking all possible key IDs; this
                        could trigger an out-of-memory condition on servers with
                        a high enough number of zones configured. Fixed by
                        tracking up to four keys per zone and rotating counters
                        when keys are replaced. This fixes the immediate problem
                        of high memory usage, but should be improved in a future
                        release by growing or shrinking the number of keys to
                        track upon key rollover events. [GL #1179]
```

(continues on next page)

(continued from previous page)

```
5372.   [bug]           Fix migration from existing DNSSEC key files
                        ("auto-dnssec maintain") to "dnssec-policy". [GL #1706]

5371.   [bug]           Improve incremental updates of the RPZ summary
                        database to reduce delays that could occur when
                        a policy zone update included a large number of
                        record deletions. [GL #1447]

5370.   [bug]           Deactivation of a netmgr handle associated with a
                        socket could be skipped in some circumstances.
                        Fixed by deactivating the netmgr handle before
                        scheduling the asynchronous close routine. [GL #1700]

5369.   [func]          Add the ability to specify whether to wait for
                        nameserver domain names to be looked up, with a new RPZ
                        modifying directive 'nsdname-wait-recurse'. [GL #1138]

5368.   [bug]           Named failed to restart if 'rndc addzone' names
                        contained special characters (e.g. '/'). [GL #1655]

5367.   [placeholder]
```

```
        --- 9.17.0 released ---

5366.   [bug]           Fix a race condition with the keymgr when the same
                        zone plus dnssec-policy is configured in multiple
                        views. [GL #1653]

5365.   [bug]           Algorithm rollover was stuck on submitting DS
                        because keymgr thought it would move to an invalid
                        state.  Fixed by checking the current key against
                        the desired state, not the existing state. [GL #1626]

5364.   [bug]           Algorithm rollover waited too long before introducing
                        zone signatures.  It waited to make sure all signatures
                        were regenerated, but when introducing a new algorithm,
                        all signatures are regenerated immediately.  Only
                        add the sign delay if there is a predecessor key.
                        [GL #1625]

5363.   [bug]           When changing a dnssec-policy, existing keys with
                        properties that no longer match were not being retired.
                        [GL #1624]

5362.   [func]          Limit the size of IXFR responses so that AXFR will
                        be used instead if it would be smaller. This is
                        controlled by the "max-ixfr-ratio" option, which
                        is a percentage representing the ratio of IXFR size
                        to the size of the entire zone. This value cannot
                        exceed 100%, which is the default. [GL #1515]
```

(continues on next page)

(continued from previous page)

```
5361.   [bug]           named might not accept new connections after
                        hitting tcp-clients quota. [GL #1643]

5360.   [bug]           delv could fail to load trust anchors in DNSKEY
                        format. [GL #1647]

5359.   [func]          "rndc nta -d" and "rndc secroots" now include
                        "validate-except" entries when listing negative
                        trust anchors. These are indicated by the keyword
                        "permanent" in place of an expiry date. [GL #1532]

5358.   [bug]           Inline master zones whose master files were touched
                        but otherwise unchanged and were subsequently reloaded
                        may have stopped re-signing. [GL !3135]

5357.   [bug]           Newly added RRSIG records with expiry times before
                        the previous earliest expiry times might not be
                        re-signed in time.  This was a side effect of 5315.
                        [GL !3137]

5356.   [func]          Update dnssec-policy configuration statements:
                        - Rename "zone-max-ttl" dnssec-policy option to
                          "max-zone-ttl" for consistency with the existing
                          zone option.
                        - Allow for "lifetime unlimited" as a synonym for
                          "lifetime PT0S".
                        - Make "key-directory" optional.
                        - Warn if specifying a key length does not make
                          sense; fail if key length is out of range for
                          the algorithm.
                        - Allow use of mnemonics when specifying key
                          algorithm (e.g. "rsasha256", "ecdsa384", etc.).
                        - Make ISO 8601 durations case-insensitive.
                        [GL #1598]

5355.   [func]          What was set with --with-tuning=large option in
                        older BIND9 versions is now a default, and
                        a --with-tuning=small option was added for small
                        (e.g. OpenWRT) systems. [GL !2989]

5354.   [bug]           dnssec-policy created new KSK keys for zones in the
                        initial stage of signing (with the DS not yet in the
                        rumoured or omnipresent states).  Fix by checking the
                        key goals rather than the active state when determining
                        whether new keys are needed. [GL #1593]

5353.   [doc]           Document port and dscp parameters in forwarders
                        configuration option. [GL #914]

5352.   [bug]           Correctly handle catalog zone entries containing
                        characters that aren't legal in filenames. [GL #1592]
```

(continues on next page)

(continued from previous page)

```
5351.   [bug]           CDS / CDNSKEY consistency checks failed to handle
                        removal records. [GL #1554]

5350.   [bug]           When a view was configured with class CHAOS, the
                        server could crash while processing a query for a
                        non-existent record. [GL #1540]

5349.   [bug]           Fix a race in task_pause/unpause. [GL #1571]

5348.   [bug]           dnssec-settime -Psync was not being honoured.
                        Thanks to Tony Finch. [GL !2893]
```

```
        --- 9.15.8 released ---

5347.   [bug]           Fixed a bug that could cause an intermittent crash
                        in validator.c when validating a negative cache
                        entry. [GL #1561]

5346.   [bug]           Make hazard pointer array allocations dynamic, fixing
                        a bug that caused named to crash on machines with more
                        than 40 cores. [GL #1493]

5345.   [func]          Key-style trust anchors and DS-style trust anchors
                        can now both be used for the same name. [GL #1237]

5344.   [bug]           Handle accept() errors properly in netmgr. [GL !2880]

5343.   [func]          Add statistics counters to the netmgr. [GL #1311]

5342.   [bug]           Disable pktinfo for IPv6 and bind to each interface
                        explicitly instead, because libuv doesn't support
                        pktinfo control messages. [GL #1558]

5341.   [func]          Simplify passing the bound TCP socket to child
                        threads by using isc_uv_export/import functions.
                        [GL !2825]

5340.   [bug]           Don't deadlock when binding to a TCP socket fails.
                        [GL #1499]

5339.   [bug]           With some libmaxminddb versions, named could erroneously
                        match an IP address not belonging to any subnet defined
                        in a given GeoIP2 database to one of the existing
                        entries in that database. [GL #1552]

5338.   [bug]           Fix line spacing in `rndc secroots`.
                        Thanks to Tony Finch. [GL !2478]

5337.   [func]          'named -V' now reports maxminddb and protobuf-c
                        versions. [GL !2686]
```

```
        --- 9.15.7 released ---

5336.   [bug]           The TCP high-water statistic could report an
                        incorrect value on startup. [GL #1392]

5335.   [func]          Make TCP listening code multithreaded. [GL !2659]

5334.   [doc]           Update documentation with dnssec-policy clarifications.
                        Also change some defaults. [GL !2711]

5333.   [bug]           Fix duration printing on Solaris when value is not
                        an ISO 8601 duration. [GL #1460]

5332.   [func]          Renamed "dnssec-keys" configuration statement
                        to the more descriptive "trust-anchors". [GL !2702]

5331.   [func]          Use compiler-provided mechanisms for thread local
                        storage, and make the requirement for such mechanisms
                        explicit in configure. [GL #1444]

5330.   [bug]           'configure --without-python' was ineffective if
                        PYTHON was set in the environment. [GL #1434]

5329.   [bug]           Reconfiguring named caused memory to be leaked when any
                        GeoIP2 database was in use. [GL #1445]

5328.   [bug]           rbtdb.c:rdataset_{get,set}ownercase failed to obtain
                        a node lock. [GL #1417]

5327.   [func]          Added a statistics counter to track queries
                        dropped because the recursive-clients quota was
                        exceeded. [GL #1399]

5326.   [bug]           Add Python dependency on 'distutils.core' to configure.
                        'distutils.core' is required for installation.
                        [GL #1397]

5325.   [bug]           Addressed several issues with TCP connections in
                        the netmgr: restored support for TCP connection
                        timeouts, restored TCP backlog support, actively
                        close all open sockets during shutdown. [GL #1312]

5324.   [bug]           Change the category of some log messages from general
                        to the more appropriate catergory of xfer-in. [GL #1394]

5323.   [bug]           Fix a bug in DNSSEC trust anchor verification.
                        [GL !2609]

5322.   [placeholder]

5321.   [bug]           Obtain write lock before updating version->records
                        and version->bytes. [GL #1341]
```

(continues on next page)

(continued from previous page)

```
5320.   [cleanup]       Silence TSAN on header->count. [GL #1344]
```

```
        --- 9.15.6 released ---

5319.   [func]          Trust anchors can now be configured using DS
                        format to represent a key digest, by using the
                        new "initial-ds" or "static-ds" keywords in
                        the "dnssec-keys" statement.

                        Note: DNSKEY-format and DS-format trust anchors
                        cannot both be used for the same domain name.
                        [GL #622]

5318.   [cleanup]       The DNSSEC validation code has been refactored
                        for clarity and to reduce code duplication.
                        [GL #622]

5317.   [func]          A new asynchronous network communications system
                        based on libuv is now used for listening for
                        incoming requests and responding to them. (The
                        old isc_socket API remains in use for sending
                        iterative queries and processing responses; this
                        will be changed too in a later release.)

                        This change will make it easier to improve
                        performance and implement new protocol layers
                        (e.g., DNS over TLS) in the future. [GL #29]

5316.   [func]          A new "dnssec-policy" option has been added to
                        named.conf to implement a key and signing policy
                        (KASP) for zones. When this option is in use,
                        named can generate new keys as needed and
                        automatically roll both ZSK and KSK keys. (Note
                        that the syntax for this statement differs from
                        the dnssec policy used by dnssec-keymgr.)

                        See the ARM for configuration details. [GL #1134]

5315.   [bug]           Apply the initial RRSIG expiration spread fixed
                        to all dynamically created records in the zone
                        including NSEC3. Also fix the signature clusters
                        when the server has been offline for prolonged
                        period of times. [GL #1256]

5314.   [func]          Added a new statistics variable "tcp-highwater"
                        that reports the maximum number of simultaneous TCP
                        clients BIND has handled while running. [GL #1206]

5313.   [bug]           The default GeoIP2 database location did not match
                        the ARM.  'named -V' now reports the default
                        location. [GL #1301]
```

(continues on next page)

(continued from previous page)

```
5312.   [bug]           Do not flush the cache for `rndc validation status`.
                        Thanks to Tony Finch. [GL !2462]

5311.   [cleanup]       Include all views in output of `rndc validation status`.
                        Thanks to Tony Finch. [GL !2461]

5310.   [bug]           TCP failures were affecting EDNS statistics. [GL #1059]

5309.   [placeholder]

5308.   [bug]           Don't log DNS_R_UNCHANGED from sync_secure_journal()
                        at ERROR level in receive_secure_serial(). [GL #1288]

5307.   [bug]           Fix hang when named-compilezone output is sent to pipe.
                        Thanks to Tony Finch. [GL !2481]

5306.   [security]      Set a limit on number of simultaneous pipelined TCP
                        queries. (CVE-2019-6477) [GL #1264]

5305.   [bug]           NSEC Aggressive Cache ("synth-from-dnssec") has been
                        disabled by default because it was found to have
                        a significant performance impact on the recursive
                        service. [GL #1265]

5304.   [bug]           "dnskey-sig-validity 0;" was not being accepted.
                        [GL #876]

5303.   [placeholder]

5302.   [bug]           Fix checking that "dnstap-output" is defined when
                        "dnstap" is specified in a view. [GL #1281]

5301.   [bug]           Detect partial prefixes / incomplete IPv4 address in
                        acls. [GL #1143]

5300.   [bug]           dig/mdig/delv: Add a colon after EDNS option names,
                        even when the option is empty, to improve
                        readability and allow correct parsing of YAML
                        output. [GL #1226]
```

```
        --- 9.15.5 released ---

5299.   [security]      A flaw in DNSSEC verification when transferring
                        mirror zones could allow data to be incorrectly
                        marked valid. (CVE-2019-6475) [GL #1252]

5298.   [security]      Named could assert if a forwarder returned a
                        referral, rather than resolving the query, when QNAME
                        minimization was enabled. (CVE-2019-6476) [GL #1051]

5297.   [bug]           Check whether a previous QNAME minimization fetch
                        is still running before starting a new one; return
```

(continues on next page)

(continued from previous page)

```
                        SERVFAIL and log an error if so. [GL #1191]

5296.   [placeholder]

5295.   [cleanup]       Split dns_name_copy() calls into dns_name_copy() and
                        dns_name_copynf() for those calls that can potentially
                        fail and those that should not fail respectively.
                        [GL !2265]

5294.   [func]          Fallback to ACE name on output in locale, which does not
                        support converting it to unicode.  [GL #846]

5293.   [bug]           On Windows, named crashed upon any attempt to fetch XML
                        statistics from it. [GL #1245]

5292.   [bug]           Queue 'rndc nsec3param' requests while signing inline
                        zone changes. [GL #1205]
```

```
       --- 9.15.4 released ---

5291.   [placeholder]

5290.   [placeholder]

5289.   [bug]           Address NULL pointer dereference in rpz.c:rpz_detach.
                        [GL #1210]

5288.   [bug]           dnssec-must-be-secure was not always honored.
                        [GL #1209]

5287.   [placeholder]

5286.   [contrib]       Address potential NULL pointer dereferences in
                        dlz_mysqldyn_mod.c. [GL #1207]

5285.   [port]          win32: implement "-T maxudpXXX". [GL #837]

5284.   [func]          Added +unexpected command line option to dig.
                        By default, dig won't accept a reply from a source
                        other than the one to which it sent the query.
                        Invoking dig with +unexpected argument will allow it
                        to process replies from unexpected sources.

5283.   [bug]           When a response-policy zone expires, ensure that
                        its policies are removed from the RPZ summary
                        database. [GL #1146]

5282.   [bug]           Fixed a bug in searching for possible wildcard matches
                        for query names in the RPZ summary database. [GL #1146]

5281.   [cleanup]       Don't escape commas when reporting named's command
                        line. [GL #1189]
```

(continues on next page)

(continued from previous page)

```
5280.   [protocol]      Add support for displaying EDNS option LLQ. [GL #1201]

5279.   [bug]           When loading, reject zones containing CDS or CDNSKEY
                        RRsets at the zone apex if they would cause DNSSEC
                        validation failures if published in the parent zone
                        as the DS RRset.  [GL #1187]

5278.   [func]          Add YAML output formats for dig, mdig and delv;
                        use the "+yaml" option to enable. [GL #1145]
```

```
        --- 9.15.3 released ---

5277.   [bug]           Cache DB statistics could underflow when serve-stale
                        was in use, because of a bug in counter maintenance
                        when RRsets become stale.

                        Functions for dumping statistics have been updated
                        to dump active, stale, and ancient statistic
                        counters.  Ancient RRset counters are prefixed
                        with '~'; stale RRset counters are still prefixed
                        with '#'. [GL #602]

5276.   [func]          DNSSEC Lookaside Validation (DLV) is now obsolete;
                        all code enabling its use has been removed from the
                        validator, "delv", and the DNSSEC tools. [GL #7]

5275.   [bug]           Mark DS records included in referral messages
                        with trust level "pending" so that they can be
                        validated and cached immediately, with no need to
                        re-query. [GL #964]

5274.   [bug]           Address potential use after free race when shutting
                        down rpz. [GL #1175]

5273.   [bug]           Check that bits [64..71] of a dns64 prefix are zero.
                        [GL #1159]

5272.   [cleanup]       Remove isc-config.sh script as the BIND 9 libraries
                        are now purely internal. [GL #1123]

5271.   [func]          The normal (non-debugging) output of dnssec-signzone
                        and dnssec-verify tools now goes to stdout, instead of
                        the combination of stderr and stdout.

5270.   [bug]           'dig +expandaaaa +short' did not work. [GL #1152]

5269.   [port]          cygwin: can return ETIMEDOUT on connect() with a
                        non-blocking socket. [GL #1133]

5268.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
5267.   [func]          Allow statistics groups display to be toggle-able.
                        [GL #1030]

5266.   [bug]           named-checkconf failed to report dnstap-output
                        missing from named.conf when dnstap was specified.
                        [GL #1136]

5265.   [bug]           DNS64 and RPZ nodata (CNAME *.) rules interacted badly
                        [GL #1106]

5264.   [func]          New DNS Cookie algorithm - siphash24 - has been added
                        to BIND 9, and the old HMAC-SHA DNS Cookie algorithms
                        have been removed. [GL #605]
```

```
        --- 9.15.2 released ---

5263.   [cleanup]       Use atomics and isc_refcount_t wherever possible.
                        [GL #1038]

5262.   [func]          Removed support for the legacy GeoIP API. [GL #1112]

5261.   [cleanup]       Remove SO_BSDCOMPAT socket option usage.

5260.   [bug]           dnstap-read was producing malformed output for large
                        packets. [GL #1093]

5259.   [func]          New option '-i' for 'named-checkconf' to ignore
                        warnings about deprecated options. [GL #1101]

5258.   [func]          Added support for the GeoIP2 API from MaxMind. This
                        will be compiled in by default if the "libmaxminddb"
                        library is found at compile time, but can be
                        suppressed using "configure --disable-geoip".

                        Certain geoip ACL settings that were available with
                        legacy GeoIP are not available when using GeoIP2.
                        [GL #182]

5257.   [bug]           Some statistics data was not being displayed.
                        Add shading to the zone tables. [GL #1030]

5256.   [bug]           Ensure that glue records are included in root
                        priming responses if "minimal-responses" is not
                        set to "yes". [GL #1092]

5255.   [bug]           Errors encountered while reloading inline-signing
                        zones could be ignored, causing the zone content to
                        be left in an incompletely updated state rather than
                        reverted. [GL #1109]

5254.   [func]          Collect metrics to report to the statistics-channel
                        DNSSEC signing operations (dnssec-sign) and refresh
```

(continues on next page)

(continued from previous page)

```
                        operations (dnssec-refresh) per zone and per keytag.
                        [GL #513]

5253.   [port]          Support platforms that don't define ULLONG_MAX.
                        [GL #1098]

5252.   [func]          Report if the last 'rndc reload/reconfig' failed in
                        rndc status. [GL !2040]

5251.   [bug]           Statistics were broken in x86 Windows builds.
                        [GL #1081]

5250.   [func]          The default size for RSA keys is now 2048 bits,
                        for both ZSKs and KSKs. [GL #1097]

5249.   [bug]           Fix a possible underflow in recursion clients
                        statistics when hitting recursive clients
                        soft quota. [GL #1067]
```

```
        --- 9.15.1 released ---

5248.   [func]          To clarify the configuration of DNSSEC keys,
                        the "managed-keys" and "trusted-keys" options
                        have both been deprecated.  The new "dnssec-keys"
                        statement can now be used for all trust anchors,
                        with the keywords "iniital-key" or "static-key"
                        to indicate whether the configured trust anchor
                        should be used for initialization of RFC 5011 key
                        management, or as a permanent trust anchor.

                        The "static-key" keyword will generate a warning if
                        used for the root zone.

                        Configurations using "trusted-keys" or "managed-keys"
                        will continue to work with no changes, but will
                        generate warnings in the log. In a future release,
                        these options will be marked obsolete. [GL #6]

5247.   [cleanup]       The 'cleaning-interval' option has been removed.
                        [GL !1731]

5246.   [func]          Log TSIG if appropriate in 'sending notify to' message.
                        [GL #1058]

5245.   [cleanup]       Reduce logging level for IXFR up-to-date poll
                        responses. [GL #1009]

5244.   [security]      Fixed a race condition in dns_dispatch_getnext()
                        that could cause an assertion failure if a
                        significant number of incoming packets were
                        rejected. (CVE-2019-6471) [GL #942]
```

(continues on next page)

(continued from previous page)

```
5243.   [bug]           Fix a possible race between dispatcher and socket
                        code in a high-load cold-cache resolver scenario.
                        [GL #943]

5242.   [bug]           In relaxed qname minimization mode, fall back to
                        normal resolution when encountering a lame
                        delegation, and use _.domain/A queries rather
                        than domain/NS. [GL #1055]

5241.   [bug]           Fix Ed448 private and public key ASN.1 prefix blobs.
                        [GL #225]

5240.   [bug]           Remove key id calculation for RSAMD5. [GL #996]

5239.   [func]          Change the json-c detection to pkg-config. [GL #855]

5238.   [bug]           Fix a possible deadlock in TCP code. [GL #1046]

5237.   [bug]           Recurse to find the root server list with 'dig +trace'.
                        [GL #1028]

5236.   [func]          Add SipHash 2-4 implementation in lib/isc/siphash.c
                        and switch isc_hash_function() to use SipHash 2-4.
                        [GL #605]

5235.   [cleanup]       Refactor lib/isc/app.c to be thread-safe, unused
                        parts of the API has been removed and the
                        isc_appctx_t data type has been changed to be
                        fully opaque. [GL #1023]

5234.   [port]          arm: just use the compiler's default support for
                        yield. [GL #981]
```

```
        --- 9.15.0 released ---

5233.   [bug]           Negative trust anchors did not work with "forward only;"
                        to validating resolvers. [GL #997]

5232.   [placeholder]

5231.   [protocol]      Add support for displaying CLIENT-TAG and SERVER-TAG.
                        [GL #960]

5230.   [protocol]      The SHA-1 hash algorithm is no longer used when
                        generating DS and CDS records. [GL #1015]

5229.   [protocol]      Enforce known SSHFP fingerprint lengths. [GL #852]

5228.   [func]          If trusted-keys and managed-keys were configured
                        simultaneously for the same name, the key could
                        not be be rolled automatically. This is now
                        a fatal configuration error. [GL #868]
```

(continues on next page)

(continued from previous page)

```
5227.   [placeholder]

5226.   [placeholder]

5225.   [func]          Allow dig to print out AAAA record fully expanded.
                        with +[no]expandaaaa. [GL #765]

5224.   [bug]           Only test provide-ixfr on TCP streams. [GL #991]

5223.   [bug]           Fixed a race in the filter-aaaa plugin accessing
                        the hash table. [GL #1005]

5222.   [bug]           'delv -t ANY' could leak memory. [GL #983]

5221.   [test]          Enable parallel execution of system tests on
                        Windows. [GL !4101]

5220.   [cleanup]       Refactor the isc_stat structure to take advantage
                        of stdatomic. [GL !1493]

5219.   [bug]           Fixed a race in the filter-aaaa plugin that could
                        trigger a crash when returning an instance object
                        to the memory pool. [GL #982]

5218.   [bug]           Conditionally include <dlfcn.h>. [GL #995]

5217.   [bug]           Restore key id calculation for RSAMD5. [GL #996]

5216.   [bug]           Fetches-per-zone counter wasn't updated correctly
                        when doing qname minimization. [GL #992]

5215.   [bug]           Change #5124 was incomplete; named could still
                        return FORMERR instead of SERVFAIL in some cases.
                        [GL #990]

5214.   [bug]           win32: named now removes its lock file upon shutdown.
                        [GL #979]

5213.   [bug]           win32: Eliminated a race which allowed named.exe running
                        as a service to be killed prematurely during shutdown.
                        [GL #978]

5212.   [placeholder]

5211.   [bug]           Allow out-of-zone additional data to be included
                        in authoritative responses if recursion is allowed
                        and "minimal-responses" is disabled.  This behavior
                        was inadvertently removed in change #4605. [GL #817]

5210.   [bug]           When dnstap is enabled and recursion is not
                        available, incoming queries are now logged
```

(continues on next page)

(continued from previous page)

```
                        as "auth". Previously, this depended on whether
                        recursion was requested by the client, not on
                        whether recursion was available. [GL #963]

5209.   [bug]           When update-check-ksk is true, add_sigs was not
                        considering offline keys, leaving record sets signed
                        with the incorrect type key. [GL #763]

5208.   [test]          Run valid rdata wire encodings through totext+fromtext
                        and tofmttext+fromtext methods to check these methods.
                        [GL #899]

5207.   [test]          Check delv and dig TTL values. [GL #965]

5206.   [bug]           Delv could print out bad TTLs. [GL #965]

5205.   [bug]           Enforce that a DS hash exists. [GL #899]

5204.   [test]          Check that dns_rdata_fromtext() produces a record that
                        will be accepted by dns_rdata_fromwire(). [GL #852]

5203.   [bug]           Enforce whether key rdata exists or not in KEY,
                        DNSKEY, CDNSKEY and RKEY. [GL #899]

5202.   [bug]           <dns/ecs.h> was missing ISC_LANG_ENDDECLS. [GL #976]

5201.   [bug]           Fix a possible deadlock in RPZ update code. [GL #973]

5200.   [security]      tcp-clients settings could be exceeded in some cases,
                        which could lead to exhaustion of file descriptors.
                        (CVE-2018-5743) [GL #615]

5199.   [security]      In certain configurations, named could crash
                        if nxdomain-redirect was in use and a redirected
                        query resulted in an NXDOMAIN from the cache.
                        (CVE-2019-6467) [GL #880]

5198.   [bug]           If a fetch context was being shut down and, at the same
                        time, we returned from qname minimization, an INSIST
                        could be hit. [GL #966]

5197.   [bug]           dig could die in best effort mode on multiple SIG(0)
                        records. Similarly on multiple OPT and multiple TSIG
                        records. [GL #920]

5196.   [bug]           make install failed with --with-dlopen=no. [GL #955]

5195.   [bug]           "allow-update" and "allow-update-forwarding" were
                        treated as configuration errors if used at the
                        options or view level. [GL #913]

5194.   [bug]           Enforce non empty ZOMEMD hash. [GL #899]
```

(continues on next page)

(continued from previous page)

```
5193.   [bug]           EID and NIMLOC failed to do multi-line output
                        correctly. [GL #899]

5192.   [placeholder]

5191.   [placeholder]

5190.   [bug]           Ignore trust anchors using disabled algorithms.
                        [GL #806]

5189.   [cleanup]       Remove revoked root DNSKEY from bind.keys. [GL #945]

5188.   [func]          The "dnssec-enable" option is deprecated and no
                        longer has any effect; DNSSEC responses are
                        always enabled. [GL #866]

5187.   [test]          Set time zone before running any tests in dnstap_test.
                        [GL #940]

5186.   [cleanup]       More dnssec-keygen manual tidying. [GL !1678]

5185.   [placeholder]

5184.   [bug]           Missing unlocks in sdlz.c. [GL #936]

5183.   [bug]           Reinitialize ECS data before reusing client
                        structures. [GL #881]

5182.   [bug]           Fix a high-load race/crash in handling of
                        isc_socket_close() in resolver. [GL #834]

5181.   [func]          Add a mechanism for a DLZ module to signal that
                        the view's allow-transfer ACL should be used to
                        determine whether transfers are allowed. [GL #803]

5180.   [bug]           delv now honors the operating system's preferred
                        ephemeral port range. [GL #925]

5179.   [cleanup]       Replace some vague type declarations with the more
                        specific dns_secalg_t and dns_dsdigest_t.
                        Thanks to Tony Finch. [GL !1498]

5178.   [bug]           Handle EDQUOT (disk quota) and ENOSPC (disk full)
                        errors when writing files. [GL #902]

5177.   [func]          Add the ability to specify in named.conf whether a
                        response-policy zone's SOA record should be added
                        to the additional section (add-soa yes/no). [GL #865]

5176.   [tests]         Remove a dependency on libxml in statschannel system
                        test. [GL #926]
```

(continues on next page)

(continued from previous page)

```
5175.   [bug]           Fixed a problem with file input in dnssec-keymgr,
                        dnssec-coverage and dnssec-checkds when using
                        python3. [GL #882]

5174.   [doc]           Tidy dnssec-keygen manual. [GL !1557]

5173.   [bug]           Fixed a race in socket code that could occur when
                        accept, send, or recv were called from an event
                        loop but the socket had been closed by another
                        thread. [RT #874]

5172.   [bug]           nsupdate now honors the operating system's preferred
                        ephemeral port range. [GL #905]

5171.   [func]          named plugins are now installed into a separate
                        directory.  Supplying a filename (a string without path
                        separators) in a "plugin" configuration stanza now
                        causes named to look for that plugin in that directory.
                        [GL #878]

5170.   [test]          Added --with-dlz-filesystem to feature-test. [GL !1587]

5169.   [bug]           The presence of certain types in an otherwise
                        empty node could cause a crash while processing a
                        type ANY query. [GL #901]

5168.   [bug]           Do not crash on shutdown when RPZ fails to load.  Also,
                        keep previous version of the database if RPZ fails to
                        load. [GL #813]

5167.   [bug]           nxdomain-redirect could sometimes lookup the wrong
                        redirect name. [GL #892]

5166.   [placeholder]

5165.   [contrib]       Removed SDB drivers from contrib; they're obsolete.
                        [GL #428]

5164.   [bug]           Correct errno to result translation in dlz filesystem
                        modules. [GL #884]

5163.   [cleanup]       Out-of-tree builds failed --enable-dnstap. [GL #836]

5162.   [cleanup]       Improve dnssec-keymgr manual. Thanks to Tony Finch.
                        [GL !1518]

5161.   [bug]           Do not require the SEP bit to be set for mirror zone
                        trust anchors. [GL #873]

5160.   [contrib]       Added DNAME support to the DLZ LDAP schema. Also
                        fixed a compilation bug affecting several DLZ
```

(continues on next page)

(continued from previous page)

```
                        modules. [GL #872]

5159.   [bug]           dnssec-coverage was incorrectly ignoring
                        names specified on the command line without
                        trailing dots. [GL !1478]

5158.   [protocol]      Add support for AMTRELAY and ZONEMD. [GL #867]

5157.   [bug]           Nslookup now errors out if there are extra command
                        line arguments. [GL #207]

5156.   [doc]           Extended and refined the section of the ARM describing
                        mirror zones. [GL #774]

5155.   [func]          "named -V" now outputs the default paths to
                        named.conf, rndc.conf, bind.keys, and other
                        files used or created by named and other tools, so
                        that the correct paths to these files can quickly be
                        determined regardless of the configure settings
                        used when BIND was built. [GL #859]

5154.   [bug]           dig: process_opt could be called twice on the same
                        message leading to a assertion failure. [GL #860]

5153.   [func]          Zone transfer statistics (size, number of records, and
                        number of messages) are now logged for outgoing
                        transfers as well as incoming ones. [GL #513]

5152.   [func]          Improved logging of DNSSEC key events:
                        - Zone signing and DNSKEY maintenance events are
                          now logged to the "dnssec" category
                        - Messages are now logged when DNSSEC keys are
                          published, activated, inactivated, deleted,
                          or revoked.
                        [GL #714]

5151.   [func]          Options that have been been marked as obsolete in
                        named.conf for a very long time are now fatal
                        configuration errors. [GL #358]

5150.   [cleanup]       Remove the ability to compile BIND with assertions
                        disabled. [GL #735]

5149.   [func]          "rndc dumpdb" now prints a line above a stale RRset
                        indicating how long the data will be retained in the
                        cache for emergency use. [GL #101]

5148.   [bug]           named did not sign the TKEY response. [GL #821]

5147.   [bug]           dnssec-keymgr: Add a five-minute margin to better
                        handle key events close to 'now'. [GL #848]
```

(continues on next page)

(continued from previous page)

```
5146.   [placeholder]

5145.   [func]          Use atomics instead of locked variables for isc_quota
                       and isc_counter. [GL !1389]

5144.   [bug]           dig now returns a non-zero exit code when a TCP
                       connection is prematurely closed by a peer more than
                       once for the same lookup.  [GL #820]

5143.   [bug]           dnssec-keymgr and dnssec-coverage failed to find
                       key files for zone names ending in ".". [GL #560]

5142.   [cleanup]       Removed "configure --disable-rpz-nsip" and
                       "--disable-rpz-nsdname" options. "nsip-enable"
                       and "nsdname-enable" both now default to yes,
                       regardless of compile-time settings. [GL #824]

5141.   [security]      Zone transfer controls for writable DLZ zones were
                       not effective as the allowzonexfr method was not being
                       called for such zones. (CVE-2019-6465) [GL #790]

5140.   [bug]           Don't immediately mark existing keys as inactive and
                       deleted when running dnssec-keymgr for the first
                       time. [GL #117]

5139.   [bug]           If possible, don't use forwarders when priming.
                       This ensures we can get root server IP addresses
                       from priming query response glue, which may not
                       be present if the forwarding server is returning
                       minimal responses. [GL #752]

5138.   [bug]           Under some circumstances named could hit an assertion
                       failure when doing qname minimization when using
                       forwarders. [GL #797]

5137.   [func]          named now logs messages whenever a mirror zone becomes
                       usable or unusable for resolution purposes. [GL #818]

5136.   [cleanup]       Check in named-checkconf that allow-update and
                       allow-update-forwarding are not set at the
                       view/options level; fix documentation. [GL #512]

5135.   [port]          sparc: Use smt_pause() instead of pause. [GL #816]

5134.   [bug]           win32: WSAStartup was not called before getservbyname
                       was called. [GL #590]

5133.   [bug]           'rndc managed-keys' didn't handle class and view
                       correctly and failed to add new lines between each
                       view. [GL !1327]

5132.   [bug]           Fix race condition in cleanup part of dns_dt_create().
```

(continues on next page)

(continued from previous page)

```
                        [GL !1323]

5131.   [cleanup]       Address Coverity warnings. [GL #801]

5130.   [cleanup]       Remove support for l10n message catalogs. [GL #709]

5129.   [contrib]       sdlz_helper.c:build_querylist was not properly
                        splitting the query string. [GL #798]

5128.   [bug]           Refreshkeytime was not being updated for managed
                        keys zones. [GL #784]

5127.   [bug]           rcode.c:maybe_numeric failed to handle NUL in text
                        regions. [GL #807]

5126.   [bug]           Named incorrectly accepted empty base64 and hex encoded
                        fields when reading master files. [GL #807]

5125.   [bug]           Allow for up to 100 records or 64k of data when caching
                        a negative response. [GL #804]

5124.   [bug]           Named could incorrectly return FORMERR rather than
                        SERVFAIL. [GL #804]

5123.   [bug]           dig could hang indefinitely after encountering an error
                        before creating a TCP socket. [GL #692]

5122.   [bug]           In a "forward first;" configuration, a forwarder
                        timeout did not prevent that forwarder from being
                        queried again after falling back to full recursive
                        resolution. [GL #315]

5121.   [contrib]       dlz_stub_driver.c fails to return ISC_R_NOTFOUND on none
                        matching zone names. [GL !1299]

5120.   [placeholder]

5119.   [placeholder]

5118.   [security]      Named could crash if it is managing a key with
                        `managed-keys` and the authoritative zone is rolling
                        the key to an unsupported algorithm. (CVE-2018-5745)
                        [GL #780]

5117.   [placeholder]

5116.   [bug]           Named/named-checkconf triggered a assertion when
                        a mirror zone's name is bad. [GL #778]

5115.   [bug]           Allow unsupported algorithms in zone when not used for
                        signing with dnssec-signzone. [GL #783]
```

(continues on next page)

(continued from previous page)

```
5114.   [func]          Include a 'reconfig/reload in progress' status line
                        in rndc status, use it in tests.

5113.   [port]          Fixed a Windows build error.

5112.   [bug]           Named/named-checkconf could dump core if there was
                        a missing masters clause and a bad notify clause.
                        [GL #779]

5111.   [bug]           Occluded DNSKEY records could make it into the
                        delegating NSEC/NSEC3 bitmap. [GL #742]

5110.   [security]      Named leaked memory if there were multiple Key Tag
                        EDNS options present. (CVE-2018-5744) [GL #772]

5109.   [cleanup]       Remove support for RSAMD5 algorithm. [GL #628]
```

```
        --- 9.13.5 released ---

5108.   [bug]           Named could fail to determine bottom of zone when
                        removing out of date keys leading to invalid NSEC
                        and NSEC3 records being added to the zone. [GL #771]

5107.   [bug]           'host -U' did not work. [GL #769]

5106.   [experimental]  A new "plugin" mechanism has been added to allow
                        extension of query processing functionality through
                        the use of dynamically loadable libraries. A
                        "filter-aaaa.so" plugin has been implemented,
                        replacing the filter-aaaa feature that was formerly
                        implemented as a native part of BIND.

                        The "filter-aaaa", "filter-aaaa-on-v4" and
                        "filter-aaaa-on-v6" options can no longer be
                        configured using native named.conf syntax. However,
                        loading the filter-aaaa.so plugin and setting its
                        parameters provides identical functionality.

                        Note that the plugin API is a work in progress and
                        is likely to evolve as further plugins are
                        implemented. [GL #15]

5105.   [bug]           Fix a race between process_fd and socketclose in
                        unix socket code. [GL #744]

5104.   [cleanup]       Log clearer informational message when a catz zone
                        is overridden by a zone in named.conf.
                        Thanks to Tony Finch. [GL !1157]

5103.   [bug]           Add missing design by contract tests to dns_catz*.
                        [GL #748]
```

(continues on next page)

(continued from previous page)

```
5102.   [bug]           dnssec-coverage failed to use the default TTL when
                        checking KSK deletion times leading to a exception.
                        [GL #585]

5101.   [bug]           Fix default installation path for Python modules and
                        remove the dnspython dependency accidentally introduced
                        by change 4970. [GL #730]

5100.   [func]          Pin resolver tasks to specific task queues. [GL !1117]

5099.   [func]          Failed mutex and conditional creations are always
                        fatal. [GL #674]
```

```
        --- 9.13.4 released ---

5098.   [func]          Failed memory allocations are now fatal. [GL #674]

5097.   [cleanup]       Remove embedded ATF unit testing framework
                        from BIND source distribution.  [GL !875]

5096.   [func]          Use multiple event loops in socket code, and
                        make network threads CPU-affinitive.  This
                        significantly improves performance on large
                        systems. [GL #666]

5095.   [test]          Converted all unit tests from ATF to CMocka;
                        removed the source code for the ATF libraries.
                        Build with "configure --with-cmocka" to enable
                        unit testing. [GL #620]

5094.   [func]          Add 'dig -r' to disable reading of .digrc. [GL !970]

5093.   [bug]           Log lame qname-minimization servers only if they're
                        really lame. [GL #671]

5092.   [bug]           Address memory leak on SIGTERM in nsupdate when using
                        GSS-TSIG. [GL #558]

5091.   [func]          Two new global and per-view options min-cache-ttl
                        and min-ncache-ttl [GL #613]

5090.   [bug]           dig and mdig failed to properly pre-parse dash value
                        pairs when value was a separate argument and started
                        with a dash. [GL #584]

5089.   [bug]           Restore localhost fallback in dig and host which is
                        used when no nameserver addresses present in
                        /etc/resolv.conf are usable due to the requested
                        address family restrictions. [GL #433]

5088.   [bug]           dig/host/nslookup could crash when interrupted close to
                        a query timeout. [GL #599]
```

(continues on next page)

(continued from previous page)

```
5087.   [test]          Check that result tables are complete. [GL #676]

5086.   [func]          Log of RPZ now includes the QTYPE and QCLASS. [GL #623]

5085.   [bug]           win32: Restore looking up nameservers, search list,
                        etc. [GL #186]

5084.   [placeholder]

5083.   [func]          Add autoconf macro AX_POSIX_SHELL, so we
                        can use POSIX-compatible shell features
                        in the scripts.

5082.   [bug]           Fixed a race that could cause a crash in
                        dig/host/nslookup. [GL #650]

5081.   [func]          Use per-worker queues in task manager, make task
                        runners CPU-affine. [GL #659]

5080.   [func]          Improvements to "rndc nta" user interface:
                        - catch and report invalid command line options
                        - when removing an NTA from all views, do not
                          abort with an error if the NTA was not found
                          in one of the views
                        - include the view name in "rndc nta -dump"
                          output, for consistency with the add and remove
                          actions
                        Thanks to Tony Finch. [GL !816]

5079.   [func]          Disable IDN processing in dig and nslookup
                        when not on a tty. [GL #653]

5078.   [cleanup]       Require python components to be explicitly disabled if
                        python is not available on unix platforms. [GL #601]

5077.   [cleanup]       Remove ip6.int support (-i) from dig and mdig.
                        [GL !969]

5076.   [bug]           "require-server-cookie" was not effective if
                        "rate-limit" was configured. [GL #617]

5075.   [bug]           Refresh nameservers from cache when sending final
                        query in qname minimization. [GL #16]

5074.   [cleanup]       Remove vector socket functions - isc_socket_recvv(),
                        isc_socket_sendtov(), isc_socket_sendtov2(),
                        isc_socket_sendv() - in order to simplify socket code.
                        [GL #645]

5073.   [bug]           Destroy a task first when destroying rpzs and catzs.
                        [GL #84]
```

(continues on next page)

(continued from previous page)

```
5072.   [bug]           Add unit tests for isc_buffer_copyregion() and fix its
                        behavior for auto-reallocated buffers. [GL #644]

5071.   [bug]           Comparison of NXT records was broken. [GL #631]

5070.   [bug]           Record types which support a empty rdata field were
                        not handling the empty rdata field case. [GL #638]

5069.   [bug]           Fix a hang on in RPZ when named is shutdown during RPZ
                        zone update. [GL !907]

5068.   [bug]           Fix a race in RPZ with min-update-interval set to 0.
                        [GL #643]

5067.   [bug]           Don't minimize qname when sending the query
                        to a forwarder. [GL #361]

5066.   [cleanup]       Allow unquoted strings to be used as a zone names
                        in response-policy statements. [GL #641]

5065.   [bug]           Only set IPV6_USE_MIN_MTU on IPv6. [GL #553]

5064.   [test]          Initialize TZ environment variable before calling
                        dns_test_begin in dnstap_test. [GL #624]

5063.   [test]          In statschannel test try a few times before failing
                        when checking if the compressed output is the same as
                        uncompressed. [GL !909]

5062.   [func]          Use non-crypto-secure PRNG to generate nonces for
                        cookies. [GL !887]

5061.   [protocol]      Add support for EID and NIMLOC. [GL #626]

5060.   [bug]           GID, UID and UINFO could not be loaded using unknown
                        record format. [GL #627]

5059.   [bug]           Display a per-view list of zones in the web interface.
                        [GL #427]

5058.   [func]          Replace old message digest and hmac APIs with more
                        generic isc_md and isc_hmac APIs, and convert their
                        respective tests to cmocka. [GL #305]

5057.   [protocol]      Add support for ATMA. [GL #619]

5056.   [placeholder]

5055.   [func]          A default list of primary servers for the root zone is
                        now built into named, allowing the "masters" statement
                        to be omitted when configuring an IANA root zone
```

(continues on next page)

(continued from previous page)

```
                        mirror. [GL #564]

5054.   [func]          Attempts to use mirror zones with recursion disabled
                        are now considered a configuration error. [GL #564]

5053.   [func]          The only valid zone-level NOTIFY settings for mirror
                        zones are now "notify no;" and "notify explicit;".
                        [GL #564]

5052.   [func]          Mirror zones are now configured using "type mirror;"
                        rather than "mirror yes;". [GL #564]

5051.   [doc]           Documentation incorrectly stated that the
                        "server-addresses" static-stub zone option accepts
                        custom port numbers. [GL #582]

5050.   [bug]           The libirs version of getaddrinfo() was unable to parse
                        scoped IPv6 addresses present in /etc/resolv.conf.
                        [GL #187]

5049.   [cleanup]       QNAME minimization has been deeply refactored. [GL #16]

5048.   [func]          Add configure option to enable and enforce FIPS mode
                        in BIND 9. [GL #506]

5047.   [bug]           Messages logged for certain query processing failures
                        now include a more specific error description if it is
                        available. [GL #572]

5046.   [bug]           named could crash during shutdown if an RPZ
                        reload was in progress. [RT #46210]

5045.   [func]          Remove support for DNSSEC algorithms 3 (DSA)
                        and 6 (DSA-NSEC3-SHA1). [GL #22]

5044.   [cleanup]       If "dnssec-enable" is no, then "dnssec-validation"
                        now also defaults to no.  [GL #388]

5043.   [bug]           Fix creating and validating EdDSA signatures. [GL #579]

5042.   [test]          Make the chained delegations in reclimit behave
                        like they would in a regular name server. [GL #578]

5041.   [test]          The chain test contains a incomplete delegation.
                        [GL #568]

5040.   [func]          Extended dnstap so that it can log UPDATE requests
                        and responses as separate message types. Thanks
                        to Greg Rabil. [GL #570]

5039.   [bug]           Named could fail to preserve owner name case of new
                        RRset. [GL #420]
```

(continues on next page)

(continued from previous page)

```
5038.   [bug]           Chaosnet addresses were compared incorrectly.
                        [GL #562]

5037.   [func]          "allow-recursion-on" and "allow-query-cache-on"
                        each now default to the other if only one of them
                        is set, in order to be more consistent with the way
                        "allow-recursion" and "allow-query-cache" work.
                        Also we now ensure that both query-cache ACLs are
                        checked when determining cache access. [GL #319]

5036.   [cleanup]       Fixed a spacing/formatting error in some RPZ-related
                        error messages in the log. [GL !805]

5035.   [test]          Fixed errors that prevented the DNSRPS subtests
                        from running in the rpz and rpzrecurse system
                        tests. [GL #503]

5034.   [bug]           A race between threads could prevent zone maintenance
                        scheduled immediately after zone load from being
                        performed. [GL #542]

5033.   [bug]           When adding NTAs to multiple views using "rndc nta",
                        the text returned via rndc was incorrectly terminated
                        after the first line, making it look as if only one
                        NTA had been added. Also, it was not possible to
                        differentiate between views with the same name but
                        different classes; this has been corrected with the
                        addition of a "-class" option. [GL #105]

5032.   [func]          Add krb5-selfsub and ms-selfsub update policy rules.
                        [GL #511]

5031.   [cleanup]       Various defines in platform.h has been either dropped
                        if always or never triggered on supported platforms
                        or replaced with config.h equivalents if the defines
                        didn't have any impact on public headers.  Workarounds
                        for LinuxThreads have been removed because NPTL is
                        available since Linux kernel 2.6.0.  [GL #525]

5030.   [bug]           Align CMSG buffers to a 64-bit boundary, fixes crash
                        on architectures with strict alignment. [GL #521]
```

```
        --- 9.13.3 released ---

5029.   [func]          Workarounds for servers that misbehave when queried
                        with EDNS have been removed, because these broken
                        servers and the workarounds for their noncompliance
                        cause unnecessary delays, increase code complexity,
                        and prevent deployment of new DNS features. See
                        https://dnsflagday.net for further details. [GL #150]
```

(continues on next page)

(continued from previous page)

```
5028.   [bug]           Spread the initial RRSIG expiration times over the
                        entire working sig-validity-interval when signing a
                        zone in named to even out re-signing and transfer
                        loads. [GL #418]

5027.   [func]          Set SO_SNDBUF size on sockets. [GL #74]

5026.   [bug]           rndc reconfig should not touch already loaded zones.
                        [GL #276]

5025.   [cleanup]       Remove isc_keyboard family of functions. [GL #178]

5024.   [func]          Replace custom assembly for atomic operations with
                        atomic support from the compiler. The code will now use
                        C11 stdatomic, or __atomic, or __sync builtins with GCC
                        or Clang compilers, and Interlocked functions with MSVC.
                        [GL #10]

5023.   [cleanup]       Remove wrappers that try to fix broken or incomplete
                        implementations of IPv6, pthreads and other core
                        functionality required and used by BIND. [GL #192]

5022.   [doc]           Update ms-self, ms-subdomain, krb5-self, and
                        krb5-subdomain documentation. [GL !708]

5021.   [bug]           dig returned a non-zero exit code when it received a
                        reply over TCP after a retry. [GL #487]

5020.   [func]          RNG uses thread-local storage instead of locks, if
                        supported by platform. [GL #496]

5019.   [cleanup]       A message is now logged when ixfr-from-differences is
                        set at zone level for an inline-signed zone. [GL #470]

5018.   [bug]           Fix incorrect sizeof arguments in lib/isc/pk11.c.
                        [GL !588]

5017.   [bug]           lib/isc/pk11.c failed to unlink the session before
                        releasing the lock which is unsafe. [GL !589]

5016.   [bug]           Named could assert with overlapping filter-aaaa and
                        dns64 acls. [GL #445]

5015.   [bug]           Reloading all zones caused zone maintenance to cease
                        for inline-signed zones. [GL #435]

5014.   [bug]           Signatures loaded from the journal for the signed
                        version of an inline-signed zone were not scheduled for
                        refresh. [GL #482]

5013.   [bug]           A referral response with a non-empty ANSWER section was
                        inadvertently being treated as an error. [GL #390]
```

(continues on next page)

(continued from previous page)

```
5012.   [bug]           Fix lock order reversal in pk11_initialize. [GL !590]

5011.   [func]          Remove support for unthreaded named. [GL #478]

5010.   [func]          New "validate-except" option specifies a list of
                        domains beneath which DNSSEC validation should not
                        be performed. [GL #237]

5009.   [bug]           Upon an OpenSSL failure, the first error in the OpenSSL
                        error queue was not logged. [GL #476]

5008.   [bug]           "rndc signing -nsec3param ..." requests were silently
                        ignored for zones which were not yet loaded or
                        transferred. [GL #468]

5007.   [cleanup]       Replace custom ISC boolean and integer data types
                        with C99 stdint.h and stdbool.h types. [GL #9]

5006.   [cleanup]       Code preparing a delegation response was extracted from
                        query_delegation() and query_zone_delegation() into a
                        separate function in order to decrease code
                        duplication. [GL #431]

5005.   [bug]           dnssec-verify, and dnssec-signzone at the verification
                        step, failed on some validly signed zones. [GL #442]

5004.   [bug]           'rndc reconfig' could cause inline zones to stop
                        re-signing. [GL #439]

5003.   [bug]           dns_acl_isinsecure did not handle geoip elements.
                        [GL #406]

5002.   [bug]           mdig: Handle malformed +ednsopt option, support 100
                        +ednsopt options per query rather than 100 total and
                        address memory leaks if +ednsopt was specified.
                        [GL #410]

5001.   [bug]           Fix refcount errors on error paths. [GL !563]

5000.   [bug]           named_server_servestale() could leave the server in
                        exclusive mode if an error occurred. [GL #441]

4999.   [cleanup]       Remove custom printf implementation in lib/isc/print.c.
                        [GL #261]

4998.   [test]          Make resolver and cacheclean tests more civilized.

4997.   [security]      named could crash during recursive processing
                        of DNAME records when "deny-answer-aliases" was
                        in use. (CVE-2018-5740) [GL #387]
```

(continues on next page)

(continued from previous page)

```
4996.   [bug]           dig: Handle malformed +ednsopt option. [GL #403]

4995.   [test]          Add tests for "tcp-self" update policy. [GL !282]

4994.   [bug]           Trust anchor telemetry queries were not being sent
                        upstream for locally served zones. [GL #392]

4993.   [cleanup]       Remove support for silently ignoring 'no-change' deltas
                        from BIND 8 when processing an IXFR stream. 'no-change'
                        deltas will now trigger a fallback to AXFR as the
                        recovery mechanism. [GL #369]

4992.   [bug]           The wrong address was being logged for trust anchor
                        telemetry queries. [GL #379]

4991.   [bug]           "rndc reconfig" was incorrectly handling zones whose
                        "mirror" setting was changed. [GL #381]

4990.   [bug]           Prevent a possible NULL reference in pkcs11-keygen.
                        [GL #401]

4989.   [cleanup]       IDN support in dig has been reworked.  IDNA2003
                        fallbacks were removed in the process. [GL #384]

4988.   [bug]           Don't synthesize NXDOMAIN from NSEC for records under
                        a DNAME.
```

```
        --- 9.13.2 released ---

4987.   [cleanup]       dns_rdataslab_tordataset() and its related
                        dns_rdatasetmethods_t callbacks were removed as they
                        were not being used by anything in BIND. [GL #371]

4986.   [func]          When built on Linux, BIND now requires the libcap
                        library to set process privileges, unless capability
                        support is explicitly overridden with "configure
                        --disable-linux-caps". [GL #321]

4985.   [func]          Add a new slave zone option, "mirror", to enable
                        serving a non-authoritative copy of a zone that
                        is subject to DNSSEC validation before being
                        used.  For now, this option is only meant to
                        facilitate deployment of an RFC 7706-style local
                        copy of the root zone. [GL #33]

4984.   [bug]           Improve handling of very large incremental
                        zone transfers to prevent journal corruption. [GL #339]

4983.   [func]          Add the ability to not return a DNS COOKIE option
                        when one is present in the request (answer-cookie no;).
                        [GL #173]
```

(continues on next page)

(continued from previous page)

```
4982.   [cleanup]       Return FORMERR if the question section is empty
                        and no COOKIE option is present; this restores
                        older behavior except in the newly specified
                        COOKIE case. [GL #260]

4981.   [bug]           Fix race in cmsg buffer usage in socket code.
                        [GL #180]

4980.   [bug]           Named-checkconf failed to detect bad in-view targets.
                        [GL #288]

4979.   [placeholder]

4978.   [test]          Fix error handling and resolver configuration in the
                        "rpz" system test. [GL #312]

4977.   [func]          When starting up, log the same details that
                        would be reported by 'named -V'. [GL #247]

4976.   [bug]           Log the label with invalid prefix length correctly
                        when loading RPZ zones. [GL #254]

4975.   [bug]           The server cookie computation for sha1 and sha256 did
                        not match the method described in RFC 7873. [GL #356]

4974.   [bug]           Restore default rrset-order to random. [GL #336]

4973.   [func]          verifyzone() and the functions it uses were moved to
                        libdns and refactored to prevent exit() from being
                        called upon failure.  A side effect of that is that
                        dnssec-signzone and dnssec-verify now check for memory
                        leaks upon shutdown. [GL #266]

4972.   [func]          Declare the 'rdata' argument for dns_rdata_tostruct()
                        to be const. [GL #341]

4971.   [bug]           dnssec-signzone and dnssec-verify did not treat records
                        below a DNAME as out-of-zone data. [GL #298]

4970.   [func]          Add QNAME minimization option to resolver. [GL #16]

4969.   [cleanup]       Refactor zone logging functions. [GL #269]
```

```
        --- 9.13.1 released ---

4968.   [bug]           If glue records are signed, attempt to validate them.
                        [GL #209]

4967.   [cleanup]       Add "answer-cookie" to the parser, marked obsolete.

4966.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
4965.   [func]          Add support for marking options as deprecated.
                        [GL #322]

4964.   [bug]           Reduce the probability of double signature when deleting
                        a DNSKEY by checking if the node is otherwise signed
                        by the algorithm of the key to be deleted. [GL #240]

4963.   [test]          ifconfig.sh now uses "ip" instead of "ifconfig",
                        if available, to configure the test interfaces on
                        linux.  [GL #302]

4962.   [cleanup]       Move 'named -T' processing to its own function.
                        [GL #316]

4961.   [protocol]      Remove support for ECC-GOST (GOST R 34.11-94).
                        [GL #295]

4960.   [security]      When recursion is enabled, but the "allow-recursion"
                        and "allow-query-cache" ACLs are not specified,
                        they should be limited to local networks,
                        but were inadvertently set to match the default
                        "allow-query", thus allowing remote queries.
                        (CVE-2018-5738) [GL #309]

4959.   [func]          NSID logging (enabled by the "request-nsid" option)
                        now has its own "nsid" category, instead of using the
                        "resolver" category. [GL !332]

4958.   [bug]           Remove redundant space from NSEC3 record. [GL #281]

4957.   [func]          The default setting for "dnssec-validation" is now
                        "auto", which activates DNSSEC validation using the
                        IANA root key. (The default can be changed back to
                        "yes", which activates DNSSEC validation only when keys
                        are explicitly configured in named.conf, by building
                        BIND with "configure --disable-auto-validation".)
                        [GL #30]

4956.   [func]          Change isc_random() to be just PRNG using xoshiro128**,
                        and add isc_nonce_buf() that uses CSPRNG. [GL #289]

4955.   [cleanup]       Silence cppcheck warnings in lib/dns/master.c.
                        [GL #286]

4954.   [func]          Messages about serving of stale answers are now
                        directed to the "serve-stale" logging category.
                        Also clarified serve-stale documentation. [GL !323]

4953.   [bug]           Removed the option to build the red black tree
                        database without a hash table; the non-hashing
                        version was buggy and is not needed. [GL #184]
```

(continues on next page)

(continued from previous page)

```
4952.   [func]          Authoritative server support in named for the
                        EDNS CLIENT-SUBNET option (which was experimental
                        and not practical to deploy) has been removed.

                        The ECS option is still supported in dig and mdig
                        via the +subnet option, and can be parsed and logged
                        when received by named, but it is no longer used
                        for ACL processing. The "geoip-use-ecs" option
                        is now obsolete; a warning will be logged if it is
                        used in named.conf. "ecs" tags in an ACL definition
                        are also obsolete and will cause the configuration
                        to fail to load.  [GL #32]

4951.   [protocol]      Add "HOME.ARPA" to list of built in empty zones as
                        per RFC 8375. [GL #273]
```

```
        --- 9.13.0 released ---

4950.   [bug]           ISC_SOCKEVENTATTR_TRUNC was not be set. [GL #238]

4949.   [placeholder]

4948.   [bug]           When request-nsid is turned on, EDNS NSID options
                        should be logged at level info. Since change 3741
                        they have been logged at debug(3) by mistake.
                        [GL !290]

4947.   [func]          Replace all random functions with isc_random(),
                        isc_random_buf() and isc_random_uniform() API.
                        [GL #221]

4946.   [bug]           Additional glue was not being returned by resolver
                        for unsigned zones since change 4596. [GL #209]

4945.   [func]          BIND can no longer be built without DNSSEC support.
                        A cryptography provider (i.e., OpenSSL or a hardware
                        service module with PKCS#11 support) must be
                        available. [GL #244]

4944.   [cleanup]       Silence cppcheck portability warnings in
                        lib/isc/tests/buffer_test.c. [GL #239]

4943.   [bug]           Change 4687 consumed too much memory when running
                        system tests with --with-tuning=large.  Reduced the
                        hash table size to 512 entries for 'named -m record'
                        restoring the previous memory footprint. [GL #248]

4942.   [cleanup]       Consolidate multiple instances of splitting of
                        batchline in dig into a single function. [GL #196]

4941.   [cleanup]       Silence clang static analyzer warnings. [GL #196]
```

(continues on next page)

(continued from previous page)

```
4940.   [cleanup]       Extract the loop in dns__zone_updatesigs() into
                        separate functions to improve code readability.
                        [GL #135]

4939.   [test]          Add basic unit tests for update_sigs(). [GL #135]

4938.   [placeholder]

4937.   [func]          Remove support for OpenSSL < 1.0.0 [GL #191]

4936.   [func]          Always use OpenSSL or PKCS#11 random data providers,
                        and remove the --{enable,disable}-crypto-rand configure
                        options. [GL #165]

4935.   [func]          Add support for LibreSSL >= 2.7.0 (some OpenSSL 1.1.0
                        call were added). [GL #191]

4934.   [security]      The serve-stale feature could cause an assertion failure
                        in rbtdb.c even when stale-answer-enable was false.
                        Simultaneous use of stale cache records and NSEC
                        aggressive negative caching could trigger a recursion
                        loop. (CVE-2018-5737) [GL #185]

4933.   [bug]           Not creating signing keys for an inline signed zone
                        prevented changes applied to the raw zone from being
                        reflected in the secure zone until signing keys were
                        made available. [GL #159]

4932.   [bug]           Bumped signed serial of an inline signed zone was
                        logged even when an error occurred while updating
                        signatures. [GL #159]

4931.   [func]          Removed the "rbtdb64" database implementation.
                        [GL #217]

4930.   [bug]           Remove a bogus check in nslookup command line
                        argument processing. [GL #206]

4929.   [func]          Add the ability to set RA and TC in queries made by
                        dig (+[no]raflag, +[no]tcflag). [GL #213]

4928.   [func]          The "dnskey-sig-validity" option allows
                        "sig-validity-interval" to be overridden for signatures
                        covering DNSKEY RRsets. [GL #145]

4927.   [placeholder]

4926.   [func]          Add root key sentinel support.  To disable, add
                        'root-key-sentinel no;' to named.conf. [GL #37]

4925.   [func]          Several configuration options that define intervals
                        can now take TTL value suffixes (for example, 2h or 1d)
```

(continues on next page)

(continued from previous page)

```
                        in addition to integer parameters. These include
                        max-cache-ttl, max-ncache-ttl, max-policy-ttl,
                        fstrm-set-reopen-interval, interface-interval, and
                        min-update-interval. [GL #203]

4924.   [cleanup]       Clean up the isc_string_* namespace and leave
                        only strlcpy and strlcat. [GL #178]

4923.   [cleanup]       Refactor socket and socket event options into
                        enum types. [GL !135]

4922.   [bug]           dnstap: Log the destination address of client
                        packets rather than the interface address.
                        [GL #197]

4921.   [cleanup]       Add dns_fixedname_initname() and refactor the caller
                        code to make usage of the new function, as a part of
                        refactoring dns_fixedname_*() macros were turned into
                        functions. [GL #183]

4920.   [cleanup]       Clean up libdns removing most of the backwards
                        compatibility wrappers.

4919.   [cleanup]       Clean up the isc_hash_* namespace and leave only
                        the FNV-1a hash implementation. [GL #178]

4918.   [bug]           Fix double free after keygen error in dnssec-keygen
                        when OpenSSL >= 1.1.0 is used and RSA_generate_key_ex
                        fails. [GL #109]

4917.   [func]          Support 64 RPZ policy zones by default. [GL #123]

4916.   [func]          Remove IDNA2003 support and the bundled idnkit-1.0
                        library.

4915.   [func]          Implement IDNA2008 support in dig by adding support
                        for libidn2.  New dig option +idnin has been added,
                        which allows to process invalid domain names much
                        like dig without IDN support.  libidn2 version 2.0
                        or higher is needed for +idnout enabled by default.

4914.   [security]      A bug in zone database reference counting could lead to
                        a crash when multiple versions of a slave zone were
                        transferred from a master in close succession.
                        (CVE-2018-5736) [GL #134]

4913.   [test]          Re-implemented older unit tests in bin/tests as ATF,
                        removed the lib/tests unit testing library. [GL #115]

4912.   [test]          Improved the reliability of the 'cds' system test.
                        [GL #136]
```

(continues on next page)

(continued from previous page)

```
4911.   [test]          Improved the reliability of the 'mkeys' system test.
                        [GL #128]

4910.   [func]          Update util/check-changes to work on release branches.
                        [GL #113]

4909.   [bug]           named-checkconf did not detect in-view zone collisions.
                        [GL #125]

4908.   [test]          Eliminated unnecessary waiting in the allow_query
                        system test. Also changed its name to allow-query.
                        [GL #81]

4907.   [test]          Improved the reliability of the 'notify' system
                        test. [GL #59]

4906.   [func]          Replace getquad() with inet_pton(), completing
                        change #4900. [GL #56]

4905.   [bug]           irs_resconf_load() ignored resolv.conf syntax errors
                        when "domain" or "search" options were present in that
                        file. [GL #110]

4904.   [bug]           Temporarily revert change #4859. [GL #124]

4903.   [bug]           "check-mx fail;" did not prevent MX records containing
                        IP addresses from being added to a zone by a dynamic
                        update. [GL #112]

4902.   [test]          Improved the reliability of the 'ixfr' system
                        test. [GL #66]

4901.   [func]          "dig +nssearch" now lists the name servers
                        for a domain that time out, as well as the servers
                        that respond. [GL #64]

4900.   [func]          Remove all uses of inet_aton().  As a result of this
                        change, IPv4 addresses are now only accepted in
                        dotted-quad format. [GL #13]

4899.   [test]          Convert most of the remaining system tests to be able
                        to run in parallel, continuing the work from change
                        #4895. To take advantage of this, use "make -jN check",
                        where N is the number of processors to use. [GL #91]

4898.   [func]          Remove libseccomp based system-call filtering. [GL #93]

4897.   [test]          Update to rpz system test so that it doesn't recurse.
                        [GL #68]

4896.   [test]          cacheclean system test was not robust. [GL #82]
```

(continues on next page)

(continued from previous page)

```
4895.   [test]          Allow some system tests to run in parallel.
                        [RT #46602]

4894.   [bug]           named could crash while rolling a dnstap output file.
                        [RT #46942]

4893.   [bug]           Address various issues reported by cppcheck. [GL #51]

4892.   [bug]           named could leak memory when "rndc reload" was invoked
                        before all zone loading actions triggered by a previous
                        "rndc reload" command were completed. [RT #47076]

4891.   [placeholder]

4890.   [func]          Remove unused ondestroy callback from libisc.
                        [isc-projects/bind9!3]

4889.   [func]          Warn about the use of old root keys without the new
                        root key being present.  Warn about dlv.isc.org's
                        key being present. Warn about both managed and
                        trusted root keys being present. [RT #43670]

4888.   [test]          Initialize sockets correctly in sample-update so
                        that the nsupdate system test will run on Windows.
                        [RT #47097]

4887.   [test]          Enable the rpzrecurse test to run on Windows.
                        [RT #47093]

4886.   [doc]           Document dig -u in manpage. [RT #47150]

4885.   [security]      update-policy rules that otherwise ignore the name
                        field now require that it be set to "." to ensure
                        that any type list present is properly interpreted.
                        [RT #47126]

4884.   [bug]           named could crash on shutdown due to a race between
                        shutdown_server() and ns__client_request(). [RT #47120]

4883.   [cleanup]       Improved debugging output from dnssec-cds. [RT #47026]

4882.   [bug]           Address potential memory leak in
                        dns_update_signaturesinc. [RT #47084]

4881.   [bug]           Only include dst_openssl.h when OpenSSL is required.
                        [RT #47068]

4880.   [bug]           Named wasn't returning the target of a cross-zone
                        CNAME between two served zones when recursion was
                        desired and available (RD=1, RA=1). (When this is
                        not the case, the CNAME target is deliberately
                        withheld to prevent accidental cache poisoning.)
```

(continues on next page)

(continued from previous page)

```
                        [RT #47078]

4879.   [bug]           dns_rdata_caa:value_len field was too small.
                        [RT #47086]

4878.   [bug]           List 'ply' as a requirement for the 'isc' python
                        package. [RT #47065]

4877.   [bug]           Address integer overflow when exponentially
                        backing off retry intervals. [RT #47041]

4876.   [bug]           Address deadlock with accessing a keytable. [RT #47000]

4875.   [bug]           Address compile failures on older systems. [RT #47015]

4874.   [bug]           Wrong time display when reporting new keywarntime.
                        [RT #47042]

4873.   [doc]           Grammars for named.conf included in the ARM are now
                        automatically generated by the configuration parser
                        itself.  As a side effect of the work needed to
                        separate zone type grammars from each other, this
                        also makes checking of zone statements in
                        named-checkconf more correct and consistent.
                        [RT #36957]

4872.   [bug]           Don't permit loading meta RR types such as TKEY
                        from master files. [RT #47009]

4871.   [bug]           Fix configure glitch in detecting stdatomic.h
                        support on systems with multiple compilers.
                        [RT #46959]

4870.   [test]          Update included ATF library to atf-0.21 preserving
                        the ATF tool. [RT #46967]

4869.   [bug]           Address some cases where NULL with zero length could
                        be passed to memmove which is undefined behavior and
                        can lead to bad optimization. [RT #46888]

4868.   [func]          dnssec-keygen can no longer generate HMAC keys.
                        Use tsig-keygen instead. [RT #46404]

4867.   [cleanup]       Normalize rndc on/off commands (validation,
                        querylog, serve-stale) so they all accept the
                        same synonyms for on/off (yes/no, true/false,
                        enable/disable). Thanks to Tony Finch. [RT #47022]

4866.   [port]          DST library initialization verifies MD5 (when MD5
                        was not disabled) and SHA-1 hash and HMAC support.
                        [RT #46764]
```

(continues on next page)

(continued from previous page)

```
4865.   [cleanup]       Simplify handling isc_socket_sendto2() return values.
                        [RT #46986]

4864.   [bug]           named acting as a slave for a catalog zone crashed if
                        the latter contained a master definition without an IP
                        address. [RT #45999]

4863.   [bug]           Fix various other bugs reported by Valgrind's
                        memcheck tool. [RT #46978]

4862.   [bug]           The rdata flags for RRSIG were not being properly set
                        when constructing a rdataslab. [RT #46978]

4861.   [bug]           The isc_crc64 unit test was not endian independent.
                        [RT #46973]

4860.   [bug]           isc_int8_t should be signed char.  [RT #46973]

4859.   [bug]           A loop was possible when attempting to validate
                        unsigned CNAME responses from secure zones;
                        this caused a delay in returning SERVFAIL and
                        also increased the chances of encountering
                        CVE-2017-3145. [RT #46839]

4858.   [security]      Addresses could be referenced after being freed
                        in resolver.c, causing an assertion failure.
                        (CVE-2017-3145) [RT #46839]

4857.   [bug]           Maintain attach/detach semantics for event->db,
                        event->node, event->rdataset and event->sigrdataset
                        in query.c. [RT #46891]

4856.   [bug]           'rndc zonestatus' reported the wrong underlying type
                        for a inline slave zone. [RT #46875]

4855.   [bug]           isc_time_formatshorttimestamp produced incorrect
                        output. [RT #46938]

4854.   [bug]           query_synthcnamewildcard should stop generating the
                        response if query_synthwildcard fails. [RT #46939]

4853.   [bug]           Add REQUIRE's and INSIST's to isc_time_formatISO8601L
                        and isc_time_formatISO8601Lms. [RT #46916]

4852.   [bug]           Handle strftime() failing in isc_time_formatISO8601ms.
                        Add REQUIRE's and INSIST's to isc_time_formattimestamp,
                        isc_time_formathttptimestamp, isc_time_formatISO8601,
                        isc_time_formatISO8601ms. [RT #46892]

4851.   [port]          Support using kyua as well as atf-run to run the unit
                        tests. [RT #46853]
```

(continues on next page)

(continued from previous page)

```
4850.   [bug]           Named failed to restart with multiple added zones in
                        lmdb database. [RT #46889]

4849.   [bug]           Duplicate zones could appear in the .nzf file if
                        addzone failed. [RT #46435]

4848.   [func]          Zone types "primary" and "secondary" can now be used
                        as synonyms for "master" and "slave" in named.conf.
                        [RT #46713]

4847.   [bug]           dnssec-dnskey-kskonly was not being honored for
                        CDS and CDNSKEY. [RT #46755]

4846.   [test]          Adjust timing values in runtime system test. Address
                        named.pid removal races in runtime system test.
                        [RT #46800]

4845.   [bug]           Dig (non iOS) should exit on malformed names.
                        [RT #46806]

4844.   [test]          Address memory leaks in libatf-c. [RT #46798]

4843.   [bug]           dnssec-signzone free hashlist on exit. [RT #46791]

4842.   [bug]           Conditionally compile opensslecdsa_link.c to avoid
                        warnings about unused function. [RT #46790]
```

```
        --- 9.12.0rc1 released ---

4841.   [bug]           Address -fsanitize=undefined warnings. [RT #46786]

4840.   [test]          Add tests to cover fallback to using ZSK on inactive
                        KSK. [RT #46787]

4839.   [bug]           zone.c:zone_sign was not properly determining
                        if there were active KSK and ZSK keys for
                        a algorithm when update-check-ksk is true
                        (default) leaving records unsigned with one or
                        more DNSKEY algorithms. [RT #46774]

4838.   [bug]           zone.c:add_sigs was not properly determining
                        if there were active KSK and ZSK keys for
                        a algorithm when update-check-ksk is true
                        (default) leaving records unsigned with one or
                        more DNSKEY algorithms. [RT #46754]

4837.   [bug]           dns_update_signatures{inc} (add_sigs) was not
                        properly determining if there were active KSK and
                        ZSK keys for a algorithm when update-check-ksk is
                        true (default) leaving records unsigned when there
                        were multiple DNSKEY algorithms for the zone.
                        [RT #46743]
```

(continues on next page)

(continued from previous page)

```
4836.   [bug]           Zones created using "rndc addzone" could
                        temporarily fail to inherit an "allow-transfer"
                        ACL that had been configured in the options
                        statement. [RT #46603]

4835.   [cleanup]       Clean up and refactor LMDB-related code. [RT #46718]

4834.   [port]          Fix LMDB support on OpenBSD. [RT #46718]

4833.   [bug]           isc_event_free should check that the event is not
                        linked when called. [RT #46725]

4832.   [bug]           Events were not being removed from zone->rss_events.
                        [RT #46725]

4831.   [bug]           Convert the RRSIG expirytime to 64 bits for
                        comparisons in diff.c:resign. [RT #46710]

4830.   [bug]           Failure to configure ATF when requested did not cause
                        an error in top-level configure script. [RT #46655]

4829.   [bug]           isc_heap_delete did not zero the index value when
                        the heap was created with a callback to do that.
                        [RT #46709]

4828.   [bug]           Do not use thread-local storage for storing LMDB reader
                        locktable slots. [RT #46556]

4827.   [misc]          Add a precommit check script util/checklibs.sh
                        [RT #46215]

4826.   [cleanup]       Prevent potential build failures in bin/confgen/ and
                        bin/named/ when using parallel make. [RT #46648]

4825.   [bug]           Prevent a bogus "error during managed-keys processing
                        (no more)" warning from being logged. [RT #46645]

4824.   [port]          Add iOS hooks to dig. [RT #42011]

4823.   [test]          Refactor reclimit system test to improve its
                        reliability and speed. [RT #46632]

4822.   [bug]           Use resign_sooner in dns_db_setsigningtime. [RT #46473]

4821.   [bug]           When resigning ensure that the SOA's expire time is
                        always later that the resigning time of other records.
                        [RT #46473]

4820.   [bug]           dns_db_subtractrdataset should transfer the resigning
                        information to the new header. [RT #46473]
```

(continues on next page)

(continued from previous page)

```
4819.   [bug]           Fully backout the transaction when adding a RRset
                        to the resigning / removal heaps fails. [RT #46473]

4818.   [test]          The logfileconfig system test could intermittently
                        report false negatives on some platforms. [RT #46615]

4817.   [cleanup]       Use DNS_NAME_INITABSOLUTE and DNS_NAME_INITNONABSOLUTE.
                        [RT #45433]

4816.   [bug]           Don't use a common array for storing EDNS options
                        in DiG as it could fill up. [RT #45611]

4815.   [bug]           rbt_test.c:insert_and_delete needed to call
                        dns_rbt_addnode instead of dns_rbt_addname. [RT #46553]

4814.   [cleanup]       Use AS_HELP_STRING for consistent help text. [RT #46521]

4813.   [bug]           Address potential read after free errors from
                        query_synthnodata, query_synthwildcard and
                        query_synthnxdomain. [RT #46547]

4812.   [bug]           Minor improvements to stability and consistency of code
                        handling managed keys. [RT #46468]

4811.   [bug]           Revert api changes to use <isc/buffer.h> inline
                        macros.  Provide a alternative mechanism to turn
                        on the use of inline macros when building BIND.
                        [RT #46520]

4810.   [test]          The chain system test failed if the IPv6 interfaces
                        were not configured. [RT #46508]
```

```
       --- 9.12.0b2 released ---

4809.   [port]          Check at configure time whether -latomic is needed
                        for stdatomic.h. [RT #46324]

4808.   [bug]           Properly test for zlib.h. [RT #46504]

4807.   [cleanup]       isc_rng_randombytes() returns a specified number of
                        bytes from the PRNG; this is now used instead of
                        calling isc_rng_random() multiple times. [RT #46230]

4806.   [func]          Log messages related to loading of zones are now
                        directed to the "zoneload" logging category.
                        [RT #41640]

4805.   [bug]           TCP4Active and TCP6Active weren't being updated
                        correctly. [RT #46454]

4804.   [port]          win32: access() does not work on directories as
                        required by POSIX.  Supply a alternative in
```

(continues on next page)

(continued from previous page)

```
                        isc_file_isdirwritable. [RT #46394]

4803.   [placeholder]

4802.   [test]          Refactor mkeys system test to make it quicker and more
                        reliable. [RT #45293]

4801.   [func]          'dnssec-lookaside auto;' and 'dnssec-lookaside .
                        trust-anchor dlv.isc.org;' now elicit warnings rather
                        than being fatal configuration errors. [RT #46410]

4800.   [bug]           When processing delzone, write one zone config per
                        line to the NZF. [RT #46323]

4799.   [cleanup]       Improve clarity of keytable unit tests. [RT #46407]

4798.   [func]          Keys specified in "managed-keys" statements
                        are tagged as "initializing" until they have been
                        updated by a key refresh query. If initialization
                        fails it will be visible from "rndc secroots".
                        [RT #46267]

4797.   [func]          Removed "isc-hmac-fixup", as the versions of BIND that
                        had the bug it worked around are long past end of
                        life. [RT #46411]

4796.   [bug]           Increase the maximum configurable TCP keepalive
                        timeout to 65535. [RT #44710]

4795.   [func]          A new statistics counter has been added to track
                        priming queries. [RT #46313]

4794.   [func]          "dnssec-checkds -s" specifies a file from which
                        to read a DS set rather than querying the parent.
                        [RT #44667]

4793.   [bug]           nsupdate -[46] could overflow the array of server
                        addresses. [RT #46402]

4792.   [bug]           Fix map file header correctness check. [RT #38418]

4791.   [doc]           Fixed outdated documentation about export libraries.
                        [RT #46341]

4790.   [bug]           nsupdate could trigger a require when sending a
                        update to the second address of the server.
                        [RT #45731]

4789.   [cleanup]       Check writability of new-zones-directory. [RT #46308]

4788.   [cleanup]       When using "update-policy local", log a warning
                        when an update matching the session key is received
```

(continues on next page)

(continued from previous page)

```
                        from a remote host. [RT #46213]

4787.   [cleanup]       Turn nsec3param_salt_totext() into a public function,
                        dns_nsec3param_salttotext(), and add unit tests for it.
                        [RT #46289]

4786.   [func]          The "filter-aaaa-on-v4" and "filter-aaaa-on-v6"
                        options are no longer conditionally compiled.
                        [RT #46340]

4785.   [func]          The hmac-md5 algorithm is no longer recommended for
                        use with RNDC keys.  The default in rndc-confgen
                        is now hmac-sha256. [RT #42272]

4784.   [func]          The use of dnssec-keygen to generate HMAC keys is
                        deprecated in favor of tsig-keygen.  dnssec-keygen
                        will print a warning when used for this purpose.
                        All HMAC algorithms will be removed from
                        dnssec-keygen in a future release. [RT #42272]

4783.   [test]          dnssec: 'check that NOTIFY is sent at the end of
                        NSEC3 chain generation failed' required more time
                        on some machines for the IXFR to complete. [RT #46388]

4782.   [test]          dnssec: 'checking positive and negative validation
                        with negative trust anchors' required more time to
                        complete on some machines. [RT #46386]

4781.   [maint]         B.ROOT-SERVERS.NET is now 199.9.14.201. [RT #45889]

4780.   [bug]           When answering ANY queries, don't include the NS
                        RRset in the authority section if it was already
                        in the answer section. [RT #44543]

4779.   [bug]           Expire NTA at the start of the second. Don't update
                        the expiry value if the record has already expired
                        after a successful check. [RT #46368]

4778.   [test]          Improve synth-from-dnssec testing. [RT #46352]

4777.   [cleanup]       Removed a redundant call to configure_view_acl().
                        [RT #46369]

4776.   [bug]           Improve portability of ht_test. [RT #46333]

4775.   [bug]           Address Coverity warnings in ht_test.c and mem_test.c
                        [RT #46281]

4774.   [bug]           <isc/util.h> was incorrectly included in several
                        header files. [RT #46311]

4773.   [doc]           Fixed generating Doxygen documentation for functions
```

(continues on next page)

(continued from previous page)

```
                        annotated using certain macros.  Miscellaneous
                        Doxygen-related cleanups. [RT #46276]
```

```
        --- 9.12.0b1 released ---

4772.   [test]          Expanded unit testing framework for libns, using
                        hooks to interrupt query flow and inspect state
                        at specified locations. [RT #46173]

4771.   [bug]           When sending RFC 5011 refresh queries, disregard
                        cached DNSKEY rrsets. [RT #46251]

4770.   [bug]           Cache additional data from priming queries as glue.
                        Previously they were ignored as unsigned
                        non-answer data from a secure zone, and never
                        actually got added to the cache, causing hints
                        to be used frequently for root-server
                        addresses, which triggered re-priming. [RT #45241]

4769.   [func]          The working directory and managed-keys directory has
                        to be writeable (and seekable). [RT #46077]

4768.   [func]          By default, memory is no longer filled with tag values
                        when it is allocated or freed; this improves
                        performance but makes debugging of certain memory
                        issues more difficult. "named -M fill" turns memory
                        filling back on. (Building "configure
                        --enable-developer", turns memory fill on by
                        default again; it can then be disabled with
                        "named -M nofill".) [RT #45123]

4767.   [func]          Add a new function, isc_buffer_printf(), which can be
                        used to append a formatted string to the used region of
                        a buffer. [RT #46201]

4766.   [cleanup]       Address Coverity warnings. [RT #46150]

4765.   [bug]           Address potential INSIST in dnssec-cds. [RT #46150]

4764.   [bug]           Address portability issues in cds system test.
                        [RT #46214]

4763.   [contrib]       Improve compatibility when building MySQL DLZ
                        module by using mysql_config if available.
                        [RT #45558]

4762.   [func]          "update-policy local" is now restricted to updates
                        from local addresses. (Previously, other addresses
                        were allowed so long as updates were signed by the
                        local session key.) [RT #45492]

4761.   [protocol]      Add support for DOA. [RT #45612]
```

(continues on next page)

(continued from previous page)

```
4760.   [func]          Add glue cache statistics counters. [RT #46028]

4759.   [func]          Add logging channel "trust-anchor-telemetry" to
                        record trust-anchor-telemetry in incoming requests.
                        Both _ta-XXXX.<anchor>/NULL and EDNS KEY-TAG options
                        are logged.  [RT #46124]

4758.   [doc]           Remove documentation of unimplemented "topology".
                        [RT #46161]

4757.   [func]          New "dnssec-cds" command creates a new parent DS
                        RRset based on CDS or CDNSKEY RRsets found in
                        a child zone, and generates either a dsset file
                        or stream of nsupdate commands to update the
                        parent. Thanks to Tony Finch. [RT #46090]

4756.   [bug]           Interrupting dig could lead to an INSIST failure after
                        certain errors were encountered while querying a host
                        whose name resolved to more than one address.  Change
                        4537 increased the odds of triggering this issue by
                        causing dig to hang indefinitely when certain error
                        paths were evaluated.  dig now also retries TCP queries
                        (once) if the server gracefully closes the connection
                        before sending a response. [RT #42832, #45159]

4755.   [cleanup]       Silence unnecessary log message when NZF file doesn't
                        exist. [RT #46186]

4754.   [bug]           dns_zone_setview needs a two stage commit to properly
                        handle errors. [RT #45841]

4753.   [contrib]       Software obtainable from known upstream locations
                        (i.e., zkt, nslint, query-loc) has been removed.
                        Links to these and other packages can be found at
                        https://www.isc.org/community/tools [RT #46182]

4752.   [test]          Add unit test for isc_net_pton. [RT #46171]

4751.   [func]          "dnssec-signzone -S" can now automatically add parent
                        synchronization records (CDS and CDNSKEY) according
                        to key metadata set using the -Psync and -Dsync
                        options to dnssec-keygen and dnssec-settime.
                        [RT #46149]

4750.   [func]          "rndc managed-keys destroy" shuts down RFC 5011 key
                        maintenance and deletes the managed-keys database.
                        If followed by "rndc reconfig" or a server restart,
                        key maintenance is reinitialized from scratch.
                        This is primarily intended for testing. [RT #32456]

4749.   [func]          The ISC DLV service has been shut down, and all
```

(continues on next page)

(continued from previous page)

```
                        DLV records have been removed from dlv.isc.org.
                        - Removed references to ISC DLV in documentation
                        - Removed DLV key from bind.keys
                        - No longer use ISC DLV by default in delv
                        - "dnssec-lookaside auto" and configuration of
                          "dnssec-lookaide" with dlv.isc.org as the trust
                          anchor are both now fatal errors.
                        [RT #46155]

4748.   [cleanup]       Sprintf to snprintf coversions. [RT #46132]

4747.   [func]          Synthesis of responses from DNSSEC-verified records.
                        Stage 3 - synthesize NODATA responses. [RT #40138]

4746.   [cleanup]       Add configured prefixes to configure summary
                        output. [RT #46153]

4745.   [test]          Add color-coded pass/fail messages to system
                        tests when running on terminals that support them.
                        [RT #45977]

4744.   [bug]           Suppress trust-anchor-telemetry queries if
                        validation is disabled. [RT #46131]

4743.   [func]          Exclude trust-anchor-telemetry queries from
                        synth-from-dnssec processing. [RT #46123]

4742.   [func]          Synthesis of responses from DNSSEC-verified records.
                        Stage 2 - synthesis of records from wildcard data.
                        If the dns64 or filter-aaaa* is configured then the
                        involved lookups are currently excluded. [RT #40138]

4741.   [bug]           Make isc_refcount_current() atomically read the
                        counter value. [RT #46074]

4740.   [cleanup]       Avoid triggering format-truncated warnings. [RT #46107]

4739.   [cleanup]       Address clang static analysis warnings. [RT #45952]

4738.   [port]          win32: strftime mishandles %Z. [RT #46039]

4737.   [cleanup]       Address Coverity warnings. [RT #46012]

4736.   [cleanup]       (a) Added comments to NSEC3-related functions in
                        lib/dns/zone.c.  (b) Refactored NSEC3 salt formatting
                        code.  (c) Minor tweaks to lock and result handling.
                        [RT #46053]

4735.   [bug]           Add @ISC_OPENSSL_LIBS@ to isc-config. [RT #46078]

4734.   [contrib]       Added sample configuration for DNS-over-TLS in
                        contrib/dnspriv.
```

(continues on next page)

(continued from previous page)

```
4733.   [bug]           Change #4706 introduced a bug causing TCP clients
                        not be reused correctly, leading to unconstrained
                        memory growth. [RT #46029]

4732.   [func]          Change default minimal-responses setting to
                        no-auth-recursive. [RT #46016]

4731.   [bug]           Fix use after free when closing an LMDB. [RT #46000]

4730.   [bug]           Fix out of bounds access in DHCID totext() method.
                        [RT #46001]

4729.   [bug]           Don't use memset() to wipe memory, as it may be
                        removed by compiler optimizations when the
                        memset() occurs on automatic stack allocation
                        just before function return. [RT #45947]

4728.   [func]          Use C11's stdatomic.h instead of isc_atomic
                        where available. [RT #40668]

4727.   [bug]           Retransferring an inline-signed slave using NSEC3
                        around the time its NSEC3 salt was changed could result
                        in an infinite signing loop. [RT #45080]

4726.   [port]          Prevent setsockopt() errors related to TCP_FASTOPEN
                        from being logged on FreeBSD if the kernel does not
                        support it.  Notify the user when the kernel does
                        support TCP_FASTOPEN, but it is disabled by sysctl.
                        Add a new configure option, --disable-tcp-fastopen, to
                        disable use of TCP_FASTOPEN altogether. [RT #44754]

4725.   [bug]           Nsupdate: "recvsoa" was incorrectly reported for
                        failures in sending the update message.  The correct
                        location to be reported is "update_completed".
                        [RT #46014]

4724.   [func]          By default, BIND now uses the random number
                        functions provided by the crypto library (i.e.,
                        OpenSSL or a PKCS#11 provider) as a source of
                        randomness rather than /dev/random.  This is
                        suitable for virtual machine environments
                        which have limited entropy pools and lack
                        hardware random number generators.

                        This can be overridden by specifying another
                        entropy source via the "random-device" option
                        in named.conf, or via the -r command line option;
                        however, for functions requiring full cryptographic
                        strength, such as DNSSEC key generation, this
                        cannot be overridden. In particular, the -r
                        command line option no longer has any effect on
```

(continues on next page)

(continued from previous page)

```
                        dnssec-keygen.

                        This can be disabled by building with
                        "configure --disable-crypto-rand".
                        [RT #31459] [RT #46047]

4723.   [bug]           Statistics counter DNSTAPdropped was misidentified
                        as DNSSECdropped. [RT #46002]

4722.   [cleanup]       Clean up uses of strcpy() and strcat() in favor of
                        strlcpy() and strlcat() for safety. [RT #45981]

4721.   [func]          'dnssec-signzone -x' and 'dnssec-dnskey-kskonly'
                        options now apply to CDNSKEY and DS records as well
                        as DNSKEY. Thanks to Tony Finch. [RT #45689]

4720.   [func]          Added a statistics counter to track prefetch
                        queries. [RT #45847]

4719.   [bug]           Address PVS static analyzer warnings. [RT #45946]

4718.   [func]          Avoid searching for a owner name compression pointer
                        more than once when writing out a RRset. [RT #45802]

4717.   [bug]           Treat replies with QCOUNT=0 as truncated if TC=1,
                        FORMERR if TC=0, and log the error correctly.
                        [RT #45836]

4716.   [placeholder]
```

```
        --- 9.12.0a1 released ---

4715.   [bug]           TreeMemMax was mis-identified as a second HeapMemMax
                        in the Json cache statistics. [RT #45980]

4714.   [port]          openbsd/libressl: add support for building with
                        --enable-openssl-hash. [RT #45982]

4713.   [func]          Added support for the DNS Response Policy Service
                        (DNSRPS) API, which allows named to use an external
                        response policy daemon when built with
                        "configure --enable-dnsrps". Thanks to Farsight
                        Security. [RT #43376]

4712.   [bug]           "dig +domain" and "dig +search" didn't retain the
                        search domain when retrying with TCP. [RT #45547]

4711.   [test]          Some RR types were missing from genzones.sh.
                        [RT #45782]

4710.   [cleanup]       Changed the --enable-openssl-hash default to yes.
                        [RT #45019]
```

(continues on next page)

(continued from previous page)

```
4709.   [cleanup]       Use dns_name_fullhash() to hash names for RRL.
                        [RT #45435]

4708.   [cleanup]       Legacy Windows builds (i.e. for XP and earlier)
                        are no longer supported. [RT #45186]

4707.   [func]          The lightweight resolver daemon and library (lwresd
                        and liblwres) have been removed. [RT #45186]

4706.   [func]          Code implementing name server query processing has
                        been moved from bin/named to a new library "libns".
                        Functions remaining in bin/named are now prefixed
                        with "named_" rather than "ns_".  This will make it
                        easier to write unit tests for name server code, or
                        link name server functionality into new tools.
                        [RT #45186]

4705.   [placeholder]

4704.   [cleanup]       Silence Visual Studio compiler warnings. [RT #45898]

4703.   [bug]           BINDInstall.exe was missing some buffer length checks.
                        [RT #45898]

4702.   [func]          Update function declarations to use
                        dns_masterstyle_flags_t for style flags. [RT #45924]

4701.   [cleanup]       Refactored lib/dns/tsig.c to reduce code
                        duplication and simplify the disabling of MD5.
                        [RT #45490]

4700.   [func]          Serving of stale answers is now supported. This
                        allows named to provide stale cached answers when
                        the authoritative server is under attack.
                        See max-stale-ttl, stale-answer-enable,
                        stale-answer-ttl. [RT #44790]

4699.   [func]          Multiple cookie-secret clauses can now be specified.
                        The first one specified is used to generate new
                        server cookies.  [RT #45672]

4698.   [port]          Add --with-python-install-dir configure option to allow
                        specifying a nonstandard installation directory for
                        Python modules. [RT #45407]

4697.   [bug]           Restore workaround for Microsoft Windows TSIG hash
                        computation bug. [RT #45854]

4696.   [port]          Enable filter-aaaa support by default on Windows
                        builds. [RT #45883]
```

(continues on next page)

(continued from previous page)

```
4695.   [bug]           cookie-secrets were not being properly checked by
                        named-checkconf. [RT #45886]

4694.   [func]          dnssec-keygen no longer uses RSASHA1 by default;
                        the signing algorithm must be specified on
                        the command line with the "-a" option.  Signing
                        scripts that rely on the existing default behavior
                        will break; use "dnssec-keygen -a RSASHA1" to
                        repair them. (The goal of this change is to make
                        it easier to find scripts using RSASHA1 so they
                        can be changed in the event of that algorithm
                        being deprecated in the future.) [RT #44755]

4693.   [func]          Synthesis of responses from DNSSEC-verified records.
                        Stage 1 covers NXDOMAIN synthesis from NSEC records.
                        This is controlled by synth-from-dnssec and is enabled
                        by default. [RT #40138]

4692.   [bug]           Fix build failures with libressl introduced in 4676.
                        [RT #45879]

4691.   [func]          Add -4/-6 command line options to nsupdate and rndc.
                        [RT #45632]

4690.   [bug]           Command line options -4/-6 were handled inconsistently
                        between tools. [RT #45632]

4689.   [cleanup]       Turn on minimal responses for CDNSKEY and CDS in
                        addition to DNSKEY and DS. Thanks to Tony Finch.
                        [RT #45690]

4688.   [protocol]      Check and display EDNS KEY TAG options (RFC 8145) in
                        messages. [RT #44804]

4687.   [func]          Refactor tracklines code. [RT #45126]

4686.   [bug]           dnssec-settime -p could print a bogus warning about
                        key deletion scheduled before its inactivation when a
                        key had an inactivation date set but no deletion date
                        set. [RT #45807]

4685.   [bug]           dnssec-settime incorrectly calculated publication and
                        activation dates for a successor key. [RT #45806]

4684.   [bug]           delv could send bogus DNS queries when an explicit
                        server address was specified on the command line along
                        with -4/-6. [RT #45804]

4683.   [bug]           Prevent nsupdate from immediately exiting on invalid
                        user input in interactive mode. [RT #28194]

4682.   [bug]           Don't report errors on records below a DNAME.
```

(continues on next page)

(continued from previous page)

```
                        [RT #44880]

4681.   [bug]           Log messages from the validator now include the
                        associated view unless the view is "_default/IN"
                        or "_dnsclient/IN". [RT #45770]

4680.   [bug]           Fix failing over to another master server address when
                        nsupdate is used with GSS-API. [RT #45380]

4679.   [cleanup]       Suggest using -o when dnssec-verify finds a SOA record
                        not at top of zone and -o is not used. [RT #45519]

4678.   [bug]           geoip-use-ecs has the wrong type when geoip support
                        is disabled at configure time. [RT #45763]

4677.   [cleanup]       Split up the main function in dig to better support
                        the iOS app version. [RT #45508]

4676.   [cleanup]       Allow BIND to be built using OpenSSL 1.0.X with
                        deprecated functions removed. [RT #45706]

4675.   [cleanup]       Don't use C++ keyword class. [RT #45726]

4674.   [func]          "dig +sigchase", and related options "+topdown" and
                        "+trusted-keys", have been removed. Use "delv" for
                        queries with DNSSEC validation. [RT #42793]

4673.   [port]          Silence GCC 7 warnings. [RT #45592]

4672.   [placeholder]

4671.   [bug]           Fix a race condition that could cause the
                        resolver to crash with assertion failure when
                        chasing DS in specific conditions with a very
                        short RTT to the upstream nameserver. [RT #45168]

4670.   [cleanup]       Ensure that a request MAC is never sent back
                        in an XFR response unless the signature was
                        verified. [RT #45494]

4669.   [func]          Iterative query logic in resolver.c has been
                        refactored into smaller functions and commented,
                        for improved readability, maintainability and
                        testability. [RT #45362]

4668.   [bug]           Use localtime_r and gmtime_r for thread safety.
                        [RT #45664]

4667.   [cleanup]       Refactor RDATA unit tests. [RT #45610]

4666.   [bug]           dnssec-keymgr: Domain names beginning with digits (0-9)
                        could cause a parser error when reading the policy
```

(continues on next page)

(continued from previous page)

```
                        file. This now works correctly so long as the domain
                        name is quoted. [RT #45641]

4665.   [protocol]      Added support for ED25519 and ED448 DNSSEC signing
                        algorithms (RFC 8080). (Note: these algorithms
                        depend on code currently in the development branch
                        of OpenSSL which has not yet been released.)
                        [RT #44696]

4664.   [func]          Add a "glue-cache" option to enable or disable the
                        glue cache. The default is "yes". [RT #45125]

4663.   [cleanup]       Clarify error message printed by dnssec-dsfromkey.
                        [RT #21731]

4662.   [performance]   Improve cache memory cleanup of zero TTL records
                        by putting them at the tail of LRU header lists.
                        [RT #45274]

4661.   [bug]           A race condition could occur if a zone was reloaded
                        while resigning, triggering a crash in
                        rbtdb.c:closeversion(). [RT #45276]

4660.   [bug]           Remove spurious "peer" from Windows socket log
                        messages. [RT #45617]

4659.   [bug]           Remove spurious log message about lmdb-mapsize
                        not being supported when parsing builtin
                        configuration file. [RT #45618]

4658.   [bug]           Clean up build directory created by "setup.py install"
                        immediately.  [RT #45628]

4657.   [bug]           rrchecker system test result could be improperly
                        determined. [RT #45602]

4656.   [bug]           Apply "port" and "dscp" values specified in catalog
                        zone's "default-masters" option to the generated
                        configuration of its member zones. [RT #45545]

4655.   [bug]           Lack of seccomp could be falsely reported. [RT #45599]

4654.   [cleanup]       Don't use C++ keywords delete, new and namespace.
                        [RT #45538]

4653.   [bug]           Reorder includes to move @DST_OPENSSL_INC@ and
                        @ISC_OPENSSL_INC@ after shipped include directories.
                        [RT #45581]

4652.   [bug]           Nsupdate could attempt to use a zeroed address on
                        server timeout. [RT #45417]
```

(continues on next page)

(continued from previous page)

```
4651.   [test]          Silence coverity warnings in tsig_test.c. [RT #45528]

4650.   [placeholder]

4649.   [bug]           The wrong zone was logged when a catalog zone is added.
                        [RT #45520]

4648.   [bug]           "rndc reconfig" on a slave no longer causes all member
                        zones of configured catalog zones to be removed from
                        configuration. [RT #45310]

4647.   [bug]           Change 4643 broke verification of TSIG signed TCP
                        message sequences where not all the messages contain
                        TSIG records.  These may be used in AXFR and IXFR
                        responses. [RT #45509]

4646.   [placeholder]

4645.   [bug]           Fix PKCS#11 RSA parsing when MD5 is disabled.
                        [RT #45300]

4644.   [placeholder]

4643.   [security]      An error in TSIG handling could permit unauthorized
                        zone transfers or zone updates. (CVE-2017-3142)
                        (CVE-2017-3143) [RT #45383]

4642.   [cleanup]       Add more logging of RFC 5011 events affecting the
                        status of managed keys: newly observed keys,
                        deletion of revoked keys, etc. [RT #45354]

4641.   [cleanup]       Parallel builds (make -j) could fail with --with-atf /
                        --enable-developer. [RT #45373]

4640.   [bug]           If query_findversion failed in query_getdb due to
                        memory failure the error status was incorrectly
                        discarded. [RT #45331]

4639.   [bug]           Fix a regression in --with-tuning reporting introduced
                        by change 4488. [RT #45396]

4638.   [bug]           Reloading or reconfiguring named could fail on
                        some platforms when LMDB was in use. [RT #45203]

4637.   [func]          "nsec3hash -r" option ("rdata order") takes arguments
                        in the same order as they appear in NSEC3 or
                        NSEC3PARAM records, so that NSEC3 parameters can
                        be cut and pasted from an existing record. Thanks
                        to Tony Finch for the contribution. [RT #45183]

4636.   [bug]           Normalize rpz policy zone names when checking for
                        existence. [RT #45358]
```

(continues on next page)

(continued from previous page)

```
4635.   [bug]           Fix RPZ NSDNAME logging that was logging
                        failures as NSIP. [RT #45052]

4634.   [contrib]       check5011.pl needs to handle optional space before
                        semi-colon in +multi-line output. [RT #45352]

4633.   [maint]         Updated AAAA (2001:500:200::b) for B.ROOT-SERVERS.NET.

4632.   [security]      The BIND installer on Windows used an unquoted
                        service path, which can enable privilege escalation.
                        (CVE-2017-3141) [RT #45229]

4631.   [security]      Some RPZ configurations could go into an infinite
                        query loop when encountering responses with TTL=0.
                        (CVE-2017-3140) [RT #45181]

4630.   [bug]           "dyndb" is dependent on dlopen existing / being
                        enabled. [RT #45291]

4629.   [bug]           dns_client_startupdate could not be called with a
                        running client. [RT #45277]

4628.   [bug]           Fixed a potential reference leak in query_getdb().
                        [RT #45247]

4627.   [placeholder]

4626.   [test]          Added more tests for handling of different record
                        ordering in CNAME and DNAME responses. [QA #430]

4625.   [bug]           Running "rndc addzone" and "rndc delzone" at close
                        to the same time could trigger a deadlock if using
                        LMDB. [RT #45209]

4624.   [placeholder]

4623.   [bug]           Use --with-protobuf-c and --with-libfstrm to find
                        protoc-c and fstrm_capture. [RT #45187]

4622.   [bug]           Remove unnecessary escaping of semicolon in CAA and
                        URI records. [RT #45216]

4621.   [port]          Force alignment of oid arrays to silence loader
                        warnings. [RT #45131]

4620.   [port]          Handle EPFNOSUPPORT being returned when probing
                        to see if a socket type is supported. [RT #45214]

4619.   [bug]           Call isc_mem_put instead of isc_mem_free in
                        bin/named/server.c:setup_newzones. [RT #45202]
```

(continues on next page)

(continued from previous page)

```
4618.   [bug]           Check isc_mem_strdup results in dns_view_setnewzones.
                        Add logging for lmdb call failures. [RT #45204]

4617.   [test]          Update rndc system test to be more delay tolerant.
                        [RT #45177]

4616.   [bug]           When using LMDB, zones deleted using "rndc delzone"
                        were not correctly removed from the new-zone
                        database. [RT #45185]

4615.   [bug]           AD could be set on truncated answer with no records
                        present in the answer and authority sections.
                        [RT #45140]

4614.   [test]          Fixed an error in the sockaddr unit test. [RT #45146]

4613.   [func]          By default, the maximum size of a zone journal file
                        is now twice the size of the zone's contents (there
                        is little benefit to a journal larger than this).
                        This can be overridden by setting "max-journal-size"
                        to "unlimited" or to an explicit value up to 2G.
                        Thanks to Tony Finch. [RT #38324]

4612.   [bug]           Silence 'may be use uninitalised' warning and simplify
                        the code in lwres/getaddinfo:process_answer.
                        [RT #45158]

4611.   [bug]           The default LMDB mapsize was too low and caused
                        errors after few thousand zones were added using
                        rndc addzone. A new config option "lmdb-mapsize"
                        has been introduced to configure the LMDB
                        mapsize depending on operational needs.
                        [RT #44954]

4610.   [func]          The "new-zones-directory" option specifies the
                        location of NZF or NZD files for storing
                        configuration of zones added by "rndc addzone".
                        Thanks to Petr Menšík. [RT #44853]

4609.   [cleanup]       Rearrange makefiles to enable parallel execution
                        (i.e. "make -j"). [RT #45078]

4608.   [func]          DiG now warns about .local queries which are reserved
                        for Multicast DNS. [RT #44783]

4607.   [bug]           The memory context's malloced and maxmalloced counters
                        were being updated without the appropriate lock being
                        held.  [RT #44869]

4606.   [port]          Stop using experimental "Experimental keys on scalar"
                        feature of perl as it has been removed. [RT #45012]
```

(continues on next page)

(continued from previous page)

```
4605.   [performance]   Improve performance for delegation heavy answers
                        and also general query performance. Removes the
                        acache feature that didn't significantly improve
                        performance. Adds a glue cache. Removes
                        additional-from-cache and additional-from-auth
                        features. Enables minimal-responses by
                        default. Improves performance of compression
                        code, owner case restoration, hash function,
                        etc. Uses inline buffer implementation by
                        default. Many other performance changes and fixes.
                        [RT #44029]

4604.   [bug]           Don't use ERR_load_crypto_strings() when building
                        with OpenSSL 1.1.0. [RT #45117]

4603.   [doc]           Automatically generate named.conf(5) man page
                        from doc/misc/options. Thanks to Tony Finch.
                        [RT #43525]

4602.   [func]          Threads are now set to human-readable
                        names to assist debugging, when supported by
                        the OS. [RT #43234]

4601.   [bug]           Reject incorrect RSA key lengths during key
                        generation and and sign/verify context
                        creation. [RT #45043]

4600.   [bug]           Adjust RPZ trigger counts only when the entry
                        being deleted exists. [RT #43386]

4599.   [bug]           Fix inconsistencies in inline signing time
                        comparison that were introduced with the
                        introduction of rdatasetheader->resign_lsb.
                        [RT #42112]

4598.   [func]          Update fuzzing code to (1) reply to a DNSKEY
                        query from named with appropriate DNSKEY used in
                        fuzzing; (2) patch the QTYPE correctly in
                        resolver fuzzing; (3) comment things so the rest
                        of us are able to understand how fuzzing is
                        implemented in named; (4) Coding style changes,
                        cleanup, etc. [RT #44787]

4597.   [bug]           The validator now ignores SHA-1 DS digest type
                        when a DS record with SHA-384 digest type is
                        present and is a supported digest type.
                        [RT #45017]

4596.   [bug]           Validate glue before adding it to the additional
                        section. This also fixes incorrect TTL capping
                        when the RRSIG expired earlier than the TTL.
                        [RT #45062]
```

(continues on next page)

(continued from previous page)

```
4595.   [func]          dnssec-keygen will no longer generate RSA keys
                        less than 1024 bits in length. dnssec-keymgr
                        was similarly updated. [RT #36895]

4594.   [func]          "dnstap-read -x" prints a hex dump of the wire
                        format of each logged DNS message. [RT #44816]

4593.   [doc]           Update README using markdown, remove outdated FAQ
                        file in favor of the knowledge base.

4592.   [bug]           A race condition on shutdown could trigger an
                        assertion failure in dispatch.c. [RT #43822]

4591.   [port]          Addressed some python 3 compatibility issues.
                        Thanks to Ville Skytta. [RT #44955] [RT #44956]

4590.   [bug]           Support for PTHREAD_MUTEX_ADAPTIVE_NP was not being
                        properly detected. [RT #44871]

4589.   [cleanup]       "configure -q" is now silent. [RT #44829]

4588.   [bug]           nsupdate could send queries for TKEY to the wrong
                        server when using GSSAPI. Thanks to Tomas Hozza.
                        [RT #39893]

4587.   [bug]           named-checkzone failed to handle occulted data below
                        DNAMEs correctly. [RT #44877]

4586.   [func]          dig, host and nslookup now use TCP for ANY queries.
                        [RT #44687]

4585.   [port]          win32: Set CompileAS value. [RT #42474]

4584.   [bug]           A number of memory usage statistics were not properly
                        reported when they exceeded 4G.  [RT #44750]

4583.   [func]          "host -A" returns most records for a name but
                        omits RRSIG, NSEC and NSEC3. (Thanks to Tony Finch.)
                        [RT #43032]

4582.   [security]      'rndc ""' could trigger a assertion failure in named.
                        (CVE-2017-3138) [RT #44924]

4581.   [port]          Linux: Add getpid and getrandom to the list of system
                        calls named uses for seccomp. [RT #44883]

4580.   [bug]           4578 introduced a regression when handling CNAME to
                        referral below the current domain. [RT #44850]

4579.   [func]          Logging channels and dnstap output files can now
                        be configured with a "suffix" option, set to
```

(continues on next page)

```
                        either "increment" or "timestamp", indicating
                        whether to use incrementing numbers or timestamps
                        as the file suffix when rolling over a log file.
                        [RT #42838]

4578.   [security]      Some chaining (CNAME or DNAME) responses to upstream
                        queries could trigger assertion failures.
                        (CVE-2017-3137) [RT #44734]

4577.   [func]          Make qtype of resolver fuzzing packet configurable
                        via command line. [RT #43540]

4576.   [func]          The RPZ implementation has been substantially
                        refactored for improved performance and reliability.
                        [RT #43449]

4575.   [security]      DNS64 with "break-dnssec yes;" can result in an
                        assertion failure. (CVE-2017-3136) [RT #44653]

4574.   [bug]           Dig leaked memory with multiple +subnet options.
                        [RT #44683]

4573.   [func]          Query logic has been substantially refactored (e.g.
                        query_find function has been split into smaller
                        functions) for improved readability, maintainability
                        and testability. [RT #43929]

4572.   [func]          The "dnstap-output" option can now take "size" and
                        "versions" parameters to indicate the maximum size
                        a dnstap log file can grow before rolling to a new
                        file, and how many old files to retain. [RT #44502]

4571.   [bug]           Out-of-tree builds of backtrace_test failed.

4570.   [cleanup]       named did not correctly fall back to the built-in
                        initializing keys if the bind.keys file was present
                        but empty. [RT #44531]

4569.   [func]          Store both local and remote addresses in dnstap
                        logging, and modify dnstap-read output format to
                        print them. [RT #43595]

4568.   [contrib]       Added a --with-bind option to the dnsperf configure
                        script to specify BIND prefix path.

4567.   [port]          Call getprotobyname and getservbyname prior to calling
                        chroot so that shared libraries get loaded. [RT #44537]

4566.   [func]          Query logging now includes the ECS option if one
                        was included in the query. [RT #44476]

4565.   [cleanup]       The inline macro versions of isc_buffer_put*()
                        did not implement automatic buffer reallocation.
```

(continues on next page)

(continued from previous page)

```
                        [RT #44216]

4564.   [maint]         Update the built in managed keys to include the
                        upcoming root KSK. [RT #44579]

4563.   [bug]           Modified zones would occasionally fail to reload.
                        [RT #39424]

4562.   [func]          Add additional memory statistics currently malloced
                        and maxmalloced per memory context. [RT #43593]

4561.   [port]          Silence a warning in strict C99 compilers. [RT #44414]

4560.   [bug]           mdig: add -m option to enable memory debugging rather
                        than having it on all the time. [RT #44509]

4559.   [bug]           openssl_link.c didn't compile if ISC_MEM_TRACKLINES
                        was turned off.  [RT #44509]

4558.   [bug]           Synthesised CNAME before matching DNAME was still
                        being cached when it should not have been.  [RT #44318]

4557.   [security]      Combining dns64 and rpz can result in dereferencing
                        a NULL pointer (read).  (CVE-2017-3135) [RT#44434]

4556.   [bug]           Sending an EDNS Padding option using "dig
                        +ednsopt" could cause a crash in dig. [RT #44462]

4555.   [func]          dig +ednsopt: EDNS options can now be specified by
                        name in addition to numeric value. [RT #44461]

4554.   [bug]           Remove double unlock in dns_dispatchmgr_setudp.
                        [RT #44336]

4553.   [bug]           Named could deadlock there were multiple changes to
                        NSEC/NSEC3 parameters for a zone being processed at
                        the same time. [RT #42770]

4552.   [bug]           Named could trigger a assertion when sending notify
                        messages. [RT #44019]

4551.   [test]          Add system tests for integrity checks of MX and
                        SRV records. [RT #43953]

4550.   [cleanup]       Increased the number of available master file
                        output style flags from 32 to 64. [RT #44043]

4549.   [func]          Added support for the EDNS TCP Keepalive option
                        (RFC 7828). [RT #42126]

4548.   [func]          Added support for the EDNS Padding option (RFC 7830).
                        [RT #42094]
```

(continues on next page)

(continued from previous page)

```
4547.   [port]          Add support for --enable-native-pkcs11 on the AEP
                        Keyper HSM. [RT #42463]

4546.   [func]          Extend the use of const declarations. [RT #43379]

4545.   [func]          Expand YAML output from dnstap-read to include
                        a detailed breakdown of the DNS message contents.
                        [RT #43642]

4544.   [bug]           Add message/payload size to dnstap-read YAML output.
                        [RT #43622]

4543.   [bug]           dns_client_startupdate now delays sending the update
                        request until isc_app_ctxrun has been called.
                        [RT #43976]

4542.   [func]          Allow rndc to manipulate redirect zones with using
                        -redirect as the zone name (use "-redirect." to
                        manipulate a zone named "-redirect"). [RT #43971]

4541.   [bug]           rndc addzone should properly reject non master/slave
                        zones. [RT #43665]

4540.   [bug]           Correctly handle ecs entries in dns_acl_isinsecure.
                        [RT #43601]

4539.   [bug]           Referencing a nonexistent zone with RPZ could lead
                        to a assertion failure when configuring. [RT #43787]

4538.   [bug]           Call dns_client_startresolve from client->task.
                        [RT #43896]

4537.   [bug]           Handle timeouts better in dig/host/nslookup. [RT #43576]

4536.   [bug]           ISC_SOCKEVENTATTR_USEMINMTU was not being cleared
                        when reusing the event structure. [RT #43885]

4535.   [bug]           Address race condition in setting / testing of
                        DNS_REQUEST_F_SENDING. [RT #43889]

4534.   [bug]           Only set RD, RA and CD in QUERY responses. [RT #43879]

4533.   [bug]           dns_client_update should terminate on prerequisite
                        failures (NXDOMAIN, YXDOMAIN, NXRRSET, YXRRSET)
                        and also on BADZONE.  [RT #43865]

4532.   [contrib]       Make gen-data-queryperf.py python 3 compatible.
                        [RT #43836]

4531.   [security]      'is_zone' was not being properly updated by redirect2
                        and subsequently preserved leading to an assertion
```

(continues on next page)

(continued from previous page)

```
                        failure. (CVE-2016-9778) [RT #43837]

4530.   [bug]           Change 4489 broke the handling of CNAME -> DNAME
                        in responses resulting in SERVFAIL being returned.
                        [RT #43779]

4529.   [cleanup]       Silence noisy log warning when DSCP probe fails
                        due to firewall rules. [RT #43847]

4528.   [bug]           Only set the flag bits for the i/o we are waiting
                        for on EPOLLERR or EPOLLHUP. [RT #43617]

4527.   [doc]           Support DocBook XSL Stylesheets v1.79.1. [RT #43831]

4526.   [doc]           Corrected errors and improved formatting of
                        grammar definitions in the ARM. [RT #43739]

4525.   [doc]           Fixed outdated documentation on managed-keys.
                        [RT #43810]

4524.   [bug]           The net zero test was broken causing IPv4 servers
                        with addresses ending in .0 to be rejected. [RT #43776]

4523.   [doc]           Expand config doc for <querysource4> and
                        <querysource6>. [RT #43768]

4522.   [bug]           Handle big gaps in log file version numbers better.
                        [RT #38688]

4521.   [cleanup]       Log it as an error if an entropy source is not
                        found and there is no fallback available. [RT #43659]

4520.   [cleanup]       Alphabetize more of the grammar when printing it
                        out. Fix unbalanced indenting. [RT #43755]

4519.   [port]          win32: handle ERROR_MORE_DATA. [RT #43534]

4518.   [func]          The "print-time" option in the logging configuration
                        can now take arguments "local", "iso8601" or
                        "iso8601-utc" to indicate the format in which the
                        date and time should be logged. For backward
                        compatibility, "yes" is a synonym for "local".
                        [RT #42585]

4517.   [security]      Named could mishandle authority sections that were
                        missing RRSIGs triggering an assertion failure.
                        (CVE-2016-9444) [RT # 43632]

4516.   [bug]           isc_socketmgr_renderjson was missing from the
                        windows build. [RT #43602]

4515.   [port]          FreeBSD: Find readline headers when they are in
```

(continues on next page)

(continued from previous page)

```
                        edit/readline/ instead of readline/. [RT #43658]

4514.   [port]          NetBSD: strip -WL, from ld command line. [RT #43204]

4513.   [cleanup]       Minimum Python versions are now 2.7 and 3.2.
                        [RT #43566]

4512.   [bug]           win32: @GEOIP_INC@ missing from delv.vcxproj.in.
                        [RT #43556]

4511.   [bug]           win32: mdig.exe-BNFT was missing Configure. [RT #43554]

4510.   [security]      Named mishandled some responses where covering RRSIG
                        records are returned without the requested data
                        resulting in a assertion failure. (CVE-2016-9147)
                        [RT #43548]

4509.   [test]          Make the rrl system test more reliable on slower
                        machines by using mdig instead of dig. [RT #43280]

4508.   [security]      Named incorrectly tried to cache TKEY records which
                        could trigger a assertion failure when there was
                        a class mismatch. (CVE-2016-9131) [RT #43522]

4507.   [bug]           Named could incorrectly log 'allows updates by IP
                        address, which is insecure' [RT #43432]

4506.   [func]          'named-checkconf -l' will now list the zones found in
                        named.conf. [RT #43154]

4505.   [port]          Use IP_PMTUDISC_OMIT if available. [RT #35494]

4504.   [security]      Allow the maximum number of records in a zone to
                        be specified.  This provides a control for issues
                        raised in CVE-2016-6170. [RT #42143]

4503.   [cleanup]       "make uninstall" now removes files installed by
                        BIND. (This currently excludes Python files
                        due to lack of support in setup.py.) [RT #42192]

4502.   [func]          Report multiple and experimental options when printing
                        grammar. [RT #43134]

4501.   [placeholder]

4500.   [bug]           Support modifier I64 in isc__print_printf. [RT #43526]

4499.   [port]          MacOSX: silence deprecated function warning
                        by using arc4random_stir() when available
                        instead of arc4random_addrandom(). [RT #43503]

4498.   [test]          Simplify prerequisite checks in system tests.
```

(continues on next page)

(continued from previous page)

```
                        [RT #43516]

4497.   [port]          Add support for OpenSSL 1.1.0. [RT #41284]

4496.   [func]          dig: add +idnout to control whether labels are
                        display in punycode or not.  Requires idn support
                        to be enabled at compile time. [RT #43398]

4495.   [bug]           A isc_mutex_init call was not being checked.
                        [RT #43391]

4494.   [bug]           Look for <editline/readline.h>. [RT #43429]

4493.   [bug]           bin/tests/system/dyndb/driver/Makefile.in should use
                        SO_TARGETS. [RT# 43336]

4492.   [bug]           irs_resconf_load failed to initialize sortlistnxt
                        causing bad writes if resolv.conf contained a
                        sortlist directive. [RT #43459]

4491.   [bug]           Improve message emitted when testing whether sendmsg
                        works with TOS/TCLASS fails. [RT #43483]

4490.   [maint]         Added AAAA (2001:500:12::d0d) for G.ROOT-SERVERS.NET.

4489.   [security]      It was possible to trigger assertions when processing
                        a response containing a DNAME answer. (CVE-2016-8864)
                        [RT #43465]

4488.   [port]          Darwin: use -framework for Kerberos. [RT #43418]

4487.   [test]          Make system tests work on Windows. [RT #42931]

4486.   [bug]           Look in $prefix/lib/pythonX.Y/site-packages for
                        the python modules we install. [RT #43330]

4485.   [bug]           Failure to find readline when requested should be
                        fatal to configure. [RT #43328]

4484.   [func]          Check prefixes in acls to make sure the address and
                        prefix lengths are consistent.  Warn only in
                        BIND 9.11 and earlier. [RT #43367]

4483.   [bug]           Address use before require check and remove extraneous
                        dns_message_gettsigkey call in dns_tsig_sign.
                        [RT #43374]

4482.   [cleanup]       Change #4455 was incomplete. [RT #43252]

4481.   [func]          dig: make +class, +crypto, +multiline, +rrcomments,
                        +onesoa, +qr, +ttlid, +ttlunits and -u per lookup
                        rather than global. [RT #42450]
```

(continues on next page)

(continued from previous page)

```
4480.   [placeholder]

4479.   [placeholder]

4478.   [func]          Add +continue option to mdig, allow continue on socket
                        errors. [RT #43281]

4477.   [test]          Fix mkeys test timing issues. [RT #41028]

4476.   [test]          Fix reclimit test on slower machines. [RT #43283]

4475.   [doc]           Update named-checkconf documentation. [RT #43153]

4474.   [bug]           win32: call WSAStartup in fromtext_in_wks so that
                        getprotobyname and getservbyname work.  [RT #43197]

4473.   [bug]           Only call fsync / _commit on regular files. [RT #43196]

4472.   [bug]           Named could fail to find the correct NSEC3 records when
                        a zone was updated between looking for the answer and
                        looking for the NSEC3 records proving nonexistence
                        of the answer. [RT #43247]
```

```
--- 9.11.0 released ---
```

```
        --- 9.11.0rc3 released ---

4471.   [cleanup]       Render client/query logging format consistent for
                        ease of log file parsing. (Note that this affects
                        "querylog" format: there is now an additional field
                        indicating the client object address.) [RT #43238]

4470.   [bug]           Reset message with intent parse before
                        calling dns_dispatch_getnext. [RT #43229]

4469.   [placeholder]
```

```
        --- 9.11.0rc2 released ---

4468.   [bug]           Address ECS option handling issues. [RT #43191]

4467.   [security]      It was possible to trigger an assertion when
                        rendering a message. (CVE-2016-2776) [RT #43139]

4466.   [bug]           Interface scanning didn't work on a Windows system
                        without a non local IPv6 addresses. [RT #43130]

4465.   [bug]           Don't use "%z" as Windows doesn't support it.
                        [RT #43131]
```

(continues on next page)

(continued from previous page)

```
4464.   [bug]           Fix windows python support. [RT #43173]

4463.   [bug]           The dnstap system test failed on some systems.
                        [RT #43129]

4462.   [bug]           Don't describe a returned EDNS COOKIE as "good"
                        when there isn't a valid server cookie. [RT #43167]

4461.   [bug]           win32: not all external data was properly marked
                        as external data for windows dll. [RT #43161]
```

```
        --- 9.11.0rc1 released ---

4460.   [test]          Add system test for dnstap using unix domain sockets.
                        [RT #42926]

4459.   [bug]           TCP client objects created to handle pipeline queries
                        were not cleaned up correctly, causing uncontrolled
                        memory growth. [RT #43106]

4458.   [cleanup]       Update assertions to be more correct, and also remove
                        use of a reserved word. [RT #43090]

4457.   [maint]         Added AAAA (2001:500:a8::e) for E.ROOT-SERVERS.NET.

4456.   [doc]           Add DOCTYPE and lang attribute to <html> tags.
                        [RT #42587]

4455.   [cleanup]       Allow dyndb modules to correctly log the filename
                        and line number when processing configuration text
                        from named.conf. [RT #43050]

4454.   [bug]           'rndc dnstap -reopen' had a race issue. [RT #43089]

4453.   [bug]           Prefetching of DS records failed to update their
                        RRSIGs. [RT #42865]

4452.   [bug]           The default key manager policy file is now
                        <sysdir>/dnssec-policy.conf (usually
                        /etc/dnssec-policy.conf). [RT #43064]

4451.   [cleanup]       Log more useful information if a PKCS#11 provider
                        library cannot be loaded. [RT #43076]

4450.   [port]          Provide more nuanced HSM support which better matches
                        the specific PKCS11 providers capabilities. [RT #42458]

4449.   [test]          Fix catalog zones test on slower systems. [RT #42997]

4448.   [bug]           win32: ::1 was not being found when iterating
                        interfaces. [RT #42993]
```

(continues on next page)

(continued from previous page)

```
4447.   [tuning]        Allow the fstrm_iothr_init() options to be set using
                        named.conf to control how dnstap manages the data
                        flow. [RT #42974]

4446.   [bug]           The cache_find() and _findrdataset() functions
                        could find rdatasets that had been marked stale.
                        [RT #42853]

4445.   [cleanup]       isc_errno_toresult() can now be used to call the
                        formerly private function isc__errno2result().
                        [RT #43050]

4444.   [bug]           Fixed some issues related to dyndb: A bug caused
                        braces to be omitted when passing configuration text
                        from named.conf to a dyndb driver, and there was a
                        use-after-free in the sample dyndb driver. [RT #43050]

4443.   [func]          Set TCP_MAXSEG in addition to IPV6_USE_MIN_MTU on
                        TCP sockets. [RT #42864]

4442.   [bug]           Fix RPZ CIDR tree insertion bug that corrupted
                        tree data structure with overlapping networks
                        (longest prefix match was ineffective).
                        [RT #43035]

4441.   [cleanup]       Alphabetize host's help output. [RT #43031]

4440.   [func]          Enable TCP fast open support when available on the
                        server side. [RT #42866]

4439.   [bug]           Address race conditions getting ownernames of nodes.
                        [RT #43005]

4438.   [func]          Use LIFO rather than FIFO when processing startup
                        notify and refresh queries. [RT #42825]

4437.   [func]          Minimal-responses now has two additional modes
                        no-auth and no-auth-recursive which suppress
                        adding the NS records to the authority section
                        as well as the associated address records for the
                        nameservers. [RT #42005]

4436.   [func]          Return TLSA records as additional data for MX and SRV
                        lookups. [RT #42894]

4435.   [tuning]        Only set IPV6_USE_MIN_MTU for UDP when the message
                        will not fit into a single IPv4 encapsulated IPv6
                        UDP packet when transmitted over a Ethernet link.
                        [RT #42871]

4434.   [protocol]      Return EDNS EXPIRE option for master zones in addition
                        to slave zones. [RT #43008]
```

(continues on next page)

(continued from previous page)

```
4433.   [cleanup]       Report an error when passing an invalid option or
                        view name to "rndc dumpdb". [RT #42958]

4432.   [test]          Hide rndc output on expected failures in logfileconfig
                        system test. [RT #27996]

4431.   [bug]           named-checkconf now checks the rate-limit clause.
                        [RT #42970]

4430.   [bug]           Lwresd died if a search list was not defined.
                        Found by 0x710DDDD At Alibaba Security. [RT #42895]

4429.   [bug]           Address potential use after free on fclose() error.
                        [RT #42976]

4428.   [bug]           The "test dispatch getnext" unit test could fail
                        in a threaded build. [RT #42979]

4427.   [bug]           The "query" and "response" parameters to the
                        "dnstap" option had their functions reversed.
```

```
        --- 9.11.0b3 released ---

4426.   [bug]           Addressed Coverity warnings. [RT #42908]

4425.   [bug]           arpaname, dnstap-read and named-rrchecker were not
                        being installed into ${prefix}/bin.  Tidy up
                        installation issues with CHANGE 4421. [RT #42910]

4424.   [experimental]  Named now sends _ta-XXXX.<trust-anchor>/NULL queries
                        to provide feedback to the trust-anchor administrators
                        about how key rollovers are progressing as per
                        draft-ietf-dnsop-edns-key-tag-02.  This can be
                        disabled using 'trust-anchor-telemetry no;'.
                        [RT #40583]

4423.   [maint]         Added missing IPv6 address 2001:500:84::b for
                        B.ROOT-SERVERS.NET. [RT #42898]

4422.   [port]          Silence clang warnings in dig.c and dighost.c.
                        [RT #42451]

4421.   [func]          When built with LMDB (Lightning Memory-mapped
                        Database), named will now use a database to store
                        the configuration for zones added by "rndc addzone"
                        instead of using a flat NZF file. This improves
                        performance of "rndc delzone" and "rndc modzone"
                        significantly. Existing NZF files will
                        automatically by converted to NZD databases.
                        To view the contents of an NZD or to roll back to
                        NZF format, use "named-nzd2nzf". To disable
```

(continues on next page)

(continued from previous page)

```
                        this feature, use "configure --without-lmdb".
                        [RT #39837]

4420.   [func]          nslookup now looks for AAAA as well as A by default.
                        [RT #40420]

4419.   [bug]           Don't cause undefined result if the label of an
                        entry in catalog zone is changed. [RT #42708]

4418.   [bug]           Fix a compiler warning in GSSAPI code. [RT #42879]

4417.   [bug]           dnssec-keymgr could fail to create successor keys
                        if the prepublication interval was set to a value
                        smaller than the default. [RT #42820]

4416.   [bug]           dnssec-keymgr: Domain names in policy files could
                        fail to match due to trailing dots. [RT #42807]

4415.   [bug]           dnssec-keymgr: Expired/deleted keys were not always
                        excluded. [RT #42884]

4414.   [bug]           Corrected a bug in the MIPS implementation of
                        isc_atomic_xadd(). [RT #41965]

4413.   [bug]           GSSAPI negotiation could fail if GSS_S_CONTINUE_NEEDED
                        was returned. [RT #42733]
```

```
        --- 9.11.0b2 released ---

4412.   [cleanup]       Make fixes for GCC 6. ISC_OFFSET_MAXIMUM macro was
                        removed. [RT #42721]

4411.   [func]          "rndc dnstap -roll" automatically rolls the
                        dnstap output file; the previous version is
                        saved with ".0" suffix, and earlier versions
                        with ".1" and so on. An optional numeric argument
                        indicates how many prior files to save. [RT #42830]

4410.   [bug]           Address use after free and memory leak with dnstap.
                        [RT #42746]

4409.   [bug]           DNS64 should exclude mapped addresses by default when
                        an exclude acl is not defined. [RT #42810]

4408.   [func]          Continue waiting for expected response when we the
                        response we get does not match the request. [RT #41026]

4407.   [performance]   Use GCC builtin for clz in RPZ lookup code.
                        [RT #42818]

4406.   [security]      getrrsetbyname with a non absolute name could
                        trigger an infinite recursion bug in lwresd
```

(continues on next page)

(continued from previous page)

```
                        and named with lwres configured if when combined
                        with a search list entry the resulting name is
                        too long. (CVE-2016-2775) [RT #42694]

4405.   [bug]           Change 4342 introduced a regression where you could
                        not remove a delegation in a NSEC3 signed zone using
                        OPTOUT via nsupdate. [RT #42702]

4404.   [misc]          Allow krb5-config to be used when configuring gssapi.
                        [RT #42580]

4403.   [bug]           Rename variables and arguments that shadow: basename,
                        clone and gai_error.

4402.   [bug]           protoc-c is now a hard requirement for --enable-dnstap.
```

```
        --- 9.11.0b1 released ---

4401.   [misc]          Change LICENSE to MPL 2.0.

4400.   [bug]           ttl policy was not being inherited in policy.py.
                        [RT #42718]

4399.   [bug]           policy.py 'ECCGOST', 'ECDSAP256SHA256', and
                        'ECDSAP384SHA384' don't have settable keysize.
                        [RT #42718]

4398.   [bug]           Correct spelling of ECDSAP256SHA256 in policy.py.
                        [RT #42718]

4397.   [bug]           Update Windows python support. [RT #42538]

4396.   [func]          dnssec-keymgr now takes a '-r randomfile' option.
                        [RT #42455]

4395.   [bug]           Improve out-of-tree installation of python modules.
                        [RT #42586]

4394.   [func]          Add rndc command "dnstap-reopen" to close and
                        reopen dnstap output files. [RT #41803]

4393.   [bug]           Address potential NULL pointer dereferences in
                        dnstap code.

4392.   [func]          Collect statistics for RSSAC02v3 traffic-volume,
                        traffic-sizes and rcode-volume reporting. [RT #41475]

4391.   [contrib]       Fix leaks in contrib DLZ code. [RT #42707]

4390.   [doc]           Description of masters with TSIG, allow-query and
                        allow-transfer options in catalog zones. [RT #42692]
```

(continues on next page)

(continued from previous page)

```
4389.   [test]          Rewritten test suite for catalog zones. [RT #42676]

4388.   [func]          Support for master entries with TSIG keys in catalog
                        zones. [RT #42577]

4387.   [bug]           Change 4336 was not complete leading to SERVFAIL
                        being return as NS records expired. [RT #42683]

4386.   [bug]           Remove shadowed overmem function/variable. [RT #42706]

4385.   [func]          Add support for allow-query and allow-transfer ACLs
                        to catalog zones. [RT #42578]

4384.   [bug]           Change 4256 accidentally disabled logging of the
                        rndc command. [RT #42654]

4383.   [bug]           Correct spelling error in stats channel description of
                        "EDNS client subnet option received". [RT #42633]

4382.   [bug]           rndc {addzone,modzone,delzone,showzone} should all
                        compare the zone name using a canonical format.
                        [RT #42630]

4381.   [bug]           Missing "zone-directory" option in catalog zone
                        definition caused BIND to crash. [RT #42579]
```

```
        --- 9.11.0a3 released ---

4380.   [experimental]  Added a "zone-directory" option to "catalog-zones"
                        syntax, allowing local masterfiles for slaves
                        that are provisioned by catalog zones to be stored
                        in a directory other than the server's working
                        directory. [RT #42527]

4379.   [bug]           An INSIST could be triggered if a zone contains
                        RRSIG records with expiry fields that loop
                        using serial number arithmetic. [RT #40571]

4378.   [contrib]       #include <isc/string.h> for strlcat in zone2ldap.c.
                        [RT #42525]

4377.   [bug]           Don't reuse zero TTL responses beyond the current
                        client set (excludes ANY/SIG/RRSIG queries).
                        [RT #42142]

4376.   [experimental]  Added support for Catalog Zones, a new method for
                        provisioning secondary servers in which a list of
                        zones to be served is stored in a DNS zone and can
                        be propagated to slaves via AXFR/IXFR. [RT #41581]

4375.   [func]          Add support for automatic reallocation of isc_buffer
                        to isc_buffer_put* functions. [RT #42394]
```

(continues on next page)

(continued from previous page)

```
4374.   [bug]           Use SAVE/RESTORE macros in query.c to reduce the
                        probability of reference counting errors as seen
                        in 4365. [RT #42405]

4373.   [bug]           Address undefined behavior in getaddrinfo. [RT #42479]

4372.   [bug]           Address undefined behavior in libt_api. [RT #42480]

4371.   [func]          New "minimal-any" option reduces the size of UDP
                        responses for qtype ANY by returning a single
                        arbitrarily selected RRset instead of all RRsets.
                        Thanks to Tony Finch. [RT #41615]

4370.   [bug]           Address python3 compatibility issues with RNDC module.
                        [RT #42499] [RT #42506]
```

```
        --- 9.11.0a2 released ---

4369.   [bug]           Fix 'make' and 'make install' out-of-tree python
                        support. [RT #42484]

4368.   [bug]           Fix a crash when calling "rndc stats" on some
                        Windows builds because some Visual Studio compilers
                        generated crashing code for the "%z" printf()
                        format specifier. [RT #42380]

4367.   [bug]           Remove unnecessary assignment of loadtime in
                        zone_touched. [RT #42440]

4366.   [bug]           Address race condition when updating rbtnode bit
                        fields. [RT #42379]

4365.   [bug]           Address zone reference counting errors involving
                        nxdomain-redirect. [RT #42258]

4364.   [port]          freebsd: add -Wl,-E to loader flags [RT #41690]

4363.   [port]          win32: Disable explicit triggering UAC when running
                        BINDInstall.

4362.   [func]          Changed rndc reconfig behavior so that newly added
                        zones are loaded asynchronously and the loading does
                        not block the server. [RT #41934]

4361.   [cleanup]       Where supported, file modification times returned
                        by isc_file_getmodtime() are now accurate to the
                        nanosecond. [RT #41968]

4360.   [bug]           Silence spurious 'bad key type' message when there is
                        a existing TSIG key. [RT #42195]
```

(continues on next page)

(continued from previous page)

```
4359.   [bug]           Inherited 'also-notify' lists were not being checked
                        by named-checkconf. [RT #42174]

4358.   [test]          Added American Fuzzy Lop harness that allows
                        feeding fuzzed packets into BIND.
                        [RT #41723]

4357.   [func]          Add the python RNDC module. [RT #42093]

4356.   [func]          Add the ability to specify whether to wait for
                        nameserver addresses to be looked up or not to
                        RPZ with a new modifying directive 'nsip-wait-recurse'.
                        [RT #35009]

4355.   [func]          "pkcs11-list" now displays the extractability
                        attribute of private or secret keys stored in
                        an HSM, as either "true", "false", or "never"
                        Thanks to Daniel Stirnimann. [RT #36557]

4354.   [bug]           Check that the received HMAC length matches the
                        expected length prior to check the contents on the
                        control channel.  This prevents a OOB read error.
                        This was reported by Lian Yihan, <lianyihan@360.cn>.
                        [RT #42215]

4353.   [cleanup]       Update PKCS#11 header files. [RT #42175]

4352.   [cleanup]       The ISC DNSSEC Lookaside Validation (DLV) service
                        is scheduled to be disabled in 2017.  A warning is
                        now logged when named is configured to use it,
                        either explicitly or via "dnssec-lookaside auto;"
                        [RT #42207]

4351.   [bug]           'dig +noignore' didn't work. [RT #42273]

4350.   [contrib]       Declare result in  dlz_filesystem_dynamic.c.

4349.   [contrib]       kasp2policy: A python script to create a DNSSEC
                        policy file from an OpenDNSSEC KASP XML file.

4348.   [func]          dnssec-keymgr: A new python-based DNSSEC key
                        management utility, which reads a policy definition
                        file and can create or update DNSSEC keys as needed
                        to ensure that a zone's keys match policy, roll over
                        correctly on schedule, etc.  Thanks to Sebastian
                        Castro for assistance in development. [RT #39211]

4347.   [port]          Corrected a build error on x86_64 Solaris. [RT #42150]

4346.   [bug]           Fixed a regression introduced in change #4337 which
                        caused signed domains with revoked KSKs to fail
                        validation. [RT #42147]
```

(continues on next page)

(continued from previous page)

```
4345.   [contrib]       perftcpdns mishandled the return values from
                        clock_nanosleep. [RT #42131]

4344.   [port]          Address openssl version differences. [RT #42059]

4343.   [bug]           dns_dnssec_syncupdate mis-declared in <dns/dnssec.h>.
                        [RT #42090]

4342.   [bug]           'rndc flushtree' could fail to clean the tree if there
                        wasn't a node at the specified name. [RT #41846]
```

```
        --- 9.11.0a1 released ---

4341.   [bug]           Correct the handling of ECS options with
                        address family 0. [RT #41377]

4340.   [performance]   Implement adaptive read-write locks, reducing the
                        overhead of locks that are only held briefly.
                        [RT #37329]

4339.   [test]          Use "mdig" to test pipelined queries. [RT #41929]

4338.   [bug]           Reimplement change 4324 as it wasn't properly doing
                        all the required book keeping. [RT #41941]

4337.   [bug]           The previous change exposed a latent flaw in
                        key refresh queries for managed-keys when
                        a cached DNSKEY had TTL 0. [RT #41986]

4336.   [bug]           Don't emit records with zero ttl unless the records
                        were learnt with a zero ttl. [RT #41687]

4335.   [bug]           zone->view could be detached too early. [RT #41942]

4334.   [func]          'named -V' now reports zlib version. [RT #41913]

4333.   [maint]         L.ROOT-SERVERS.NET is now 199.7.83.42 and
                        2001:500:9f::42.

4332.   [placeholder]

4331.   [func]          When loading managed signed zones detect if the
                        RRSIG's inception time is in the future and regenerate
                        the RRSIG immediately. [RT #41808]

4330.   [protocol]      Identify the PAD option as "PAD" when printing out
                        a message.

4329.   [func]          Warn about a common misconfiguration when forwarding
                        RFC 1918 zones. [RT #41441]
```

(continues on next page)

(continued from previous page)

```
4328.   [performance]   Add dns_name_fromwire() benchmark test. [RT #41694]

4327.   [func]          Log query and depth counters during fetches when
                        querytrace (./configure --enable-querytrace) is
                        enabled (helps in diagnosing).  [RT #41787]

4326.   [protocol]      Add support for AVC. [RT #41819]

4325.   [func]          Add a line to "rndc status" indicating the
                        hostname and operating system details. [RT #41610]

4324.   [bug]           When deleting records from a zone database, interior
                        nodes could be left empty but not deleted, damaging
                        search performance afterward. [RT #40997]

4323.   [bug]           Improve HTTP header processing on statschannel.
                        [RT #41674]

4322.   [security]      Duplicate EDNS COOKIE options in a response could
                        trigger an assertion failure. (CVE-2016-2088)
                        [RT #41809]

4321.   [bug]           Zones using mapped files containing out-of-zone data
                        could return SERVFAIL instead of the expected NODATA
                        or NXDOMAIN results. [RT #41596]

4320.   [bug]           Insufficient memory allocation when handling
                        "none" ACL could cause an assertion failure in
                        named when parsing ACL configuration. [RT #41745]

4319.   [security]      Fix resolver assertion failure due to improper
                        DNAME handling when parsing fetch reply messages.
                        (CVE-2016-1286) [RT #41753]

4318.   [security]      Malformed control messages can trigger assertions
                        in named and rndc. (CVE-2016-1285) [RT #41666]

4317.   [bug]           Age all unused servers on fetch timeout. [RT #41597]

4316.   [func]          Add option to tools to print RRs in unknown
                        presentation format [RT #41595].

4315.   [bug]           Check that configured view class isn't a meta class.
                        [RT #41572].

4314.   [contrib]       Added 'dnsperf-2.1.0.0-1', a set of performance
                        testing tools provided by Nominum, Inc.

4313.   [bug]           Handle ns_client_replace failures in test mode.
                        [RT #41190]

4312.   [bug]           dig's unknown DNS and EDNS flags (MBZ value) logging
```

(continues on next page)

(continued from previous page)

```
                        was not consistent. [RT #41600]

4311.   [bug]           Prevent "rndc delzone" from being used on
                        response-policy zones. [RT #41593]

4310.   [performance]   Use __builtin_expect() where available to annotate
                        conditions with known behavior. [RT #41411]

4309.   [cleanup]       Remove the spurious "none" filename from log messages
                        when processing built-in configuration. [RT #41594]

4308.   [func]          Added operating system details to "named -V"
                        output. [RT #41452]

4307.   [bug]           "dig +subnet" and "mdig +subnet" could send
                        incorrectly-formatted Client Subnet options
                        if the prefix length was not divisible by 8.
                        Also fixed a memory leak in "mdig". [RT #45178]

4306.   [maint]         Added a PKCS#11 openssl patch supporting
                        version 1.0.2f [RT #38312]

4305.   [bug]           dnssec-signzone was not removing unnecessary rrsigs
                        from the zone's apex. [RT #41483]

4304.   [port]          xfer system test failed as 'tail -n +value' is not
                        portable. [RT #41315]

4303.   [bug]           "dig +subnet" was unable to send a prefix length of
                        zero, as it was incorrectly changed to 32 for v4
                        prefixes or 128 for v6 prefixes. In addition to
                        fixing this, "dig +subnet=0" has been added as a
                        short form for 0.0.0.0/0. The same changes have
                        also been made in "mdig". [RT #41553]

4302.   [port]          win32: fixed a build error in VS 2015. [RT #41426]

4301.   [bug]           dnssec-settime -p [DP]sync was not working. [RT #41534]

4300.   [bug]           A flag could be set in the wrong field when setting
                        up non-recursive queries; this could cause the
                        SERVFAIL cache to cache responses it shouldn't.
                        New querytrace logging has been added which
                        identified this error. [RT #41155]

4299.   [bug]           Check that exactly totallen bytes are read when
                        reading a RRset from raw files in both single read
                        and incremental modes. [RT #41402]

4298.   [bug]           dns_rpz_add errors in loadzone were not being
                        propagated up the call stack. [RT #41425]
```

(continues on next page)

(continued from previous page)

```
4297.   [test]          Ensure delegations in RPZ zones fail robustly.
                        [RT #41518]

4296.   [bug]           TCP packet sizes were calculated incorrectly in the
                        stats channel; they could be counted in the wrong
                        histogram bucket. [RT #40587]

4295.   [bug]           An unchecked result in dns_message_pseudosectiontotext()
                        could allow incorrect text formatting of EDNS EXPIRE
                        options. [RT #41437]

4294.   [bug]           Fixed a regression in which "rndc stop -p" failed
                        to print the PID. [RT #41513]

4293.   [bug]           Address memory leak on priming query creation failure.
                        [RT #41512]

4292.   [placeholder]

4291.   [cleanup]       Added a required include to dns/forward.h. [RT #41474]

4290.   [func]          The timers returned by the statistics channel
                        (indicating current time, server boot time, and
                        most recent reconfiguration time) are now reported
                        with millisecond accuracy. [RT #40082]

4289.   [bug]           The server could crash due to memory being used
                        after it was freed if a zone transfer timed out.
                        [RT #41297]

4288.   [bug]           Fixed a regression in resolver.c:possibly_mark()
                        which caused known-bogus servers to be queried
                        anyway. [RT #41321]

4287.   [bug]           Silence an overly noisy log message when message
                        parsing fails. [RT #41374]

4286.   [security]      render_ecs errors were mishandled when printing out
                        a OPT record resulting in a assertion failure.
                        (CVE-2015-8705) [RT #41397]

4285.   [security]      Specific APL data could trigger a INSIST.
                        (CVE-2015-8704) [RT #41396]

4284.   [bug]           Some GeoIP options were incorrectly documented
                        using abbreviated forms which were not accepted by
                        named.  The code has been updated to allow both
                        long and abbreviated forms. [RT #41381]

4283.   [bug]           OPENSSL_config is no longer re-callable. [RT #41348]

4282.   [func]          'dig +[no]mapped' determine whether the use of mapped
```

(continues on next page)

(continued from previous page)

```
                        IPv4 addresses over IPv6 is permitted or not.  The
                        default is +mapped.  [RT #41307]

4281.   [bug]           Teach dns_message_totext about BADCOOKIE. [RT #41257]

4280.   [performance]   Use optimal message sizes to improve compression
                        in AXFRs. This reduces network traffic. [RT #40996]

4279.   [test]          Don't use fixed ports when unit testing. [RT #41194]

4278.   [bug]           'delv +short +[no]split[=##]' didn't work as expected.
                        [RT #41238]

4277.   [performance]   Improve performance of the RBT, the central zone
                        datastructure: The aux hashtable was improved,
                        hash function was updated to perform more
                        uniform mapping, uppernode was added to
                        dns_rbtnode, and other cleanups and performance
                        improvements were made. [RT #41165]

4276.   [protocol]      Add support for SMIMEA. [RT #40513]

4275.   [performance]   Lazily initialize dns_compress->table only when
                        compression is enabled. [RT #41189]

4274.   [performance]   Speed up typemap processing from text. [RT #41196]

4273.   [bug]           Only call dns_test_begin() and dns_test_end() once each
                        in nsec3_test as it fails with GOST if called multiple
                        times.

4272.   [bug]           dig: the +norrcomments option didn't work with +multi.
                        [RT #41234]

4271.   [test]          Unit tests could deadlock in isc__taskmgr_pause().
                        [RT #41235]

4270.   [security]      Update allowed OpenSSL versions as named is
                        potentially vulnerable to CVE-2015-3193.

4269.   [bug]           Zones using "map" format master files currently
                        don't work as policy zones.  This limitation has
                        now been documented; attempting to use such zones
                        in "response-policy" statements is now a
                        configuration error.  [RT #38321]

4268.   [func]          "rndc status" now reports the path to the
                        configuration file. [RT #36470]

4267.   [test]          Check sdlz error handling. [RT #41142]

4266.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
4265.   [bug]           Address unchecked isc_mem_get calls. [RT #41187]

4264.   [bug]           Check const of strchr/strrchr assignments match
                        argument's const status. [RT #41150]

4263.   [contrib]       Address compiler warnings in mysqldyn module.
                        [RT #41130]

4262.   [bug]           Fixed a bug in epoll socket code that caused
                        sockets to not be registered for ready
                        notification in some cases, causing named to not
                        read from or write to them, resulting in what
                        appear to the user as blocked connections.
                        [RT #41067]

4261.   [maint]         H.ROOT-SERVERS.NET is 198.97.190.53 and 2001:500:1::53.
                        [RT #40556]

4260.   [security]      Insufficient testing when parsing a message allowed
                        records with an incorrect class to be be accepted,
                        triggering a REQUIRE failure when those records
                        were subsequently cached. (CVE-2015-8000) [RT #40987]

4259.   [func]          Add an option for non-destructive control channel
                        access using a "read-only" clause. In such
                        cases, a restricted set of rndc commands are
                        allowed for querying information from named.
                        [RT #40498]

4258.   [bug]           Limit rndc query message sizes to 32 KiB. This should
                        not break any legitimate rndc commands, but will
                        prevent a rogue rndc query from allocating too
                        much memory. [RT #41073]

4257.   [cleanup]       Python scripts reported incorrect version. [RT #41080]

4256.   [bug]           Allow rndc command arguments to be quoted so as
                        to allow spaces. [RT #36665]

4255.   [performance]   Add 'message-compression' option to disable DNS
                        compression in responses. [RT #40726]

4254.   [bug]           Address missing lock when getting zone's serial.
                        [RT #41072]

4253.   [security]      Address fetch context reference count handling error
                        on socket error. (CVE-2015-8461)  [RT#40945]

4252.   [func]          Add support for automating the generation CDS and
                        CDNSKEY rrsets to named and dnssec-signzone.
                        [RT #40424]
```

(continues on next page)

(continued from previous page)

```
4251.   [bug]           NTAs were deleted when the server was reconfigured
                        or reloaded. [RT #41058]

4250.   [func]          Log the TSIG key in use during inbound zone
                        transfers. [RT #41075]

4249.   [func]          Improve error reporting of TSIG / SIG(0) records in
                        the wrong location. [RT #41030]

4248.   [performance]   Add an isc_atomic_storeq() function, use it in
                        stats counters to improve performance.
                        [RT #39972] [RT #39979]

4247.   [port]          Require both HAVE_JSON and JSON_C_VERSION to be
                        defined to report json library version. [RT #41045]

4246.   [test]          Ensure the statschannel system test runs when BIND
                        is not built with libjson. [RT #40944]

4245.   [placeholder]

4244.   [bug]           The parser was not reporting that use-ixfr is obsolete.
                        [RT #41010]

4243.   [func]          Improved stats reporting from Timothe Litt. [RT #38941]

4242.   [bug]           Replace the client if not already replaced when
                        prefetching. [RT #41001]

4241.   [doc]           Improved the TSIG, TKEY, and SIG(0) sections in
                        the ARM. [RT #40955]

4240.   [port]          Fix LibreSSL compatibility. [RT #40977]

4239.   [func]          Changed default servfail-ttl value to 1 second from 10.
                        Also, the maximum value is now 30 instead of 300.
                        [RT #37556]

4238.   [bug]           Don't send to servers on net zero (0.0.0.0/8).
                        [RT #40947]

4237.   [doc]           Upgraded documentation toolchain to use DocBook 5
                        and dblatex. [RT #40766]

4236.   [performance]   On machines with 2 or more processors (CPU), the
                        default value for the number of UDP listeners
                        has been changed to the number of detected
                        processors minus one. [RT #40761]

4235.   [func]          Added support in named for "dnstap", a fast method of
                        capturing and logging DNS traffic, and a new command
```

(continues on next page)

(continued from previous page)

```
                        "dnstap-read" to read a dnstap log file.  Use
                        "configure --enable-dnstap" to enable this
                        feature (note that this requires libprotobuf-c
                        and libfstrm). See the ARM for configuration details.

                        Thanks to Robert Edmonds of Farsight Security.
                        [RT #40211]

4234.   [func]          Add deflate compression in statistics channel HTTP
                        server. [RT #40861]

4233.   [test]          Add tests for CDS and CDNSKEY with delegation-only.
                        [RT #40597]

4232.   [contrib]       Address unchecked memory allocation calls in
                        query-loc and zone2ldap. [RT #40789]

4231.   [contrib]       Address unchecked calloc call in dlz_mysqldyn_mod.c.
                        [RT #40840]

4230.   [contrib]       dlz_wildcard_dynamic.c:dlz_create could return a
                        uninitialized result. [RT #40839]

4229.   [bug]           A variable could be used uninitialized in
                        dns_update_signaturesinc. [RT #40784]

4228.   [bug]           Address race condition in dns_client_destroyrestrans.
                        [RT #40605]

4227.   [bug]           Silence static analysis warnings. [RT #40828]

4226.   [bug]           Address a theoretical shutdown race in
                        zone.c:notify_send_queue(). [RT #38958]

4225.   [port]          freebsd/openbsd:  Use '${CC} -shared' for building
                        shared libraries. [RT #39557]

4224.   [func]          Added support for "dyndb", a new interface for loading
                        zone data from an external database, developed by
                        Red Hat for the FreeIPA project.

                        DynDB drivers fully implement the BIND database
                        API, and are capable of significantly better
                        performance and functionality than DLZ drivers,
                        while taking advantage of advanced database
                        features not available in BIND such as multi-master
                        replication.

                        Thanks to Adam Tkac and Petr Spacek of Red Hat.
                        [RT #35271]

4223.   [func]          Add support for setting max-cache-size to percentage
```

(continues on next page)

(continued from previous page)

```
                        of available physical memory, set default to 90%.
                        [RT #38442]

4222.   [func]          Bias IPv6 servers when selecting the next server to
                        query. [RT #40836]

4221.   [bug]           Resource leak on DNS_R_NXDOMAIN in fctx_create.
                        [RT #40583]

4220.   [doc]           Improve documentation for zone-statistics.
                        [RT #36955]

4219.   [bug]           Set event->result to ISC_R_WOULDBLOCK on EWOULDBLOCK,
                        EGAIN when these soft error are not retried for
                        isc_socket_send*().

4218.   [bug]           Potential null pointer dereference on out of memory
                        if mmap is not supported. [RT #40777]

4217.   [protocol]      Add support for CSYNC. [RT #40532]

4216.   [cleanup]       Silence static analysis warnings. [RT #40649]

4215.   [bug]           nsupdate: skip to next request on GSSTKEY create
                        failure. [RT #40685]

4214.   [protocol]      Add support for TALINK.  [RT #40544]

4213.   [bug]           Don't reuse a cache across multiple classes.
                        [RT #40205]

4212.   [func]          Re-query if we get a bad client cookie returned over
                        UDP. [RT #40748]

4211.   [bug]           Ensure that lwresd gets at least one task to work
                        with if enabled. [RT #40652]

4210.   [cleanup]       Silence use after free false positive. [RT #40743]

4209.   [bug]           Address resource leaks in dlz modules. [RT #40654]

4208.   [bug]           Address null pointer dereferences on out of memory.
                        [RT #40764]

4207.   [bug]           Handle class mismatches with raw zone files.
                        [RT #40746]

4206.   [bug]           contrib: fixed a possible NULL dereference in
                        DLZ wildcard module. [RT #40745]

4205.   [bug]           'named-checkconf -p' could include unwanted spaces
                        when printing tuples with unset optional fields.
```

(continues on next page)

```
                        [RT #40731]

4204.   [bug]           'dig +trace' failed to lookup the correct type if
                        the initial root NS query was retried. [RT #40296]

4203.   [test]          The rrchecker system test now tests conversion
                        to and from unknown-type format. [RT #40584]

4202.   [bug]           isccc_cc_fromwire() could return an incorrect
                        result. [RT #40614]

4201.   [func]          The default preferred-glue is now the address record
                        type of the transport the query was received
                        over.  [RT #40468]

4200.   [cleanup]       win32: update BINDinstall to be BIND release
                        independent. [RT #38915]

4199.   [protocol]      Add support for NINFO, RKEY, SINK, TA.
                        [RT #40545] [RT #40547] [RT #40561] [RT #40563]

4198.   [placeholder]

4197.   [bug]           'named-checkconf -z' didn't handle 'in-view' clauses.
                        [RT #40603]

4196.   [doc]           Improve how "enum + other" types are documented.
                        [RT #40608]

4195.   [bug]           'max-zone-ttl unlimited;' was broken. [RT #40608]

4194.   [bug]           named-checkconf -p failed to properly print a port
                        range.  [RT #40634]

4193.   [bug]           Handle broken servers that return BADVERS incorrectly.
                        [RT #40427]

4192.   [bug]           The default rrset-order of random was not always being
                        applied. [RT #40456]

4191.   [protocol]      Accept DNS-SD non LDH PTR records in reverse zones
                        as per RFC 6763. [RT #37889]

4190.   [protocol]      Accept Active Directory gc._msdcs.<forest> name as
                        valid with check-names.  <forest> still needs to be
                        LDH. [RT #40399]

4189.   [cleanup]       Don't exit on overly long tokens in named.conf.
                        [RT #40418]

4188.   [bug]           Support HTTP/1.0 client properly on the statistics
                        channel. [RT #40261]
```

(continues on next page)

(continued from previous page)

```
4187.   [func]          When any RR type implementation doesn't
                        implement totext() for the RDATA's wire
                        representation and returns ISC_R_NOTIMPLEMENTED,
                        such RDATA is now printed in unknown
                        presentation format (RFC 3597). RR types affected
                        include LOC(29) and APL(42). [RT #40317].

4186.   [bug]           Fixed an RPZ bug where a QNAME would be matched
                        against a policy RR with wildcard owner name
                        (trigger) where the QNAME was the wildcard owner
                        name's parent. For example, the bug caused a query
                        with QNAME "example.com" to match a policy RR with
                        "*.example.com" as trigger. [RT #40357]

4185.   [bug]           Fixed an RPZ bug where a policy RR with wildcard
                        owner name (trigger) would prevent another policy RR
                        with its parent owner name from being
                        loaded. For example, the bug caused a policy RR
                        with trigger "example.com" to not have any
                        effect when a previous policy RR with trigger
                        "*.example.com" existed in that RPZ zone.
                        [RT #40357]

4184.   [bug]           Fixed a possible memory leak in name compression
                        when rendering long messages. (Also, improved
                        wire_test for testing such messages.) [RT #40375]

4183.   [cleanup]       Use timing-safe memory comparisons in cryptographic
                        code. Also, the timing-safe comparison functions have
                        been renamed to avoid possible confusion with
                        memcmp(). Thanks to Loganaden Velvindron of
                        AFRINIC. [RT #40148]

4182.   [cleanup]       Use mnemonics for RR class and type comparisons.
                        [RT #40297]

4181.   [bug]           Queued notify messages could be dequeued from the
                        wrong rate limiter queue. [RT #40350]

4180.   [bug]           Error responses in pipelined queries could
                        cause a crash in client.c. [RT #40289]

4179.   [bug]           Fix double frees in getaddrinfo() in libirs.
                        [RT #40209]

4178.   [bug]           Fix assertion failure in parsing UNSPEC(103) RR from
                        text. [RT #40274]

4177.   [bug]           Fix assertion failure in parsing NSAP records from
                        text. [RT #40285]

4176.   [bug]           Address race issues with lwresd. [RT #40284]
```

(continues on next page)

(continued from previous page)

```
4175.   [bug]           TKEY with GSS-API keys needed bigger buffers.
                        [RT #40333]

4174.   [bug]           "dnssec-coverage -r" didn't handle time unit
                        suffixes correctly. [RT #38444]

4173.   [bug]           dig +sigchase was not properly matching the trusted
                        key. [RT #40188]

4172.   [bug]           Named / named-checkconf didn't handle a view of CLASS0.
                        [RT #40265]

4171.   [bug]           Fixed incorrect class checks in TSIG RR
                        implementation. [RT #40287]

4170.   [security]      An incorrect boundary check in the OPENPGPKEY
                        rdatatype could trigger an assertion failure.
                        (CVE-2015-5986) [RT #40286]

4169.   [test]          Added a 'wire_test -d' option to read input as
                        raw binary data, for use as a fuzzing harness.
                        [RT #40312]

4168.   [security]      A buffer accounting error could trigger an
                        assertion failure when parsing certain malformed
                        DNSSEC keys. (CVE-2015-5722) [RT #40212]

4167.   [func]          Update rndc's usage output to include recently added
                        commands. Thanks to Tony Finch for submitting a
                        patch. [RT #40010]

4166.   [func]          Print informative output from rndc showzone when
                        allow-new-zones is not enabled for a view. Thanks to
                        Tony Finch for submitting a patch. [RT #40009]

4165.   [security]      A failure to reset a value to NULL in tkey.c could
                        result in an assertion failure. (CVE-2015-5477)
                        [RT #40046]

4164.   [bug]           Don't rename slave files and journals on out of memory.
                        [RT #40033]

4163.   [bug]           Address compiler warnings. [RT #40024]

4162.   [bug]           httpdmgr->flags was not being initialized. [RT #40017]

4161.   [test]          Add JSON test for traffic size stats; also test
                        for consistency between "rndc stats" and the XML
                        and JSON statistics channel contents. [RT #38700]

4160.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
4159.   [cleanup]       Alphabetize dig's help output. [RT #39966]

4158.   [placeholder]

4157.   [placeholder]

4156.   [func]          Added statistics counters to track the sizes
                        of incoming queries and outgoing responses in
                        histogram buckets, as specified in RSSAC002.
                        [RT #39049]

4155.   [func]          Allow RPZ rewrite logging to be configured on a
                        per-zone basis using a newly introduced log clause in
                        the response-policy option. [RT #39754]

4154.   [bug]           A OPT record should be included with the FORMERR
                        response when there is a malformed EDNS option.
                        [RT #39647]

4153.   [bug]           Dig should zero non significant +subnet bits.  Check
                        that non significant ECS bits are zero on receipt.
                        [RT #39647]

4152.   [func]          Implement DNS COOKIE option.  This replaces the
                        experimental SIT option of BIND 9.10.  The following
                        named.conf directives are available: send-cookie,
                        cookie-secret, cookie-algorithm, nocookie-udp-size
                        and require-server-cookie.  The following dig options
                        are available: +[no]cookie[=value] and +[no]badcookie.
                        [RT #39928]

4151.   [bug]           'rndc flush' could cause a deadlock. [RT #39835]

4150.   [bug]           win32: listen-on-v6 { any; }; was not working.  Apply
                        minimal fix.  [RT #39667]

4149.   [bug]           Fixed a race condition in the getaddrinfo()
                        implementation in libirs, which caused the delv
                        utility to crash with an assertion failure when using
                        the '@server' syntax with a hostname argument.
                        [RT #39899]

4148.   [bug]           Fix a bug when printing zone names with '/' character
                        in XML and JSON statistics output. [RT #39873]

4147.   [bug]           Filter-aaaa / filter-aaaa-on-v4 / filter-aaaa-on-v6
                        was returning referrals rather than nodata responses
                        when the AAAA records were filtered.  [RT #39843]

4146.   [bug]           Address reference leak that could prevent a clean
                        shutdown. [RT #37125]
```

(continues on next page)

(continued from previous page)

```
4145.   [bug]           Not all unassociated adb entries where being printed.
                        [RT #37125]

4144.   [func]          Add statistics counters for nxdomain redirections.
                        [RT #39790]

4143.   [placeholder]

4142.   [bug]           rndc addzone with view specified saved NZF config
                        that could not be read back by named. This has now
                        been fixed. [RT #39845]

4141.   [bug]           A formatting bug caused rndc zonestatus to print
                        negative numbers for large serial values. This has
                        now been fixed. [RT #39854]

4140.   [cleanup]       Remove redundant nzf_remove() call during delzone.
                        [RT #39844]

4139.   [doc]           Fix rpz-client-ip documentation. [RT #39783]

4138.   [security]      An uninitialized value in validator.c could result
                        in an assertion failure. (CVE-2015-4620) [RT #39795]

4137.   [bug]           Make rndc reconfig report configuration errors the
                        same way rndc reload does. [RT #39635]

4136.   [bug]           Stale statistics counters with the leading
                        '#' prefix (such as #NXDOMAIN) were not being
                        updated correctly. This has been fixed. [RT #39141]

4135.   [cleanup]       Log expired NTA at startup. [RT #39680]

4134.   [cleanup]       Include client-ip rules when logging the number
                        of RPZ rules of each type. [RT #39670]

4133.   [port]          Update how various json libraries are handled.
                        [RT #39646]

4132.   [cleanup]       dig: added +rd as a synonym for +recurse,
                        added +class as an unabbreviated alternative
                        to +cl. [RT #39686]

4131.   [bug]           Addressed further problems with reloading RPZ
                        zones. [RT #39649]

4130.   [bug]           The compatibility shim for *printf() misprinted some
                        large numbers. [RT #39586]

4129.   [port]          Address API changes in OpenSSL 1.1.0. [RT #39532]
```

(continues on next page)

(continued from previous page)

```
4128.   [bug]           Address issues raised by Coverity 7.6. [RT #39537]

4127.   [protocol]      CDS and CDNSKEY need to be signed by the key signing
                        key as per RFC 7344, Section 4.1. [RT #37215]

4126.   [bug]           Addressed a regression introduced in change #4121.
                        [RT #39611]

4125.   [test]          Added tests for dig, renamed delv test to digdelv.
                        [RT #39490]

4124.   [func]          Log errors or warnings encountered when parsing the
                        internal default configuration.  Clarify the logging
                        of errors and warnings encountered in rndc
                        addzone or modzone parameters. [RT #39440]

4123.   [port]          Added %z (size_t) format options to the portable
                        internal printf/sprintf implementation. [RT #39586]

4122.   [bug]           The server could match a shorter prefix than what was
                        available in CLIENT-IP policy triggers, and so, an
                        unexpected action could be taken. This has been
                        corrected. [RT #39481]

4121.   [bug]           On servers with one or more policy zones
                        configured as slaves, if a policy zone updated
                        during regular operation (rather than at
                        startup) using a full zone reload, such as via
                        AXFR, a bug could allow the RPZ summary data to
                        fall out of sync, potentially leading to an
                        assertion failure in rpz.c when further
                        incremental updates were made to the zone, such
                        as via IXFR. [RT #39567]

4120.   [bug]           A bug in RPZ could cause the server to crash if
                        policy zones were updated while recursion was
                        pending for RPZ processing of an active query.
                        [RT #39415]

4119.   [test]          Allow dig to set the message opcode. [RT #39550]

4118.   [bug]           Teach isc-config.sh about irs. [RT #39213]

4117.   [protocol]      Add EMPTY.AS112.ARPA as per RFC 7534.

4116.   [bug]           Fix a bug in RPZ that could cause some policy
                        zones that did not specifically require
                        recursion to be treated as if they did;
                        consequently, setting qname-wait-recurse no; was
                        sometimes ineffective. [RT #39229]

4115.   [func]          "rndc -r" now prints the result code (e.g.,
```

(continues on next page)

(continued from previous page)

```
                        ISC_R_SUCCESS, ISC_R_TIMEOUT, etc) after
                        running the requested command. [RT #38913]

4114.   [bug]           Fix a regression in radix tree implementation
                        introduced by ECS code. This bug was never
                        released, but it was reported by a user testing
                        master. [RT #38983]

4113.   [test]          Check for Net::DNS is some system test
                        prerequisites. [RT #39369]

4112.   [bug]           Named failed to load when "root-delegation-only"
                        was used without a list of domains to exclude.
                        [RT #39380]

4111.   [doc]           Alphabetize rndc man page. [RT #39360]

4110.   [bug]           Address memory leaks / null pointer dereferences
                        on out of memory. [RT #39310]

4109.   [port]          linux: support reading the local port range from
                        net.ipv4.ip_local_port_range. [RT # 39379]

4108.   [func]          An additional NXDOMAIN redirect method (option
                        "nxdomain-redirect") has been added, allowing
                        redirection to a specified DNS namespace instead
                        of a single redirect zone. [RT #37989]

4107.   [bug]           Address potential deadlock when updating zone content.
                        [RT #39269]

4106.   [port]          Improve readline support. [RT #38938]

4105.   [port]          Misc fixes for Microsoft Visual Studio
                        2015 CTP6 in 64 bit mode. [RT #39308]

4104.   [bug]           Address uninitialized elements. [RT #39252]

4103.   [port]          Misc fixes for Microsoft Visual Studio
                        2015 CTP6. [RT #39267]

4102.   [bug]           Fix a use after free bug introduced in change
                        #4094.  [RT #39281]

4101.   [bug]           dig: the +split and +rrcomments options didn't
                        work with +short. [RT #39291]

4100.   [bug]           Inherited owernames on the line immediately following
                        a $INCLUDE were not working.  [RT #39268]

4099.   [port]          clang: make unknown commandline options hard errors
                        when determining what options are supported.
```

(continues on next page)

(continued from previous page)

```
                        [RT #39273]

4098.   [bug]           Address use-after-free issue when using a
                        predecessor key with dnssec-settime. [RT #39272]

4097.   [func]          Add additional logging about xfrin transfer status.
                        [RT #39170]

4096.   [bug]           Fix a use after free of query->sendevent.
                        [RT #39132]

4095.   [bug]           zone->options2 was not being properly initialized.
                        [RT #39228]

4094.   [bug]           A race during shutdown or reconfiguration could
                        cause an assertion in mem.c. [RT #38979]

4093.   [func]          Dig now learns the SIT value from truncated
                        responses when it retries over TCP. [RT #39047]

4092.   [bug]           'in-view' didn't work for zones beneath a empty zone.
                        [RT #39173]

4091.   [cleanup]       Some cleanups in isc mem code. [RT #38896]

4090.   [bug]           Fix a crash while parsing malformed CAA RRs in
                        presentation format, i.e., from text such as
                        from master files. Thanks to John Van de
                        Meulebrouck Brendgard for discovering and
                        reporting this problem. [RT #39003]

4089.   [bug]           Send notifies immediately for slave zones during
                        startup. [RT #38843]

4088.   [port]          Fixed errors when building with libressl. [RT #38899]

4087.   [bug]           Fix a crash due to use-after-free due to sequencing
                        of tasks actions. [RT #38495]

4086.   [bug]           Fix out-of-srcdir build with native pkcs11. [RT #38831]

4085.   [bug]           ISC_PLATFORM_HAVEXADDQ could be inconsistently set.
                        [RT #38828]

4084.   [bug]           Fix a possible race in updating stats counters.
                        [RT #38826]

4083.   [cleanup]       Print the number of CPUs and UDP listeners
                        consistently in the log and in "rndc status"
                        output; indicate whether threads are supported
                        in "named -V" output. [RT #38811]
```

(continues on next page)

(continued from previous page)

```
4082.   [bug]           Incrementally sign large inline zone deltas.
                        [RT #37927]

4081.   [cleanup]       Use dns_rdatalist_init consistently. [RT #38759]

4080.   [func]          Completed change #4022, adding a "lock-file" option
                        to named.conf to override the default lock file,
                        in addition to the "named -X <filename>" command
                        line option.  Setting the lock file to "none"
                        using either method disables the check completely.
                        [RT #37908]

4079.   [func]          Preserve the case of the owner name of records to
                        the RRset level. [RT #37442]

4078.   [bug]           Handle the case where CMSG_SPACE(sizeof(int)) !=
                        CMSG_SPACE(sizeof(char)). [RT #38621]

4077.   [test]          Add static-stub regression test for DS NXDOMAIN
                        return making the static stub disappear. [RT #38564]

4076.   [bug]           Named could crash on shutdown with outstanding
                        reload / reconfig events. [RT #38622]

4075.   [placeholder]

4074.   [cleanup]       Cleaned up more warnings from gcc -Wshadow. [RT #38708]

4073.   [cleanup]       Add libjson-c version number reporting to
                        "named -V"; normalize version number formatting.
                        [RT #38056]

4072.   [func]          Add a --enable-querytrace configure switch for
                        very verbose query trace logging. (This option
                        has a negative performance impact and should be
                        used only for debugging.) [RT #37520]

4071.   [cleanup]       Initialize pthread mutex attrs just once, instead of
                        doing it per mutex creation. [RT #38547]

4070.   [bug]           Fix a segfault in nslookup in a query such as
                        "nslookup isc.org AMS.SNS-PB.ISC.ORG -all".
                        [RT #38548]

4069.   [doc]           Reorganize options in the nsupdate man page.
                        [RT #38515]

4068.   [bug]           Omit unknown serial number from JSON zone statistics.
                        [RT #38604]

4067.   [cleanup]       Reduce noise from RRL when query logging is
                        disabled. [RT #38648]
```

(continues on next page)

(continued from previous page)

```
4066.   [doc]           Reorganize options in the dig man page. [RT #38516]

4065.   [test]          Additional RFC 5011 tests. [RT #38569]

4064.   [contrib]       dnssec-keyset.sh: Generates a specified number
                        of DNSSEC keys with timing set to implement a
                        pre-publication key rollover strategy. Thanks
                        to Jeffry A. Spain. [RT #38459]

4063.   [bug]           Asynchronous zone loads were not handled
                        correctly when the zone load was already in
                        progress; this could trigger a crash in zt.c.
                        [RT #37573]

4062.   [bug]           Fix an out-of-bounds read in RPZ code. If the
                        read succeeded, it doesn't result in a bug
                        during operation. If the read failed, named
                        could segfault. [RT #38559]

4061.   [bug]           Handle timeout in legacy system test. [RT #38573]

4060.   [bug]           dns_rdata_freestruct could be called on a
                        uninitialized structure when handling a error.
                        [RT #38568]

4059.   [bug]           Addressed valgrind warnings. [RT #38549]

4058.   [bug]           UDP dispatches could use the wrong pseudorandom
                        number generator context. [RT #38578]

4057.   [bug]           'dnssec-dsfromkey -T 0' failed to add ttl field.
                        [RT #38565]

4056.   [bug]           Expanded automatic testing of trust anchor
                        management and fixed several small bugs including
                        a memory leak and a possible loss of key state
                        information. [RT #38458]

4055.   [func]          "rndc managed-keys" can be used to check status
                        of trust anchors or to force keys to be refreshed,
                        Also, the managed keys data file has easier-to-read
                        comments.  [RT #38458]

4054.   [func]          Added a new tool 'mdig', a lightweight clone of
                        dig able to send multiple pipelined queries.
                        [RT #38261]

4053.   [security]      Revoking a managed trust anchor and supplying
                        an untrusted replacement could cause named
                        to crash with an assertion failure.
                        (CVE-2015-1349) [RT #38344]
```

(continues on next page)

(continued from previous page)

```
4052.   [bug]           Fix a leak of query fetchlock. [RT #38454]

4051.   [bug]           Fix a leak of pthread_mutexattr_t. [RT #38454]

4050.   [bug]           RPZ could send spurious SERVFAILs in response
                        to duplicate queries. [RT #38510]

4049.   [bug]           CDS and CDNSKEY had the wrong attributes. [RT #38491]

4048.   [bug]           adb hash table was not being grown. [RT #38470]

4047.   [cleanup]       "named -V" now reports the current running versions
                        of OpenSSL and the libxml2 libraries, in addition to
                        the versions that were in use at build time.

4046.   [bug]           Accounting of "total use" in memory context
                        statistics was not correct. [RT #38370]

4045.   [bug]           Skip to next master on dns_request_createvia4 failure.
                        [RT #25185]

4044.   [bug]           Change 3955 was not complete, resulting in an assertion
                        failure if the timing was just right. [RT #38352]

4043.   [func]          "rndc modzone" can be used to modify the
                        configuration of an existing zone, using similar
                        syntax to "rndc addzone". [RT #37895]

4042.   [bug]           zone.c:iszonesecure was being called too late.
                        [RT #38371]

4041.   [func]          TCP sockets can now be shared while connecting.
                        (This will be used to enable client-side support
                        of pipelined queries.) [RT #38231]

4040.   [func]          Added server-side support for pipelined TCP
                        queries. Clients may continue sending queries via
                        TCP while previous queries are being processed
                        in parallel.  (The new "keep-response-order"
                        option allows clients to be specified for which
                        the old behavior will still be used.) [RT #37821]

4039.   [cleanup]       Cleaned up warnings from gcc -Wshadow. [RT #37381]

4038.   [bug]           Add 'rpz' flag to node and use it to determine whether
                        to call dns_rpz_delete.  This should prevent unbalanced
                        add / delete calls. [RT #36888]

4037.   [bug]           also-notify was ignoring the tsig key when checking
                        for duplicates resulting in some expected notify
                        messages not being sent. [RT #38369]
```

(continues on next page)

(continued from previous page)

```
4036.   [bug]           Make call to open a temporary file name safe during
                        NZF creation. [RT #38331]

4035.   [bug]           Close temporary and NZF FILE pointers before moving
                        the former into the latter's place, as required on
                        Windows. [RT #38332]

4034.   [func]          When added, negative trust anchors (NTA) are now
                        saved to files (viewname.nta), in order to
                        persist across restarts of the named server.
                        [RT #37087]

4033.   [bug]           Missing out of memory check in request.c:req_send.
                        [RT #38311]

4032.   [bug]           Built-in "empty" zones did not correctly inherit the
                        "allow-transfer" ACL from the options or view.
                        [RT #38310]

4031.   [bug]           named-checkconf -z failed to report a missing file
                        with a hint zone. [RT #38294]

4030.   [func]          "rndc delzone" is now applicable to zones that were
                        configured in named.conf, as well as zones that
                        were added via "rndc addzone". (Note, however, that
                        if named.conf is not also modified, the deleted zone
                        will return when named is reloaded.) [RT #37887]

4029.   [func]          "rndc showzone" displays the current configuration
                        of a specified zone. [RT #37887]

4028.   [bug]           $GENERATE with a zero step was not being caught as a
                        error.  A $GENERATE with a / but no step was not being
                        caught as a error. [RT #38262]

4027.   [port]          Net::DNS 0.81 compatibility. [RT #38165]

4026.   [bug]           Fix RFC 3658 reference in dig +sigchase. [RT #38173]

4025.   [port]          bsdi: failed to build. [RT #38047]

4024.   [bug]           dns_rdata_opt_first, dns_rdata_opt_next,
                        dns_rdata_opt_current, dns_rdata_txt_first,
                        dns_rdata_txt_next and dns_rdata_txt_current were
                        documented but not implemented.  These have now been
                        implemented.

                        dns_rdata_spf_first, dns_rdata_spf_next and
                        dns_rdata_spf_current were documented but not
                        implemented.  The prototypes for these
                        functions have been removed. [RT #38068]
```

(continues on next page)

(continued from previous page)

```
4023.   [bug]           win32: socket handling with explicit ports and
                        invoking named with -4 was broken for some
                        configurations. [RT #38068]

4022.   [func]          Stop multiple spawns of named by limiting number of
                        processes to 1. This is done by using a lockfile and
                        checking whether we can listen on any configured
                        TCP interfaces. [RT #37908]

4021.   [bug]           Adjust max-recursion-queries to accommodate
                        the need for more queries when the cache is
                        empty. [RT #38104]

4020.   [bug]           Change 3736 broke nsupdate's SOA MNAME discovery
                        resulting in updates being sent to the wrong server.
                        [RT #37925]

4019.   [func]          If named is not configured to validate the answer
                        then allow fallback to plain DNS on timeout even
                        when we know the server supports EDNS. [RT #37978]

4018.   [placeholder]

4017.   [test]          Add system test to check lookups to legacy servers
                        with broken DNS behavior. [RT #37965]

4016.   [bug]           Fix a dig segfault due to bad linked list usage.
                        [RT #37591]

4015.   [bug]           Nameservers that are skipped due to them being
                        CNAMEs were not being logged. They are now logged
                        to category 'cname' as per BIND 8. [RT #37935]

4014.   [bug]           When including a master file origin_changed was
                        not being properly set leading to a potentially
                        spurious 'inherited owner' warning. [RT #37919]

4013.   [func]          Add a new tcp-only option to server (config) /
                        peer (struct) to use TCP transport to send
                        queries (in place of UDP transport with a
                        TCP fallback on truncated (TC set) response).
                        [RT #37800]

4012.   [cleanup]       Check returned status of OpenSSL digest and HMAC
                        functions when they return one. Note this applies
                        only to FIPS capable OpenSSL libraries put in
                        FIPS mode and MD5. [RT #37944]

4011.   [bug]           master's list port and dscp inheritance was not
                        properly implemented. [RT #37792]
```

(continues on next page)

(continued from previous page)

```
4010.   [cleanup]       Clear the prefetchable state when initiating a
                        prefetch. [RT #37399]

4009.   [func]          delv: added a +tcp option. [RT #37855]

4008.   [contrib]       Updated zkt to latest version (1.1.3). [RT #37886]

4007.   [doc]           Remove acl forward reference restriction. [RT #37772]

4006.   [security]      A flaw in delegation handling could be exploited
                        to put named into an infinite loop.  This has
                        been addressed by placing limits on the number
                        of levels of recursion named will allow (default 7),
                        and the number of iterative queries that it will
                        send (default 50) before terminating a recursive
                        query (CVE-2014-8500).

                        The recursion depth limit is configured via the
                        "max-recursion-depth" option, and the query limit
                        via the "max-recursion-queries" option.  [RT #37580]

4005.   [func]          The buffer used for returning text from rndc
                        commands is now dynamically resizable, allowing
                        arbitrarily large amounts of text to be sent back
                        to the client. (Prior to this change, it was
                        possible for the output of "rndc tsig-list" to be
                        truncated.) [RT #37731]

4004.   [bug]           When delegations had AAAA glue but not A, a
                        reference could be leaked causing an assertion
                        failure on shutdown. [RT #37796]

4003.   [security]      When geoip-directory was reconfigured during
                        named run-time, the previously loaded GeoIP
                        data could remain, potentially causing wrong
                        ACLs to be used or wrong results to be served
                        based on geolocation (CVE-2014-8680). [RT #37720]

4002.   [security]      Lookups in GeoIP databases that were not
                        loaded could cause an assertion failure
                        (CVE-2014-8680). [RT #37679]

4001.   [security]      The caching of GeoIP lookups did not always
                        handle address families correctly, potentially
                        resulting in an assertion failure (CVE-2014-8680).
                        [RT #37672]

4000.   [bug]           NXDOMAIN redirection incorrectly handled NXRRSET
                        from the redirect zone. [RT #37722]
```

```
3999.   [func]          "mkeys" and "nzf" files are now named after
                        their corresponding views, unless the view name
```

(continues on next page)

(continued from previous page)

```
                        contains characters that would be incompatible
                        with use in a filename (i.e., slash, backslash,
                        or capital letters). If a view name does contain
                        these characters, the files will still be named
                        using a cryptographic hash of the view name.
                        Regardless of this, if a file using the old name
                        format is found to exist, it will continue to be
                        used. [RT #37704]

3998.   [bug]           isc_radix_search was returning matches that were
                        too precise. [RT #37680]

3997.   [protocol]      Add OPENGPGKEY record. [RT# 37671]

3996.   [bug]           Address use after free on out of memory error in
                        keyring_add. [RT #37639]

3995.   [bug]           receive_secure_serial holds the zone lock for too
                        long. [RT #37626]

3994.   [func]          Dig now supports setting the last unassigned DNS
                        header flag bit (dig +zflag). [RT #37421]

3993.   [func]          Dig now supports EDNS negotiation by default.
                        (dig +[no]ednsnegotiation).

                        Note:  This is disabled by default in BIND 9.10
                        and enabled by default in BIND 9.11.  [RT #37604]

3992.   [func]          DiG can now send queries without questions
                        (dig +header-only). [RT #37599]

3991.   [func]          Add the ability to buffer logging output by specifying
                        "buffered yes;" when defining a channel. [RT #26561]

3990.   [test]          Add tests for unknown DNSSEC algorithm handling.
                        [RT #37541]

3989.   [cleanup]       Remove redundant dns_db_resigned calls. [RT #35748]

3988.   [func]          Allow the zone serial of a dynamically updatable
                        zone to be updated via "rndc signing -serial".
                        [RT #37404]

3987.   [port]          Handle future Visual Studio 14 incompatible changes.
                        [RT #37380]

3986.   [doc]           Add the BIND version number to page footers
                        in the ARM. [RT #37398]

3985.   [doc]           Describe how +ndots and +search interact in dig.
                        [RT #37529]
```

(continues on next page)

(continued from previous page)

```
3984.   [func]          Accept 256 byte long PINs in native PKCS#11
                        crypto. [RT #37410]

3983.   [bug]           Change #3940 was incomplete: negative trust anchors
                        could be set to last up to a week, but the
                        "nta-lifetime" and "nta-recheck" options were
                        still limited to one day. [RT #37522]

3982.   [doc]           Include release notes in product documentation.
                        [RT #37272]

3981.   [bug]           Cache DS/NXDOMAIN independently of other query types.
                        [RT #37467]

3980.   [bug]           Improve --with-tuning=large by self tuning of SO_RCVBUF
                        size. [RT #37187]

3979.   [bug]           Negative trust anchor fetches were not properly
                        managed. [RT #37488]

3978.   [test]          Added a unit test for Diffie-Hellman key
                        computation, completing change #3974. [RT #37477]

3977.   [cleanup]       "rndc secroots" reported a "not found" error when
                        there were no negative trust anchors set. [RT #37506]

3976.   [bug]           When refreshing managed-key trust anchors, clear
                        any cached trust so that they will always be
                        revalidated with the current set of secure
                        roots. [RT #37506]

3975.   [bug]           Don't populate or use the bad cache for queries that
                        don't request or use recursion. [RT #37466]

3974.   [bug]           Handle DH_compute_key() failure correctly in
                        openssldh_link.c. [RT #37477]

3973.   [test]          Added hooks for Google Performance Tools CPU profiler,
                        including real-time/wall-clock profiling. Use
                        "configure --with-gperftools-profiler" to enable.
                        [RT #37339]

3972.   [bug]           Fix host's usage statement. [RT #37397]

3971.   [bug]           Reduce the cascading failures due to a bad $TTL line
                        in named-checkconf / named-checkzone. [RT #37138]

3970.   [contrib]       Fixed a use after free bug in the SDB LDAP driver.
                        [RT #37237]

3969.   [test]          Added 'delv' system test. [RT #36901]
```

(continues on next page)

(continued from previous page)

```
3968.   [bug]           Silence spurious log messages when using 'named -[46]'.
                        [RT #37308]

3967.   [test]          Add test for inlined signed zone in multiple views
                        with different DNSKEY sets. [RT #35759]

3966.   [bug]           Missing dns_db_closeversion call in receive_secure_db.
                        [RT #35746]

3965.   [func]          Log outgoing packets and improve packet logging to
                        support logging the remote address. [RT #36624]

3964.   [func]          nsupdate now performs check-names processing.
                        [RT #36266]

3963.   [test]          Added NXRRSET test cases to the "dlzexternal"
                        system test. [RT #37344]

3962.   [bug]           'dig +topdown +trace +sigchase' address unhandled error
                        conditions. [RT #34663]

3961.   [bug]           Forwarding of SIG(0) signed UPDATE messages failed with
                        BADSIG.  [RT #37216]

3960.   [bug]           'dig +sigchase' could loop forever. [RT #37220]

3959.   [bug]           Updates could be lost if they arrived immediately
                        after a rndc thaw. [RT #37233]

3958.   [bug]           Detect when writeable files have multiple references
                        in named.conf. [RT #37172]

3957.   [bug]           "dnssec-keygen -S" failed for ECCGOST, ECDSAP256SHA256
                        and ECDSAP384SHA384. [RT #37183]

3956.   [func]          Notify messages are now rate limited by notify-rate and
                        startup-notify-rate instead of serial-query-rate.
                        [RT #24454]

3955.   [bug]           Notify messages due to changes are no longer queued
                        behind startup notify messages. [RT #24454]

3954.   [bug]           Unchecked mutex init in dlz_dlopen_driver.c [RT #37112]

3953.   [bug]           Don't escape semi-colon in TXT fields. [RT #37159]

3952.   [bug]           dns_name_fullcompare failed to set *nlabelsp when the
                        two name pointers were the same. [RT #37176]

3951.   [func]          Add the ability to set yet-to-be-defined EDNS flags
                        to dig (+ednsflags=#). [RT #37142]
```

(continues on next page)

(continued from previous page)

```
3950.   [port]          Changed the bin/python Makefile to work around a
                        bmake bug in FreeBSD 10 and NetBSD 6. [RT #36993]

3949.   [experimental]  Experimental support for draft-andrews-edns1 by sending
                        EDNS(1) queries (define DRAFT_ANDREWS_EDNS1 when
                        building).  Add support for limiting the EDNS version
                        advertised to servers: server { edns-version 0; };
                        Log the EDNS version received in the query log.
                        [RT #35864]

3948.   [port]          solaris: RCVBUFSIZE was too large on Solaris with
                        --with-tuning=large. [RT #37059]

3947.   [cleanup]       Set the executable bit on libraries when using
                        libtool. [RT #36786]

3946.   [cleanup]       Improved "configure" search for a python interpreter.
                        [RT #36992]

3945.   [bug]           Invalid wildcard expansions could be incorrectly
                        accepted by the validator. [RT #37093]

3944.   [test]          Added a regression test for "server-id". [RT #37057]

3943.   [func]          SERVFAIL responses can now be cached for a
                        limited time (configured by "servfail-ttl",
                        default 10 seconds, limit 30). This can reduce
                        the frequency of retries when an authoritative
                        server is known to be failing, e.g., due to
                        ongoing DNSSEC validation problems. [RT #21347]

3942.   [bug]           Wildcard responses from a optout range should be
                        marked as insecure. [RT #37072]

3941.   [doc]           Include the BIND version number in the ARM. [RT #37067]

3940.   [func]          "rndc nta" now allows negative trust anchors to be
                        set for up to one week. [RT #37069]

3939.   [func]          Improve UPDATE forwarding performance by allowing TCP
                        connections to be shared. [RT #37039]

3938.   [func]          Added quotas to be used in recursive resolvers
                        that are under high query load for names in zones
                        whose authoritative servers are nonresponsive or
                        are experiencing a denial of service attack.

                        - "fetches-per-server" limits the number of
                          simultaneous queries that can be sent to any
                          single authoritative server.  The configured
                          value is a starting point; it is automatically
```

(continues on next page)

(continued from previous page)

```
                        adjusted downward if the server is partially or
                        completely non-responsive. The algorithm used to
                        adjust the quota can be configured via the
                        "fetch-quota-params" option.
                      - "fetches-per-zone" limits the number of
                        simultaneous queries that can be sent for names
                        within a single domain.  (Note: Unlike
                        "fetches-per-server", this value is not
                        self-tuning.)
                      - New stats counters have been added to count
                        queries spilled due to these quotas.

                      See the ARM for details of these options. [RT #37125]

3937.   [func]        Added some debug logging to better indicate the
                      conditions causing SERVFAILs when resolving.
                      [RT #35538]

3936.   [func]        Added authoritative support for the EDNS Client
                      Subnet (ECS) option.

                      ACLs can now include "ecs" elements which specify
                      an address or network prefix; if an ECS option is
                      included in a DNS query, then the address encoded
                      in the option will be matched against "ecs" ACL
                      elements.

                      Also, if an ECS address is included in a query,
                      then it will be used instead of the client source
                      address when matching "geoip" ACL elements.  This
                      behavior can be overridden with "geoip-use-ecs no;".
                      (Note: to enable "geoip" ACLs, use "configure
                      --with-geoip". This requires libGeoIP version
                      1.5.0 or higher.)

                      When "ecs" or "geoip" ACL elements are used to
                      select a view for a query, the response will include
                      an ECS option to indicate which client network the
                      answer is valid for.

                      (Thanks to Vincent Bernat.) [RT #36781]

3935.   [bug]         "geoip asnum" ACL elements would not match unless
                      the full organization name was specified.  They
                      can now match against the AS number alone (e.g.,
                      AS1234). [RT #36945]

3934.   [bug]         Catch bad 'sit-secret' in named-checkconf.  Improve
                      sit-secret documentation. [RT #36980]

3933.   [bug]         Corrected the implementation of dns_rdata_casecompare()
                      for the HIP rdata type.  [RT #36911]
```

(continues on next page)

(continued from previous page)

```
3932.   [test]          Improved named-checkconf tests. [RT #36911]

3931.   [cleanup]       Cleanup how dlz grammar is defined. [RT #36879]

3930.   [bug]           "rndc nta -r" could cause a server hang if the
                        NTA was not found. [RT #36909]

3929.   [bug]           'host -a' needed to clear idnoptions. [RT #36963]

3928.   [test]          Improve rndc system test. [RT #36898]

3927.   [bug]           dig: report PKCS#11 error codes correctly when
                        compiled with --enable-native-pkcs11. [RT #36956]

3926.   [doc]           Added doc for geoip-directory. [RT #36877]

3925.   [bug]           DS lookup of RFC 1918 empty zones failed. [RT #36917]

3924.   [bug]           Improve 'rndc addzone' error reporting. [RT #35187]

3923.   [bug]           Sanity check the xml2-config output. [RT #22246]

3922.   [bug]           When resigning, dnssec-signzone was removing
                        all signatures from delegation nodes. It now
                        retains DS and (if applicable) NSEC signatures.
                        [RT #36946]

3921.   [bug]           AD was inappropriately set on RPZ responses. [RT #36833]

3920.   [doc]           Added doc for masterfile-style. [RT #36823]

3919.   [bug]           dig: continue to next line if a address lookup fails
                        in batch mode. [RT #36755]

3918.   [doc]           Update check-spf documentation. [RT #36910]

3917.   [bug]           dig, nslookup and host now continue on names that are
                        too long after applying a search list elements.
                        [RT #36892]

3916.   [contrib]       zone2sqlite checked wrong result code.  Address
                        compiler warnings. [RT #36931]

3915.   [bug]           Address a assertion if a route event arrived while
                        shutting down. [RT #36887]

3914.   [bug]           Allow the URI target and CAA value fields to
                        be zero length. [RT #36737]

3913.   [bug]           Address race issue in dispatch. [RT #36731]
```

(continues on next page)

(continued from previous page)

```
3912.   [bug]           Address some unrecoverable lookup failures. [RT #36330]

3911.   [func]          Implement EDNS EXPIRE option client side, allowing
                        a slave server to set the expiration timer correctly
                        when transferring zone data from another slave
                        server. [RT #35925]

3910.   [bug]           Fix races to free event during shutdown. [RT #36720]

3909.   [bug]           When computing the number of elements required for a
                        acl count_acl_elements could have a short count leading
                        to a assertion failure.  Also zero out new acl elements
                        in dns_acl_merge.  [RT #36675]

3908.   [bug]           rndc now differentiates between a zone in multiple
                        views and a zone that doesn't exist at all. [RT #36691]

3907.   [cleanup]       Alphabetize rndc help. [RT #36683]

3906.   [protocol]      Update URI record format to comply with
                        draft-faltstrom-uri-08. [RT #36642]

3905.   [bug]           Address deadlock between view.c and adb.c. [RT #36341]

3904.   [func]          Add the RPZ SOA to the additional section. [RT36507]

3903.   [bug]           Improve the accuracy of DiG's reported round trip
                        time. [RT 36611]

3902.   [bug]           liblwres wasn't handling link-local addresses in
                        nameserver clauses in resolv.conf. [RT #36039]

3901.   [protocol]      Added support for CAA record type (RFC 6844).
                        [RT #36625]

3900.   [bug]           Fix a crash in PostgreSQL DLZ driver. [RT #36637]

3899.   [bug]           "request-ixfr" is only applicable to slave and redirect
                        zones. [RT #36608]

3898.   [bug]           Too small a buffer in tohexstr() calls in test code.
                        [RT #36598]

3897.   [bug]           RPZ summary information was not properly being updated
                        after a AXFR resulting in changes sometimes being
                        ignored.  [RT #35885]

3896.   [bug]           Address performance issues with DSCP code on some
                        platforms. [RT #36534]

3895.   [func]          Add the ability to set the DSCP code point to dig.
                        [RT #36546]
```

(continues on next page)

(continued from previous page)

```
3894.   [bug]           Buffers in isc_print_vsnprintf were not properly
                        initialized leading to potential overflows when
                        printing out quad values. [RT #36505]

3893.   [bug]           Peer DSCP values could be returned without being set.
                        [RT #36538]

3892.   [bug]           Setting '-t aaaa' in .digrc had unintended side
                        effects. [RT #36452]

3891.   [bug]           Use ${INSTALL_SCRIPT} rather than ${INSTALL_PROGRAM}
                        to install python programs.

3890.   [bug]           RRSIG sets that were not loaded in a single transaction
                        at start up where not being correctly added to
                        re-signing heaps.  [RT #36302]

3889.   [port]          hurd: configure fixes as per:
                        https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=746540

3888.   [func]          'rndc status' now reports the number of automatic
                        zones. [RT #36015]

3887.   [cleanup]       Make all static symbols in rbtdb64 end in "64" so
                        they are easier to use in a debugger. [RT #36373]

3886.   [bug]           rbtdb_write_header should use a once to initialize
                        FILE_VERSION. [RT #36374]

3885.   [port]          Use 'open()' rather than 'file()' to open files in
                        python.

3884.   [protocol]      Add CDS and CDNSKEY record types. [RT #36333]

3883.   [placeholder]

3882.   [func]          By default, negative trust anchors will be tested
                        periodically to see whether data below them can be
                        validated, and if so, they will be allowed to
                        expire early. The "rndc nta -force" option
                        overrides this behavior.  The default NTA lifetime
                        and the recheck frequency can be configured by the
                        "nta-lifetime" and "nta-recheck" options. [RT #36146]

3881.   [bug]           Address memory leak with UPDATE error handling.
                        [RT #36303]

3880.   [test]          Update ans.pl to work with new TSIG support in
                        Net::DNS; add additional Net::DNS version prerequisite
                        checks. [RT #36327]
```

(continues on next page)

(continued from previous page)

```
3879.   [func]          Add version printing option to various BIND utilities.
                        [RT #10686]

3878.   [bug]           Using the incorrect filename for a DLZ module
                        caused a segmentation fault on startup. [RT #36286]

3877.   [bug]           Inserting and deleting parent and child nodes
                        in response policy zones could trigger an assertion
                        failure. [RT #36272]

3876.   [bug]           Improve efficiency of DLZ redirect zones by
                        suppressing unnecessary database lookups. [RT #35835]

3875.   [cleanup]       Clarify log message when unable to read private
                        key files. [RT #24702]

3874.   [test]          Check that only "check-names master" is needed for
                        updates to be accepted.

3873.   [protocol]      Only warn for SPF without TXT spf record. [RT #36210]

3872.   [bug]           Address issues found by static analysis. [RT #36209]

3871.   [bug]           Don't publish an activated key automatically before
                        its publish time. [RT #35063]

3870.   [func]          Updated the random number generator used in
                        the resolver to use the updated ChaCha based one
                        (similar to OpenBSD's changes). Also moved the
                        RNG to libisc and added unit tests for it.
                        [RT #35942]

3869.   [doc]           Document that in-view zones cannot be used for
                        response policy zones. [RT #35941]

3868.   [bug]           isc_mem_setwater incorrectly cleared hi_called
                        potentially leaving over memory cleaner running.
                        [RT #35270]

3867.   [func]          "rndc nta" can now be used to set a temporary
                        negative trust anchor, which disables DNSSEC
                        validation below a specified name for a specified
                        period of time (not exceeding 24 hours).  This
                        can be used when validation for a domain is known
                        to be failing due to a configuration error on
                        the part of the domain owner rather than a
                        spoofing attack. [RT #29358]

3866.   [bug]           Named could die on disk full in generate_session_key.
                        [RT #36119]

3865.   [test]          Improved testability of the red-black tree
```

(continues on next page)

```
                        implementation and added unit tests. [RT #35904]

3864.   [bug]           RPZ didn't work well when being used as forwarder.
                        [RT #36060]

3863.   [bug]           The "E" flag was missing from the query log as a
                        unintended side effect of code rearrangement to
                        support EDNS EXPIRE. [RT #36117]

3862.   [cleanup]       Return immediately if we are not going to log the
                        message in ns_client_dumpmessage.

3861.   [security]      Missing isc_buffer_availablelength check results
                        in a REQUIRE assertion when printing out a packet
                        (CVE-2014-3859).  [RT #36078]

3860.   [bug]           ioctl(DP_POLL) array size needs to be determined
                        at run time as it is limited to {OPEN_MAX}.
                        [RT #35878]

3859.   [placeholder]

3858.   [bug]           Disable GCC 4.9 "delete null pointer check".
                        [RT #35968]

3857.   [bug]           Make it harder for a incorrect NOEDNS classification
                        to be made. [RT #36020]

3856.   [bug]           Configuring libjson without also configuring libxml
                        resulted in a REQUIRE assertion when retrieving
                        statistics using json. [RT #36009]

3855.   [bug]           Limit smoothed round trip time aging to no more than
                        once a second. [RT #32909]

3854.   [cleanup]       Report unrecognized options, if any, in the final
                        configure summary. [RT #36014]

3853.   [cleanup]       Refactor dns_rdataslab_fromrdataset to separate out
                        the handling of a rdataset with no records. [RT #35968]

3852.   [func]          Increase the default number of clients available
                        for servicing lightweight resolver queries, and
                        make them configurable via the "lwres-tasks" and
                        "lwres-clients" options.  (Thanks to Tomas Hozza.)
                        [RT #35857]

3851.   [func]          Allow libseccomp based system-call filtering
                        on Linux; use "configure --enable-seccomp" to
                        turn it on.  Thanks to Loganaden Velvindron
                        of AFRINIC for the contribution. [RT #35347]

3850.   [bug]           Disabling forwarding could trigger a REQUIRE assertion.
```

(continues on next page)

(continued from previous page)

```
                        [RT #35979]

3849.   [doc]           Alphabetized dig's +options. [RT #35992]

3848.   [bug]           Adjust 'statistics-channels specified but not effective'
                        error message to account for JSON support. [RT #36008]

3847.   [bug]           'configure --with-dlz-postgres' failed to fail when
                        there is not support available.

3846.   [bug]           "dig +notcp ixfr=<serial>" should result in a UDP
                        ixfr query. [RT #35980]

3845.   [placeholder]

3844.   [bug]           Use the x64 version of the Microsoft Visual C++
                        Redistributable when built for 64 bit Windows.
                        [RT #35973]

3843.   [protocol]      Check EDNS EXPIRE option in dns_rdata_fromwire.
                        [RT #35969]

3842.   [bug]           Adjust RRL log-only logging category. [RT #35945]

3841.   [cleanup]       Refactor zone.c:add_opt to use dns_message_buildopt.
                        [RT #35924]

3840.   [port]          Check for arc4random_addrandom() before using it;
                        it's been removed from OpenBSD 5.5. [RT #35907]

3839.   [test]          Use only posix-compatible shell in system tests.
                        [RT #35625]

3838.   [protocol]      EDNS EXPIRE as been assigned a code point of 9.

3837.   [security]      A NULL pointer is passed to query_prefetch resulting
                        a REQUIRE assertion failure when a fetch is actually
                        initiated (CVE-2014-3214).  [RT #35899]

3836.   [bug]           Address C++ keyword usage in header file.

3835.   [bug]           Geoip ACL elements didn't work correctly when
                        referenced via named or nested ACLs. [RT #35879]

3834.   [bug]           The re-signing heaps were not being updated soon enough
                        leading to multiple re-generations of the same RRSIG
                        when a zone transfer was in progress. [RT #35273]

3833.   [bug]           Cross compiling was broken due to calling genrandom at
                        build time. [RT #35869]

3832.   [func]          "named -L <filename>" causes named to send log
```

(continues on next page)

(continued from previous page)

```
                        messages to the specified file by default instead
                        of to the system log. (Thanks to Tony Finch.)
                        [RT #35845]

3831.   [cleanup]       Reduce logging noise when EDNS state changes occur.
                        [RT #35843]

3830.   [func]          When query logging is enabled, log query errors at
                        the same level ('info') as the queries themselves.
                        [RT #35844]

3829.   [func]          "dig +ttlunits" causes dig to print TTL values
                        with time-unit suffixes: w, d, h, m, s for
                        weeks, days, hours, minutes, and seconds. (Thanks
                        to Tony Finch.) [RT #35823]

3828.   [func]          "dnssec-signzone -N date" updates serial number
                        to the current date in YYYYMMDDNN format.
                        [RT #35800]

3827.   [placeholder]

3826.   [bug]           Corrected bad INSIST logic in isc_radix_remove().
                        [RT #35870]

3825.   [bug]           Address sign extension bug in isc_regex_validate.
                        [RT #35758]

3824.   [bug]           A collision between two flag values could cause
                        problems with cache cleaning when SIT was enabled.
                        [RT #35858]

3823.   [func]          Log the rpz cname target when rewriting. [RT #35667]

3822.   [bug]           Log the correct type of static-stub zones when
                        removing them. [RT #35842]

3821.   [contrib]       Added a new "mysqldyn" DLZ module with dynamic
                        update and transaction support. Thanks to Marty
                        Lee for the contribution. [RT #35656]

3820.   [func]          The DLZ API doesn't pass the database version to
                        the lookup() function; this can cause DLZ modules
                        that allow dynamic updates to mishandle prerequisite
                        checks. This has been corrected by adding a
                        'dbversion' field to the dns_clientinfo_t
                        structure. [RT #35656]

3819.   [bug]           NSEC3 hashes need to be able to be entered and
                        displayed without padding.  This is not a issue for
                        currently defined algorithms but may be for future
                        hash algorithms. [RT #27925]
```

(continues on next page)

(continued from previous page)

```
3818.   [bug]           Stop lying to the optimizer that 'void *arg' is a
                        constant in isc_event_allocate.

3817.   [func]          The "delve" command is now spelled "delv" to avoid
                        a namespace collision with the Xapian project.
                        [RT #35801]

3816.   [func]          "dig +qr" now reports query size. (Thanks to
                        Tony Finch.) [RT #35822]

3815.   [doc]           Clarify "nsupdate -y" usage in man page. [RT #35808]

3814.   [func]          The "masterfile-style" zone option controls the
                        formatting of dumped zone files. Options are
                        "relative" (multiline format) and "full" (one
                        record per line). The default is "relative".
                        [RT #20798]

3813.   [func]          "host" now recognizes the "timeout", "attempts" and
                        "debug" options when set in /etc/resolv.conf.
                        (Thanks to Adam Tkac at RedHat.) [RT #21885]

3812.   [func]          Dig now supports sending arbitrary EDNS options from
                        the command line (+ednsopt=code[:value]). [RT #35584]

3811.   [func]          "serial-update-method date;" sets serial number
                        on dynamic update to today's date in YYYYMMDDNN
                        format. (Thanks to Bradley Forschinger.) [RT #24903]

3810.   [bug]           Work around broken nameservers that fail to ignore
                        unknown EDNS options. [RT #35766]

3809.   [doc]           Fix SIT and NSID documentation.

3808.   [doc]           Clean up "prefetch" documentation. [RT #35751]

3807.   [bug]           Fix sign extension bug in dns_name_fromtext when
                        lowercase is set. [RT #35743]

3806.   [test]          Improved system test portability. [RT #35625]

3805.   [contrib]       Added contrib/perftcpdns, a performance testing tool
                        for DNS over TCP. [RT #35710]
```

```
        --- 9.10.0rc1 released ---

3804.   [bug]           Corrected a race condition in dispatch.c in which
                        portentry could be reset leading to an assertion
                        failure in socket_search(). (Change #3708
                        addressed the same issue but was incomplete.)
                        [RT #35128]
```

(continues on next page)

(continued from previous page)

```
3803.   [bug]           "named-checkconf -z" incorrectly rejected zones
                        using alternate data sources for not having a "file"
                        option. [RT #35685]

3802.   [bug]           Various header files were not being installed.

3801.   [port]          Fix probing for gssapi support on FreeBSD. [RT #35615]

3800.   [bug]           A pending event on the route socket could cause an
                        assertion failure when shutting down named. [RT #35674]

3799.   [bug]           Improve named's command line error reporting.
                        [RT #35603]

3798.   [bug]           'rndc zonestatus' was reporting the wrong re-signing
                        time. [RT #35659]

3797.   [port]          netbsd: geoip support probing was broken. [RT #35642]

3796.   [bug]           Register dns and pkcs#11 error codes. [RT #35629]

3795.   [bug]           Make named-checkconf detect raw masterfiles for
                        hint zones and reject them. [RT #35268]

3794.   [maint]         Added AAAA for C.ROOT-SERVERS.NET.

3793.   [bug]           zone.c:save_nsec3param() could assert when out of
                        memory. [RT #35621]

3792.   [func]          Provide links to the alternate statistics views when
                        displaying in a browser.  [RT #35605]

3791.   [placeholder]

3790.   [bug]           Handle broken nameservers that send BADVERS in
                        response to unknown EDNS options.  Maintain
                        statistics on BADVERS responses.

3789.   [bug]           Null pointer dereference on rbt creation failure.

3788.   [bug]           dns_peer_getrequestsit was returning request_nsid by
                        mistake.
```

```
        --- 9.10.0b2 released ---

3787.   [bug]           The code that checks whether "auto-dnssec" is
                        allowed was ignoring "allow-update" ACLs set at
                        the options or view level. [RT #29536]

3786.   [func]          Provide more detailed error codes when using
                        native PKCS#11. "pkcs11-tokens" now fails robustly
```

(continues on next page)

(continued from previous page)

```
                        rather than asserting when run against an HSM with
                        an incomplete PKCS#11 API implementation. [RT #35479]

3785.   [bug]           Debugging code dumphex didn't accept arbitrarily long
                        input (only compiled with -DDEBUG). [RT #35544]

3784.   [bug]           Using "rrset-order fixed" when it had not been
                        enabled at compile time caused inconsistent
                        results. It now works as documented, defaulting
                        to cyclic mode. [RT #28104]

3783.   [func]          "tsig-keygen" is now available as an alternate
                        command name for "ddns-confgen".  It generates
                        a TSIG key in named.conf format without comments.
                        [RT #35503]

3782.   [func]          Specifying "auto" as the salt when using
                        "rndc signing -nsec3param" causes named to
                        generate a 64-bit salt at random. [RT #35322]

3781.   [tuning]        Use adaptive mutex locks when available; this
                        has been found to improve performance under load
                        on many systems. "configure --with-locktype=standard"
                        restores conventional mutex locks. [RT #32576]

3780.   [bug]           $GENERATE handled negative numbers incorrectly.
                        [RT #25528]

3779.   [cleanup]       Clarify the error message when using an option
                        that was not enabled at compile time. [RT #35504]

3778.   [bug]           Log a warning when the wrong address family is
                        used in "listen-on" or "listen-on-v6". [RT #17848]

3777.   [bug]           EDNS EXPIRE code could dump core when processing
                        DLZ queries. [RT #35493]

3776.   [func]          "rndc -q" suppresses output from successful
                        rndc commands. Errors are printed on stderr.
                        [RT #21393]

3775.   [bug]           dlz_dlopen driver could return the wrong error
                        code on API version mismatch, leading to a segfault.
                        [RT #35495]

3774.   [func]          When using "request-nsid", log the NSID value in
                        printable form as well as hex. [RT #20864]

3773.   [func]          "host", "nslookup" and "nsupdate" now have
                        options to print the version number and exit.
                        [RT #26057]
```

(continues on next page)

(continued from previous page)

```
3772.   [contrib]       Added sqlite3 dynamically-loadable DLZ module.
                        (Based in part on a contribution from Tim Tessier.)
                        [RT #20822]

3771.   [cleanup]       Adjusted log level for "using built-in key"
                        messages. [RT #24383]

3770.   [bug]           "dig +trace" could fail with an assertion when it
                        needed to fall back to TCP due to a truncated
                        response. [RT #24660]

3769.   [doc]           Improved documentation of "rndc signing -list".
                        [RT #30652]

3768.   [bug]           "dnssec-checkds" was missing the SHA-384 digest
                        algorithm. [RT #34000]

3767.   [func]          Log explicitly when using rndc.key to configure
                        command channel. [RT #35316]

3766.   [cleanup]       Fixed problems with building outside the source
                        tree when using native PKCS#11. [RT #35459]

3765.   [bug]           Fixed a bug in "rndc secroots" that could crash
                        named when dumping an empty keynode. [RT #35469]

3764.   [bug]           The dnssec-keygen/settime -S and -i options
                        (to set up a successor key and set the prepublication
                        interval) were missing from dnssec-keyfromlabel.
                        [RT #35394]

3763.   [bug]           delve: Cache DNSSEC records to avoid the need to
                        re-fetch them when restarting validation. [RT #35476]

3762.   [bug]           Address build problems with --pkcs11-native +
                        --with-openssl with ECDSA support. [RT #35467]

3761.   [bug]           Address dangling reference bug in dns_keytable_add.
                        [RT #35471]

3760.   [bug]           Improve SIT with native PKCS#11 and on Windows.
                        [RT #35433]

3759.   [port]          Enable delve on Windows. [RT #35441]

3758.   [port]          Enable export library APIs on Windows. [RT #35382]

3757.   [port]          Enable Python tools (dnssec-coverage,
                        dnssec-checkds) to run on Windows. [RT #34355]

3756.   [bug]           GSSAPI Kerberos realm checking was broken in
                        check_config leading to spurious messages being
```

(continues on next page)

(continued from previous page)

```
                        logged.  [RT #35443]
```

```
        --- 9.10.0b1 released ---

3755.   [func]          Add stats counters for known EDNS options + others.
                        [RT #35447]

3754.   [cleanup]       win32: Installer now places files in the
                        Program Files area rather than system services.
                        [RT #35361]

3753.   [bug]           allow-notify was ignoring keys. [RT #35425]

3752.   [bug]           Address potential REQUIRE failure if
                        DNS_STYLEFLAG_COMMENTDATA is set when printing out
                        a rdataset.

3751.   [tuning]        The default setting for the -U option (setting
                        the number of UDP listeners per interface) has
                        been adjusted to improve performance. [RT #35417]

3750.   [experimental]  Partially implement EDNS EXPIRE option as described
                        in draft-andrews-dnsext-expire-00.  Retrieval of
                        the remaining time until expiry for slave zones
                        is supported.

                        EXPIRE uses an experimental option code (65002),
                        which is subject to change. [RT #35416]

3749.   [func]          "dig +subnet" sends an EDNS client subnet option
                        containing the specified address/prefix when
                        querying. (Thanks to Wilmer van der Gaast.)
                        [RT #35415]

3748.   [test]          Use delve to test dns_client interfaces. [RT #35383]

3747.   [bug]           A race condition could lead to a core dump when
                        destroying a resolver fetch object. [RT #35385]

3746.   [func]          New "max-zone-ttl" option enforces maximum
                        TTLs for zones. If loading a zone containing a
                        higher TTL, the load fails. DDNS updates with
                        higher TTLs are accepted but the TTL is truncated.
                        (Note: Currently supported for master zones only;
                        inline-signing slaves will be added.) [RT #38405]

3745.   [func]          "configure --with-tuning=large" adjusts various
                        compiled-in constants and default settings to
                        values suited to large servers with abundant
                        memory. [RT #29538]

3744.   [experimental]  SIT: send and process Source Identity Tokens
```

(continues on next page)

(continued from previous page)

```
                        (similar to DNS Cookies by Donald Eastlake 3rd),
                        which are designed to help clients detect off-path
                        spoofed responses and for servers to identify
                        legitimate clients.

                        SIT uses an experimental EDNS option code (65001),
                        which will be changed to an IANA-assigned value
                        if the experiment is deemed a success.

                        SIT can be enabled via "configure --enable-sit" (or
                        --enable-developer). It is enabled by default in
                        Windows.

                        Servers can be configured to send smaller responses
                        to clients that have not identified themselves via
                        SIT.  RRL processing has also been updated;
                        legitimate clients are not subject to rate
                        limiting. [RT #35389]

3743.   [bug]           delegation-only flag wasn't working in forward zone
                        declarations despite being documented.  This is
                        needed to support turning off forwarding and turning
                        on delegation only at the same name.  [RT #35392]

3742.   [port]          linux: libcap support: declare curval at start of
                        block. [RT #35387]

3741.   [func]          "delve" (domain entity lookup and validation engine):
                        A new tool with dig-like semantics for performing DNS
                        lookups, with internal DNSSEC validation, using the
                        same resolver and validator logic as named. This
                        allows easy validation of DNSSEC data in environments
                        with untrustworthy resolvers, and assists with
                        troubleshooting of DNSSEC problems. [RT #32406]

3740.   [contrib]       Minor fixes to configure --with-dlz-bdb,
                        --with-dlz-postgres and --with-dlz-odbc. [RT #35340]

3739.   [func]          Added per-zone stats counters to track TCP and
                        UDP queries. [RT #35375]

3738.   [bug]           --enable-openssl-hash failed to build. [RT #35343]

3737.   [bug]           'rndc retransfer' could trigger a assertion failure
                        with inline zones. [RT #35353]

3736.   [bug]           nsupdate: When specifying a server by name,
                        fall back to alternate addresses if the first
                        address for that name is not reachable. [RT #25784]

3735.   [cleanup]       Merged the libiscpk11 library into libisc
                        to simplify dependencies. [RT #35205]
```

(continues on next page)

(continued from previous page)

```
3734.   [bug]           Improve building with libtool. [RT #35314]

3733.   [func]          Improve interface scanning support.  Interface
                        information will be automatically updated if the
                        OS supports routing sockets (MacOS, *BSD, Linux).
                        Use "automatic-interface-scan no;" to disable.

                        Add "rndc scan" to trigger a scan. [RT #23027]

3732.   [contrib]       Fixed a type mismatch causing the ODBC DLZ
                        driver to dump core on 64-bit systems. [RT #35324]

3731.   [func]          Added a "no-case-compress" ACL, which causes
                        named to use case-insensitive compression
                        (disabling change #3645) for specified
                        clients. (This is useful when dealing
                        with broken client implementations that
                        use case-sensitive name comparisons,
                        rejecting responses that fail to match the
                        capitalization of the query that was sent.)
                        [RT #35300]

3730.   [cleanup]       Added "never" as a synonym for "none" when
                        configuring key event dates in the dnssec tools.
                        [RT #35277]

3729.   [bug]           dnssec-keygen could set the publication date
                        incorrectly when only the activation date was
                        specified on the command line. [RT #35278]

3728.   [doc]           Expanded native-PKCS#11 documentation,
                        specifically pkcs11: URI labels. [RT #35287]

3727.   [func]          The isc_bitstring API is no longer used and
                        has been removed from libisc. [RT #35284]

3726.   [cleanup]       Clarified the error message when attempting
                        to configure more than 32 response-policy zones.
                        [RT #35283]

3725.   [contrib]       Updated zkt and nslint to newest versions,
                        cleaned up and rearranged the contrib
                        directory, and added a README.
```

```
       --- 9.10.0a2 released ---

3724.   [bug]           win32: Fixed a bug that prevented dig and
                        host from exiting properly after completing
                        a UDP query. [RT #35288]

3723.   [cleanup]       Imported keys are now handled the same way
```

(continues on next page)

(continued from previous page)

```
                        regardless of DNSSEC algorithm. [RT #35215]

3722.   [bug]           Using geoip ACLs in a blackhole statement
                        could cause a segfault. [RT #35272]

3721.   [doc]           Improved documentation of the EDNS processing
                        enhancements introduced in change #3593. [RT #35275]

3720.   [bug]           Address compiler warnings. [RT #35261]

3719.   [bug]           Address memory leak in in peer.c. [RT #35255]

3718.   [bug]           A missing ISC_LINK_INIT in log.c. [RT #35260]

3717.   [port]          hpux: Treat EOPNOTSUPP as a expected error code when
                        probing to see if it is possible to set dscp values
                        on a per packet basis. [RT #35252]

3716.   [bug]           The dns_request code was setting dcsp values when not
                        requested.  [RT #35252]

3715.   [bug]           The region and city databases could fail to
                        initialize when using some versions of libGeoIP,
                        causing assertion failures when named was
                        configured to use them. [RT #35427]

3714.   [test]          System tests that need to test for cryptography
                        support before running can now use a common
                        "testcrypto.sh" script to do so. [RT #35213]

3713.   [bug]           Save memory by not storing "also-notify" addresses
                        in zone objects that are configured not to send
                        notify requests. [RT #35195]

3712.   [placeholder]

3711.   [placeholder]

3710.   [bug]           Address double dns_zone_detach when switching to
                        using automatic empty zones from regular zones.
                        [RT #35177]

3709.   [port]          Use built-in versions of strptime() and timegm()
                        on all platforms to avoid portability issues.
                        [RT #35183]

3708.   [bug]           Address a portentry locking issue in dispatch.c.
                        [RT #35128]

3707.   [bug]           irs_resconf_load now returns ISC_R_FILENOTFOUND
                        on a missing resolv.conf file and initializes the
                        structure as if it had been configured with:
```

(continues on next page)

(continued from previous page)

```
                        nameserver ::1
                        nameserver 127.0.0.1

                Note: Callers will need to be updated to treat
                ISC_R_FILENOTFOUND as a qualified success or else
                they will leak memory. The following code fragment
                will work with both old and new versions without
                changing the behaviour of the existing code.

                resconf = NULL;
                result = irs_resconf_load(mctx, "/etc/resolv.conf",
                                          &resconf);
                if (result != ISC_SUCCESS) {
                        if (resconf != NULL)
                                irs_resconf_destroy(&resconf);
                        ....
                }

                [RT #35194]

3706.   [contrib]       queryperf: Fixed a possible integer overflow when
                printing results. [RT #35182]

3705.   [func]          "configure --enable-native-pkcs11" enables BIND
                to use the PKCS#11 API for all cryptographic
                functions, so that it can drive a hardware service
                module directly without the need to use a modified
                OpenSSL as intermediary (so long as the HSM's vendor
                provides a complete-enough implementation of the
                PKCS#11 interface). This has been tested successfully
                with the Thales nShield HSM and with SoftHSMv2 from
                the OpenDNSSEC project. [RT #29031]

3704.   [protocol]      Accept integer timestamps in RRSIG records. [RT #35185]

3703.   [func]          To improve recursive resolver performance, cache
                records which are still being requested by clients
                can now be automatically refreshed from the
                authoritative server before they expire, reducing
                or eliminating the time window in which no answer
                is available in the cache. See the "prefetch" option
                for more details. [RT #35041]

3702.   [func]          'dnssec-coverage -l' option specifies a length
                of time to check for coverage; events further into
                the future are ignored.  'dnssec-coverage -z'
                checks only ZSK events, and 'dnssec-coverage -k'
                checks only KSK events.  (Thanks to Peter Palfrader.)
                [RT #35168]

3701.   [func]          named-checkconf can now obscure shared secrets
```

(continues on next page)

(continued from previous page)

```
                        when printing by specifying '-x'. [RT #34465]

3700.   [func]          Allow access to subgroups of XML statistics via
                        special URLs http://<server>:<port>/xml/v3/server,
                        /zones, /net, /tasks, /mem, and /status.  [RT #35115]

3699.   [bug]           Improvements to statistics channel XSL stylesheet:
                        the stylesheet can now be cached by the browser;
                        section headers are omitted from the stats display
                        when there is no data in those sections to be
                        displayed; counters are now right-justified for
                        easier readability. [RT #35117]

3698.   [cleanup]       Replaced all uses of memcpy() with memmove().
                        [RT #35120]

3697.   [bug]           Handle "." as a search list element when IDN support
                        is enabled. [RT #35133]

3696.   [bug]           dig failed to handle AXFR style IXFR responses which
                        span multiple messages. [RT #35137]

3695.   [bug]           Address a possible race in dispatch.c. [RT #35107]

3694.   [bug]           Warn when a key-directory is configured for a zone,
                        but does not exist or is not a directory. [RT #35108]

3693.   [security]      memcpy was incorrectly called with overlapping
                        ranges resulting in malformed names being generated
                        on some platforms.  This could cause INSIST failures
                        when serving NSEC3 signed zones (CVE-2014-0591).
                        [RT #35120]

3692.   [bug]           Two calls to dns_db_getoriginnode were fatal if there
                        was no data at the node. [RT #35080]

3691.   [contrib]       Address null pointer dereference in LDAP and
                        MySQL DLZ modules.

3690.   [bug]           Iterative responses could be missed when the source
                        port for an upstream query was the same as the
                        listener port (53). [RT #34925]

3689.   [bug]           Fixed a bug causing an insecure delegation from one
                        static-stub zone to another to fail with a broken
                        trust chain. [RT #35081]

3688.   [bug]           loadnode could return a freed node on out of memory.
                        [RT #35106]

3687.   [bug]           Address null pointer dereference in zone_xfrdone.
                        [RT #35042]
```

(continues on next page)

(continued from previous page)

```
3686.   [func]          "dnssec-signzone -Q" drops signatures from keys
                        that are still published but no longer active.
                        [RT #34990]

3685.   [bug]           "rndc refresh" didn't work correctly with slave
                        zones using inline-signing. [RT #35105]

3684.   [bug]           The list of included files would grow on reload.
                        [RT 35090]

3683.   [cleanup]       Add a more detailed "not found" message to rndc
                        commands which specify a zone name. [RT #35059]

3682.   [bug]           Correct the behavior of rndc retransfer to allow
                        inline-signing slave zones to retain NSEC3 parameters
                        instead of reverting to NSEC. [RT #34745]

3681.   [port]          Update the Windows build system to support feature
                        selection and WIN64 builds.  This is a work in
                        progress. [RT #34160]

3680.   [bug]           Ensure buffer space is available in "rndc zonestatus".
                        [RT #35084]

3679.   [bug]           dig could fail to clean up TCP sockets still
                        waiting on connect(). [RT #35074]

3678.   [port]          Update config.guess and config.sub. [RT #35060]

3677.   [bug]           'nsupdate' leaked memory if 'realm' was used multiple
                        times.  [RT #35073]

3676.   [bug]           "named-checkconf -z" now checks zones of type
                        hint and redirect as well as master. [RT #35046]

3675.   [misc]          Provide a place for third parties to add version
                        information for their extensions in the version
                        file by setting the EXTENSIONS variable.
```

```
        --- 9.10.0a1 released ---

3674.   [bug]           RPZ zeroed ttls if the query type was '*'. [RT #35026]

3673.   [func]          New "in-view" zone option allows direct sharing
                        of zones between views. [RT #32968]

3672.   [func]          Local address can now be specified when using
                        dns_client API. [RT #34811]

3671.   [bug]           Don't allow dnssec-importkey overwrite a existing
                        non-imported private key.
```

(continues on next page)

(continued from previous page)

```
3670.   [bug]           Address read after free in server side of
                        lwres_getrrsetbyname. [RT #29075]

3669.   [port]          freebsd: --with-gssapi needs -lhx509. [RT #35001]

3668.   [bug]           Fix cast in lex.c which could see 0xff treated as eof.
                        [RT #34993]

3667.   [test]          dig: add support to keep the TCP socket open between
                        successive queries (+[no]keepopen).  [RT #34918]

3666.   [func]          Add a tool, named-rrchecker, for checking the syntax
                        of individual resource records.  This tool is intended
                        to be called by provisioning systems so that the front
                        end does not need to be upgraded to support new DNS
                        record types. [RT #34778]

3665.   [bug]           Failure to release lock on error in receive_secure_db.
                        [RT #34944]

3664.   [bug]           Updated OpenSSL PKCS#11 patches to fix active list
                        locking and other bugs. [RT #34855]

3663.   [bug]           Address bugs in dns_rdata_fromstruct and
                        dns_rdata_tostruct for WKS and ISDN types. [RT #34910]

3662.   [bug]           'host' could die if a UDP query timed out. [RT #34870]

3661.   [bug]           Address lock order reversal deadlock with inline zones.
                        [RT #34856]

3660.   [cleanup]       Changed the name of "isc-config.sh" to "bind9-config".
                        [RT #23825]

3659.   [port]          solaris: don't add explicit dependencies/rules for
                        python programs as make won't use the implicit rules.
                        [RT #34835]

3658.   [port]          linux: Address platform specific compilation issue
                        when libcap-devel is installed. [RT #34838]

3657.   [port]          Some readline clones don't accept NULL pointers when
                        calling add_history. [RT #34842]

3656.   [security]      Treat an all zero netmask as invalid when generating
                        the localnets acl. (The prior behavior could
                        allow unexpected matches when using some versions
                        of Winsock: CVE-2013-6320.) [RT #34687]

3655.   [cleanup]       Simplify TCP message processing when requesting a
                        zone transfer.  [RT #34825]
```

(continues on next page)

(continued from previous page)

```
3654.   [bug]           Address race condition with manual notify requests.
                        [RT #34806]

3653.   [func]          Create delegations for all "children" of empty zones
                        except "forward first". [RT #34826]

3652.   [bug]           Address bug with rpz-drop policy. [RT #34816]

3651.   [tuning]        Adjust when a master server is deemed unreachable.
                        [RT #27075]

3650.   [tuning]        Use separate rate limiting queues for refresh and
                        notify requests. [RT #30589]

3649.   [cleanup]       Include a comment in .nzf files, giving the name of
                        the associated view. [RT #34765]

3648.   [test]          Updated the ATF test framework to version 0.17.
                        [RT #25627]

3647.   [bug]           Address a race condition when shutting down a zone.
                        [RT #34750]

3646.   [bug]           Journal filename string could be set incorrectly,
                        causing garbage in log messages. [RT #34738]

3645.   [protocol]      Use case sensitive compression when responding to
                        queries. [RT #34737]

3644.   [protocol]      Check that EDNS subnet client options are well formed.
                        [RT #34718]

3643.   [doc]           Clarify RRL "slip" documentation.

3642.   [func]          Allow externally generated DNSKEY to be imported
                        into the DNSKEY management framework.  A new tool
                        dnssec-importkey is used to do this. [RT #34698]

3641.   [bug]           Handle changes to sig-validity-interval settings
                        better. [RT #34625]

3640.   [bug]           ndots was not being checked when searching.  Only
                        continue searching on NXDOMAIN responses.  Add the
                        ability to specify ndots to nslookup. [RT #34711]

3639.   [bug]           Treat type 65533 (KEYDATA) as opaque except when used
                        in a key zone. [RT #34238]

3638.   [cleanup]       Add the ability to handle ENOPROTOOPT in case it is
                        encountered. [RT #34668]
```

(continues on next page)

(continued from previous page)

```
3637.   [bug]           'allow-query-on' was checking the source address
                        rather than the destination address. [RT #34590]

3636.   [bug]           Automatic empty zones now behave better with
                        forward only "zones" beneath them. [RT #34583]

3635.   [bug]           Signatures were not being removed from a zone with
                        only KSK keys for a algorithm. [RT #34439]

3634.   [func]          Report build-id in rndc status. Report build-id
                        when building from a git repository. [RT #20422]

3633.   [cleanup]       Refactor OPT processing in named to make it easier
                        to support new EDNS options. [RT #34414]

3632.   [bug]           Signature from newly inactive keys were not being
                        removed. [RT #32178]

3631.   [bug]           Remove spurious warning about missing signatures when
                        qtype is SIG. [RT #34600]

3630.   [bug]           Ensure correct ID computation for MD5 keys. [RT #33033]

3629.   [func]          Allow the printing of cryptographic fields in DNSSEC
                        records by dig to be suppressed (dig +nocrypto).
                        [RT #34534]

3628.   [func]          Report DNSKEY key id's when dumping the cache.
                        [RT #34533]

3627.   [bug]           RPZ changes were not effective on slaves. [RT #34450]

3626.   [func]          dig: NSID output now easier to read. [RT #21160]

3625.   [bug]           Don't send notify messages to machines outside of the
                        test setup.

3624.   [bug]           Look for 'json_object_new_int64' when looking for a
                        the json library. [RT #34449]

3623.   [placeholder]

3622.   [tuning]        Eliminate an unnecessary lock when incrementing
                        cache statistics. [RT #34339]

3621.   [security]      Incorrect bounds checking on private type 'keydata'
                        can lead to a remotely triggerable REQUIRE failure
                        (CVE-2013-4854). [RT #34238]

3620.   [func]          Added "rpz-client-ip" policy triggers, enabling
                        RPZ responses to be configured on the basis of
                        the client IP address; this can be used, for
```

(continues on next page)

(continued from previous page)

```
                        example, to blacklist misbehaving recursive
                        or stub resolvers. [RT #33605]

3619.   [bug]           Fixed a bug in RPZ with "recursive-only no;"
                        [RT #33776]

3618.   [func]          "rndc reload" now checks modification times of
                        include files as well as master files to determine
                        whether to skip reloading a zone. [RT #33936]

3617.   [bug]           Named was failing to answer queries during
                        "rndc reload" [RT #34098]

3616.   [bug]           Change #3613 was incomplete. [RT #34177]

3615.   [cleanup]       "configure" now finishes by printing a summary
                        of optional BIND features and whether they are
                        active or inactive. ("configure --enable-full-report"
                        increases the verbosity of the summary.) [RT #31777]

3614.   [port]          Check for <linux/types.h>. [RT #34162]

3613.   [bug]           named could crash when deleting inline-signing
                        zones with "rndc delzone". [RT #34066]

3612.   [port]          Check whether to use -ljson or -ljson-c. [RT #34115]

3611.   [bug]           Improved resistance to a theoretical authentication
                        attack based on differential timing.  [RT #33939]

3610.   [cleanup]       win32: Some executables had been omitted from the
                        installer. [RT #34116]

3609.   [bug]           Corrected a possible deadlock in applications using
                        the export version of the isc_app API. [RT #33967]

3608.   [port]          win32: added todos.pl script to ensure all text files
                        the win32 build depends on are converted to DOS
                        newline format. [RT #22067]

3607.   [bug]           dnssec-keygen had broken 'Invalid keyfile' error
                        message. [RT #34045]

3606.   [func]          "rndc flushtree" now flushes matching
                        records in the address database and bad cache
                        as well as the DNS cache. (Previously only the
                        DNS cache was flushed.) [RT #33970]

3605.   [port]          win32: Addressed several compatibility issues
                        with newer versions of Visual Studio. [RT #33916]

3604.   [bug]           Fixed a compile-time error when building with
```

(continues on next page)

(continued from previous page)

```
                        JSON but not XML. [RT #33959]

3603.   [bug]           Install <isc/stat.h>. [RT #33956]

3602.   [contrib]       Added DLZ Perl module, allowing Perl scripts to
                        integrate with named and serve DNS data.
                        (Contributed by John Eaglesham of Yahoo.)

3601.   [bug]           Added to PKCS#11 openssl patches a value len
                        attribute in DH derive key. [RT #33928]

3600.   [cleanup]       dig: Fixed a typo in the warning output when receiving
                        an oversized response. [RT #33910]

3599.   [tuning]        Check for pointer equivalence in name comparisons.
                        [RT #18125]

3598.   [cleanup]       Improved portability of map file code. [RT #33820]

3597.   [bug]           Ensure automatic-resigning heaps are reconstructed
                        when loading zones in map format. [RT #33381]

3596.   [port]          Updated win32 build documentation, added
                        dnssec-verify. [RT #22067]

3595.   [port]          win32: Fix build problems introduced by change #3550.
                        [RT #33807]

3594.   [maint]         Update config.guess and config.sub. [RT #33816]

3593.   [func]          Update EDNS processing to better track remote server
                        capabilities. [RT #30655]

3592.   [doc]           Moved documentation of rndc command options to the
                        rndc man page. [RT #33506]

3591.   [func]          Use CRC-64 to detect map file corruption at load
                        time. [RT #33746]

3590.   [bug]           When using RRL on recursive servers, defer
                        rate-limiting until after recursion is complete;
                        also, use correct rcode for slipped NXDOMAIN
                        responses.  [RT #33604]

3589.   [func]          Report serial numbers in when starting zone transfers.
                        Report accepted NOTIFY requests including serial.
                        [RT #33037]

3588.   [bug]           dig: addressed a memory leak in the sigchase code
                        that could cause a shutdown crash.  [RT #33733]

3587.   [func]          'named -g' now checks the logging configuration but
```

(continues on next page)

(continued from previous page)

```
                        does not use it. [RT #33473]

3586.   [bug]           Handle errors in xmlDocDumpFormatMemoryEnc. [RT #33706]

3585.   [func]          "rndc delzone -clean" option removes zone files
                        when deleting a zone. [RT #33570]

3584.   [security]      Caching data from an incompletely signed zone could
                        trigger an assertion failure in resolver.c
                        (CVE-2013-3919). [RT #33690]

3583.   [bug]           Address memory leak in GSS-API processing [RT #33574]

3582.   [bug]           Silence false positive warning regarding missing file
                        directive for inline slave zones.  [RT #33662]

3581.   [bug]           Changed the tcp-listen-queue default to 10. [RT #33029]

3580.   [bug]           Addressed a possible race in acache.c [RT #33602]

3579.   [maint]         Updates to PKCS#11 openssl patches, supporting
                        versions 0.9.8y, 1.0.0k, 1.0.1e [RT #33463]

3578.   [bug]           'rndc -c file' now fails if 'file' does not exist.
                        [RT #33571]

3577.   [bug]           Handle zero TTL values better. [RT #33411]

3576.   [bug]           Address a shutdown race when validating. [RT #33573]

3575.   [func]          Changed the logging category for RRL events from
                        'queries' to 'query-errors'. [RT #33540]

3574.   [doc]           The 'hostname' keyword was missing from server-id
                        description in the named.conf man page. [RT #33476]

3573.   [bug]           "rndc addzone" and "rndc delzone" incorrectly handled
                        zone names containing punctuation marks and other
                        nonstandard characters. [RT #33419]

3572.   [func]          Threads are now enabled by default on most
                        operating systems. [RT #25483]

3571.   [bug]           Address race condition in dns_client_startresolve().
                        [RT #33234]

3570.   [bug]           Check internal pointers are valid when loading map
                        files. [RT #33403]

3569.   [contrib]       Ported mysql DLZ driver to dynamically-loadable
                        module, and added multithread support. [RT #33394]
```

(continues on next page)

(continued from previous page)

```
3568.   [cleanup]       Add a product description line to the version file,
                        to be reported by named -v/-V. [RT #33366]

3567.   [bug]           Silence clang static analyzer warnings. [RT #33365]

3566.   [func]          Log when forwarding updates to master. [RT #33240]

3565.   [placeholder]

3564.   [bug]           Improved handling of corrupted map files. [RT #33380]

3563.   [contrib]       zone2sqlite failed with some table names. [RT #33375]

3562.   [func]          Update map file header format to include a SHA-1 hash
                        of the database content, so that corrupted map files
                        can be rejected at load time. [RT #32459]

3561.   [bug]           dig: issue a warning if an EDNS query returns FORMERR
                        or NOTIMP.  Adjust usage message. [RT #33363]

3560.   [bug]           isc-config.sh did not honor includedir and libdir
                        when set via configure. [RT #33345]

3559.   [func]          Check that both forms of Sender Policy Framework
                        records exist or do not exist. [RT #33355]

3558.   [bug]           IXFR of a DLZ stored zone was broken. [RT #33331]

3557.   [bug]           Reloading redirect zones was broken. [RT #33292]

3556.   [maint]         Added AAAA for D.ROOT-SERVERS.NET.

3555.   [bug]           Address theoretical race conditions in acache.c
                        (change #3553 was incomplete). [RT #33252]

3554.   [bug]           RRL failed to correctly rate-limit upward
                        referrals and failed to count dropped error
                        responses in the statistics. [RT #33225]

3553.   [bug]           Address suspected double free in acache. [RT #33252]

3552.   [bug]           Wrong getopt option string for 'nsupdate -r'.
                        [RT #33280]

3551.   [bug]           resolver.querydscp[46] were uninitialized.  [RT #32686]

3550.   [func]          Unified the internal and export versions of the
                        BIND libraries, allowing external clients to use
                        the same libraries as BIND. [RT #33131]

3549.   [doc]           Documentation for "request-nsid" was missing.
                        [RT #33153]
```

(continues on next page)

(continued from previous page)

```
3548.   [bug]           The NSID request code in resolver.c was broken
                        resulting in invalid EDNS options being sent.
                        [RT #33153]

3547.   [bug]           Some malformed unknown rdata records were not properly
                        detected and rejected. [RT #33129]

3546.   [func]          Add EUI48 and EUI64 types. [RT #33082]

3545.   [bug]           RRL slip behavior was incorrect when set to 1.
                        [RT #33111]

3544.   [contrib]       check5011.pl: Script to report the status of
                        managed keys as recorded in managed-keys.bind.
                        Contributed by Tony Finch <dot@dotat.at>

3543.   [bug]           Update socket structure before attaching to socket
                        manager after accept. [RT #33084]

3542.   [placeholder]

3541.   [bug]           Parts of libdns were not properly initialized when
                        built in libexport mode. [RT #33028]

3540.   [test]          libt_api: t_info and t_assert were not thread safe.

3539.   [port]          win32: timestamp format didn't match other platforms.

3538.   [test]          Running "make test" now requires loopback interfaces
                        to be set up. [RT #32452]

3537.   [tuning]        Slave zones, when updated, now send NOTIFY messages
                        to peers before being dumped to disk rather than
                        after. [RT #27242]

3536.   [func]          Add support for setting Differentiated Services Code
                        Point (DSCP) values in named.  Most configuration
                        options which take a "port" option (e.g.,
                        listen-on, forwarders, also-notify, masters,
                        notify-source, etc) can now also take a "dscp"
                        option specifying a code point for use with
                        outgoing traffic, if supported by the underlying
                        OS. [RT #27596]

3535.   [bug]           Minor win32 cleanups. [RT #32962]

3534.   [bug]           Extra text after an embedded NULL was ignored when
                        parsing zone files. [RT #32699]

3533.   [contrib]       query-loc-0.4.0: memory leaks. [RT #32960]
```

(continues on next page)

(continued from previous page)

```
3532.   [contrib]       zkt: fixed buffer overrun, resource leaks. [RT #32960]

3531.   [bug]           win32: A uninitialized value could be returned on out
                        of memory. [RT #32960]

3530.   [contrib]       Better RTT tracking in queryperf. [RT #30128]

3529.   [func]          Named now listens on both IPv4 and IPv6 interfaces
                        by default.  Named previously only listened on IPv4
                        interfaces by default unless named was running in
                        IPv6 only mode.  [RT #32945]

3528.   [func]          New "dnssec-coverage" command scans the timing
                        metadata for a set of DNSSEC keys and reports if a
                        lapse in signing coverage has been scheduled
                        inadvertently. (Note: This tool depends on python;
                        it will not be built or installed on systems that
                        do not have a python interpreter.) [RT #28098]

3527.   [compat]        Add a URI to allow applications to explicitly
                        request a particular XML schema from the statistics
                        channel, returning 404 if not supported. [RT #32481]

3526.   [cleanup]       Set up dependencies for unit tests correctly during
                        build. [RT #32803]

3525.   [func]          Support for additional signing algorithms in rndc:
                        hmac-sha1, -sha224, -sha256, -sha384, and -sha512.
                        The -A option to rndc-confgen can be used to
                        select the algorithm for the generated key.
                        (The default is still hmac-md5; this may
                        change in a future release.) [RT #20363]

3524.   [func]          Added an alternate statistics channel in JSON format,
                        when the server is built with the json-c library:
                        http://[address]:[port]/json. [RT #32630]

3523.   [contrib]       Ported filesystem and ldap DLZ drivers to
                        dynamically-loadable modules, and added the
                        "wildcard" module based on a contribution from
                        Vadim Goncharov <vgoncharov@nic.ru>. [RT #23569]

3522.   [bug]           DLZ lookups could fail to return SERVFAIL when
                        they ought to. [RT #32685]

3521.   [bug]           Address memory leak in opensslecdsa_link.c. [RT #32249]

3520.   [bug]           'mctx' was not being referenced counted in some places
                        where it should have been.  [RT #32794]

3519.   [func]          Full replay protection via four-way handshake is
                        now mandatory for rndc clients. Very old versions
```

(continues on next page)

```
                        of rndc will no longer work. [RT #32798]

3518.   [bug]           Increase the size of dns_rrl_key.s.rtype by one bit
                        so that all dns_rrl_rtype_t enum values fit regardless
                        of whether it is treated as signed or unsigned by
                        the compiler. [RT #32792]

3517.   [bug]           Reorder destruction to avoid shutdown race. [RT #32777]

3516.   [placeholder]

3515.   [port]          '%T' is not portable in strftime(). [RT #32763]

3514.   [bug]           The ranges for valid key sizes in ddns-confgen and
                        rndc-confgen were too constrained. Keys up to 512
                        bits are now allowed for most algorithms, and up
                        to 1024 bits for hmac-sha384 and hmac-sha512.
                        [RT #32753]

3513.   [func]          "dig -u" prints times in microseconds rather than
                        milliseconds. [RT #32704]

3512.   [func]          "rndc validation check" reports the current status
                        of DNSSEC validation. [RT #21397]

3511.   [doc]           Improve documentation of redirect zones. [RT #32756]

3510.   [func]          "rndc status" and XML statistics channel now report
                        server start and reconfiguration times. [RT #21048]

3509.   [cleanup]       Added a product line to version file to allow for
                        easy naming of different products (BIND
                        vs BIND ESV, for example). [RT #32755]

3508.   [contrib]       queryperf was incorrectly rejecting the -T option.
                        [RT #32338]

3507.   [bug]           Statistics channel XSL had a glitch when attempting
                        to chart query data before any queries had been
                        received. [RT #32620]

3506.   [func]          When setting "max-cache-size" and "max-acache-size",
                        the keyword "unlimited" is no longer defined as equal
                        to 4 gigabytes (except on 32-bit platforms); it
                        means literally unlimited. [RT #32358]

3505.   [bug]           When setting "max-cache-size" and "max-acache-size",
                        larger values than 4 gigabytes could not be set
                        explicitly, though larger sizes were available
                        when setting cache size to 0. This has been
                        corrected; the full range is now available.
                        [RT #32358]
```

(continues on next page)

(continued from previous page)

```
3504.   [func]          Add support for ACLs based on geographic location,
                        using MaxMind GeoIP databases. Based on code
                        contributed by Ken Brownfield <kb@slide.com>.
                        [RT #30681]

3503.   [doc]           Clarify size_spec syntax. [RT #32449]

3502.   [func]          zone-statistics: "no" is now a synonym for "none",
                        instead of "terse". [RT #29165]

3501.   [func]          zone-statistics now takes three options: full,
                        terse, and none. "yes" and "no" are retained as
                        synonyms for full and terse, respectively. [RT #29165]

3500.   [security]      Support NAPTR regular expression validation on
                        all platforms without using libregex, which
                        can be vulnerable to memory exhaustion attack
                        (CVE-2013-2266). [RT #32688]
```

```
3499.   [doc]           Corrected ARM documentation of built-in zones.
                        [RT #32694]

3498.   [bug]           zone statistics for zones which matched a potential
                        empty zone could have their zone-statistics setting
                        overridden.

3497.   [func]          When deleting a slave/stub zone using 'rndc delzone'
                        report the files that were being used so they can
                        be cleaned up if desired. [RT #27899]

3496.   [placeholder]

3495.   [func]          Support multiple response-policy zones (up to 32),
                        while improving RPZ performance.  "response-policy"
                        syntax now includes a "min-ns-dots" clause, with
                        default 1, to exclude top-level domains from
                        NSIP and NSDNAME checking. --enable-rpz-nsip and
                        --enable-rpz-nsdname are now the default. [RT #32251]

3494.   [func]          DNS RRL: Blunt the impact of DNS reflection and
                        amplification attacks by rate-limiting substantially-
                        identical responses. [RT #28130]

3493.   [contrib]       Added BDBHPT dynamically-loadable DLZ module,
                        contributed by Mark Goldfinch. [RT #32549]

3492.   [bug]           Fixed a regression in zone loading performance
                        due to lock contention. [RT #30399]

3491.   [bug]           Slave zones using inline-signing must specify a
                        file name. [RT #31946]
```

(continues on next page)

(continued from previous page)

```
3490.   [bug]           When logging RDATA during update, truncate if it's
                        too long. [RT #32365]

3489.   [bug]           --enable-developer now turns on ISC_LIST_CHECKINIT.
                        dns_dlzcreate() failed to properly initialize
                        dlzdb.link.  When cloning a rdataset do not copy
                        the link contents.  [RT #32651]

3488.   [bug]           Use after free error with DH generated keys. [RT #32649]

3487.   [bug]           Change 3444 was not complete.  There was a additional
                        place where the NOQNAME proof needed to be saved.
                        [RT #32629]

3486.   [bug]           named could crash when using TKEY-negotiated keys
                        that had been deleted and then recreated. [RT #32506]

3485.   [cleanup]       Only compile openssl_gostlink.c if we support GOST.

3484.   [bug]           Some statistics were incorrectly rendered in XML.
                        [RT #32587]

3483.   [placeholder]

3482.   [func]          dig +nssearch now prints name servers that don't
                        have address records (missing AAAA or A, or the name
                        doesn't exist). [RT #29348]

3481.   [cleanup]       Removed use of const const in atf.

3480.   [bug]           Silence logging noise when setting up zone
                        statistics. [RT #32525]

3479.   [bug]           Address potential memory leaks in gssapi support
                        code. [RT #32405]

3478.   [port]          Fix a build failure in strict C99 environments
                        [RT #32475]

3477.   [func]          Expand logging when adding records via DDNS update
                        [RT #32365]

3476.   [bug]           "rndc zonestatus" could report a spurious "not
                        found" error on inline-signing zones. [RT #29226]

3475.   [cleanup]       Changed name of 'map' zone file format (previously
                        'fast'). [RT #32458]

3474.   [bug]           nsupdate could assert when the local and remote
                        address families didn't match. [RT #22897]

3473.   [bug]           dnssec-signzone/verify could incorrectly report
```

(continues on next page)

(continued from previous page)

```
                        an error condition due to an empty node above an
                        opt-out delegation lacking an NSEC3. [RT #32072]

3472.   [bug]           The active-connections counter in the socket
                        statistics could underflow. [RT #31747]

3471.   [bug]           The number of UDP dispatches now defaults to
                        the number of CPUs even if -n has been set to
                        a higher value. [RT #30964]

3470.   [bug]           Slave zones could fail to dump when successfully
                        refreshing after an initial failure. [RT #31276]

3469.   [bug]           Handle DLZ lookup failures more gracefully. Improve
                        backward compatibility between versions of DLZ dlopen
                        API. [RT #32275]

3468.   [security]      RPZ rules to generate A records (but not AAAA records)
                        could trigger an assertion failure when used in
                        conjunction with DNS64 (CVE-2012-5689). [RT #32141]

3467.   [bug]           Added checks in dnssec-keygen and dnssec-settime
                        to check for delete date < inactive date. [RT #31719]

3466.   [contrib]       Corrected the DNS_CLIENTINFOMETHODS_VERSION check
                        in DLZ example driver. [RT #32275]

3465.   [bug]           Handle isolated reserved ports. [RT #31778]

3464.   [maint]         Updates to PKCS#11 openssl patches, supporting
                        versions 0.9.8x, 1.0.0j, 1.0.1c [RT #29749]

3463.   [doc]           Clarify managed-keys syntax in ARM. [RT #32232]

3462.   [doc]           Clarify server selection behavior of dig when using
                        -4 or -6 options. [RT #32181]

3461.   [bug]           Negative responses could incorrectly have AD=1
                        set. [RT #32237]

3460.   [bug]           Only link against readline where needed. [RT #29810]

3459.   [func]          Added -J option to named-checkzone/named-compilezone
                        to specify the path to the journal file. [RT #30958]

3458.   [bug]           Return FORMERR when presented with a overly long
                        domain named in a request. [RT #29682]

3457.   [protocol]      Add ILNP records (NID, LP, L32, L64). [RT #31836]

3456.   [port]          g++47: ATF failed to compile. [RT #32012]
```

(continues on next page)

(continued from previous page)

```
3455.   [contrib]       queryperf: fix getopt option list. [RT #32338]

3454.   [port]          sparc64: improve atomic support. [RT #25182]

3453.   [bug]           'rndc addzone' of a zone with 'inline-signing yes;'
                        failed. [RT #31960]

3452.   [bug]           Accept duplicate singleton records. [RT #32329]

3451.   [port]          Increase per thread stack size from 64K to 1M.
                        [RT #32230]

3450.   [bug]           Stop logfileconfig system test spam system logs.
                        [RT #32315]

3449.   [bug]           gen.c: use the pre-processor to construct format
                        strings so that compiler can perform sanity checks;
                        check the snprintf results. [RT #17576]

3448.   [bug]           The allow-query-on ACL was not processed correctly.
                        [RT #29486]

3447.   [port]          Add support for libxml2-2.9.x [RT #32231]

3446.   [port]          win32: Add source ID (see change #3400) to build.
                        [RT #31683]

3445.   [bug]           Warn about zone files with blank owner names
                        immediately after $ORIGIN directives. [RT #31848]

3444.   [bug]           The NOQNAME proof was not being returned from cached
                        insecure responses. [RT #21409]

3443.   [bug]           ddns-confgen: Some TSIG algorithms were incorrectly
                        rejected when generating keys. [RT #31927]

3442.   [port]          Net::DNS 0.69 introduced a non backwards compatible
                        change. [RT #32216]

3441.   [maint]         D.ROOT-SERVERS.NET is now 199.7.91.13.

3440.   [bug]           Reorder get_key_struct to not trigger a assertion when
                        cleaning up due to out of memory error. [RT #32131]

3439.   [placeholder]

3438.   [bug]           Don't accept unknown data escape in quotes. [RT #32031]

3437.   [bug]           isc_buffer_init -> isc_buffer_constinit to initialize
                        buffers with constant data. [RT #32064]

3436.   [bug]           Check malloc/calloc return values. [RT #32088]
```

(continues on next page)

(continued from previous page)

```
3435.   [bug]           Cross compilation support in configure was broken.
                        [RT #32078]

3434.   [bug]           Pass client info to the DLZ findzone() entry
                        point in addition to lookup().  This makes it
                        possible for a database to answer differently
                        whether it's authoritative for a name depending
                        on the address of the client.  [RT #31775]

3433.   [bug]           dlz_findzone() did not correctly handle
                        ISC_R_NOMORE. [RT #31172]

3432.   [func]          Multiple DLZ databases can now be configured.
                        DLZ databases are searched in the order configured,
                        unless set to "search no", in which case a
                        zone can be configured to be retrieved from a
                        particular DLZ database by using a "dlz <name>"
                        option in the zone statement.  DLZ databases can
                        support type "master" and "redirect" zones.
                        [RT #27597]

3431.   [bug]           ddns-confgen: Some valid key algorithms were
                        not accepted. [RT #31927]

3430.   [bug]           win32: isc_time_formatISO8601 was missing the
                        'T' between the date and time. [RT #32044]

3429.   [bug]           dns_zone_getserial2 could a return success without
                        returning a valid serial. [RT #32007]

3428.   [cleanup]       dig: Add timezone to date output. [RT #2269]

3427.   [bug]           dig +trace incorrectly displayed name server
                        addresses instead of names. [RT #31641]

3426.   [bug]           dnssec-checkds: Clearer output when records are not
                        found. [RT #31968]

3425.   [bug]           "acacheentry" reference counting was broken resulting
                        in use after free. [RT #31908]

3424.   [func]          dnssec-dsfromkey now emits the hash without spaces.
                        [RT #31951]

3423.   [bug]           "rndc signing -nsec3param" didn't accept the full
                        range of possible values.  Address portability issues.
                        [RT #31938]

3422.   [bug]           Added a clear error message for when the SOA does not
                        match the referral. [RT #31281]
```

(continues on next page)

(continued from previous page)

```
3421.   [bug]           Named loops when re-signing if all keys are offline.
                        [RT #31916]

3420.   [bug]           Address VPATH compilation issues. [RT #31879]

3419.   [bug]           Memory leak on validation cancel. [RT #31869]

3418.   [func]          New XML schema (version 3.0) for the statistics channel
                        adds query type statistics at the zone level, and
                        flattens the XML tree and uses compressed format to
                        optimize parsing. Includes new XSL that permits
                        charting via the Google Charts API on browsers that
                        support javascript in XSL.  The old XML schema has been
                        deprecated. [RT #30023]

3417.   [placeholder]

3416.   [bug]           Named could die on shutdown if running with 128 UDP
                        dispatches per interface. [RT #31743]

3415.   [bug]           named could die with a REQUIRE failure if a validation
                        was canceled. [RT #31804]

3414.   [bug]           Address locking issues found by Coverity. [RT #31626]

3413.   [func]          Record the number of DNS64 AAAA RRsets that have been
                        synthesized. [RT #27636]

3412.   [bug]           Copy timeval structure from control message data.
                        [RT #31548]

3411.   [tuning]        Use IPV6_USE_MIN_MTU or equivalent with TCP in addition
                        to UDP. [RT #31690]

3410.   [bug]           Addressed Coverity warnings. [RT #31626]

3409.   [contrib]       contrib/dane/mkdane.sh: Tool to generate TLSA RR's
                        from X.509 certificates, for use with DANE
                        (DNS-based Authentication of Named Entities).
                        [RT #30513]

3408.   [bug]           Some DNSSEC-related options (update-check-ksk,
                        dnssec-loadkeys-interval, dnssec-dnskey-kskonly)
                        are now legal in slave zones as long as
                        inline-signing is in use. [RT #31078]

3407.   [placeholder]

3406.   [bug]           mem.c: Fix compilation errors when building with
                        ISC_MEM_TRACKLINES or ISC_MEMPOOL_NAMES disabled.
                        Also, ISC_MEM_DEBUG is no longer optional. [RT #31559]
```

(continues on next page)

(continued from previous page)

```
3405.   [bug]           Handle time going backwards in acache. [RT #31253]

3404.   [bug]           dnssec-signzone: When re-signing a zone, remove
                        RRSIG and NSEC records from nodes that used to be
                        in-zone but are now below a zone cut. [RT #31556]

3403.   [bug]           Silence noisy OpenSSL logging. [RT #31497]

3402.   [test]          The IPv6 interface numbers used for system
                        tests were incorrect on some platforms. [RT #25085]

3401.   [bug]           Addressed Coverity warnings. [RT #31484]

3400.   [cleanup]       "named -V" can now report a source ID string, defined
                        in the "srcid" file in the build tree and normally set
                        to the most recent git hash.  [RT #31494]

3399.   [port]          netbsd: rename 'bool' parameter to avoid namespace
                        clash.  [RT #31515]

3398.   [bug]           SOA parameters were not being updated with inline
                        signed zones if the zone was modified while the
                        server was offline. [RT #29272]

3397.   [bug]           dig crashed when using +nssearch with +tcp. [RT #25298]

3396.   [bug]           OPT records were incorrectly removed from signed,
                        truncated responses. [RT #31439]

3395.   [protocol]      Add RFC 6598 reverse zones to built in empty zones
                        list, 64.100.IN-ADDR.ARPA ... 127.100.IN-ADDR.ARPA.
                        [RT #31336]

3394.   [bug]           Adjust 'successfully validated after lower casing
                        signer' log level and category. [RT #31414]

3393.   [bug]           'host -C' could core dump if REFUSED was received.
                        [RT #31381]

3392.   [func]          Keep statistics on REFUSED responses. [RT #31412]

3391.   [bug]           A DNSKEY lookup that encountered a CNAME failed.
                        [RT #31262]

3390.   [bug]           Silence clang compiler warnings. [RT #30417]

3389.   [bug]           Always return NOERROR (not 0) in TSIG. [RT #31275]

3388.   [bug]           Fixed several Coverity warnings.
                        Note: This change includes a fix for a bug that
                        was subsequently determined to be an exploitable
                        security vulnerability, CVE-2012-5688: named could
```

(continues on next page)

(continued from previous page)

```
                        die on specific queries with dns64 enabled.
                        [RT #30996]

3387.   [func]          DS digest can be disabled at runtime with
                        disable-ds-digests. [RT #21581]

3386.   [bug]           Address locking violation when generating new NSEC /
                        NSEC3 chains. [RT #31224]

3385.   [bug]           named-checkconf didn't detect missing master lists
                        in also-notify clauses. [RT #30810]

3384.   [bug]           Improved logging of crypto errors. [RT #30963]

3383.   [security]      A certain combination of records in the RBT could
                        cause named to hang while populating the additional
                        section of a response. [RT #31090]

3382.   [bug]           SOA query from slave used use-v6-udp-ports range,
                        if set, regardless of the address family in use.
                        [RT #24173]

3381.   [contrib]       Update queryperf to support more RR types.
                        [RT #30762]

3380.   [bug]           named could die if a nonexistent master list was
                        referenced in a also-notify. [RT #31004]

3379.   [bug]           isc_interval_zero and isc_time_epoch should be
                        "const (type)* const". [RT #31069]

3378.   [bug]           Handle missing 'managed-keys-directory' better.
                        [RT #30625]

3377.   [bug]           Removed spurious newline from NSEC3 multiline
                        output. [RT #31044]

3376.   [bug]           Lack of EDNS support was being recorded without a
                        successful response. [RT #30811]

3375.   [bug]           'rndc dumpdb' failed on empty caches. [RT #30808]

3374.   [bug]           isc_parse_uint32 failed to return a range error on
                        systems with 64 bit longs. [RT #30232]

3373.   [bug]           win32: open raw files in binary mode. [RT #30944]

3372.   [bug]           Silence spurious "deleted from unreachable cache"
                        messages.  [RT #30501]

3371.   [bug]           AD=1 should behave like DO=1 when deciding whether to
                        add NS RRsets to the additional section or not.
```

(continues on next page)

(continued from previous page)

```
                        [RT #30479]

3370.   [bug]           Address use after free while shutting down. [RT #30241]

3369.   [bug]           nsupdate terminated unexpectedly in interactive mode
                        if built with readline support. [RT #29550]

3368.   [bug]           <dns/iptable.h>, <dns/private.h> and <dns/zone.h>
                        were not C++ safe.

3367.   [bug]           dns_dnsseckey_create() result was not being checked.
                        [RT #30685]

3366.   [bug]           Fixed Read-After-Write dependency violation for IA64
                        atomic operations. [RT #25181]

3365.   [bug]           Removed spurious newlines from log messages in
                        zone.c [RT #30675]

3364.   [security]      Named could die on specially crafted record.
                        [RT #30416]

3363.   [bug]           Need to allow "forward" and "fowarders" options
                        in static-stub zones; this had been overlooked.
                        [RT #30482]

3362.   [bug]           Setting some option values to 0 in named.conf
                        could trigger an assertion failure on startup.
                        [RT #27730]

3361.   [bug]           "rndc signing -nsec3param" didn't work correctly
                        when salt was set to '-' (no salt). [RT #30099]

3360.   [bug]           'host -w' could die.  [RT #18723]

3359.   [bug]           An improperly-formed TSIG secret could cause a
                        memory leak. [RT #30607]

3358.   [placeholder]

3357.   [port]          Add support for libxml2-2.8.x [RT #30440]

3356.   [bug]           Cap the TTL of signed RRsets when RRSIGs are
                        approaching their expiry, so they don't remain
                        in caches after expiry. [RT #26429]

3355.   [port]          Use more portable awk in verify system test.

3354.   [func]          Improve OpenSSL error logging. [RT #29932]

3353.   [bug]           Use a single task for task exclusive operations.
                        [RT #29872]
```

(continues on next page)

```
3352.   [bug]           Ensure that learned server attributes timeout of the
                        adb cache. [RT #29856]

3351.   [bug]           isc_mem_put and isc_mem_putanddetach didn't report
                        caller if either ISC_MEM_DEBUGSIZE or ISC_MEM_DEBUGCTX
                        memory debugging flags are set. [RT #30243]

3350.   [bug]           Memory read overrun in isc___mem_reallocate if
                        ISC_MEM_DEBUGCTX memory debugging flag is set.
                        [RT #30240]

3349.   [bug]           Change #3345 was incomplete. [RT #30233]

3348.   [bug]           Prevent RRSIG data from being cached if a negative
                        record matching the covering type exists at a higher
                        trust level. Such data already can't be retrieved from
                        the cache since change 3218 -- this prevents it
                        being inserted into the cache as well. [RT #26809]

3347.   [bug]           dnssec-settime: Issue a warning when writing a new
                        private key file would cause a change in the
                        permissions of the existing file. [RT #27724]

3346.   [security]      Bad-cache data could be used before it was
                        initialized, causing an assert. [RT #30025]

3345.   [bug]           Addressed race condition when removing the last item
                        or inserting the first item in an ISC_QUEUE.
                        [RT #29539]

3344.   [func]          New "dnssec-checkds" command checks a zone to
                        determine which DS records should be published
                        in the parent zone, or which DLV records should be
                        published in a DLV zone, and queries the DNS to
                        ensure that it exists. (Note: This tool depends
                        on python; it will not be built or installed on
                        systems that do not have a python interpreter.)
                        [RT #28099]

3343.   [placeholder]

3342.   [bug]           Change #3314 broke saving of stub zones to disk
                        resulting in excessive cpu usage in some cases.
                        [RT #29952]

3341.   [func]          New "dnssec-verify" command checks a signed zone
                        to ensure correctness of signatures and of NSEC/NSEC3
                        chains. [RT #23673]

3340.   [func]          Added new 'map' zone file format, which is an image
                        of a zone database that can be loaded directly into
                        memory via mmap(), allowing much faster zone loading.
```

(continues on next page)

(continued from previous page)

```
                        (Note: Because of pointer sizes and other
                        considerations, this file format is platform-dependent;
                        'map' zone files cannot always be transferred from one
                        server to another.) [RT #25419]

3339.   [func]          Allow the maximum supported rsa exponent size to be
                        specified: "max-rsa-exponent-size <value>;" [RT #29228]

3338.   [bug]           Address race condition in units tests: asyncload_zone
                        and asyncload_zt. [RT #26100]

3337.   [bug]           Change #3294 broke support for the multiple keys
                        in controls. [RT #29694]

3336.   [func]          Maintain statistics for RRsets tagged as "stale".
                        [RT #29514]

3335.   [func]          nslookup: return a nonzero exit code when unable
                        to get an answer. [RT #29492]

3334.   [bug]           Hold a zone table reference while performing a
                        asynchronous load of a zone. [RT #28326]

3333.   [bug]           Setting resolver-query-timeout too low can cause
                        named to not recover if it loses connectivity.
                        [RT #29623]

3332.   [bug]           Re-use cached DS rrsets if possible. [RT #29446]

3331.   [security]      dns_rdataslab_fromrdataset could produce bad
                        rdataslabs. [RT #29644]

3330.   [func]          Fix missing signatures on NOERROR results despite
                        RPZ rewriting.  Also
                         - add optional "recursive-only yes|no" to the
                           response-policy statement
                         - add optional "max-policy-ttl" to the response-policy
                            statement to limit the false data that
                            "recursive-only no" can introduce into
                            resolvers' caches
                         - add a RPZ performance test to bin/tests/system/rpz
                             when queryperf is available.
                         - the encoding of PASSTHRU action to "rpz-passthru".
                             (The old encoding is still accepted.)
                        [RT #26172]


3329.   [bug]           Handle RRSIG signer-name case consistently: We
                        generate RRSIG records with the signer-name in
                        lower case.  We accept them with any case, but if
                        they fail to validate, we try again in lower case.
                        [RT #27451]
```

(continues on next page)

(continued from previous page)

```
3328.   [bug]           Fixed inconsistent data checking in dst_parse.c.
                        [RT #29401]

3327.   [func]          Added 'filter-aaaa-on-v6' option; this is similar
                        to 'filter-aaaa-on-v4' but applies to IPv6
                        connections.  (Use "configure --enable-filter-aaaa"
                        to enable this option.)  [RT #27308]

3326.   [func]          Added task list statistics: task model, worker
                        threads, quantum, tasks running, tasks ready.
                        [RT #27678]

3325.   [func]          Report cache statistics: memory use, number of
                        nodes, number of hash buckets, hit and miss counts.
                        [RT #27056]

3324.   [test]          Add better tests for ADB stats [RT #27057]

3323.   [func]          Report the number of buckets the resolver is using.
                        [RT #27020]

3322.   [func]          Monitor the number of active TCP and UDP dispatches.
                        [RT #27055]

3321.   [func]          Monitor the number of recursive fetches and the
                        number of open sockets, and report these values in
                        the statistics channel. [RT #27054]

3320.   [func]          Added support for monitoring of recursing client
                        count. [RT #27009]

3319.   [func]          Added support for monitoring of ADB entry count and
                        hash size. [RT #27057]

3318.   [tuning]        Reduce the amount of work performed while holding a
                        bucket lock when finished with a fetch context.
                        [RT #29239]

3317.   [func]          Add ECDSA support (RFC 6605). [RT #21918]

3316.   [tuning]        Improved locking performance when recursing.
                        [RT #28836]

3315.   [tuning]        Use multiple dispatch objects for sending upstream
                        queries; this can improve performance on busy
                        multiprocessor systems by reducing lock contention.
                        [RT #28605]

3314.   [bug]           The masters list could be updated while stub_callback
                        or refresh_callback were using it. [RT #26732]
```

(continues on next page)

(continued from previous page)

```
3313.   [protocol]      Add TLSA record type. [RT #28989]

3312.   [bug]           named-checkconf didn't detect a bad dns64 clients acl.
                        [RT #27631]

3311.   [bug]           Abort the zone dump if zone->db is NULL in
                        zone.c:zone_gotwritehandle. [RT #29028]

3310.   [test]          Increase table size for mutex profiling. [RT #28809]

3309.   [bug]           resolver.c:fctx_finddone() was not thread safe.
                        [RT #27995]

3308.   [placeholder]

3307.   [bug]           Add missing ISC_LANG_BEGINDECLS and ISC_LANG_ENDDECLS.
                        [RT #28956]

3306.   [bug]           Improve DNS64 reverse zone performance. [RT #28563]

3305.   [func]          Add wire format lookup method to sdb. [RT #28563]

3304.   [bug]           Use hmctx, not mctx when freeing rbtdb->heaps.
                        [RT #28571]

3303.   [bug]           named could die when reloading. [RT #28606]

3302.   [bug]           dns_dnssec_findmatchingkeys could fail to find
                        keys if the zone name contained character that
                        required special mappings. [RT #28600]

3301.   [contrib]       Update queryperf to build on darwin.  Add -R flag
                        for non-recursive queries. [RT #28565]

3300.   [bug]           Named could die if gssapi was enabled in named.conf
                        but was not compiled in. [RT #28338]

3299.   [bug]           Make SDB handle errors from database drivers better.
                        [RT #28534]

3298.   [bug]           Named could dereference a NULL pointer in
                        zmgr_start_xfrin_ifquota if the zone was being removed.
                        [RT #28419]

3297.   [bug]           Named could die on a malformed master file. [RT #28467]

3296.   [bug]           Named could die with a INSIST failure in
                        client.c:exit_check. [RT #28346]

3295.   [bug]           Adjust isc_time_secondsastimet range check to be more
                        portable. [RT # 26542]
```

(continues on next page)

(continued from previous page)

```
3294.   [bug]           isccc/cc.c:table_fromwire failed to free alist on
                        error. [RT #28265]

3293.   [func]          nsupdate: list supported type. [RT #28261]

3292.   [func]          Log messages in the axfr stream at debug 10.
                        [RT #28040]

3291.   [port]          Fixed a build error on systems without ENOTSUP.
                        [RT #28200]

3290.   [bug]           <isc/hmacsha.h> was not being installed. [RT #28169]

3289.   [bug]           'rndc retransfer' failed for inline zones. [RT #28036]

3288.   [bug]           dlz_destroy() function wasn't correctly registered
                        by the DLZ dlopen driver. [RT #28056]

3287.   [port]          Update ans.pl to work with Net::DNS 0.68. [RT #28028]

3286.   [bug]           Managed key maintenance timer could fail to start
                        after 'rndc reconfig'. [RT #26786]

3285.   [bug]           val-frdataset was incorrectly disassociated in
                        proveunsecure after calling startfinddlvsep.
                        [RT #27928]

3284.   [bug]           Address race conditions with the handling of
                        rbtnode.deadlink. [RT #27738]

3283.   [bug]           Raw zones with with more than 512 records in a RRset
                        failed to load. [RT #27863]

3282.   [bug]           Restrict the TTL of NS RRset to no more than that
                        of the old NS RRset when replacing it.
                        [RT #27792] [RT #27884]

3281.   [bug]           SOA refresh queries could be treated as cancelled
                        despite succeeding over the loopback interface.
                        [RT #27782]

3280.   [bug]           Potential double free of a rdataset on out of memory
                        with DNS64. [RT #27762]

3279.   [bug]           Hold a internal reference to the zone while performing
                        a asynchronous load.  Address potential memory leak
                        if the asynchronous is cancelled. [RT #27750]

3278.   [bug]           Make sure automatic key maintenance is started
                        when "auto-dnssec maintain" is turned on during
                        "rndc reconfig". [RT #26805]
```

(continues on next page)

(continued from previous page)

```
3277.   [bug]           win32: isc_socket_dup is not implemented. [RT #27696]

3276.   [bug]           win32: ns_os_openfile failed to return NULL on
                        safe_open failure. [RT #27696]

3275.   [bug]           Corrected rndc -h output; the 'rndc sync -clean'
                        option had been misspelled as '-clear'.  (To avoid
                        future confusion, both options now work.) [RT #27173]

3274.   [placeholder]

3273.   [bug]           AAAA responses could be returned in the additional
                        section even when filter-aaaa-on-v4 was in use.
                        [RT #27292]

3272.   [func]          New "rndc zonestatus" command prints information
                        about the specified zone. [RT #21671]

3271.   [port]          darwin: mksymtbl is not always stable, loop several
                        times before giving up.  mksymtbl was using non
                        portable perl to covert 64 bit hex strings. [RT #27653]
```

```
        --- 9.9.0rc2 released ---

3270.   [bug]           "rndc reload" didn't reuse existing zones correctly
                        when inline-signing was in use. [RT #27650]

3269.   [port]          darwin 11 and later now built threaded by default.

3268.   [bug]           Convert RRSIG expiry times to 64 timestamps to work
                        out the earliest expiry time. [RT #23311]

3267.   [bug]           Memory allocation failures could be mis-reported as
                        unexpected error.  New ISC_R_UNSET result code.
                        [RT #27336]

3266.   [bug]           The maximum number of NSEC3 iterations for a
                        DNSKEY RRset was not being properly computed.
                        [RT #26543]

3265.   [bug]           Corrected a problem with lock ordering in the
                        inline-signing code. [RT #27557]

3264.   [bug]           Automatic regeneration of signatures in an
                        inline-signing zone could stall when the server
                        was restarted. [RT #27344]

3263.   [bug]           "rndc sync" did not affect the unsigned side of an
                        inline-signing zone. [RT #27337]

3262.   [bug]           Signed responses were handled incorrectly by RPZ.
                        [RT #27316]
```

(continues on next page)

(continued from previous page)

```
3261.   [func]          RRset ordering now defaults to random. [RT #27174]

3260.   [bug]           "rrset-order cyclic" could appear not to rotate
                        for some query patterns.  [RT #27170/27185]
```

```
        --- 9.9.0rc1 released ---

3259.   [bug]           named-compilezone: Suppress "dump zone to <file>"
                        message when writing to stdout. [RT #27109]

3258.   [test]          Add "forcing full sign with unreadable keys" test.
                        [RT #27153]

3257.   [bug]           Do not generate a error message when calling fsync()
                        in a pipe or socket. [RT #27109]

3256.   [bug]           Disable empty zones for lwresd -C. [RT #27139]

3255.   [func]          No longer require that a empty zones be explicitly
                        enabled or that a empty zone is disabled for
                        RFC 1918 empty zones to be configured. [RT #27139]

3254.   [bug]           Set isc_socket_ipv6only() on the IPv6 control channels.
                        [RT #22249]

3253.   [bug]           Return DNS_R_SYNTAX when the input to a text field is
                        too long. [RT #26956]

3252.   [bug]           When master zones using inline-signing were
                        updated while the server was offline, the source
                        zone could fall out of sync with the signed
                        copy. They can now resynchronize. [RT #26676]

3251.   [bug]           Enforce a upper bound (65535 bytes) on the amount of
                        memory dns_sdlz_putrr() can allocate per record to
                        prevent run away memory consumption on ISC_R_NOSPACE.
                        [RT #26956]

3250.   [func]          'configure --enable-developer'; turn on various
                        configure options, normally off by default, that
                        we want developers to build and test with. [RT #27103]

3249.   [bug]           Update log message when saving slave zones files for
                        analysis after load failures. [RT #27087]

3248.   [bug]           Configure options --enable-fixed-rrset and
                        --enable-exportlib were incompatible with each
                        other. [RT #27087]

3247.   [bug]           'raw' format zones failed to preserve load order
                        breaking 'fixed' sort order. [RT #27087]
```

(continues on next page)

(continued from previous page)

```
3246.   [bug]           Named failed to start with a empty also-notify list.
                        [RT #27087]

3245.   [bug]           Don't report a error unchanged serials unless there
                        were other changes when thawing a zone with
                        ixfr-fromdifferences. [RT #26845]

3244.   [func]          Added readline support to nslookup and nsupdate.
                        Also simplified nsupdate syntax to make "update"
                        and "prereq" optional. [RT #24659]

3243.   [port]          freebsd,netbsd,bsdi: the thread defaults were not
                        being properly set.

3242.   [func]          Extended the header of raw-format master files to
                        include the serial number of the zone from which
                        they were generated, if different (as in the case
                        of inline-signing zones).  This is to be used in
                        inline-signing zones, to track changes between the
                        unsigned and signed versions of the zone, which may
                        have different serial numbers.

                        (Note: raw zonefiles generated by this version of
                        BIND are no longer compatible with prior versions.
                        To generate a backward-compatible raw zonefile
                        using dnssec-signzone or named-compilezone, specify
                        output format "raw=0" instead of simply "raw".)
                        [RT #26587]

3241.   [bug]           Address race conditions in the resolver code.
                        [RT #26889]

3240.   [bug]           DNSKEY state change events could be missed. [RT #26874]

3239.   [bug]           dns_dnssec_findmatchingkeys needs to use a consistent
                        timestamp. [RT #26883]

3238.   [bug]           keyrdata was not being reinitialized in
                        lib/dns/rbtdb.c:iszonesecure. [RT #26913]

3237.   [bug]           dig -6 didn't work with +trace. [RT #26906]

3236.   [bug]           Backed out changes #3182 and #3202, related to
                        EDNS(0) fallback behavior. [RT #26416]

3235.   [func]          dns_db_diffx, a extended dns_db_diff which returns
                        the generated diff and optionally writes it to a
                        journal. [RT #26386]

3234.   [bug]           'make depend' produced invalid makefiles. [RT #26830]
```

(continues on next page)

(continued from previous page)

```
3233.   [bug]           'rndc freeze/thaw' didn't work for inline zones.
                        [RT #26632]

3232.   [bug]           Zero zone->curmaster before return in
                        dns_zone_setmasterswithkeys(). [RT #26732]

3231.   [bug]           named could fail to send a incompressible zone.
                        [RT #26796]

3230.   [bug]           'dig axfr' failed to properly handle a multi-message
                        axfr with a serial of 0. [RT #26796]

3229.   [bug]           Fix local variable to struct var assignment
                        found by CLANG warning.

3228.   [tuning]        Dynamically grow symbol table to improve zone
                        loading performance. [RT #26523]

3227.   [bug]           Interim fix to make WKS's use of getprotobyname()
                        and getservbyname() self thread safe. [RT #26232]

3226.   [bug]           Address minor resource leakages. [RT #26624]

3225.   [bug]           Silence spurious "setsockopt(517, IPV6_V6ONLY) failed"
                        messages. [RT #26507]

3224.   [bug]           'rndc signing' argument parsing was broken. [RT #26684]

3223.   [bug]           'task_test privilege_drop' generated false positives.
                        [RT #26766]

3222.   [cleanup]       Replace dns_journal_{get,set}_bitws with
                        dns_journal_{get,set}_sourceserial. [RT #26634]

3221.   [bug]           Fixed a potential core dump on shutdown due to
                        referencing fetch context after it's been freed.
                        [RT #26720]
```

```
        --- 9.9.0b2 released ---

3220.   [bug]           Change #3186 was incomplete; dns_db_rpz_findips()
                        could fail to set the database version correctly,
                        causing an assertion failure. [RT #26180]

3219.   [bug]           Disable NOEDNS caching following a timeout.

3218.   [security]      Cache lookup could return RRSIG data associated with
                        nonexistent records, leading to an assertion
                        failure. [RT #26590]

3217.   [cleanup]       Fix build problem with --disable-static. [RT #26476]
```

(continues on next page)

(continued from previous page)

```
3216.   [bug]           resolver.c:validated() was not thread-safe. [RT #26478]

3215.   [bug]           'rndc recursing' could cause a core dump. [RT #26495]

3214.   [func]          Add 'named -U' option to set the number of UDP
                        listener threads per interface. [RT #26485]

3213.   [doc]           Clarify ixfr-from-differences behavior. [RT #25188]

3212.   [bug]           rbtdb.c: failed to remove a node from the deadnodes
                        list prior to adding a reference to it leading a
                        possible assertion failure. [RT #23219]

3211.   [func]          dnssec-signzone: "-f -" prints to stdout; "-O full"
                        option prints in single-line-per-record format.
                        [RT #20287]

3210.   [bug]           Canceling the oldest query due to recursive-client
                        overload could trigger an assertion failure. [RT #26463]

3209.   [func]          Add "dnssec-lookaside 'no'".  [RT #24858]

3208.   [bug]           'dig -y' handle unknown tsig algorithm better.
                        [RT #25522]

3207.   [contrib]       Fixed build error in Berkeley DB DLZ module. [RT #26444]

3206.   [cleanup]       Add ISC information to log at start time. [RT #25484]

3205.   [func]          Upgrade dig's defaults to better reflect modern
                        nameserver behavior.  Enable "dig +adflag" and
                        "dig +edns=0" by default.  Enable "+dnssec" when
                        running "dig +trace". [RT #23497]

3204.   [bug]           When a master server that has been marked as
                        unreachable sends a NOTIFY, mark it reachable
                        again. [RT #25960]

3203.   [bug]           Increase log level to 'info' for validation failures
                        from expired or not-yet-valid RRSIGs. [RT #21796]

3202.   [bug]           NOEDNS caching on timeout was too aggressive.
                        [RT #26416]

3201.   [func]          'rndc querylog' can now be given an on/off parameter
                        instead of only being used as a toggle. [RT #18351]

3200.   [doc]           Some rndc functions were undocumented or were
                        missing from 'rndc -h' output. [RT #25555]

3199.   [func]          When logging client information, include the name
                        being queried. [RT #25944]
```

(continues on next page)

(continued from previous page)

```
3198.   [doc]           Clarified that dnssec-settime can alter keyfile
                        permissions. [RT #24866]

3197.   [bug]           Don't try to log the filename and line number when
                        the config parser can't open a file. [RT #22263]

3196.   [bug]           nsupdate: return nonzero exit code when target zone
                        doesn't exist. [RT #25783]

3195.   [cleanup]       Silence "file not found" warnings when loading
                        managed-keys zone. [RT #26340]

3194.   [doc]           Updated RFC references in the 'empty-zones-enable'
                        documentation. [RT #25203]

3193.   [cleanup]       Changed MAXZONEKEYS to DNS_MAXZONEKEYS, moved to
                        dnssec.h. [RT #26415]

3192.   [bug]           A query structure could be used after being freed.
                        [RT #22208]

3191.   [bug]           Print NULL records using "unknown" format. [RT #26392]

3190.   [bug]           Underflow in error handling in isc_mutexblock_init.
                        [RT #26397]

3189.   [test]          Added a summary report after system tests. [RT #25517]

3188.   [bug]           zone.c:zone_refreshkeys() could fail to detach
                        references correctly when errors occurred, causing
                        a hang on shutdown. [RT #26372]

3187.   [port]          win32: support for Visual Studio 2008.  [RT #26356]
```

```
        --- 9.9.0b1 released ---

3186.   [bug]           Version/db mismatch in rpz code. [RT #26180]

3185.   [func]          New 'rndc signing' option for auto-dnssec zones:
                         - 'rndc signing -list' displays the current
                           state of signing operations
                         - 'rndc signing -clear' clears the signing state
                           records for keys that have fully signed the zone
                         - 'rndc signing -nsec3param' sets the NSEC3
                           parameters for the zone
                        The 'rndc keydone' syntax is removed. [RT #23729]

3184.   [bug]           named had excessive cpu usage when a redirect zone was
                        configured. [RT #26013]

3183.   [bug]           Added RTLD_GLOBAL flag to dlopen call. [RT #26301]
```

(continues on next page)

(continued from previous page)

```
3182.   [bug]           Auth servers behind firewalls which block packets
                        greater than 512 bytes may cause other servers to
                        perform poorly. Now, adb retains edns information
                        and caches noedns servers. [RT #23392/24964]

3181.   [func]          Inline-signing is now supported for master zones.
                        [RT #26224]

3180.   [func]          Local copies of slave zones are now saved in raw
                        format by default, to improve startup performance.
                        'masterfile-format text;' can be used to override
                        the default, if desired. [RT #25867]

3179.   [port]          kfreebsd: build issues. [RT #26273]

3178.   [bug]           A race condition introduced by change #3163 could
                        cause an assertion failure on shutdown. [RT #26271]

3177.   [func]          'rndc keydone', remove the indicator record that
                        named has finished signing the zone with the
                        corresponding key.  [RT #26206]

3176.   [doc]           Corrected example code and added a README to the
                        sample external DLZ module in contrib/dlz/example.
                        [RT #26215]

3175.   [bug]           Fix how DNSSEC positive wildcard responses from a
                        NSEC3 signed zone are validated.  Stop sending a
                        unnecessary NSEC3 record when generating such
                        responses. [RT #26200]

3174.   [bug]           Always compute to revoked key tag from scratch.
                        [RT #26186]

3173.   [port]          Correctly validate root DS responses. [RT #25726]

3172.   [port]          darwin 10.* and freebsd [89] are now built threaded by
                        default.

3171.   [bug]           Exclusively lock the task when adding a zone using
                        'rndc addzone'.  [RT #25600]
```

```
       --- 9.9.0a3 released ---

3170.   [func]          RPZ update:
                        - fix precedence among competing rules
                        - improve ARM text including documenting rule precedence
                        - try to rewrite CNAME chains until first hit
                        - new "rpz" logging channel
                        - RDATA for CNAME rules can include wildcards
                        - replace "NO-OP" named.conf policy override with
```

(continues on next page)

(continued from previous page)

```
                          "PASSTHRU" and add "DISABLED" override ("NO-OP"
                          is still recognized)
                        [RT #25172]

3169.   [func]          Catch db/version mis-matches when calling dns_db_*().
                        [RT #26017]

3168.   [bug]           Nxdomain redirection could trigger an assert with
                        a ANY query. [RT #26017]

3167.   [bug]           Negative answers from forwarders were not being
                        correctly tagged making them appear to not be cached.
                        [RT #25380]

3166.   [bug]           Upgrading a zone to support inline-signing failed.
                        [RT #26014]

3165.   [bug]           dnssec-signzone could generate new signatures when
                        resigning, even when valid signatures were already
                        present. [RT #26025]

3164.   [func]          Enable DLZ modules to retrieve client information,
                        so that responses can be changed depending on the
                        source address of the query. [RT #25768]

3163.   [bug]           Use finer-grained locking in client.c to address
                        concurrency problems with large numbers of threads.
                        [RT #26044]

3162.   [test]          start.pl: modified to allow for "named.args" in
                        ns*/ subdirectory to override stock arguments to
                        named. Largely from RT #26044, but no separate ticket.

3161.   [bug]           zone.c:del_sigs failed to always reset rdata leading
                        assertion failures. [RT #25880]

3160.   [bug]           When printing out a NSEC3 record in multiline form
                        the newline was not being printed causing type codes
                        to be run together. [RT #25873]

3159.   [bug]           On some platforms, named could assert on startup
                        when running in a chrooted environment without
                        /proc. [RT #25863]

3158.   [bug]           Recursive servers would prefer a particular UDP
                        socket instead of using all available sockets.
                        [RT #26038]

3157.   [tuning]        Reduce the time spent in "rndc reconfig" by parsing
                        the config file before pausing the server. [RT #21373]

3156.   [placeholder]
```

```
        --- 9.9.0a2 released ---

3155.   [bug]           Fixed a build failure when using contrib DLZ
                        drivers (e.g., mysql, postgresql, etc). [RT #25710]

3154.   [bug]           Attempting to print an empty rdataset could trigger
                        an assert. [RT #25452]

3153.   [func]          Extend request-ixfr to zone level and remove the
                        side effect of forcing an AXFR. [RT #25156]

3152.   [cleanup]       Some versions of gcc and clang failed due to
                        incorrect use of __builtin_expect. [RT #25183]

3151.   [bug]           Queries for type RRSIG or SIG could be handled
                        incorrectly.  [RT #21050]

3150.   [func]          Improved startup and reconfiguration time by
                        enabling zones to load in multiple threads. [RT #25333]

3149.   [placeholder]

3148.   [bug]           Processing of normal queries could be stalled when
                        forwarding a UPDATE message. [RT #24711]

3147.   [func]          Initial inline signing support.  [RT #23657]
```

```
        --- 9.9.0a1 released ---

3146.   [test]          Fixed gcc4.6.0 errors in ATF. [RT #25598]

3145.   [test]          Capture output of ATF unit tests in "./atf.out" if
                        there were any errors while running them. [RT #25527]

3144.   [bug]           dns_dbiterator_seek() could trigger an assert when
                        used with a nonexistent database node. [RT #25358]

3143.   [bug]           Silence clang compiler warnings. [RT #25174]

3142.   [bug]           NAPTR is class agnostic. [RT #25429]

3141.   [bug]           Silence spurious "zone serial (0) unchanged" messages
                        associated with empty zones. [RT #25079]

3140.   [func]          New command "rndc flushtree <name>" clears the
                        specified name from the server cache along with
                        all names under it. [RT #19970]

3139.   [test]          Added tests from RFC 6234, RFC 2202, and RFC 1321
                        for the hashing algorithms (md5, sha1 - sha512, and
                        their hmac counterparts).  [RT #25067]
```

(continues on next page)

(continued from previous page)

```
3138.   [bug]           Address memory leaks and out-of-order operations when
                        shutting named down. [RT #25210]

3137.   [func]          Improve hardware scalability by allowing multiple
                        worker threads to process incoming UDP packets.
                        This can significantly increase query throughput
                        on some systems.  [RT #22992]

3136.   [func]          Add RFC 1918 reverse zones to the list of built-in
                        empty zones switched on by the 'empty-zones-enable'
                        option. [RT #24990]

3135.   [port]          FreeBSD: workaround broken IPV6_USE_MIN_MTU processing.
                        See http://www.freebsd.org/cgi/query-pr.cgi?pr=158307
                        [RT #24950]

3134.   [bug]           Improve the accuracy of dnssec-signzone's signing
                        statistics. [RT #16030]

3133.   [bug]           Change #3114 was incomplete. [RT #24577]

3132.   [placeholder]

3131.   [tuning]        Improve scalability by allocating one zone task
                        per 100 zones at startup time, rather than using a
                        fixed-size task table. [RT #24406]

3130.   [func]          Support alternate methods for managing a dynamic
                        zone's serial number.  Two methods are currently
                        defined using serial-update-method, "increment"
                        (default) and "unixtime".  [RT #23849]

3129.   [bug]           Named could crash on 'rndc reconfig' when
                        allow-new-zones was set to yes and named ACLs
                        were used. [RT #22739]

3128.   [func]          Inserting an NSEC3PARAM via dynamic update in an
                        auto-dnssec zone that has not been signed yet
                        will cause it to be signed with the specified NSEC3
                        parameters when keys are activated.  The
                        NSEC3PARAM record will not appear in the zone until
                        it is signed, but the parameters will be stored.
                        [RT #23684]

3127.   [bug]           'rndc thaw' will now remove a zone's journal file
                        if the zone serial number has been changed and
                        ixfr-from-differences is not in use.  [RT #24687]

3126.   [security]      Using DNAME record to generate replacements caused
                        RPZ to exit with a assertion failure. [RT #24766]

3125.   [security]      Using wildcard CNAME records as a replacement with
```

(continues on next page)

(continued from previous page)

```
                        RPZ caused named to exit with a assertion failure.
                        [RT #24715]

3124.   [bug]           Use an rdataset attribute flag to indicate
                        negative-cache records rather than using rrtype 0;
                        this will prevent problems when that rrtype is
                        used in actual DNS packets. [RT #24777]

3123.   [security]      Change #2912 exposed a latent flaw in
                        dns_rdataset_totext() that could cause named to
                        crash with an assertion failure. [RT #24777]

3122.   [cleanup]       dnssec-settime: corrected usage message. [RT #24664]

3121.   [security]      An authoritative name server sending a negative
                        response containing a very large RRset could
                        trigger an off-by-one error in the ncache code
                        and crash named. [RT #24650]

3120.   [bug]           Named could fail to validate zones listed in a DLV
                        that validated insecure without using DLV and had
                        DS records in the parent zone. [RT #24631]

3119.   [bug]           When rolling to a new DNSSEC key, a private-type
                        record could be created and never marked complete.
                        [RT #23253]

3118.   [bug]           nsupdate could dump core on shutdown when using
                        SIG(0) keys. [RT #24604]

3117.   [cleanup]       Remove doc and parser references to the
                        never-implemented 'auto-dnssec create' option.
                        [RT #24533]

3116.   [func]          New 'dnssec-update-mode' option controls updates
                        of DNSSEC records in signed dynamic zones.  Set to
                        'no-resign' to disable automatic RRSIG regeneration
                        while retaining the ability to sign new or changed
                        data. [RT #24533]

3115.   [bug]           Named could fail to return requested data when
                        following a CNAME that points into the same zone.
                        [RT #24455]

3114.   [bug]           Retain expired RRSIGs in dynamic zones if key is
                        inactive and there is no replacement key. [RT #23136]

3113.   [doc]           Document the relationship between serial-query-rate
                        and NOTIFY messages.

3112.   [doc]           Add missing descriptions of the update policy name
                        types "ms-self", "ms-subdomain", "krb5-self" and
```

(continues on next page)

(continued from previous page)

```
                        "krb5-subdomain", which allow machines to update
                        their own records, to the BIND 9 ARM.

3111.   [bug]           Improved consistency checks for dnssec-enable and
                        dnssec-validation, added test cases to the
                        checkconf system test. [RT #24398]

3110.   [bug]           dnssec-signzone: Wrong error message could appear
                        when attempting to sign with no KSK. [RT #24369]

3109.   [func]          The also-notify option now uses the same syntax
                        as a zone's masters clause.  This means it is
                        now possible to specify a TSIG key to use when
                        sending notifies to a given server, or to include
                        an explicit named masters list in an also-notify
                        statement.  [RT #23508]

3108.   [cleanup]       dnssec-signzone: Clarified some error and
                        warning messages; removed #ifdef ALLOW_KSKLESS_ZONES
                        code (use -P instead). [RT #20852]

3107.   [bug]           dnssec-signzone: Report the correct number of ZSKs
                        when using -x. [RT #20852]

3106.   [func]          When logging client requests, include the name of
                        the TSIG key if any. [RT #23619]

3105.   [bug]           GOST support can be suppressed by "configure
                        --without-gost" [RT #24367]

3104.   [bug]           Better support for cross-compiling. [RT #24367]

3103.   [bug]           Configuring 'dnssec-validation auto' in a view
                        instead of in the options statement could trigger
                        an assertion failure in named-checkconf. [RT #24382]

3102.   [func]          New 'dnssec-loadkeys-interval' option configures
                        how often, in minutes, to check the key repository
                        for updates when using automatic key maintenance.
                        Default is every 60 minutes (formerly hard-coded
                        to 12 hours). [RT #23744]

3101.   [bug]           Zones using automatic key maintenance could fail
                        to check the key repository for updates. [RT #23744]

3100.   [security]      Certain response policy zone configurations could
                        trigger an INSIST when receiving a query of type
                        RRSIG. [RT #24280]

3099.   [test]          "dlz" system test now runs but gives R:SKIPPED if
                        not compiled with --with-dlz-filesystem.  [RT #24146]
```

(continues on next page)

(continued from previous page)

```
3098.   [bug]           DLZ zones were answering without setting the AA bit.
                        [RT #24146]

3097.   [test]          Add a tool to test handling of malformed packets.
                        [RT #24096]

3096.   [bug]           Set KRB5_KTNAME before calling log_cred() in
                        dst_gssapi_acceptctx(). [RT #24004]

3095.   [bug]           Handle isolated reserved ports in the port range.
                        [RT #23957]

3094.   [doc]           Expand dns64 documentation.

3093.   [bug]           Fix gssapi/kerberos dependencies [RT #23836]

3092.   [bug]           Signatures for records at the zone apex could go
                        stale due to an incorrect timer setting. [RT #23769]

3091.   [bug]           Fixed a bug in which zone keys that were published
                        and then subsequently activated could fail to trigger
                        automatic signing. [RT #22911]

3090.   [func]          Make --with-gssapi default [RT #23738]

3089.   [func]          dnssec-dsfromkey now supports reading keys from
                        standard input "dnssec-dsfromkey -f -". [RT #20662]

3088.   [bug]           Remove bin/tests/system/logfileconfig/ns1/named.conf
                        and add setup.sh in order to resolve changing
                        named.conf issue.  [RT #23687]

3087.   [bug]           DDNS updates using SIG(0) with update-policy match
                        type "external" could cause a crash. [RT #23735]

3086.   [bug]           Running dnssec-settime -f on an old-style key will
                        now force an update to the new key format even if no
                        other change has been specified, using "-P now -A now"
                        as default values.  [RT #22474]

3085.   [func]          New '-R' option in dnssec-signzone forces removal
                        of signatures which have not yet expired but
                        were generated by a key that no longer exists.
                        [RT #22471]

3084.   [func]          A new command "rndc sync" dumps pending changes in
                        a dynamic zone to disk; "rndc sync -clean" also
                        removes the journal file after syncing.  Also,
                        "rndc freeze" no longer removes journal files.
                        [RT #22473]

3083.   [bug]           NOTIFY messages were not being sent when generating
```

(continues on next page)

(continued from previous page)

```
                        a NSEC3 chain incrementally. [RT #23702]

3082.   [port]          strtok_r is threads only. [RT #23747]

3081.   [bug]           Failure of DNAME substitution did not return
                        YXDOMAIN. [RT #23591]

3080.   [cleanup]       Replaced compile time constant by STDTIME_ON_32BITS.
                        [RT #23587]

3079.   [bug]           Handle isc_event_allocate failures in t_tasks.
                        [RT #23572]

3078.   [func]          Added a new include file with function typedefs
                        for the DLZ "dlopen" driver. [RT #23629]

3077.   [bug]           zone.c:zone_refreshkeys() incorrectly called
                        dns_zone_attach(), use zone->irefs instead. [RT #23303]

3076.   [func]          New '-L' option in dnssec-keygen, dnsset-settime, and
                        dnssec-keyfromlabel sets the default TTL of the
                        key.  When possible, automatic signing will use that
                        TTL when the key is published.  [RT #23304]

3075.   [bug]           dns_dnssec_findzonekeys{2} used a inconsistent
                        timestamp when determining which keys are active.
                        [RT #23642]

3074.   [bug]           Make the adb cache read through for zone data and
                        glue learn for zone named is authoritative for.
                        [RT #22842]

3073.   [bug]           managed-keys changes were not properly being recorded.
                        [RT #20256]

3072.   [bug]           dns_dns64_aaaaok() potential NULL pointer dereference.
                        [RT #20256]

3071.   [bug]           has_nsec could be used uninitialized in
                        update.c:next_active. [RT #20256]

3070.   [bug]           dnssec-signzone potential NULL pointer dereference.
                        [RT #20256]

3069.   [cleanup]       Silence warnings messages from clang static analysis.
                        [RT #20256]

3068.   [bug]           Named failed to build with a OpenSSL without engine
                        support. [RT #23473]

3067.   [bug]           ixfr-from-differences {master|slave}; failed to
                        select the master/slave zones.  [RT #23580]
```

(continues on next page)

(continued from previous page)

```
3066.   [func]          The DLZ "dlopen" driver is now built by default,
                        no longer requiring a configure option.  To
                        disable it, use "configure --without-dlopen".
                        Driver also supported on win32.  [RT #23467]

3065.   [bug]           RRSIG could have time stamps too far in the future.
                        [RT #23356]

3064.   [bug]           powerpc: add sync instructions to the end of atomic
                        operations. [RT #23469]

3063.   [contrib]       More verbose error reporting from DLZ LDAP. [RT #23402]

3062.   [func]          Made several changes to enhance human readability
                        of DNSSEC data in dig output and in generated
                        zone files:
                         - DNSKEY record comments are more verbose, no
                           longer used in multiline mode only
                         - multiline RRSIG records reformatted
                         - multiline output mode for NSEC3PARAM records
                         - "dig +norrcomments" suppresses DNSKEY comments
                         - "dig +split=X" breaks hex/base64 records into
                           fields of width X; "dig +nosplit" disables this.
                        [RT #22820]

3061.   [func]          New option "dnssec-signzone -D", only write out
                        generated DNSSEC records. [RT #22896]

3060.   [func]          New option "dnssec-signzone -X <date>" allows
                        specification of a separate expiration date
                        for DNSKEY RRSIGs and other RRSIGs. [RT #22141]

3059.   [test]          Added a regression test for change #3023.

3058.   [bug]           Cause named to terminate at startup or rndc reconfig/
                        reload to fail, if a log file specified in the conf
                        file isn't a plain file. [RT #22771]

3057.   [bug]           "rndc secroots" would abort after the first error
                        and so could miss some views. [RT #23488]

3056.   [func]          Added support for URI resource record. [RT #23386]

3055.   [placeholder]

3054.   [bug]           Added elliptic curve support check in
                        GOST OpenSSL engine detection. [RT #23485]

3053.   [bug]           Under a sustained high query load with a finite
                        max-cache-size, it was possible for cache memory
                        to be exhausted and not recovered. [RT #23371]
```

(continues on next page)

(continued from previous page)

```
3052.   [test]          Fixed last autosign test report. [RT #23256]

3051.   [bug]           NS records obscure DNAME records at the bottom of the
                        zone if both are present. [RT #23035]

3050.   [bug]           The autosign system test was timing dependent.
                        Wait for the initial autosigning to complete
                        before running the rest of the test. [RT #23035]

3049.   [bug]           Save and restore the gid when creating creating
                        named.pid at startup. [RT #23290]

3048.   [bug]           Fully separate view key management. [RT #23419]

3047.   [bug]           DNSKEY NODATA responses not cached fixed in
                        validator.c. Tests added to dnssec system test.
                        [RT #22908]

3046.   [bug]           Use RRSIG original TTL to compute validated RRset
                        and RRSIG TTL. [RT #23332]

3045.   [removed]       Replaced by change #3050.

3044.   [bug]           Hold the socket manager lock while freeing the socket.
                        [RT #23333]

3043.   [test]          Merged in the NetBSD ATF test framework (currently
                        version 0.12) for development of future unit tests.
                        Use configure --with-atf to build ATF internally
                        or configure --with-atf=prefix to use an external
                        copy.  [RT #23209]

3042.   [bug]           dig +trace could fail attempting to use IPv6
                        addresses on systems with only IPv4 connectivity.
                        [RT #23297]

3041.   [bug]           dnssec-signzone failed to generate new signatures on
                        ttl changes. [RT #23330]

3040.   [bug]           Named failed to validate insecure zones where a node
                        with a CNAME existed between the trust anchor and the
                        top of the zone. [RT #23338]

3039.   [func]          Redirect on NXDOMAIN support. [RT #23146]

3038.   [bug]           Install <dns/rpz.h>.  [RT #23342]

3037.   [doc]           Update COPYRIGHT to contain all the individual
                        copyright notices that cover various parts.

3036.   [bug]           Check built-in zone arguments to see if the zone
```

(continues on next page)

(continued from previous page)

```
                        is re-usable or not. [RT #21914]

3035.   [cleanup]       Simplify by using strlcpy. [RT #22521]

3034.   [cleanup]       nslookup: use strlcpy instead of safecopy. [RT #22521]

3033.   [cleanup]       Add two INSIST(bucket != DNS_ADB_INVALIDBUCKET).
                        [RT #22521]

3032.   [bug]           rdatalist.c: add missing REQUIREs. [RT #22521]

3031.   [bug]           dns_rdataclass_format() handle a zero sized buffer.
                        [RT #22521]

3030.   [bug]           dns_rdatatype_format() handle a zero sized buffer.
                        [RT #22521]

3029.   [bug]           isc_netaddr_format() handle a zero sized buffer.
                        [RT #22521]

3028.   [bug]           isc_sockaddr_format() handle a zero sized buffer.
                        [RT #22521]

3027.   [bug]           Add documented REQUIREs to cfg_obj_asnetprefix() to
                        catch NULL pointer dereferences before they happen.
                        [RT #22521]

3026.   [bug]           lib/isc/httpd.c: check that we have enough space
                        after calling grow_headerspace() and if not
                        re-call grow_headerspace() until we do. [RT #22521]

3025.   [bug]           Fixed a possible deadlock due to zone resigning.
                        [RT #22964]

3024.   [func]          RTT Banding removed due to minor security increase
                        but major impact on resolver latency. [RT #23310]

3023.   [bug]           Named could be left in an inconsistent state when
                        receiving multiple AXFR response messages that were
                        not all TSIG-signed. [RT #23254]

3022.   [bug]           Fixed rpz SERVFAILs after failed zone transfers
                        [RT #23246]

3021.   [bug]           Change #3010 was incomplete. [RT #22296]

3020.   [bug]           auto-dnssec failed to correctly update the zone when
                        changing the DNSKEY RRset. [RT #23232]

3019.   [test]          Test: check apex NSEC3 records after adding DNSKEY
                        record via UPDATE. [RT #23229]
```

(continues on next page)

(continued from previous page)

```
3018.   [bug]           Named failed to check for the "none;" acl when deciding
                        if a zone may need to be re-signed. [RT #23120]

3017.   [doc]           dnssec-keyfromlabel -I was not properly documented.
                        [RT #22887]

3016.   [bug]           rndc usage missing '-b'. [RT #22937]

3015.   [port]          win32: fix IN6_IS_ADDR_LINKLOCAL and
                        IN6_IS_ADDR_SITELOCAL macros. [RT #22724]

3014.   [placeholder]

3013.   [bug]           The DNS64 ttl was not always being set as expected.
                        [RT #23034]

3012.   [bug]           Remove DNSKEY TTL change pairs before generating
                        signing records for any remaining DNSKEY changes.
                        [RT #22590]

3011.   [func]          Change the default query timeout from 30 seconds
                        to 10.  Allow setting this in named.conf using the new
                        'resolver-query-timeout' option, which specifies a max
                        time in seconds.  0 means 'default' and anything longer
                        than 30 will be silently set to 30. [RT #22852]

3010.   [bug]           Fixed a bug where "rndc reconfig" stopped the timer
                        for refreshing managed-keys. [RT #22296]

3009.   [bug]           clients-per-query code didn't work as expected with
                        particular query patterns. [RT #22972]
```

```
        --- 9.8.0b1 released ---

3008.   [func]          Response policy zones (RPZ) support. [RT #21726]

3007.   [bug]           Named failed to preserve the case of domain names in
                        rdata which is not compressible when writing master
                        files.  [RT #22863]

3006.   [func]          Allow dynamically generated TSIG keys to be preserved
                        across restarts of named.  Initially this is for
                        TSIG keys generated using GSSAPI. [RT #22639]

3005.   [port]          Solaris: Work around the lack of
                        gsskrb5_register_acceptor_identity() by setting
                        the KRB5_KTNAME environment variable to the
                        contents of tkey-gssapi-keytab.  Also fixed
                        test errors on MacOSX.  [RT #22853]

3004.   [func]          DNS64 reverse support. [RT #22769]
```

(continues on next page)

(continued from previous page)

```
3003.   [experimental]  Added update-policy match type "external",
                        enabling named to defer the decision of whether to
                        allow a dynamic update to an external daemon.
                        (Contributed by Andrew Tridgell.) [RT #22758]

3002.   [bug]           isc_mutex_init_errcheck() failed to destroy attr.
                        [RT #22766]

3001.   [func]          Added a default trust anchor for the root zone, which
                        can be switched on by setting "dnssec-validation auto;"
                        in the named.conf options. [RT #21727]

3000.   [bug]           More TKEY/GSS fixes:
                         - nsupdate can now get the default realm from
                           the user's Kerberos principal
                         - corrected gsstest compilation flags
                         - improved documentation
                         - fixed some NULL dereferences
                        [RT #22795]

2999.   [func]          Add GOST support (RFC 5933). [RT #20639]

2998.   [func]          Add isc_task_beginexclusive and isc_task_endexclusive
                        to the task api. [RT #22776]

2997.   [func]          named -V now reports the OpenSSL and libxml2 versions
                        it was compiled against. [RT #22687]

2996.   [security]      Temporarily disable SO_ACCEPTFILTER support.
                        [RT #22589]

2995.   [bug]           The Kerberos realm was not being correctly extracted
                        from the signer's identity. [RT #22770]

2994.   [port]          NetBSD: use pthreads by default on NetBSD >= 5.0, and
                        do not use threads on earlier versions.  Also kill
                        the unproven-pthreads, mit-pthreads, and ptl2 support.

2993.   [func]          Dynamically grow adb hash tables. [RT #21186]

2992.   [contrib]       contrib/check-secure-delegation.pl:  A simple tool
                        for looking at a secure delegation. [RT #22059]

2991.   [contrib]       contrib/zone-edit.sh: A simple zone editing tool for
                        dynamic zones. [RT #22365]

2990.   [bug]           'dnssec-settime -S' no longer tests prepublication
                        interval validity when the interval is set to 0.
                        [RT #22761]

2989.   [func]          Added support for writable DLZ zones. (Contributed
                        by Andrew Tridgell of the Samba project.) [RT #22629]
```

(continues on next page)

(continued from previous page)

```
2988.   [experimental]  Added a "dlopen" DLZ driver, allowing the creation
                        of external DLZ drivers that can be loaded as
                        shared objects at runtime rather than linked with
                        named.  Currently this is switched on via a
                        compile-time option, "configure --with-dlz-dlopen".
                        Note: the syntax for configuring DLZ zones
                        is likely to be refined in future releases.
                        (Contributed by Andrew Tridgell of the Samba
                        project.) [RT #22629]

2987.   [func]          Improve ease of configuring TKEY/GSS updates by
                        adding a "tkey-gssapi-keytab" option.  If set,
                        updates will be allowed with any key matching
                        a principal in the specified keytab file.
                        "tkey-gssapi-credential" is no longer required
                        and is expected to be deprecated.  (Contributed
                        by Andrew Tridgell of the Samba project.)
                        [RT #22629]

2986.   [func]          Add new zone type "static-stub".  It's like a stub
                        zone, but the nameserver names and/or their IP
                        addresses are statically configured. [RT #21474]

2985.   [bug]           Add a regression test for change #2896. [RT #21324]

2984.   [bug]           Don't run MX checks when the target of the MX record
                        is ".".  [RT #22645]

2983.   [bug]           Include "loadkeys" in rndc help output. [RT #22493]
```

```
        --- 9.8.0a1 released ---

2982.   [bug]           Reference count dst keys.  dst_key_attach() can be used
                        increment the reference count.

                        Note: dns_tsigkey_createfromkey() callers should now
                        always call dst_key_free() rather than setting it
                        to NULL on success. [RT #22672]

2981.   [func]          Partial DNS64 support (AAAA synthesis). [RT #21991]

2980.   [bug]           named didn't properly handle UPDATES that changed the
                        TTL of the NSEC3PARAM RRset. [RT #22363]

2979.   [bug]           named could deadlock during shutdown if two
                        "rndc stop" commands were issued at the same
                        time. [RT #22108]

2978.   [port]          hpux: look for <devpoll.h> [RT #21919]

2977.   [bug]           'nsupdate -l' report if the session key is missing.
```

(continues on next page)

(continued from previous page)

```
                        [RT #21670]

2976.   [bug]           named could die on exit after negotiating a GSS-TSIG
                        key. [RT #22573]

2975.   [bug]           rbtdb.c:cleanup_dead_nodes_callback() acquired the
                        wrong lock which could lead to server deadlock.
                        [RT #22614]

2974.   [bug]           Some valid UPDATE requests could fail due to a
                        consistency check examining the existing version
                        of the zone rather than the new version resulting
                        from the UPDATE. [RT #22413]

2973.   [bug]           bind.keys.h was being removed by the "make clean"
                        at the end of configure resulting in build failures
                        where there is very old version of perl installed.
                        Move it to "make maintainer-clean". [RT #22230]

2972.   [bug]           win32: address windows socket errors. [RT #21906]

2971.   [bug]           Fixed a bug that caused journal files not to be
                        compacted on Windows systems as a result of
                        non-POSIX-compliant rename() semantics. [RT #22434]

2970.   [security]      Adding a NO DATA negative cache entry failed to clear
                        any matching RRSIG records.  A subsequent lookup of
                        of NO DATA cache entry could trigger a INSIST when the
                        unexpected RRSIG was also returned with the NO DATA
                        cache entry.

                        CVE-2010-3613, VU#706148. [RT #22288]

2969.   [security]      Fix acl type processing so that allow-query works
                        in options and view statements.  Also add a new
                        set of tests to verify proper functioning.

                        CVE-2010-3615, VU#510208. [RT #22418]

2968.   [security]      Named could fail to prove a data set was insecure
                        before marking it as insecure.  One set of conditions
                        that can trigger this occurs naturally when rolling
                        DNSKEY algorithms.

                        CVE-2010-3614, VU#837744. [RT #22309]

2967.   [bug]           'host -D' now turns on debugging messages earlier.
                        [RT #22361]

2966.   [bug]           isc_print_vsnprintf() failed to check if there was
                        space available in the buffer when adding a left
                        justified character with a non zero width,
```

(continues on next page)

(continued from previous page)

```
                        (e.g. "%-1c"). [RT #22270]

2965.   [func]          Test HMAC functions using test data from RFC 2104 and
                        RFC 4634. [RT #21702]

2964.   [placeholder]

2963.   [security]      The allow-query acl was being applied instead of the
                        allow-query-cache acl to cache lookups. [RT #22114]

2962.   [port]          win32: add more dependencies to BINDBuild.dsw.
                        [RT #22062]

2961.   [bug]           Be still more selective about the non-authoritative
                        answers we apply change 2748 to. [RT #22074]

2960.   [func]          Check that named accepts non-authoritative answers.
                        [RT #21594]

2959.   [func]          Check that named starts with a missing masterfile.
                        [RT #22076]

2958.   [bug]           named failed to start with a missing master file.
                        [RT #22076]

2957.   [bug]           entropy_get() and entropy_getpseudo() failed to match
                        the API for RAND_bytes() and RAND_pseudo_bytes()
                        respectively. [RT #21962]

2956.   [port]          Enable atomic operations on the PowerPC64. [RT #21899]

2955.   [func]          Provide more detail in the recursing log. [RT #22043]

2954.   [bug]           contrib: dlz_mysql_driver.c bad error handling on
                        build_sqldbinstance failure. [RT #21623]

2953.   [bug]           Silence spurious "expected covering NSEC3, got an
                        exact match" message when returning a wildcard
                        no data response. [RT #21744]

2952.   [port]          win32: named-checkzone and named-checkconf failed
                        to initialize winsock. [RT #21932]

2951.   [bug]           named failed to generate a correct signed response
                        in a optout, delegation only zone with no secure
                        delegations. [RT #22007]

2950.   [bug]           named failed to perform a SOA up to date check when
                        falling back to TCP on UDP timeouts when
                        ixfr-from-differences was set. [RT #21595]

2949.   [bug]           dns_view_setnewzones() contained a memory leak if
```

(continues on next page)

(continued from previous page)

```
                        it was called multiple times. [RT #21942]

2948.   [port]          MacOS: provide a mechanism to configure the test
                        interfaces at reboot. See bin/tests/system/README
                        for details.

2947.   [placeholder]

2946.   [doc]           Document the default values for the minimum and maximum
                        zone refresh and retry values in the ARM. [RT #21886]

2945.   [doc]           Update empty-zones list in ARM. [RT #21772]

2944.   [maint]         Remove ORCHID prefix from built in empty zones.
                        [RT #21772]

2943.   [func]          Add support to load new keys into managed zones
                        without signing immediately with "rndc loadkeys".
                        Add support to link keys with "dnssec-keygen -S"
                        and "dnssec-settime -S".  [RT #21351]

2942.   [contrib]       zone2sqlite failed to setup the entropy sources.
                        [RT #21610]

2941.   [bug]           sdb and sdlz (dlz's zone database) failed to support
                        DNAME at the zone apex.  [RT #21610]

2940.   [port]          Remove connection aborted error message on
                        Windows. [RT #21549]

2939.   [func]          Check that named successfully skips NSEC3 records
                        that fail to match the NSEC3PARAM record currently
                        in use. [RT #21868]

2938.   [bug]           When generating signed responses, from a signed zone
                        that uses NSEC3, named would use a uninitialized
                        pointer if it needed to skip a NSEC3 record because
                        it didn't match the selected NSEC3PARAM record for
                        zone. [RT #21868]

2937.   [bug]           Worked around an apparent race condition in over
                        memory conditions.  Without this fix a DNS cache DB or
                        ADB could incorrectly stay in an over memory state,
                        effectively refusing further caching, which
                        subsequently made a BIND 9 caching server unworkable.
                        This fix prevents this problem from happening by
                        polling the state of the memory context, rather than
                        making a copy of the state, which appeared to cause
                        a race.  This is a "workaround" in that it doesn't
                        solve the possible race per se, but several experiments
                        proved this change solves the symptom.  Also, the
                        polling overhead hasn't been reported to be an issue.
```

(continues on next page)

(continued from previous page)

```
                        This bug should only affect a caching server that
                        specifies a finite max-cache-size.  It's also quite
                        likely that the bug happens only when enabling threads,
                        but it's not confirmed yet. [RT #21818]

2936.   [func]          Improved configuration syntax and multiple-view
                        support for addzone/delzone feature (see change
                        #2930).  Removed "new-zone-file" option, replaced
                        with "allow-new-zones (yes|no)".  The new-zone-file
                        for each view is now created automatically, with
                        a filename generated from a hash of the view name.
                        It is no longer necessary to "include" the
                        new-zone-file in named.conf; this happens
                        automatically.  Zones that were not added via
                        "rndc addzone" can no longer be removed with
                        "rndc delzone". [RT #19447]

2935.   [bug]           nsupdate: improve 'file not found' error message.
                        [RT #21871]

2934.   [bug]           Use ANSI C compliant shift range in lib/isc/entropy.c.
                        [RT #21871]

2933.   [bug]           'dig +nsid' used stack memory after it went out of
                        scope.  This could potentially result in a unknown,
                        potentially malformed, EDNS option being sent instead
                        of the desired NSID option. [RT #21781]

2932.   [cleanup]       Corrected a numbering error in the "dnssec" test.
                        [RT #21597]

2931.   [bug]           Temporarily and partially disable change 2864
                        because it would cause infinite attempts of RRSIG
                        queries.  This is an urgent care fix; we'll
                        revisit the issue and complete the fix later.
                        [RT #21710]

2930.   [experimental]  New "rndc addzone" and "rndc delzone" commands
                        allow dynamic addition and deletion of zones.
                        To enable this feature, specify a "new-zone-file"
                        option at the view or options level in named.conf.
                        Zone configuration information for the new zones
                        will be written into that file.  To make the new
                        zones persist after a restart, "include" the file
                        into named.conf in the appropriate view.  (Note:
                        This feature is not yet documented, and its syntax
                        is expected to change.) [RT #19447]

2929.   [bug]           Improved handling of GSS security contexts:
                         - added LRU expiration for generated TSIGs
                         - added the ability to use a non-default realm
                         - added new "realm" keyword in nsupdate
```

(continues on next page)

(continued from previous page)

```
                         - limited lifetime of generated keys to 1 hour
                           or the lifetime of the context (whichever is
                           smaller)
                        [RT #19737]

2928.   [bug]           Be more selective about the non-authoritative
                        answer we apply change 2748 to. [RT #21594]

2927.   [placeholder]

2926.   [placeholder]

2925.   [bug]           Named failed to accept uncachable negative responses
                        from insecure zones. [RT #21555]

2924.   [func]          'rndc  secroots'  dump a combined summary of the
                        current managed keys combined with trusted keys.
                        [RT #20904]

2923.   [bug]           'dig +trace' could drop core after "connection
                        timeout". [RT #21514]

2922.   [contrib]       Update zkt to version 1.0.

2921.   [bug]           The resolver could attempt to destroy a fetch context
                        too soon.  [RT #19878]

2920.   [func]          Allow 'filter-aaaa-on-v4' to be applied selectively
                        to IPv4 clients.  New acl 'filter-aaaa' (default any).

2919.   [func]          Add autosign-ksk and autosign-zsk virtual time tests.
                        [RT #20840]

2918.   [maint]         Add AAAA address for I.ROOT-SERVERS.NET.

2917.   [func]          Virtual time test framework. [RT #20801]

2916.   [func]          Add framework to use IPv6 in tests.
                        fd92:7065:b8e:ffff::1 ... fd92:7065:b8e:ffff::7

2915.   [cleanup]       Be smarter about which objects we attempt to compile
                        based on configure options. [RT #21444]

2914.   [bug]           Make the "autosign" system test more portable.
                        [RT #20997]

2913.   [func]          Add pkcs#11 system tests. [RT #20784]

2912.   [func]          Windows clients don't like UPDATE responses that clear
                        the zone section. [RT #20986]

2911.   [bug]           dnssec-signzone didn't handle out of zone records well.
```

(continues on next page)

(continued from previous page)

```
                        [RT #21367]

2910.   [func]          Sanity check Kerberos credentials. [RT #20986]

2909.   [bug]           named-checkconf -p could die if "update-policy local;"
                        was specified in named.conf. [RT #21416]

2908.   [bug]           It was possible for re-signing to stop after removing
                        a DNSKEY. [RT #21384]

2907.   [bug]           The export version of libdns had undefined references.
                        [RT #21444]

2906.   [bug]           Address RFC 5011 implementation issues. [RT #20903]

2905.   [port]          aix: set use_atomic=yes with native compiler.
                        [RT #21402]

2904.   [bug]           When using DLV, sub-zones of the zones in the DLV,
                        could be incorrectly marked as insecure instead of
                        secure leading to negative proofs failing.  This was
                        a unintended outcome from change 2890. [RT #21392]

2903.   [bug]           managed-keys-directory missing from namedconf.c.
                        [RT #21370]

2902.   [func]          Add regression test for change 2897. [RT #21040]

2901.   [port]          Use AC_C_FLEXIBLE_ARRAY_MEMBER. [RT #21316]

2900.   [bug]           The placeholder negative caching element was not
                        properly constructed triggering a INSIST in
                        dns_ncache_towire(). [RT #21346]

2899.   [port]          win32: Support linking against OpenSSL 1.0.0.

2898.   [bug]           nslookup leaked memory when -domain=value was
                        specified. [RT #21301]

2897.   [bug]           NSEC3 chains could be left behind when transitioning
                        to insecure. [RT #21040]

2896.   [bug]           "rndc sign" failed to properly update the zone
                        when adding a DNSKEY for publication only. [RT #21045]

2895.   [func]          genrandom: add support for the generation of multiple
                        files.  [RT #20917]

2894.   [contrib]       DLZ LDAP support now use '$' not '%'. [RT #21294]

2893.   [bug]           Improve managed keys support.  New named.conf option
                        managed-keys-directory. [RT #20924]
```

(continues on next page)

(continued from previous page)

```
2892.   [bug]           Handle REVOKED keys better. [RT #20961]

2891.   [maint]         Update empty-zones list to match
                        draft-ietf-dnsop-default-local-zones-13. [RT #21099]

2890.   [bug]           Handle the introduction of new trusted-keys and
                        DS, DLV RRsets better. [RT #21097]

2889.   [bug]           Elements of the grammar where not properly reported.
                        [RT #21046]

2888.   [bug]           Only the first EDNS option was displayed. [RT #21273]

2887.   [bug]           Report the keytag times in UTC in the .key file,
                        local time is presented as a comment within the
                        comment.  [RT #21223]

2886.   [bug]           ctime() is not thread safe. [RT #21223]

2885.   [bug]           Improve -fno-strict-aliasing support probing in
                        configure. [RT #21080]

2884.   [bug]           Insufficient validation in dns_name_getlabelsequence().
                        [RT #21283]

2883.   [bug]           'dig +short' failed to handle really large datasets.
                        [RT #21113]

2882.   [bug]           Remove memory context from list of active contexts
                        before clearing 'magic'. [RT #21274]

2881.   [bug]           Reduce the amount of time the rbtdb write lock
                        is held when closing a version. [RT #21198]

2880.   [cleanup]       Make the output of dnssec-keygen and dnssec-revoke
                        consistent. [RT #21078]

2879.   [contrib]       DLZ bdbhpt driver fails to close correct cursor.
                        [RT #21106]

2878.   [func]          Incrementally write the master file after performing
                        a AXFR.  [RT #21010]

2877.   [bug]           The validator failed to skip obviously mismatching
                        RRSIGs. [RT #21138]

2876.   [bug]           Named could return SERVFAIL for negative responses
                        from unsigned zones. [RT #21131]

2875.   [bug]           dns_time64_fromtext() could accept non digits.
                        [RT #21033]
```

(continues on next page)

(continued from previous page)

```
2874.   [bug]           Cache lack of EDNS support only after the server
                        successfully responds to the query using plain DNS.
                        [RT #20930]

2873.   [bug]           Canceling a dynamic update via the dns/client module
                        could trigger an assertion failure. [RT #21133]

2872.   [bug]           Modify dns/client.c:dns_client_createx() to only
                        require one of IPv4 or IPv6 rather than both.
                        [RT #21122]

2871.   [bug]           Type mismatch in mem_api.c between the definition and
                        the header file, causing build failure with
                        --enable-exportlib. [RT #21138]

2870.   [maint]         Add AAAA address for L.ROOT-SERVERS.NET.

2869.   [bug]           Fix arguments to dns_keytable_findnextkeynode() call.
                        [RT #20877]

2868.   [cleanup]       Run "make clean" at the end of configure to ensure
                        any changes made by configure are integrated.
                        Use --with-make-clean=no to disable.  [RT #20994]

2867.   [bug]           Don't set GSS_C_SEQUENCE_FLAG as Windows DNS servers
                        don't like it.  [RT #20986]

2866.   [bug]           Windows does not like the TSIG name being compressed.
                        [RT #20986]

2865.   [bug]           memset to zero event.data.  [RT #20986]

2864.   [bug]           Direct SIG/RRSIG queries were not handled correctly.
                        [RT #21050]

2863.   [port]          linux: disable IPv6 PMTUD and use network minimum MTU.
                        [RT #21056]

2862.   [bug]           nsupdate didn't default to the parent zone when
                        updating DS records. [RT #20896]

2861.   [doc]           dnssec-settime man pages didn't correctly document the
                        inactivation time. [RT #21039]

2860.   [bug]           named-checkconf's usage was out of date. [RT #21039]

2859.   [bug]           When canceling validation it was possible to leak
                        memory. [RT #20800]

2858.   [bug]           RTT estimates were not being adjusted on ICMP errors.
                        [RT #20772]
```

(continues on next page)

(continued from previous page)

```
2857.   [bug]           named-checkconf did not fail on a bad trusted key.
                        [RT #20705]

2856.   [bug]           The size of a memory allocation was not always properly
                        recorded. [RT #20927]

2855.   [func]          nsupdate will now preserve the entered case of domain
                        names in update requests it sends. [RT #20928]

2854.   [func]          dig: allow the final soa record in a axfr response to
                        be suppressed, dig +onesoa. [RT #20929]

2853.   [bug]           add_sigs() could run out of scratch space. [RT #21015]

2852.   [bug]           Handle broken DNSSEC trust chains better. [RT #15619]

2851.   [doc]           nslookup.1, removed <informalexample> from the docbook
                        source as it produced bad nroff.  [RT #21007]

2850.   [bug]           If isc_heap_insert() failed due to memory shortage
                        the heap would have corrupted entries. [RT #20951]

2849.   [bug]           Don't treat errors from the xml2 library as fatal.
                        [RT #20945]

2848.   [doc]           Moved README.dnssec, README.libdns, README.pkcs11 and
                        README.rfc5011 into the ARM. [RT #20899]

2847.   [cleanup]       Corrected usage message in dnssec-settime. [RT #20921]

2846.   [bug]           EOF on unix domain sockets was not being handled
                        correctly. [RT #20731]

2845.   [bug]           RFC 5011 client could crash on shutdown. [RT #20903]

2844.   [doc]           notify-delay default in ARM was wrong.  It should have
                        been five (5) seconds.

2843.   [func]          Prevent dnssec-keygen and dnssec-keyfromlabel from
                        creating key files if there is a chance that the new
                        key ID will collide with an existing one after
                        either of the keys has been revoked.  (To override
                        this in the case of dnssec-keyfromlabel, use the -y
                        option.  dnssec-keygen will simply create a
                        different, non-colliding key, so an override is
                        not necessary.) [RT #20838]

2842.   [func]          Added "smartsign" and improved "autosign" and
                        "dnssec" regression tests. [RT #20865]

2841.   [bug]           Change 2836 was not complete. [RT #20883]
```

(continues on next page)

```
2840.   [bug]           Temporary fixed pkcs11-destroy usage check.
                        [RT #20760]

2839.   [bug]           A KSK revoked by named could not be deleted.
                        [RT #20881]

2838.   [placeholder]

2837.   [port]          Prevent Linux spurious warnings about fwrite().
                        [RT #20812]

2836.   [bug]           Keys that were scheduled to become active could
                        be delayed. [RT #20874]

2835.   [bug]           Key inactivity dates were inadvertently stored in
                        the private key file with the outdated tag
                        "Unpublish" rather than "Inactive".  This has been
                        fixed; however, any existing keys that had Inactive
                        dates set will now need to have them reset, using
                        'dnssec-settime -I'. [RT #20868]

2834.   [bug]           HMAC-SHA* keys that were longer than the algorithm
                        digest length were used incorrectly, leading to
                        interoperability problems with other DNS
                        implementations.  This has been corrected.
                        (Note: If an oversize key is in use, and
                        compatibility is needed with an older release of
                        BIND, the new tool "isc-hmac-fixup" can convert
                        the key secret to a form that will work with all
                        versions.) [RT #20751]

2833.   [cleanup]       Fix usage messages in dnssec-keygen and dnssec-settime.
                        [RT #20851]

2832.   [bug]           Modify "struct stat" in lib/export/samples/nsprobe.c
                        to avoid redefinition in some OSs [RT 20831]

2831.   [security]      Do not attempt to validate or cache
                        out-of-bailiwick data returned with a secure
                        answer; it must be re-fetched from its original
                        source and validated in that context. [RT #20819]

2830.   [bug]           Changing the OPTOUT setting could take multiple
                        passes. [RT #20813]

2829.   [bug]           Fixed potential node inconsistency in rbtdb.c.
                        [RT #20808]

2828.   [security]      Cached CNAME or DNAME RR could be returned to clients
                        without DNSSEC validation. [RT #20737]

2827.   [security]      Bogus NXDOMAIN could be cached as if valid. [RT #20712]
```

(continues on next page)

(continued from previous page)

```
2826.   [bug]           NSEC3->NSEC transitions could fail due to a lock not
                        being released.  [RT #20740]

2825.   [bug]           Changing the setting of OPTOUT in a NSEC3 chain that
                        was in the process of being created was not properly
                        recorded in the zone. [RT #20786]

2824.   [bug]           "rndc sign" was not being run by the correct task.
                        [RT #20759]

2823.   [bug]           rbtdb.c:getsigningtime() was missing locks. [RT #20781]

2822.   [bug]           rbtdb.c:loadnode() could return the wrong result.
                        [RT #20802]

2821.   [doc]           Add note that named-checkconf doesn't automatically
                        read rndc.key and bind.keys [RT #20758]

2820.   [func]          Handle read access failure of OpenSSL configuration
                        file more user friendly (PKCS#11 engine patch).
                        [RT #20668]

2819.   [cleanup]       Removed unnecessary DNS_POINTER_MAXHOPS define.
                        [RT #20771]

2818.   [cleanup]       rndc could return an incorrect error code
                        when a zone was not found. [RT #20767]

2817.   [cleanup]       Removed unnecessary isc_task_endexclusive() calls.
                        [RT #20768]

2816.   [bug]           previous_closest_nsec() could fail to return
                        data for NSEC3 nodes [RT #29730]

2815.   [bug]           Exclusively lock the task when freezing a zone.
                        [RT #19838]

2814.   [func]          Provide a definitive error message when a master
                        zone is not loaded. [RT #20757]

2813.   [bug]           Better handling of unreadable DNSSEC key files.
                        [RT #20710]

2812.   [bug]           Make sure updates can't result in a zone with
                        NSEC-only keys and NSEC3 records. [RT #20748]

2811.   [cleanup]       Add "rndc sign" to list of commands in rndc usage
                        output. [RT #20733]

2810.   [doc]           Clarified the process of transitioning an NSEC3 zone
                        to insecure. [RT #20746]
```

(continues on next page)

(continued from previous page)

```
2809.   [cleanup]       Restored accidentally-deleted text in usage output
                        in dnssec-settime and dnssec-revoke [RT #20739]

2808.   [bug]           Remove the attempt to install atomic.h from lib/isc.
                        atomic.h is correctly installed by the architecture
                        specific subdirectories.  [RT #20722]

2807.   [bug]           Fixed a possible ASSERT when reconfiguring zone
                        keys. [RT #20720]
```

```
        --- 9.7.0rc1 released ---

2806.   [bug]           "rdnc sign" could delay re-signing the DNSKEY
                        when it had changed. [RT #20703]

2805.   [bug]           Fixed namespace problems encountered when building
                        external programs using non-exported BIND9 libraries
                        (i.e., built without --enable-exportlib). [RT #20679]

2804.   [bug]           Send notifies when a zone is signed with "rndc sign"
                        or as a result of a scheduled key change. [RT #20700]

2803.   [port]          win32: Install named-journalprint, nsec3hash, arpaname
                        and genrandom under windows. [RT #20670]

2802.   [cleanup]       Rename journalprint to named-journalprint. [RT #20670]

2801.   [func]          Detect and report records that are different according
                        to DNSSEC but are semantically equal according to plain
                        DNS.  Apply plain DNS comparisons rather than DNSSEC
                        comparisons when processing UPDATE requests.
                        dnssec-signzone now removes such semantically duplicate
                        records prior to signing the RRset.

                        named-checkzone -r {ignore|warn|fail} (default warn)
                        named-compilezone -r {ignore|warn|fail} (default warn)

                        named.conf: check-dup-records {ignore|warn|fail};

2800.   [func]          Reject zones which have NS records which refer to
                        CNAMEs, DNAMEs or don't have address record (class IN
                        only).  Reject UPDATEs which would cause the zone
                        to fail the above checks if committed. [RT #20678]

2799.   [cleanup]       Changed the "secure-to-insecure" option to
                        "dnssec-secure-to-insecure", and "dnskey-ksk-only"
                        to "dnssec-dnskey-kskonly", for clarity. [RT #20586]

2798.   [bug]           Addressed bugs in managed-keys initialization
                        and rollover. [RT #20683]
```

(continues on next page)

(continued from previous page)

```
2797.   [bug]           Don't decrement the dispatch manager's maxbuffers.
                        [RT #20613]

2796.   [bug]           Missing dns_rdataset_disassociate() call in
                        dns_nsec3_delnsec3sx(). [RT #20681]

2795.   [cleanup]       Add text to differentiate "update with no effect"
                        log messages. [RT #18889]

2794.   [bug]           Install <isc/namespace.h>.  [RT #20677]

2793.   [func]          Add "autosign" and "metadata" tests to the
                        automatic tests. [RT #19946]

2792.   [func]          "filter-aaaa-on-v4" can now be set in view
                        options (if compiled in).  [RT #20635]

2791.   [bug]           The installation of isc-config.sh was broken.
                        [RT #20667]

2790.   [bug]           Handle DS queries to stub zones. [RT #20440]

2789.   [bug]           Fixed an INSIST in dispatch.c [RT #20576]

2788.   [bug]           dnssec-signzone could sign with keys that were
                        not requested [RT #20625]

2787.   [bug]           Spurious log message when zone keys were
                        dynamically reconfigured. [RT #20659]

2786.   [bug]           Additional could be promoted to answer. [RT #20663]
```

```
        --- 9.7.0b3 released ---

2785.   [bug]           Revoked keys could fail to self-sign [RT #20652]

2784.   [bug]           TC was not always being set when required glue was
                        dropped. [RT #20655]

2783.   [func]          Return minimal responses to EDNS/UDP queries with a UDP
                        buffer size of 512 or less.  [RT #20654]

2782.   [port]          win32: use getaddrinfo() for hostname lookups.
                        [RT #20650]

2781.   [bug]           Inactive keys could be used for signing. [RT #20649]

2780.   [bug]           dnssec-keygen -A none didn't properly unset the
                        activation date in all cases. [RT #20648]

2779.   [bug]           Dynamic key revocation could fail. [RT #20644]
```

(continues on next page)

(continued from previous page)

```
2778.   [bug]           dnssec-signzone could fail when a key was revoked
                        without deleting the unrevoked version. [RT #20638]

2777.   [contrib]       DLZ MYSQL auto reconnect support discovery was wrong.

2776.   [bug]           Change #2762 was not correct. [RT #20647]

2775.   [bug]           Accept RSASHA256 and RSASHA512 as NSEC3 compatible
                        in dnssec-keyfromlabel. [RT #20643]

2774.   [bug]           Existing cache DB wasn't being reused after
                        reconfiguration. [RT #20629]

2773.   [bug]           In autosigned zones, the SOA could be signed
                        with the KSK. [RT #20628]

2772.   [security]      When validating, track whether pending data was from
                        the additional section or not and only return it if
                        validates as secure. [RT #20438]

2771.   [bug]           dnssec-signzone: DNSKEY records could be
                        corrupted when importing from key files [RT #20624]

2770.   [cleanup]       Add log messages to resolver.c to indicate events
                        causing FORMERR responses. [RT #20526]

2769.   [cleanup]       Change #2742 was incomplete. [RT #19589]

2768.   [bug]           dnssec-signzone: -S no longer implies -g [RT #20568]

2767.   [bug]           named could crash on startup if a zone was
                        configured with auto-dnssec and there was no
                        key-directory. [RT #20615]

2766.   [bug]           isc_socket_fdwatchpoke() should only update the
                        socketmgr state if the socket is not pending on a
                        read or write.  [RT #20603]

2765.   [bug]           Skip masters for which the TSIG key cannot be found.
                        [RT #20595]

2764.   [bug]           "rndc-confgen -a" could trigger a REQUIRE. [RT #20610]

2763.   [bug]           "rndc sign" didn't create an NSEC chain. [RT #20591]

2762.   [bug]           DLV validation failed with a local slave DLV zone.
                        [RT #20577]

2761.   [cleanup]       Enable internal symbol table for backtrace only for
                        systems that are known to work.  Currently, BSD
                        variants, Linux and Solaris are supported. [RT #20202]
```

(continues on next page)

(continued from previous page)

```
2760.   [cleanup]       Corrected named-compilezone usage summary. [RT #20533]

2759.   [doc]           Add information about .jbk/.jnw files to
                        the ARM. [RT #20303]

2758.   [bug]           win32: Added a workaround for a windows 2008 bug
                        that could cause the UDP client handler to shut
                        down. [RT #19176]

2757.   [bug]           dig: assertion failure could occur in connect
                        timeout. [RT #20599]

2756.   [bug]           Fixed corrupt logfile message in update.c. [RT #20597]

2755.   [placeholder]

2754.   [bug]           Secure-to-insecure transitions failed when zone
                        was signed with NSEC3. [RT #20587]

2753.   [bug]           Removed an unnecessary warning that could appear when
                        building an NSEC chain. [RT #20589]

2752.   [bug]           Locking violation. [RT #20587]

2751.   [bug]           Fixed a memory leak in dnssec-keyfromlabel. [RT #20588]

2750.   [bug]           dig: assertion failure could occur when a server
                        didn't have an address. [RT #20579]

2749.   [bug]           ixfr-from-differences generated a non-minimal ixfr
                        for NSEC3 signed zones. [RT #20452]

2748.   [func]          Identify bad answers from GTLD servers and treat them
                        as referrals. [RT #18884]

2747.   [bug]           Journal roll forwards failed to set the re-signing
                        time of RRSIGs correctly. [RT #20541]

2746.   [port]          hpux: address signed/unsigned expansion mismatch of
                        dns_rbtnode_t.nsec. [RT #20542]

2745.   [bug]           configure script didn't probe the return type of
                        gai_strerror(3) correctly. [RT #20573]

2744.   [func]          Log if a query was over TCP. [RT #19961]

2743.   [bug]           RRSIG could be incorrectly set in the NSEC3 record
                        for a insecure delegation.
```

```
        --- 9.7.0b2 released ---

2742.   [cleanup]       Clarify some DNSSEC-related log messages in
```

(continues on next page)

(continued from previous page)

```
                        validator.c. [RT #19589]

2741.   [func]          Allow the dnssec-keygen progress messages to be
                        suppressed (dnssec-keygen -q).  Automatically
                        suppress the progress messages when stdin is not
                        a tty. [RT #20474]

2740.   [placeholder]

2739.   [cleanup]       Clean up API for initializing and clearing trust
                        anchors for a view. [RT #20211]

2738.   [func]          Add RSASHA256 and RSASHA512 tests to the dnssec system
                        test. [RT #20453]

2737.   [func]          UPDATE requests can leak existence information.
                        [RT #17261]

2736.   [func]          Improve the performance of NSEC signed zones with
                        more than a normal amount of glue below a delegation.
                        [RT #20191]

2735.   [bug]           dnssec-signzone could fail to read keys
                        that were specified on the command line with
                        full paths, but weren't in the current
                        directory. [RT #20421]

2734.   [port]          cygwin: arpaname did not compile. [RT #20473]

2733.   [cleanup]       Clean up coding style in pkcs11-* tools. [RT #20355]

2732.   [func]          Add optional filter-aaaa-on-v4 option, available
                        if built with './configure --enable-filter-aaaa'.
                        Filters out AAAA answers to clients connecting
                        via IPv4.  (This is NOT recommended for general
                        use.) [RT #20339]

2731.   [func]          Additional work on change 2709.  The key parser
                        will now ignore unrecognized fields when the
                        minor version number of the private key format
                        has been increased.  It will reject any key with
                        the major version number increased. [RT #20310]

2730.   [func]          Have dnssec-keygen display a progress indication
                        a la 'openssl genrsa' on standard error. Note
                        when the first '.' is followed by a long stop
                        one has the choice between slow generation vs.
                        poor random quality, i.e., '-r /dev/urandom'.
                        [RT #20284]

2729.   [func]          When constructing a CNAME from a DNAME use the DNAME
                        TTL. [RT #20451]
```

(continues on next page)

(continued from previous page)

```
2728.   [bug]           dnssec-keygen, dnssec-keyfromlabel and
                        dnssec-signzone now warn immediately if asked to
                        write into a nonexistent directory. [RT #20278]

2727.   [func]          The 'key-directory' option can now specify a relative
                        path. [RT #20154]

2726.   [func]          Added support for SHA-2 DNSSEC algorithms,
                        RSASHA256 and RSASHA512. [RT #20023]

2725.   [doc]           Added information about the file "managed-keys.bind"
                        to the ARM. [RT #20235]

2724.   [bug]           Updates to a existing node in secure zone using NSEC
                        were failing. [RT #20448]

2723.   [bug]           isc_base32_totext(), isc_base32hex_totext(), and
                        isc_base64_totext(), didn't always mark regions of
                        memory as fully consumed after conversion.  [RT #20445]

2722.   [bug]           Ensure that the memory associated with the name of
                        a node in a rbt tree is not altered during the life
                        of the node. [RT #20431]

2721.   [port]          Have dst__entropy_status() prime the random number
                        generator. [RT #20369]

2720.   [bug]           RFC 5011 trust anchor updates could trigger an
                        assert if the DNSKEY record was unsigned. [RT #20406]

2719.   [func]          Skip trusted/managed keys for unsupported algorithms.
                        [RT #20392]

2718.   [bug]           The space calculations in opensslrsa_todns() were
                        incorrect. [RT #20394]

2717.   [bug]           named failed to update the NSEC/NSEC3 record when
                        the last private type record was removed as a result
                        of completing the signing the zone with a key.
                        [RT #20399]

2716.   [bug]           nslookup debug mode didn't return the ttl. [RT #20414]
```

```
        --- 9.7.0b1 released ---

2715.   [bug]           Require OpenSSL support to be explicitly disabled.
                        [RT #20288]

2714.   [port]          aix/powerpc: 'asm("ics");' needs non standard assembler
                        flags.
```

(continues on next page)

(continued from previous page)

```
2713.   [bug]           powerpc: atomic operations missing asm("ics") /
                        __isync() calls.

2712.   [func]          New 'auto-dnssec' zone option allows zone signing
                        to be fully automated in zones configured for
                        dynamic DNS.  'auto-dnssec allow;' permits a zone
                        to be signed by creating keys for it in the
                        key-directory and using 'rndc sign <zone>'.
                        'auto-dnssec maintain;' allows that too, plus it
                        also keeps the zone's DNSSEC keys up to date
                        according to their timing metadata. [RT #19943]

2711.   [port]          win32: Add the bin/pkcs11 tools into the full
                        build. [RT #20372]

2710.   [func]          New 'dnssec-signzone -x' flag and 'dnskey-ksk-only'
                        zone option cause a zone to be signed with only KSKs
                        signing the DNSKEY RRset, not ZSKs.  This reduces
                        the size of a DNSKEY answer.  [RT #20340]

2709.   [func]          Added some data fields, currently unused, to the
                        private key file format, to allow implementation
                        of explicit key rollover in a future release
                        without impairing backward or forward compatibility.
                        [RT #20310]

2708.   [func]          Insecure to secure and NSEC3 parameter changes via
                        update are now fully supported and no longer require
                        defines to enable.  We now no longer overload the
                        NSEC3PARAM flag field, nor the NSEC OPT bit at the
                        apex.  Secure to insecure changes are controlled by
                        by the named.conf option 'secure-to-insecure'.

                        Warning: If you had previously enabled support by
                        adding defines at compile time to BIND 9.6 you should
                        ensure that all changes that are in progress have
                        completed prior to upgrading to BIND 9.7.  BIND 9.7
                        is not backwards compatible.

2707.   [func]          dnssec-keyfromlabel no longer require engine name
                        to be specified in the label if there is a default
                        engine or the -E option has been used.  Also, it
                        now uses default algorithms as dnssec-keygen does
                        (i.e., RSASHA1, or NSEC3RSASHA1 if -3 is used).
                        [RT #20371]

2706.   [bug]           Loading a zone with a very large NSEC3 salt could
                        trigger an assert. [RT #20368]

2705.   [placeholder]

2704.   [bug]           Serial of dynamic and stub zones could be inconsistent
```

(continues on next page)

(continued from previous page)

```
                        with their SOA serial.  [RT #19387]

2703.   [func]          Introduce an OpenSSL "engine" argument with -E
                        for all binaries which can take benefit of
                        crypto hardware. [RT #20230]

2702.   [func]          Update PKCS#11 tools (bin/pkcs11) [RT #20225 & all]

2701.   [doc]           Correction to ARM: hmac-md5 is no longer the only
                        supported TSIG key algorithm. [RT #18046]

2700.   [doc]           The match-mapped-addresses option is discouraged.
                        [RT #12252]

2699.   [bug]           Missing lock in rbtdb.c. [RT #20037]

2698.   [placeholder]

2697.   [port]          win32: ensure that S_IFMT, S_IFDIR, S_IFCHR and
                        S_IFREG are defined after including <isc/stat.h>.
                        [RT #20309]

2696.   [bug]           named failed to successfully process some valid
                        acl constructs. [RT #20308]

2695.   [func]          DHCP/DDNS - update fdwatch code for use by
                        DHCP.  Modify the api to isc_sockfdwatch_t (the
                        callback function for isc_socket_fdwatchcreate)
                        to include information about the direction (read
                        or write) and add isc_socket_fdwatchpoke.
                        [RT #20253]

2694.   [bug]           Reduce default NSEC3 iterations from 100 to 10.
                        [RT #19970]

2693.   [port]          Add some noreturn attributes. [RT #20257]

2692.   [port]          win32: 32/64 bit cleanups. [RT #20335]

2691.   [func]          dnssec-signzone: retain the existing NSEC or NSEC3
                        chain when re-signing a previously-signed zone.
                        Use -u to modify NSEC3 parameters or switch
                        between NSEC and NSEC3. [RT #20304]

2690.   [bug]           win32: fix isc_thread_key_getspecific() prototype.
                        [RT #20315]

2689.   [bug]           Correctly handle snprintf result. [RT #20306]

2688.   [bug]           Use INTERFACE_F_POINTTOPOINT, not IFF_POINTOPOINT,
                        to decide to fetch the destination address. [RT #20305]
```

(continues on next page)

(continued from previous page)

```
2687.   [bug]           Fixed dnssec-signzone -S handling of revoked keys.
                        Also, added warnings when revoking a ZSK, as this is
                        not defined by protocol (but is legal).  [RT #19943]

2686.   [bug]           dnssec-signzone should clean the old NSEC chain when
                        signing with NSEC3 and vice versa. [RT #20301]

2685.   [contrib]       Update contrib/zkt to version 0.99c. [RT #20054]

2684.   [cleanup]       dig: formalize +ad and +cd as synonyms for
                        +adflag and +cdflag.  [RT #19305]

2683.   [bug]           dnssec-signzone should clean out old NSEC3 chains when
                        the NSEC3 parameters used to sign the zone change.
                        [RT #20246]

2682.   [bug]           "configure --enable-symtable=all" failed to
                        build. [RT #20282]

2681.   [bug]           IPSECKEY RR of gateway type 3 was not correctly
                        decoded. [RT #20269]

2680.   [func]          Move contrib/pkcs11-keygen to bin/pkcs11. [RT #20067]

2679.   [func]          dig -k can now accept TSIG keys in named.conf
                        format.  [RT #20031]

2678.   [func]          Treat DS queries as if "minimal-response yes;"
                        was set. [RT #20258]

2677.   [func]          Changes to key metadata behavior:
                        - Keys without "publish" or "active" dates set will
                          no longer be used for smart signing.  However,
                          those dates will be set to "now" by default when
                          a key is created; to generate a key but not use
                          it yet, use dnssec-keygen -G.
                        - New "inactive" date (dnssec-keygen/settime -I)
                          sets the time when a key is no longer used for
                          signing but is still published.
                        - The "unpublished" date (-U) is deprecated in
                          favor of "deleted" (-D).
                        [RT #20247]

2676.   [bug]           --with-export-installdir should have been
                        --with-export-includedir. [RT #20252]

2675.   [bug]           dnssec-signzone could crash if the key directory
                        did not exist. [RT #20232]
```

```
        --- 9.7.0a3 released ---

2674.   [bug]           "dnssec-lookaside auto;" crashed if named was built
```

(continues on next page)

(continued from previous page)

```
                        without openssl. [RT #20231]

2673.   [bug]           The managed-keys.bind zone file could fail to
                        load due to a spurious result from sync_keyzone()
                        [RT #20045]

2672.   [bug]           Don't enable searching in 'host' when doing reverse
                        lookups. [RT #20218]

2671.   [bug]           Add support for PKCS#11 providers not returning
                        the public exponent in RSA private keys
                        (OpenCryptoki for instance) in
                        dnssec-keyfromlabel. [RT #19294]

2670.   [bug]           Unexpected connect failures failed to log enough
                        information to be useful. [RT #20205]

2669.   [func]          Update PKCS#11 support to support Keyper HSM.
                        Update PKCS#11 patch to be against openssl-0.9.8i.

2668.   [func]          Several improvements to dnssec-* tools, including:
                        - dnssec-keygen and dnssec-settime can now set key
                          metadata fields 0 (to unset a value, use "none")
                        - dnssec-revoke sets the revocation date in
                          addition to the revoke bit
                        - dnssec-settime can now print individual metadata
                          fields instead of always printing all of them,
                          and can print them in unix epoch time format for
                          use by scripts
                        [RT #19942]

2667.   [func]          Add support for logging stack backtrace on assertion
                        failure (not available for all platforms). [RT #19780]

2666.   [func]          Added an 'options' argument to dns_name_fromstring()
                        (API change from 9.7.0a2). [RT #20196]

2665.   [func]          Clarify syntax for managed-keys {} statement, add
                        ARM documentation about RFC 5011 support. [RT #19874]

2664.   [bug]           create_keydata() and minimal_update() in zone.c
                        didn't properly check return values for some
                        functions.  [RT #19956]

2663.   [func]          win32:  allow named to run as a service using
                        "NT AUTHORITY\LocalService" as the account. [RT #19977]

2662.   [bug]           lwres_getipnodebyname() and lwres_getipnodebyaddr()
                        returned a misleading error code when lwresd was
                        down. [RT #20028]

2661.   [bug]           Check whether socket fd exceeds FD_SETSIZE when
```

(continues on next page)

(continued from previous page)

```
                        creating lwres context. [RT #20029]

2660.   [func]          Add a new set of DNS libraries for non-BIND9
                        applications.  See README.libdns. [RT #19369]

2659.   [doc]           Clarify dnssec-keygen doc: key name must match zone
                        name for DNSSEC keys. [RT #19938]

2658.   [bug]           dnssec-settime and dnssec-revoke didn't process
                        key file paths correctly. [RT #20078]

2657.   [cleanup]       Lower "journal file <path> does not exist, creating it"
                        log level to debug 1. [RT #20058]

2656.   [func]          win32: add a "tools only" check box to the installer
                        which causes it to only install dig, host, nslookup,
                        nsupdate and relevant DLLs.  [RT #19998]

2655.   [doc]           Document that key-directory does not affect
                        bind.keys, rndc.key or session.key.  [RT #20155]

2654.   [bug]           Improve error reporting on duplicated names for
                        deny-answer-xxx. [RT #20164]

2653.   [bug]           Treat ENGINE_load_private_key() failures as key
                        not found rather than out of memory.  [RT #18033]

2652.   [func]          Provide more detail about what record is being
                        deleted. [RT #20061]

2651.   [bug]           Dates could print incorrectly in K*.key files on
                        64-bit systems. [RT #20076]

2650.   [bug]           Assertion failure in dnssec-signzone when trying
                        to read keyset-* files. [RT #20075]

2649.   [bug]           Set the domain for forward only zones. [RT #19944]

2648.   [port]          win32: isc_time_seconds() was broken. [RT #19900]

2647.   [bug]           Remove unnecessary SOA updates when a new KSK is
                        added. [RT #19913]

2646.   [bug]           Incorrect cleanup on error in socket.c. [RT #19987]

2645.   [port]          "gcc -m32" didn't work on amd64 and x86_64 platforms
                        which default to 64 bits. [RT #19927]
```

```
        --- 9.7.0a2 released ---

2644.   [bug]           Change #2628 caused a regression on some systems;
                        named was unable to write the PID file and would
```

(continues on next page)

(continued from previous page)

```
                        fail on startup. [RT #20001]

2643.   [bug]           Stub zones interacted badly with NSEC3 support.
                        [RT #19777]

2642.   [bug]           nsupdate could dump core on solaris when reading
                        improperly formatted key files.  [RT #20015]

2641.   [bug]           Fixed an error in parsing update-policy syntax,
                        added a regression test to check it. [RT #20007]

2640.   [security]      A specially crafted update packet will cause named
                        to exit. [RT #20000]

2639.   [bug]           Silence compiler warnings in gssapi code. [RT #19954]

2638.   [bug]           Install arpaname. [RT #19957]

2637.   [func]          Rationalize dnssec-signzone's signwithkey() calling.
                        [RT #19959]

2636.   [func]          Simplify zone signing and key maintenance with the
                        dnssec-* tools.  Major changes:
                        - all dnssec-* tools now take a -K option to
                          specify a directory in which key files will be
                          stored
                        - DNSSEC can now store metadata indicating when
                          they are scheduled to be published, activated,
                          revoked or removed; these values can be set by
                          dnssec-keygen or overwritten by the new
                          dnssec-settime command
                        - dnssec-signzone -S (for "smart") option reads key
                          metadata and uses it to determine automatically
                          which keys to publish to the zone, use for
                          signing, revoke, or remove from the zone
                        [RT #19816]

2635.   [bug]           isc_inet_ntop() incorrectly handled 0.0/16 addresses.
                        [RT #19716]

2634.   [port]          win32: Add support for libxml2, enable
                        statschannel. [RT #19773]

2633.   [bug]           Handle 15 bit rand() functions. [RT #19783]

2632.   [func]          util/kit.sh: warn if documentation appears to be out of
                        date.  [RT #19922]

2631.   [bug]           Handle "//", "/./" and "/../" in mkdirpath().
                        [RT #19926 ]

2630.   [func]          Improved syntax for DDNS autoconfiguration:  use
```

(continues on next page)

(continued from previous page)

```
                        "update-policy local;" to switch on local DDNS in a
                        zone. (The "ddns-autoconf" option has been removed.)
                        [RT #19875]

2629.   [port]          Check for seteuid()/setegid(), use setresuid()/
                        setresgid() if not present. [RT #19932]

2628.   [port]          linux: Allow /var/run/named/named.pid to be opened
                        at startup with reduced capabilities in operation.
                        [RT #19884]

2627.   [bug]           Named aborted if the same key was included in
                        trusted-keys more than once. [RT #19918]

2626.   [bug]           Multiple trusted-keys could trigger an assertion
                        failure. [RT #19914]

2625.   [bug]           Missing UNLOCK in rbtdb.c. [RT #19865]

2624.   [func]          'named-checkconf -p' will print out the parsed
                        configuration. [RT #18871]

2623.   [bug]           Named started searches for DS non-optimally. [RT #19915]

2622.   [bug]           Printing of named.conf grammar was broken. [RT #19919]

2621.   [doc]           Made copyright boilerplate consistent.  [RT #19833]

2620.   [bug]           Delay thawing the zone until the reload of it has
                        completed successfully.  [RT #19750]

2619.   [func]          Add support for RFC 5011, automatic trust anchor
                        maintenance.  The new "managed-keys" statement can
                        be used in place of "trusted-keys" for zones which
                        support this protocol.  (Note: this syntax is
                        expected to change prior to 9.7.0 final.) [RT #19248]

2618.   [bug]           The sdb and sdlz db_interator_seek() methods could
                        loop infinitely. [RT #19847]

2617.   [bug]           ifconfig.sh failed to emit an error message when
                        run from the wrong location. [RT #19375]

2616.   [bug]           'host' used the nameservers from resolv.conf even
                        when a explicit nameserver was specified. [RT #19852]

2615.   [bug]           "__attribute__((unused))" was in the wrong place
                        for ia64 gcc builds. [RT #19854]

2614.   [port]          win32: 'named -v' should automatically be executed
                        in the foreground. [RT #19844]
```

(continues on next page)

(continued from previous page)

```
2613.   [placeholder]
```

```
        --- 9.7.0a1 released ---

2612.   [func]          Add default values for the arguments to
                        dnssec-keygen.  Without arguments, it will now
                        generate a 1024-bit RSASHA1 zone-signing key,
                        or with the -f KSK option, a 2048-bit RSASHA1
                        key-signing key. [RT #19300]

2611.   [func]          Add -l option to dnssec-dsfromkey to generate
                        DLV records instead of DS records. [RT #19300]

2610.   [port]          sunos: Change #2363 was not complete. [RT #19796]

2609.   [func]          Simplify the configuration of dynamic zones:
                        - add ddns-confgen command to generate
                          configuration text for named.conf
                        - add zone option "ddns-autoconf yes;", which
                          causes named to generate a TSIG session key
                          and allow updates to the zone using that key
                        - add '-l' (localhost) option to nsupdate, which
                          causes nsupdate to connect to a locally-running
                          named process using the session key generated
                          by named
                        [RT #19284]

2608.   [func]          Perform post signing verification checks in
                        dnssec-signzone.  These can be disabled with -P.

                        The post sign verification test ensures that for each
                        algorithm in use there is at least one non revoked
                        self signed KSK key.  That all revoked KSK keys are
                        self signed.  That all records in the zone are signed
                        by the algorithm.  [RT #19653]

2607.   [bug]           named could incorrectly delete NSEC3 records for
                        empty nodes when processing a update request.
                        [RT #19749]

2606.   [bug]           "delegation-only" was not being accepted in
                        delegation-only type zones. [RT #19717]

2605.   [bug]           Accept DS responses from delegation only zones.
                        [RT # 19296]

2604.   [func]          Add support for DNS rebinding attack prevention through
                        new options, deny-answer-addresses and
                        deny-answer-aliases.  Based on contributed code from
                        JD Nurmi, Google. [RT #18192]

2603.   [port]          win32: handle .exe extension of named-checkzone and
```

(continues on next page)

(continued from previous page)

```
                      named-comilezone argv[0] names under windows.
                      [RT #19767]

2602.   [port]        win32: fix debugging command line build of libisccfg.
                      [RT #19767]

2601.   [doc]         Mention file creation mode mask in the
                      named manual page.

2600.   [doc]         ARM: miscellaneous reformatting for different
                      page widths. [RT #19574]

2599.   [bug]         Address rapid memory growth when validation fails.
                      [RT #19654]

2598.   [func]        Reserve the -F flag. [RT #19657]

2597.   [bug]         Handle a validation failure with a insecure delegation
                      from a NSEC3 signed master/slave zone.  [RT #19464]

2596.   [bug]         Stale tree nodes of cache/dynamic rbtdb could stay
                      long, leading to inefficient memory usage or rejecting
                      newer cache entries in the worst case. [RT #19563]

2595.   [bug]         Fix unknown extended rcodes in dig. [RT #19625]

2594.   [func]        Have rndc warn if using its default configuration
                      file when the key file also exists. [RT #19424]

2593.   [bug]         Improve a corner source of SERVFAILs [RT #19632]

2592.   [bug]         Treat "any" as a type in nsupdate. [RT #19455]

2591.   [bug]         named could die when processing a update in
                      removed_orphaned_ds(). [RT #19507]

2590.   [func]        Report zone/class of "update with no effect".
                      [RT #19542]

2589.   [bug]         dns_db_unregister() failed to clear '*dbimp'.
                      [RT #19626]

2588.   [bug]         SO_REUSEADDR could be set unconditionally after failure
                      of bind(2) call.  This should be rare and mostly
                      harmless, but may cause interference with other
                      processes that happen to use the same port. [RT #19642]

2587.   [func]        Improve logging by reporting serial numbers for
                      when zone serial has gone backwards or unchanged.
                      [RT #19506]

2586.   [bug]         Missing cleanup of SIG rdataset in searching a DLZ DB
```

(continues on next page)

(continued from previous page)

```
                        or SDB. [RT #19577]

2585.   [bug]           Uninitialized socket name could be referenced via a
                        statistics channel, triggering an assertion failure in
                        XML rendering. [RT #19427]

2584.   [bug]           alpha: gcc optimization could break atomic operations.
                        [RT #19227]

2583.   [port]          netbsd: provide a control to not add the compile
                        date to the version string, -DNO_VERSION_DATE.

2582.   [bug]           Don't emit warning log message when we attempt to
                        remove non-existent journal. [RT #19516]

2581.   [contrib]       dlz/mysql set MYSQL_OPT_RECONNECT option on connection.
                        Requires MySQL 5.0.19 or later. [RT #19084]

2580.   [bug]           UpdateRej statistics counter could be incremented twice
                        for one rejection. [RT #19476]

2579.   [bug]           DNSSEC lookaside validation failed to handle unknown
                        algorithms. [RT #19479]

2578.   [bug]           Changed default sig-signing-type to 65534, because
                        65535 turns out to be reserved.  [RT #19477]

2577.   [doc]           Clarified some statistics counters. [RT #19454]

2576.   [bug]           NSEC record were not being correctly signed when
                        a zone transitions from insecure to secure.
                        Handle such incorrectly signed zones. [RT #19114]

2575.   [func]          New functions dns_name_fromstring() and
                        dns_name_tostring(), to simplify conversion
                        of a string to a dns_name structure and vice
                        versa. [RT #19451]

2574.   [doc]           Document nsupdate -g and -o. [RT #19351]

2573.   [bug]           Replacing a non-CNAME record with a CNAME record in a
                        single transaction in a signed zone failed. [RT #19397]

2572.   [func]          Simplify DLV configuration, with a new option
                        "dnssec-lookaside auto;"  This is the equivalent
                        of "dnssec-lookaside . trust-anchor dlv.isc.org;"
                        plus setting a trusted-key for dlv.isc.org.

                        Note: The trusted key is hard-coded into named,
                        but is also stored in (and can be overridden
                        by) $sysconfdir/bind.keys.  As the ISC DLV key
                        rolls over it can be kept up to date by replacing
```

(continues on next page)

(continued from previous page)

```
                        the bind.keys file with a key downloaded from
                        https://www.isc.org/solutions/dlv. [RT #18685]

2571.   [func]          Add a new tool "arpaname" which translates IP addresses
                        to the corresponding IN-ADDR.ARPA or IP6.ARPA name.
                        [RT #18976]

2570.   [func]          Log the destination address the query was sent to.
                        [RT #19209]

2569.   [func]          Move journalprint, nsec3hash, and genrandom
                        commands from bin/tests into bin/tools;
                        "make install" will put them in $sbindir. [RT #19301]

2568.   [bug]           Report when the write to indicate a otherwise
                        successful start fails. [RT #19360]

2567.   [bug]           dst__privstruct_writefile() could miss write errors.
                        write_public_key() could miss write errors.
                        dnssec-dsfromkey could miss write errors.
                        [RT #19360]

2566.   [cleanup]       Clarify logged message when an insecure DNSSEC
                        response arrives from a zone thought to be secure:
                        "insecurity proof failed" instead of "not
                        insecure". [RT #19400]

2565.   [func]          Add support for HIP record.  Includes new functions
                        dns_rdata_hip_first(), dns_rdata_hip_next()
                        and dns_rdata_hip_current().  [RT #19384]

2564.   [bug]           Only take EDNS fallback steps when processing timeouts.
                        [RT #19405]

2563.   [bug]           Dig could leak a socket causing it to wait forever
                        to exit. [RT #19359]

2562.   [doc]           ARM: miscellaneous improvements, reorganization,
                        and some new content.

2561.   [doc]           Add isc-config.sh(1) man page. [RT #16378]

2560.   [bug]           Add #include <config.h> to iptable.c. [RT #18258]

2559.   [bug]           dnssec-dsfromkey could compute bad DS records when
                        reading from a K* files.  [RT #19357]

2558.   [func]          Set the ownership of missing directories created
                        for pid-file if -u has been specified on the command
                        line. [RT #19328]

2557.   [cleanup]       PCI compliance:
```

(continues on next page)

(continued from previous page)

```
                        * new libisc log module file
                        * isc_dir_chroot() now also changes the working
                          directory to "/".
                        * additional INSISTs
                        * additional logging when files can't be removed.

2556.   [port]          Solaris: mkdir(2) on tmpfs filesystems does not do the
                        error checks in the correct order resulting in the
                        wrong error code sometimes being returned. [RT #19249]

2555.   [func]          dig: when emitting a hex dump also display the
                        corresponding characters. [RT #19258]

2554.   [bug]           Validation of uppercase queries from NSEC3 zones could
                        fail. [RT #19297]

2553.   [bug]           Reference leak on DNSSEC validation errors. [RT #19291]

2552.   [bug]           zero-no-soa-ttl-cache was not being honored.
                        [RT #19340]

2551.   [bug]           Potential Reference leak on return. [RT #19341]

2550.   [bug]           Check --with-openssl=<path> finds <openssl/opensslv.h>.
                        [RT #19343]

2549.   [port]          linux: define NR_OPEN if not currently defined.
                        [RT #19344]

2548.   [bug]           Install iterated_hash.h. [RT #19335]

2547.   [bug]           openssl_link.c:mem_realloc() could reference an
                        out-of-range area of the source buffer.  New public
                        function isc_mem_reallocate() was introduced to address
                        this bug. [RT #19313]

2546.   [func]          Add --enable-openssl-hash configure flag to use
                        OpenSSL (in place of internal routine) for hash
                        functions (MD5, SHA[12] and HMAC). [RT #18815]

2545.   [doc]           ARM: Legal hostname checking (check-names) is
                        for SRV RDATA too. [RT #19304]

2544.   [cleanup]       Removed unused structure members in adb.c. [RT #19225]

2543.   [contrib]       Update contrib/zkt to version 0.98. [RT #19113]

2542.   [doc]           Update the description of dig +adflag. [RT #19290]

2541.   [bug]           Conditionally update dispatch manager statistics.
                        [RT #19247]
```

(continues on next page)

(continued from previous page)

```
2540.   [func]          Add a nibble mode to $GENERATE. [RT #18872]

2539.   [security]      Update the interaction between recursion, allow-query,
                        allow-query-cache and allow-recursion.  [RT #19198]

2538.   [bug]           cache/ADB memory could grow over max-cache-size,
                        especially with threads and smaller max-cache-size
                        values. [RT #19240]

2537.   [func]          Added more statistics counters including those on socket
                        I/O events and query RTT histograms. [RT #18802]

2536.   [cleanup]       Silence some warnings when -Werror=format-security is
                        specified. [RT #19083]

2535.   [bug]           dig +showsearch and +trace interacted badly. [RT #19091]

2534.   [func]          Check NAPTR records regular expressions and
                        replacement strings to ensure they are syntactically
                        valid and consistent. [RT #18168]

2533.   [doc]           ARM: document @ (at-sign). [RT #17144]

2532.   [bug]           dig: check the question section of the response to
                        see if it matches the asked question. [RT #18495]

2531.   [bug]           Change #2207 was incomplete. [RT #19098]

2530.   [bug]           named failed to reject insecure to secure transitions
                        via UPDATE. [RT #19101]

2529.   [cleanup]       Upgrade libtool to silence complaints from recent
                        version of autoconf. [RT #18657]

2528.   [cleanup]       Silence spurious configure warning about
                        --datarootdir [RT #19096]

2527.   [placeholder]

2526.   [func]          New named option "attach-cache" that allows multiple
                        views to share a single cache to save memory and
                        improve lookup efficiency.  Based on contributed code
                        from Barclay Osborn, Google. [RT #18905]

2525.   [func]          New logging category "query-errors" to provide detailed
                        internal information about query failures, especially
                        about server failures. [RT #19027]

2524.   [port]          sunos: dnssec-signzone needs strtoul(). [RT #19129]

2523.   [bug]           Random type rdata freed by dns_nsec_typepresent().
                        [RT #19112]
```

(continues on next page)

(continued from previous page)

```
2522.   [security]      Handle -1 from DSA_do_verify() and EVP_VerifyFinal().

2521.   [bug]           Improve epoll cross compilation support. [RT #19047]

2520.   [bug]           Update xml statistics version number to 2.0 as change
                        #2388 made the schema incompatible to the previous
                        version. [RT #19080]

2519.   [bug]           dig/host with -4 or -6 didn't work if more than two
                        nameserver addresses of the excluded address family
                        preceded in resolv.conf. [RT #19081]

2518.   [func]          Add support for the new CERT types from RFC 4398.
                        [RT #19077]

2517.   [bug]           dig +trace with -4 or -6 failed when it chose a
                        nameserver address of the excluded address type.
                        [RT #18843]

2516.   [bug]           glue sort for responses was performed even when not
                        needed. [RT #19039]

2515.   [port]          win32: build dnssec-dsfromkey and dnssec-keyfromlabel.
                        [RT #19063]

2514.   [bug]           dig/host failed with -4 or -6 when resolv.conf contains
                        a nameserver of the excluded address family.
                        [RT #18848]

2513.   [bug]           Fix windows cli build. [RT #19062]

2512.   [func]          Print a summary of the cached records which make up
                        the negative response.  [RT #18885]

2511.   [cleanup]       dns_rdata_tofmttext() add const to linebreak.
                        [RT #18885]

2510.   [bug]           "dig +sigchase" could trigger REQUIRE failures.
                        [RT #19033]

2509.   [bug]           Specifying a fixed query source port was broken.
                        [RT #19051]

2508.   [placeholder]

2507.   [func]          Log the recursion quota values when killing the
                        oldest query or refusing to recurse due to quota.
                        [RT #19022]

2506.   [port]          solaris: Check at configure time if
                        hack_shutup_pthreadonceinit is needed. [RT #19037]
```

(continues on next page)

(continued from previous page)

```
2505.   [port]          Treat amd64 similarly to x86_64 when determining
                        atomic operation support. [RT #19031]

2504.   [bug]           Address race condition in the socket code. [RT #18899]

2503.   [port]          linux: improve compatibility with Linux Standard
                        Base. [RT #18793]

2502.   [cleanup]       isc_radix: Improve compliance with coding style,
                        document function in <isc/radix.h>. [RT #18534]

2501.   [func]          $GENERATE now supports all rdata types.  Multi-field
                        rdata types need to be quoted.  See the ARM for
                        details. [RT #18368]

2500.   [contrib]       contrib/sdb/pgsql/zonetodb.c called non-existent
                        function. [RT #18582]

2499.   [port]          solaris: lib/lwres/getaddrinfo.c namespace clash.
                        [RT #18837]
```

```
        --- 9.6.0rc1 released ---

2498.   [bug]           Removed a bogus function argument used with
                        ISC_SOCKET_USE_POLLWATCH: it could cause compiler
                        warning or crash named with the debug 1 level
                        of logging. [RT #18917]

2497.   [bug]           Don't add RRSIG bit to NSEC3 bit map for insecure
                        delegation.

2496.   [bug]           Add sanity length checks to NSID option. [RT #18813]

2495.   [bug]           Tighten RRSIG checks. [RT #18795]

2494.   [bug]           isc/radix.h, dns/sdlz.h and dns/dlz.h were not being
                        installed. [RT #18826]

2493.   [bug]           The linux capabilities code was not correctly cleaning
                        up after itself. [RT #18767]

2492.   [func]          Rndc status now reports the number of cpus discovered
                        and the number of worker threads when running
                        multi-threaded. [RT #18273]

2491.   [func]          Attempt to re-use a local port if we are already using
                        the port. [RT #18548]

2490.   [port]          aix: work around a kernel bug where IPV6_RECVPKTINFO
                        is cleared when IPV6_V6ONLY is set. [RT #18785]
```

(continues on next page)

(continued from previous page)

```
2489.   [port]          solaris: Workaround Solaris's kernel bug about
                        /dev/poll:
                        http://bugs.opensolaris.org/view_bug.do?bug_id=6724237
                        Define ISC_SOCKET_USE_POLLWATCH at build time to enable
                        this workaround. [RT #18870]

2488.   [func]          Added a tool, dnssec-dsfromkey, to generate DS records
                        from keyset and .key files. [RT #18694]

2487.   [bug]           Give TCP connections longer to complete. [RT #18675]

2486.   [func]          The default locations for named.pid and lwresd.pid
                        are now /var/run/named/named.pid and
                        /var/run/lwresd/lwresd.pid respectively.

                        This allows the owner of the containing directory
                        to be set, for "named -u" support, and allows there
                        to be a permanent symbolic link in the path, for
                        "named -t" support.  [RT #18306]

2485.   [bug]           Change update's the handling of obscured RRSIG
                        records.  Not all orphaned DS records were being
                        removed. [RT #18828]

2484.   [bug]           It was possible to trigger a REQUIRE failure when
                        adding NSEC3 proofs to the response in
                        query_addwildcardproof().  [RT #18828]

2483.   [port]          win32: chroot() is not supported. [RT #18805]

2482.   [port]          libxml2: support versions 2.7.* in addition
                        to 2.6.*. [RT #18806]
```

```
        --- 9.6.0b1 released ---

2481.   [bug]           rbtdb.c:matchparams() failed to handle NSEC3 chain
                        collisions.  [RT #18812]

2480.   [bug]           named could fail to emit all the required NSEC3
                        records.  [RT #18812]

2479.   [bug]           xfrout:covers was not properly initialized. [RT #18801]

2478.   [bug]           'addresses' could be used uninitialized in
                        configure_forward(). [RT #18800]

2477.   [bug]           dig: the global option to print the command line is
                        +cmd not print_cmd.  Update the output to reflect
                        this. [RT #17008]

2476.   [doc]           ARM: improve documentation for max-journal-size and
                        ixfr-from-differences. [RT #15909] [RT #18541]
```

(continues on next page)

(continued from previous page)

```
2475.   [bug]           LRU cache cleanup under overmem condition could purge
                        particular entries more aggressively. [RT #17628]

2474.   [bug]           ACL structures could be allocated with insufficient
                        space, causing an array overrun. [RT #18765]

2473.   [port]          linux: raise the limit on open files to the possible
                        maximum value before spawning threads; 'files'
                        specified in named.conf doesn't seem to work with
                        threads as expected. [RT #18784]

2472.   [port]          linux: check the number of available cpu's before
                        calling chroot as it depends on "/proc". [RT #16923]

2471.   [bug]           named-checkzone was not reporting missing mandatory
                        glue when sibling checks were disabled. [RT #18768]

2470.   [bug]           Elements of the isc_radix_node_t could be incorrectly
                        overwritten.  [RT #18719]

2469.   [port]          solaris: Work around Solaris's select() limitations.
                        [RT #18769]

2468.   [bug]           Resolver could try unreachable servers multiple times.
                        [RT #18739]

2467.   [bug]           Failure of fcntl(F_DUPFD) wasn't logged. [RT #18740]

2466.   [doc]           ARM: explain max-cache-ttl 0 SERVFAIL issue.
                        [RT #18302]

2465.   [bug]           Adb's handling of lame addresses was different
                        for IPv4 and IPv6. [RT #18738]

2464.   [port]          linux: check that a capability is present before
                        trying to set it. [RT #18135]

2463.   [port]          linux: POSIX doesn't include the IPv6 Advanced Socket
                        API and glibc hides parts of the IPv6 Advanced Socket
                        API as a result.  This is stupid as it breaks how the
                        two halves (Basic and Advanced) of the IPv6 Socket API
                        were designed to be used but we have to live with it.
                        Define _GNU_SOURCE to pull in the IPv6 Advanced Socket
                        API. [RT #18388]

2462.   [doc]           Document -m (enable memory usage debugging)
                        option for dig. [RT #18757]

2461.   [port]          sunos: Change #2363 was not complete. [RT #17513]
```

```
        --- 9.6.0a1 released ---

2460.   [bug]           Don't call dns_db_getnsec3parameters() on the cache.
                        [RT #18697]

2459.   [contrib]       Import dnssec-zkt to contrib/zkt. [RT #18448]

2458.   [doc]           ARM: update and correction for max-cache-size.
                        [RT #18294]

2457.   [tuning]        max-cache-size is reverted to 0, the previous
                        default.  It should be safe because expired cache
                        entries are also purged. [RT #18684]

2456.   [bug]           In ACLs, ::/0 and 0.0.0.0/0 would both match any
                        address, regardless of family.  They now correctly
                        distinguish IPv4 from IPv6.  [RT #18559]

2455.   [bug]           Stop metadata being transferred via axfr/ixfr.
                        [RT #18639]

2454.   [func]          nsupdate: you can now set a default ttl. [RT #18317]

2453.   [bug]           Remove NULL pointer dereference in dns_journal_print().
                        [RT #18316]

2452.   [func]          Improve bin/test/journalprint. [RT #18316]

2451.   [port]          solaris: handle runtime linking better. [RT #18356]

2450.   [doc]           Fix lwresd docbook problem for manual page.
                        [RT #18672]

2449.   [placeholder]

2448.   [func]          Add NSEC3 support. [RT #15452]

2447.   [cleanup]       libbind has been split out as a separate product.

2446.   [func]          Add a new log message about build options on startup.
                        A new command-line option '-V' for named is also
                        provided to show this information. [RT #18645]

2445.   [doc]           ARM out-of-date on empty reverse zones (list includes
                        RFC1918 address, but these are not yet compiled in).
                        [RT #18578]

2444.   [port]          Linux, FreeBSD, AIX: Turn off path mtu discovery
                        (clear DF) for UDP responses and requests.

2443.   [bug]           win32: UDP connect() would not generate an event,
                        and so connected UDP sockets would never clean up.
                        Fix this by doing an immediate WSAConnect() rather
```

(continues on next page)

(continued from previous page)

```
                        than an io completion port type for UDP.

2442.   [bug]           A lock could be destroyed twice. [RT #18626]

2441.   [bug]           isc_radix_insert() could copy radix tree nodes
                        incompletely. [RT #18573]

2440.   [bug]           named-checkconf used an incorrect test to determine
                        if an ACL was set to none.

2439.   [bug]           Potential NULL dereference in dns_acl_isanyornone().
                        [RT #18559]

2438.   [bug]           Timeouts could be logged incorrectly under win32.

2437.   [bug]           Sockets could be closed too early, leading to
                        inconsistent states in the socket module. [RT #18298]

2436.   [security]      win32: UDP client handler can be shutdown. [RT #18576]

2435.   [bug]           Fixed an ACL memory leak affecting win32.

2434.   [bug]           Fixed a minor error-reporting bug in
                        lib/isc/win32/socket.c.

2433.   [tuning]        Set initial timeout to 800ms.

2432.   [bug]           More Windows socket handling improvements.  Stop
                        using I/O events and use IO Completion Ports
                        throughout.  Rewrite the receive path logic to make
                        it easier to support multiple simultaneous
                        requesters in the future.  Add stricter consistency
                        checking as a compile-time option (define
                        ISC_SOCKET_CONSISTENCY_CHECKS; defaults to off).

2431.   [bug]           Acl processing could leak memory. [RT #18323]

2430.   [bug]           win32: isc_interval_set() could round down to
                        zero if the input was less than NS_INTERVAL
                        nanoseconds.  Round up instead. [RT #18549]

2429.   [doc]           nsupdate should be in section 1 of the man pages.
                        [RT #18283]

2428.   [bug]           dns_iptable_merge() mishandled merges of negative
                        tables. [RT #18409]

2427.   [func]          Treat DNSKEY queries as if "minimal-response yes;"
                        was set. [RT #18528]

2426.   [bug]           libbind: inet_net_pton() can sometimes return the
                        wrong value if excessively large net masks are
```

(continues on next page)

(continued from previous page)

```
                        supplied. [RT #18512]

2425.   [bug]           named didn't detect unavailable query source addresses
                        at load time. [RT #18536]

2424.   [port]          configure now probes for a working epoll
                        implementation.  Allow the use of kqueue,
                        epoll and /dev/poll to be selected at compile
                        time. [RT #18277]

2423.   [security]      Randomize server selection on queries, so as to
                        make forgery a little more difficult.  Instead of
                        always preferring the server with the lowest RTT,
                        pick a server with RTT within the same 128
                        millisecond band.  [RT #18441]

2422.   [bug]           Handle the special return value of a empty node as
                        if it was a NXRRSET in the validator. [RT #18447]

2421.   [func]          Add new command line option '-S' for named to specify
                        the max number of sockets. [RT #18493]
                        Use caution: this option may not work for some
                        operating systems without rebuilding named.

2420.   [bug]           Windows socket handling cleanup.  Let the io
                        completion event send out canceled read/write
                        done events, which keeps us from writing to memory
                        we no longer have ownership of.  Add debugging
                        socket_log() function.  Rework TCP socket handling
                        to not leak sockets.

2419.   [cleanup]       Document that isc_socket_create() and isc_socket_open()
                        should not be used for isc_sockettype_fdwatch sockets.
                        [RT #18521]

2418.   [bug]           AXFR request on a DLZ could trigger a REQUIRE failure
                        [RT #18430]

2417.   [bug]           Connecting UDP sockets for outgoing queries could
                        unexpectedly fail with an 'address already in use'
                        error. [RT #18411]

2416.   [func]          Log file descriptors that cause exceeding the
                        internal maximum. [RT #18460]

2415.   [bug]           'rndc dumpdb' could trigger various assertion failures
                        in rbtdb.c. [RT #18455]

2414.   [bug]           A masterdump context held the database lock too long,
                        causing various troubles such as dead lock and
                        recursive lock acquisition. [RT #18311, #18456]
```

(continues on next page)

(continued from previous page)

```
2413.   [bug]           Fixed an unreachable code path in socket.c. [RT #18442]

2412.   [bug]           win32: address a resource leak. [RT #18374]

2411.   [bug]           Allow using a larger number of sockets than FD_SETSIZE
                        for select().  To enable this, set ISC_SOCKET_MAXSOCKETS
                        at compilation time.  [RT #18433]

                        Note: with changes #2469 and #2421 above, there is no
                        need to tweak ISC_SOCKET_MAXSOCKETS at compilation time
                        any more.

2410.   [bug]           Correctly delete m_versionInfo. [RT #18432]

2409.   [bug]           Only log that we disabled EDNS processing if we were
                        subsequently successful.  [RT #18029]

2408.   [bug]           A duplicate TCP dispatch event could be sent, which
                        could then trigger an assertion failure in
                        resquery_response().  [RT #18275]

2407.   [port]          hpux: test for sys/dyntune.h. [RT #18421]

2406.   [placeholder]

2405.   [cleanup]       The default value for dnssec-validation was changed to
                        "yes" in 9.5.0-P1 and all subsequent releases; this
                        was inadvertently omitted from CHANGES at the time.

2404.   [port]          hpux: files unlimited support.

2403.   [bug]           TSIG context leak. [RT #18341]

2402.   [port]          Support Solaris 2.11 and over. [RT #18362]

2401.   [bug]           Expect to get E[MN]FILE errno internal_accept()
                        (from accept() or fcntl() system calls). [RT #18358]

2400.   [bug]           Log if kqueue()/epoll_create()/open(/dev/poll) fails.
                        [RT #18297]

2399.   [placeholder]

2398.   [bug]           Improve file descriptor management.  New,
                        temporary, named.conf option reserved-sockets,
                        default 512. [RT #18344]

2397.   [bug]           gssapi_functions had too many elements. [RT #18355]

2396.   [bug]           Don't set SO_REUSEADDR for randomized ports.
                        [RT #18336]
```

(continues on next page)

(continued from previous page)

```
2395.   [port]          Avoid warning and no effect from "files unlimited"
                        on Linux when running as root. [RT #18335]

2394.   [bug]           Default configuration options set the limit for
                        open files to 'unlimited' as described in the
                        documentation. [RT #18331]

2393.   [bug]           nested acls containing keys could trigger an
                        assertion in acl.c. [RT #18166]

2392.   [bug]           remove 'grep -q' from acl test script, some platforms
                        don't support it. [RT #18253]

2391.   [port]          hpux: cover additional recvmsg() error codes.
                        [RT #18301]

2390.   [bug]           dispatch.c could make a false warning on 'odd socket'.
                        [RT #18301].

2389.   [bug]           Move the "working directory writable" check to after
                        the ns_os_changeuser() call. [RT #18326]

2388.   [bug]           Avoid using tables for layout purposes in
                        statistics XSL [RT #18159].

2387.   [bug]           Silence compiler warnings in lib/isc/radix.c.
                        [RT #18147] [RT #18258]

2386.   [func]          Add warning about too small 'open files' limit.
                        [RT #18269]

2385.   [bug]           A condition variable in socket.c could leak in
                        rare error handling [RT #17968].

2384.   [security]      Fully randomize UDP query ports to improve
                        forgery resilience. [RT #17949, #18098]

2383.   [bug]           named could double queries when they resulted in
                        SERVFAIL due to overkilling EDNS0 failure detection.
                        [RT #18182]

2382.   [doc]           Add descriptions of DHCID, IPSECKEY, SPF and SSHFP
                        to ARM.

2381.   [port]          dlz/mysql: support multiple install layouts for
                        mysql.  <prefix>/include/{,mysql/}mysql.h and
                        <prefix>/lib/{,mysql/}. [RT #18152]

2380.   [bug]           dns_view_find() was not returning NXDOMAIN/NXRRSET
                        proofs which, in turn, caused validation failures
                        for insecure zones immediately below a secure zone
                        the server was authoritative for. [RT #18112]
```

(continues on next page)

(continued from previous page)

```
2379.   [contrib]       queryperf/gen-data-queryperf.py: removed redundant
                        TLDs and supported RRs with TTLs [RT #17972]

2378.   [bug]           gssapi_functions{} had a redundant member in BIND 9.5.
                        [RT #18169]

2377.   [bug]           Address race condition in dnssec-signzone. [RT #18142]

2376.   [bug]           Change #2144 was not complete.

2375.   [placeholder]

2374.   [bug]           "blackhole" ACLs could cause named to segfault due
                        to some uninitialized memory. [RT #18095]

2373.   [bug]           Default values of zone ACLs were re-parsed each time a
                        new zone was configured, causing an overconsumption
                        of memory. [RT #18092]

2372.   [bug]           Fixed incorrect TAG_HMACSHA256_BITS value [RT #18047]

2371.   [doc]           Add +nsid option to dig man page. [RT #18039]

2370.   [bug]           "rndc freeze" could trigger an assertion in named
                        when called on a nonexistent zone. [RT #18050]

2369.   [bug]           libbind: Array bounds overrun on read in bitncmp().
                        [RT #18054]

2368.   [port]          Linux: use libcap for capability management if
                        possible. [RT #18026]

2367.   [bug]           Improve counting of dns_resstatscounter_retry
                        [RT #18030]

2366.   [bug]           Adb shutdown race. [RT #18021]

2365.   [bug]           Fix a bug that caused dns_acl_isany() to return
                        spurious results. [RT #18000]

2364.   [bug]           named could trigger a assertion when serving a
                        malformed signed zone. [RT #17828]

2363.   [port]          sunos: pre-set "lt_cv_sys_max_cmd_len=4096;".
                        [RT #17513]

2362.   [cleanup]       Make "rrset-order fixed" a compile-time option.
                        settable by "./configure --enable-fixed-rrset".
                        Disabled by default. [RT #17977]

2361.   [bug]           "recursion" statistics counter could be counted
```

(continues on next page)

(continued from previous page)

```
                        multiple times for a single query.  [RT #17990]

2360.   [bug]           Fix a condition where we release a database version
                        (which may acquire a lock) while holding the lock.

2359.   [bug]           Fix NSID bug. [RT #17942]

2358.   [doc]           Update host's default query description. [RT #17934]

2357.   [port]          Don't use OpenSSL's engine support in versions before
                        OpenSSL 0.9.7f. [RT #17922]

2356.   [bug]           Built in mutex profiler was not scalable enough.
                        [RT #17436]

2355.   [func]          Extend the number statistics counters available.
                        [RT #17590]

2354.   [bug]           Failed to initialize some rdatasetheader_t elements.
                        [RT #17927]

2353.   [func]          Add support for Name Server ID (RFC 5001).
                        'dig +nsid' requests NSID from server.
                        'request-nsid yes;' causes recursive server to send
                        NSID requests to upstream servers.  Server responds
                        to NSID requests with the string configured by
                        'server-id' option.  [RT #17091]

2352.   [bug]           Various GSS_API fixups. [RT #17729]

2351.   [bug]           convertxsl.pl generated very long lines. [RT #17906]

2350.   [port]          win32: IPv6 support. [RT #17797]

2349.   [func]          Provide incremental re-signing support for secure
                        dynamic zones. [RT #1091]

2348.   [func]          Use the EVP interface to OpenSSL. Add PKCS#11 support.
                        Documentation is in the new README.pkcs11 file.
                        New tool, dnssec-keyfromlabel, which takes the
                        label of a key pair in a HSM and constructs a DNS
                        key pair for use by named and dnssec-signzone.
                        [RT #16844]

2347.   [bug]           Delete now traverses the RB tree in the canonical
                        order. [RT #17451]

2346.   [func]          Memory statistics now cover all active memory contexts
                        in increased detail. [RT #17580]

2345.   [bug]           named-checkconf failed to detect when forwarders
                        were set at both the options/view level and in
```

(continues on next page)

(continued from previous page)

```
                        a root zone. [RT #17671]

2344.   [bug]           Improve "logging{ file ...; };" documentation.
                        [RT #17888]

2343.   [bug]           (Seemingly) duplicate IPv6 entries could be
                        created in ADB. [RT #17837]

2342.   [func]          Use getifaddrs() if available under Linux. [RT #17224]

2341.   [bug]           libbind: add missing -I../include for off source
                        tree builds. [RT #17606]

2340.   [port]          openbsd: interface configuration. [RT #17700]

2339.   [port]          tru64: support for libbind. [RT #17589]

2338.   [bug]           check_ds() could be called with a non DS rdataset.
                        [RT #17598]

2337.   [bug]           BUILD_LDFLAGS was not being correctly set.  [RT #17614]

2336.   [func]          If "named -6" is specified then listen on all IPv6
                        interfaces if there are not listen-on-v6 clauses in
                        named.conf.  [RT #17581]

2335.   [port]          sunos:  libbind and *printf() support for long long.
                        [RT #17513]

2334.   [bug]           Bad REQUIRES in fromstruct_in_naptr(),  off by one
                        bug in fromstruct_txt(). [RT #17609]

2333.   [bug]           Fix off by one error in isc_time_nowplusinterval().
                        [RT #17608]

2332.   [contrib]       query-loc-0.4.0. [RT #17602]

2331.   [bug]           Failure to regenerate any signatures was not being
                        reported nor being past back to the UPDATE client.
                        [RT #17570]

2330.   [bug]           Remove potential race condition when handling
                        over memory events. [RT #17572]

                        WARNING: API CHANGE: over memory callback
                        function now needs to call isc_mem_waterack().
                        See <isc/mem.h> for details.

2329.   [bug]           Clearer help text for dig's '-x' and '-i' options.

2328.   [maint]         Add AAAA addresses for A.ROOT-SERVERS.NET,
                        F.ROOT-SERVERS.NET, H.ROOT-SERVERS.NET,
```

(continues on next page)

(continued from previous page)

```
                        J.ROOT-SERVERS.NET, K.ROOT-SERVERS.NET and
                        M.ROOT-SERVERS.NET.

2327.   [bug]           It was possible to dereference a NULL pointer in
                        rbtdb.c.  Implement dead node processing in zones as
                        we do for caches. [RT #17312]

2326.   [bug]           It was possible to trigger a INSIST in the acache
                        processing.

2325.   [port]          Linux: use capset() function if available. [RT #17557]

2324.   [bug]           Fix IPv6 matching against "any;". [RT #17533]

2323.   [port]          tru64: namespace clash. [RT #17547]

2322.   [port]          MacOS: work around the limitation of setrlimit()
                        for RLIMIT_NOFILE. [RT #17526]

2321.   [placeholder]

2320.   [func]          Make statistics counters thread-safe for platforms
                        that support certain atomic operations. [RT #17466]

2319.   [bug]           Silence Coverity warnings in
                        lib/dns/rdata/in_1/apl_42.c. [RT #17469]

2318.   [port]          sunos fixes for libbind.  [RT #17514]

2317.   [bug]           "make distclean" removed bind9.xsl.h. [RT #17518]

2316.   [port]          Missing #include <isc/print.h> in lib/dns/gssapictx.c.
                        [RT #17513]

2315.   [bug]           Used incorrect address family for mapped IPv4
                        addresses in acl.c. [RT #17519]

2314.   [bug]           Uninitialized memory use on error path in
                        bin/named/lwdnoop.c.  [RT #17476]

2313.   [cleanup]       Silence Coverity warnings. Handle private stacks.
                        [RT #17447] [RT #17478]

2312.   [cleanup]       Silence Coverity warning in lib/isc/unix/socket.c.
                        [RT #17458]

2311.   [bug]           IPv6 addresses could match IPv4 ACL entries and
                        vice versa. [RT #17462]

2310.   [bug]           dig, host, nslookup: flush stdout before emitting
                        debug/fatal messages.  [RT #17501]
```

(continues on next page)

```
2309.   [cleanup]       Fix Coverity warnings in lib/dns/acl.c and iptable.c.
                        [RT #17455]

2308.   [cleanup]       Silence Coverity warning in bin/named/controlconf.c.
                        [RT #17495]

2307.   [bug]           Remove infinite loop from lib/dns/sdb.c. [RT #17496]

2306.   [bug]           Remove potential race from lib/dns/resolver.c.
                        [RT #17470]

2305.   [security]      inet_network() buffer overflow. CVE-2008-0122.

2304.   [bug]           Check returns from all dns_rdata_tostruct() calls.
                        [RT #17460]

2303.   [bug]           Remove unnecessary code from bin/named/lwdgnba.c.
                        [RT #17471]

2302.   [bug]           Fix memset() calls in lib/tests/t_api.c. [RT #17472]

2301.   [bug]           Remove resource leak and fix error messages in
                        bin/tests/system/lwresd/lwtest.c. [RT #17474]

2300.   [bug]           Fixed failure to close open file in
                        bin/tests/names/t_names.c. [RT #17473]

2299.   [bug]           Remove unnecessary NULL check in
                        bin/nsupdate/nsupdate.c. [RT #17475]

2298.   [bug]           isc_mutex_lock() failure not caught in
                        bin/tests/timers/t_timers.c. [RT #17468]

2297.   [bug]           isc_entropy_createfilesource() failure not caught in
                        bin/tests/dst/t_dst.c. [RT #17467]

2296.   [port]          Allow docbook stylesheet location to be specified to
                        configure. [RT #17457]

2295.   [bug]           Silence static overrun error in bin/named/lwaddr.c.
                        [RT #17459]

2294.   [func]          Allow the experimental statistics channels to have
                        multiple connections and ACL.
                        Note: the stats-server and stats-server-v6 options
                        available in the previous beta releases are replaced
                        with the generic statistics-channels statement.

2293.   [func]          Add ACL regression test. [RT #17375]

2292.   [bug]           Log if the working directory is not writable.
                        [RT #17312]
```

(continues on next page)

(continued from previous page)

```
2291.   [bug]           PR_SET_DUMPABLE may be set too late.  Also report
                        failure to set PR_SET_DUMPABLE. [RT #17312]

2290.   [bug]           Let AD in the query signal that the client wants AD
                        set in the response. [RT #17301]

2289.   [func]          named-checkzone now reports the out-of-zone CNAME
                        found. [RT #17309]

2288.   [port]          win32: mark service as running when we have finished
                        loading.  [RT #17441]

2287.   [bug]           Use 'volatile' if the compiler supports it. [RT #17413]

2286.   [func]          Allow a TCP connection to be used as a weak
                        authentication method for reverse zones.
                        New update-policy methods tcp-self and 6to4-self.
                        [RT #17378]

2285.   [func]          Test framework for client memory context management.
                        [RT #17377]

2284.   [bug]           Memory leak in UPDATE prerequisite processing.
                        [RT #17377]

2283.   [bug]           TSIG keys were not attaching to the memory
                        context.  TSIG keys should use the rings
                        memory context rather than the clients memory
                        context. [RT #17377]

2282.   [bug]           Acl code fixups. [RT #17346] [RT #17374]

2281.   [bug]           Attempts to use undefined acls were not being logged.
                        [RT #17307]

2280.   [func]          Allow the experimental http server to be reached
                        over IPv6 as well as IPv4. [RT #17332]

2279.   [bug]           Use setsockopt(SO_NOSIGPIPE), when available,
                        to protect applications from receiving spurious
                        SIGPIPE signals when using the resolver.

2278.   [bug]           win32: handle the case where Windows returns no
                        search list or DNS suffix. [RT #17354]

2277.   [bug]           Empty zone names were not correctly being caught at
                        in the post parse checks. [RT #17357]

2276.   [bug]           Install <dst/gssapi.h>.  [RT #17359]

2275.   [func]          Add support to dig to perform IXFR queries over UDP.
                        [RT #17235]
```

(continues on next page)

(continued from previous page)

```
2274.   [func]          Log zone transfer statistics. [RT #17336]

2273.   [bug]           Adjust log level to WARNING when saving inconsistent
                        stub/slave master and journal files. [RT #17279]

2272.   [bug]           Handle illegal dnssec-lookaside trust-anchor names.
                        [RT #17262]

2271.   [bug]           Fix a memory leak in http server code [RT #17100]

2270.   [bug]           dns_db_closeversion() version->writer could be reset
                        before it is tested. [RT #17290]

2269.   [contrib]       dbus memory leaks and missing va_end calls. [RT #17232]

2268.   [bug]           0.IN-ADDR.ARPA was missing from the empty zones
                        list.
```

```
        --- 9.5.0b1 released ---

2267.   [bug]           Radix tree node_num value could be set incorrectly,
                        causing positive ACL matches to look like negative
                        ones.  [RT #17311]

2266.   [bug]           client.c:get_clientmctx() returned the same mctx
                        once the pool of mctx's was filled. [RT #17218]

2265.   [bug]           Test that the memory context's basic_table is non NULL
                        before freeing.  [RT #17265]

2264.   [bug]           Server prefix length was being ignored. [RT #17308]

2263.   [bug]           "named-checkconf -z" failed to set default value
                        for "check-integrity".  [RT #17306]

2262.   [bug]           Error status from all but the last view could be
                        lost. [RT #17292]

2261.   [bug]           Fix memory leak with "any" and "none" ACLs [RT #17272]

2260.   [bug]           Reported wrong clients-per-query when increasing the
                        value. [RT #17236]

2259.   [placeholder]
```

```
        --- 9.5.0a7 released ---

2258.   [bug]           Fallback from IXFR/TSIG to SOA/AXFR/TSIG broken.
                        [RT #17241]
```

(continues on next page)

(continued from previous page)

```
2257.   [bug]           win32: Use the full path to vcredist_x86.exe when
                        calling it. [RT #17222]

2256.   [bug]           win32: Correctly register the installation location of
                        bindevt.dll. [RT #17159]

2255.   [maint]         L.ROOT-SERVERS.NET is now 199.7.83.42.

2254.   [bug]           timer.c:dispatch() failed to lock timer->lock
                        when reading timer->idle allowing it to see
                        intermediate values as timer->idle was reset by
                        isc_timer_touch(). [RT #17243]

2253.   [func]          "max-cache-size" defaults to 32M.
                        "max-acache-size" defaults to 16M.

2252.   [bug]           Fixed errors in sortlist code [RT #17216]

2251.   [placeholder]

2250.   [func]          New flag 'memstatistics' to state whether the
                        memory statistics file should be written or not.
                        Additionally named's -m option will cause the
                        statistics file to be written. [RT #17113]

2249.   [bug]           Only set Authentic Data bit if client requested
                        DNSSEC, per RFC 3655 [RT #17175]

2248.   [cleanup]       Fix several errors reported by Coverity. [RT #17160]

2247.   [doc]           Sort doc/misc/options. [RT #17067]

2246.   [bug]           Make the startup of test servers (ans.pl) more
                        robust. [RT #17147]

2245.   [bug]           Validating lack of DS records at trust anchors wasn't
                        working. [RT #17151]

2244.   [func]          Allow the check of nameserver names against the
                        SOA MNAME field to be disabled by specifying
                        'notify-to-soa yes;'.  [RT #17073]

2243.   [func]          Configuration files without a newline at the end now
                        parse without error. [RT #17120]

2242.   [bug]           nsupdate: GSS-TSIG support using the Heimdal Kerberos
                        library could require a source of random data.
                        [RT #17127]

2241.   [func]          nsupdate: add a interactive 'help' command. [RT #17099]

2240.   [bug]           Cleanup nsupdates GSS-TSIG support.  Convert
```

(continues on next page)

(continued from previous page)

```
                        a number of INSIST()s into plain fatal() errors
                        which report the triggering result code.
                        The 'key' command wasn't disabling GSS-TSIG.
                        [RT #17099]

2239.   [func]          Ship a pre built bin/named/bind9.xsl.h. [RT #17114]

2238.   [bug]           It was possible to trigger a REQUIRE when a
                        validation was canceled. [RT #17106]

2237.   [bug]           libbind: res_init() was not thread aware. [RT #17123]

2236.   [bug]           dnssec-signzone failed to preserve the case of
                        of wildcard owner names. [RT #17085]

2235.   [bug]           <isc/atomic.h> was not being installed. [RT #17135]

2234.   [port]          Correct some compiler warnings on SCO OSr5 [RT #17134]

2233.   [func]          Add support for O(1) ACL processing, based on
                        radix tree code originally written by Kevin
                        Brintnall. [RT #16288]

2232.   [bug]           dns_adb_findaddrinfo() could fail and return
                        ISC_R_SUCCESS. [RT #17137]

2231.   [bug]           Building dlzbdb (contrib/dlz/bin/dlzbdb) was broken.
                        [RT #17088]

2230.   [bug]           We could INSIST reading a corrupted journal.
                        [RT #17132]

2229.   [bug]           Null pointer dereference on query pool creation
                        failure. [RT #17133]

2228.   [contrib]       contrib: Change 2188 was incomplete.

2227.   [cleanup]       Tidied up the FAQ. [RT #17121]

2226.   [placeholder]

2225.   [bug]           More support for systems with no IPv4 addresses.
                        [RT #17111]

2224.   [bug]           Defer journal compaction if a xfrin is in progress.
                        [RT #17119]

2223.   [bug]           Make a new journal when compacting. [RT #17119]

2222.   [func]          named-checkconf now checks server key references.
                        [RT #17097]
```

(continues on next page)

(continued from previous page)

```
2221.   [bug]           Set the event result code to reflect the actual
                        record turned to caller when a cache update is
                        rejected due to a more credible answer existing.
                        [RT #17017]

2220.   [bug]           win32: Address a race condition in final shutdown of
                        the Windows socket code. [RT #17028]

2219.   [bug]           Apply zone consistency checks to additions, not
                        removals, when updating. [RT #17049]

2218.   [bug]           Remove unnecessary REQUIRE from dns_validator_create().
                        [RT #16976]

2217.   [func]          Adjust update log levels. [RT #17092]

2216.   [cleanup]       Fix a number of errors reported by Coverity.
                        [RT #17094]

2215.   [bug]           Bad REQUIRE check isc_hmacsha1_verify(). [RT #17094]

2214.   [bug]           Deregister OpenSSL lock callback when cleaning
                        up.  Reorder OpenSSL cleanup so that RAND_cleanup()
                        is called before the locks are destroyed. [RT #17098]

2213.   [bug]           SIG0 diagnostic failure messages were looking at the
                        wrong status code. [RT #17101]

2212.   [func]          'host -m' now causes memory statistics and active
                        memory to be printed at exit. [RT 17028]

2211.   [func]          Update "dynamic update temporarily disabled" message.
                        [RT #17065]

2210.   [bug]           Deleting class specific records via UPDATE could
                        fail.  [RT #17074]

2209.   [port]          osx: linking against user supplied static OpenSSL
                        libraries failed as the system ones were still being
                        found. [RT #17078]

2208.   [port]          win32: make sure both build methods produce the
                        same output. [RT #17058]

2207.   [port]          Some implementations of getaddrinfo() fail to set
                        ai_canonname correctly. [RT #17061]
```

```
        --- 9.5.0a6 released ---

2206.   [security]      "allow-query-cache" and "allow-recursion" now
                        cross inherit from each other.
```

(continues on next page)

(continued from previous page)

```
                        If allow-query-cache is not set in named.conf then
                        allow-recursion is used if set, otherwise allow-query
                        is used if set, otherwise the default (localnets;
                        localhost;) is used.

                        If allow-recursion is not set in named.conf then
                        allow-query-cache is used if set, otherwise allow-query
                        is used if set, otherwise the default (localnets;
                        localhost;) is used.

                        [RT #16987]

2205.   [bug]           libbind: change #2119 broke thread support. [RT #16982]

2204.   [bug]           "rndc flushname name unknown-view" caused named
                        to crash. [RT #16984]

2203.   [security]      Query id generation was cryptographically weak.
                        [RT # 16915]

2202.   [security]      The default acls for allow-query-cache and
                        allow-recursion were not being applied. [RT #16960]

2201.   [bug]           The build failed in a separate object directory.
                        [RT #16943]

2200.   [bug]           The search for cached NSEC records was stopping to
                        early leading to excessive DLV queries. [RT #16930]

2199.   [bug]           win32: don't call WSAStartup() while loading dlls.
                        [RT #16911]

2198.   [bug]           win32: RegCloseKey() could be called when
                        RegOpenKeyEx() failed. [RT #16911]

2197.   [bug]           Add INSIST to catch negative responses which are
                        not setting the event result code appropriately.
                        [RT #16909]

2196.   [port]          win32: yield processor while waiting for once to
                        to complete. [RT #16958]

2195.   [func]          dnssec-keygen now defaults to nametype "ZONE"
                        when generating DNSKEYs. [RT #16954]

2194.   [bug]           Close journal before calling 'done' in xfrin.c.
```

```
        --- 9.5.0a5 released ---

2193.   [port]          win32: BINDInstall.exe is now linked statically.
                        [RT #16906]
```

(continues on next page)

(continued from previous page)

```
2192.   [port]          win32: use vcredist_x86.exe to install Visual
                        Studio's redistributable dlls if building with
                        Visual Stdio 2005 or later.

2191.   [func]          named-checkzone now allows dumping to stdout (-).
                        named-checkconf now has -h for help.
                        named-checkzone now has -h for help.
                        rndc now has -h for help.
                        Better handling of '-?' for usage summaries.
                        [RT #16707]

2190.   [func]          Make fallback to plain DNS from EDNS due to timeouts
                        more visible.  New logging category "edns-disabled".
                        [RT #16871]

2189.   [bug]           Handle socket() returning EINTR. [RT #15949]

2188.   [contrib]       queryperf: autoconf changes to make the search for
                        libresolv or libbind more robust. [RT #16299]

2187.   [bug]           query_addds(), query_addwildcardproof() and
                        query_addnxrrsetnsec() should take a version
                        argument. [RT #16368]

2186.   [port]          cygwin: libbind: check for struct sockaddr_storage
                        independently of IPv6. [RT #16482]

2185.   [port]          sunos: libbind: check for ssize_t, memmove() and
                        memchr(). [RT #16463]

2184.   [bug]           bind9.xsl.h didn't build out of the source tree.
                        [RT #16830]

2183.   [bug]           dnssec-signzone didn't handle offline private keys
                        well.  [RT #16832]

2182.   [bug]           dns_dispatch_createtcp() and dispatch_createudp()
                        could return ISC_R_SUCCESS when they ran out of
                        memory. [RT #16365]

2181.   [port]          sunos: libbind: add paths.h from BIND 8. [RT #16462]

2180.   [cleanup]       Remove bit test from 'compress_test' as they
                        are no longer needed. [RT #16497]

2179.   [func]          'rndc command zone' will now find 'zone' if it is
                        unique to all the views. [RT #16821]

2178.   [bug]           'rndc reload' of a slave or stub zone resulted in
                        a reference leak. [RT #16867]

2177.   [bug]           Array bounds overrun on read (rcodetext) at
```

(continues on next page)

(continued from previous page)

```
                        debug level 10+. [RT #16798]

2176.   [contrib]       dbus update to handle race condition during
                        initialization (Bugzilla 235809). [RT #16842]

2175.   [bug]           win32: windows broadcast condition variable support
                        was broken. [RT #16592]

2174.   [bug]           I/O errors should always be fatal when reading
                        master files. [RT #16825]

2173.   [port]          win32: When compiling with MSVS 2005 SP1 we also
                        need to ship Microsoft.VC80.MFCLOC.
```

```
        --- 9.5.0a4 released ---

2172.   [bug]           query_addsoa() was being called with a non zone db.
                        [RT #16834]

2171.   [bug]           Handle breaks in DNSSEC trust chains where the parent
                        servers are not DS aware (DS queries to the parent
                        return a referral to the child).

2170.   [func]          Add acache processing to test suite. [RT #16711]

2169.   [bug]           host, nslookup: when reporting NXDOMAIN report the
                        given name and not the last name searched for.
                        [RT #16763]

2168.   [bug]           nsupdate: in non-interactive mode treat syntax errors
                        as fatal errors. [RT #16785]

2167.   [bug]           When re-using a automatic zone named failed to
                        attach it to the new view. [RT #16786]
```

```
        --- 9.5.0a3 released ---

2166.   [bug]           When running in batch mode, dig could misinterpret
                        a server address as a name to be looked up, causing
                        unexpected output. [RT #16743]

2165.   [func]          Allow the destination address of a query to determine
                        if we will answer the query or recurse.
                        allow-query-on, allow-recursion-on and
                        allow-query-cache-on. [RT #16291]

2164.   [bug]           The code to determine how named-checkzone /
                        named-compilezone was called failed under windows.
                        [RT #16764]

2163.   [bug]           If only one of query-source and query-source-v6
```

(continues on next page)

(continued from previous page)

```
                        specified a port the query pools code broke (change
                        2129).  [RT #16768]

2162.   [func]          Allow "rrset-order fixed" to be disabled at compile
                        time. [RT #16665]

2161.   [bug]           Fix which log messages are emitted for 'rndc flush'.
                        [RT #16698]

2160.   [bug]           libisc wasn't handling NULL ifa_addr pointers returned
                        from getifaddrs(). [RT #16708]
```

```
        --- 9.5.0a2 released ---

2159.   [bug]           Array bounds overrun in acache processing. [RT #16710]

2158.   [bug]           ns_client_isself() failed to initialize key
                        leading to a REQUIRE failure. [RT #16688]

2157.   [func]          dns_db_transfernode() created. [RT #16685]

2156.   [bug]           Fix node reference leaks in lookup.c:lookup_find(),
                        resolver.c:validated() and resolver.c:cache_name().
                        Fix a memory leak in rbtdb.c:free_noqname().
                        Make lookup.c:lookup_find() robust against
                        event leaks. [RT #16685]

2155.   [contrib]       SQLite sdb module from jaboydjr@netwalk.com.
                        [RT #16694]

2154.   [func]          Scoped (e.g. IPv6 link-local) addresses may now be
                        matched in acls by omitting the scope. [RT #16599]

2153.   [bug]           nsupdate could leak memory. [RT #16691]

2152.   [cleanup]       Use sizeof(buf) instead of fixed number in
                        dighost.c:get_trusted_key(). [RT #16678]

2151.   [bug]           Missing newline in usage message for journalprint.
                        [RT #16679]

2150.   [bug]           'rrset-order cyclic' uniformly distribute the
                        starting point for the first response for a given
                        RRset. [RT #16655]

2149.   [bug]           isc_mem_checkdestroyed() failed to abort on
                        if there were still active memory contexts.
                        [RT #16672]

2148.   [func]          Add positive logging for rndc commands. [RT #14623]

2147.   [bug]           libbind: remove potential buffer overflow from
```

(continues on next page)

(continued from previous page)

```
                        hmac_link.c. [RT #16437]

2146.   [cleanup]       Silence Linux's spurious "obsolete setsockopt
                        SO_BSDCOMPAT" message. [RT #16641]

2145.   [bug]           Check DS/DLV digest lengths for known digests.
                        [RT #16622]

2144.   [cleanup]       Suppress logging of SERVFAIL from forwarders.
                        [RT #16619]

2143.   [bug]           We failed to restart the IPv6 client when the
                        kernel failed to return the destination the
                        packet was sent to. [RT #16613]

2142.   [bug]           Handle master files with a modification time that
                        matches the epoch. [RT #16612]

2141.   [bug]           dig/host should not be setting IDN_ASCCHECK (IDN
                        equivalent of LDH checks).  [RT #16609]

2140.   [bug]           libbind: missing unlock on pthread_key_create()
                        failures. [RT #16654]

2139.   [bug]           dns_view_find() was being called with wrong type
                        in adb.c. [RT #16670]

2138.   [bug]           Lock order reversal in resolver.c. [RT #16653]

2137.   [port]          Mips little endian and/or mips 64 bit are now
                        supported for atomic operations. [RT #16648]

2136.   [bug]           nslookup/host looped if there was no search list
                        and the host didn't exist. [RT #16657]

2135.   [bug]           Uninitialized rdataset in sdlz.c. [RT #16656]

2134.   [func]          Additional statistics support. [RT #16666]

2133.   [port]          powerpc:  Support both IBM and MacOS Power PC
                        assembler syntaxes. [RT #16647]

2132.   [bug]           Missing unlock on out of memory in
                        dns_dispatchmgr_setudp().

2131.   [contrib]       dlz/mysql: AXFR was broken. [RT #16630]

2130.   [func]          Log if CD or DO were set. [RT #16640]

2129.   [func]          Provide a pool of UDP sockets for queries to be
                        made over. See use-queryport-pool, queryport-pool-ports
                        and queryport-pool-updateinterval.  [RT #16415]
```

(continues on next page)

(continued from previous page)

```
2128.   [doc]           xsltproc --nonet, update DTD versions.  [RT #16635]

2127.   [port]          Improved OpenSSL 0.9.8 support. [RT #16563]

2126.   [security]      Serialize validation of type ANY responses. [RT #16555]

2125.   [bug]           dns_zone_getzeronosoattl() REQUIRE failure if DLZ
                        was defined. [RT #16574]

2124.   [security]      It was possible to dereference a freed fetch
                        context. [RT #16584]
```

```
        --- 9.5.0a1 released ---

2123.   [func]          Use Doxygen to generate internal documentation.
                        [RT #11398]

2122.   [func]          Experimental http server and statistics support
                        for named via xml.

2121.   [func]          Add a 10 slot dead masters cache (LRU) with a 600
                        second timeout. [RT #16553]

2120.   [doc]           Fix markup on nsupdate man page. [RT #16556]

2119.   [compat]        libbind: allow res_init() to succeed enough to
                        return the default domain even if it was unable
                        to allocate memory.

2118.   [bug]           Handle response with long chains of domain name
                        compression pointers which point to other compression
                        pointers. [RT #16427]

2117.   [bug]           DNSSEC fixes: named could fail to cache NSEC records
                        which could lead to validation failures.  named didn't
                        handle negative DS responses that were in the process
                        of being validated.  Check CNAME bit before accepting
                        NODATA proof. To be able to ignore a child NSEC there
                        must be SOA (and NS) set in the bitmap. [RT #16399]

2116.   [bug]           'rndc reload' could cause the cache to continually
                        be cleaned. [RT #16401]

2115.   [bug]           'rndc reconfig' could trigger a INSIST if the
                        number of masters for a zone was reduced. [RT #16444]

2114.   [bug]           dig/host/nslookup: searches for names with multiple
                        labels were failing. [RT #16447]

2113.   [bug]           nsupdate: if a zone is specified it should be used
                        for server discover. [RT #16455]
```

(continues on next page)

(continued from previous page)

```
2112.   [security]      Warn if weak RSA exponent is used. [RT #16460]

2111.   [bug]           Fix a number of errors reported by Coverity.
                        [RT #16507]

2110.   [bug]           "minimal-responses yes;" interacted badly with BIND 8
                        priming queries. [RT #16491]

2109.   [port]          libbind: silence aix 5.3 compiler warnings. [RT #16502]

2108.   [func]          DHCID support. [RT #16456]

2107.   [bug]           dighost.c: more cleanup of buffers. [RT #16499]

2106.   [func]          'rndc status' now reports named's version. [RT #16426]

2105.   [func]          GSS-TSIG support (RFC 3645).

2104.   [port]          Fix Solaris SMF error message.

2103.   [port]          Add /usr/sfw to list of locations for OpenSSL
                        under Solaris.

2102.   [port]          Silence Solaris 10 warnings.

2101.   [bug]           OpenSSL version checks were not quite right.
                        [RT #16476]

2100.   [port]          win32: copy libeay32.dll to Build\Debug.
                        Copy Debug\named-checkzone to Debug\named-compilezone.

2099.   [port]          win32: more manifest issues.

2098.   [bug]           Race in rbtdb.c:no_references(), which occasionally
                        triggered an INSIST failure about the node lock
                        reference.  [RT #16411]

2097.   [bug]           named could reference a destroyed memory context
                        after being reloaded / reconfigured. [RT #16428]

2096.   [bug]           libbind: handle applications that fail to detect
                        res_init() failures better.

2095.   [port]          libbind: always prototype inet_cidr_ntop_ipv6() and
                        net_cidr_ntop_ipv6(). [RT #16388]

2094.   [contrib]       Update named-bootconf.  [RT #16404]

2093.   [bug]           named-checkzone -s was broken.

2092.   [bug]           win32: dig, host, nslookup.  Use registry config
```

(continues on next page)

(continued from previous page)

```
                        if resolv.conf does not exist or no nameservers
                        listed. [RT #15877]

2091.   [port]          dighost.c: race condition on cleanup. [RT #16417]

2090.   [port]          win32: Visual C++ 2005 command line manifest support.
                        [RT #16417]

2089.   [security]      Raise the minimum safe OpenSSL versions to
                        OpenSSL 0.9.7l and OpenSSL 0.9.8d.  Versions
                        prior to these have known security flaws which
                        are (potentially) exploitable in named. [RT #16391]

2088.   [security]      Change the default RSA exponent from 3 to 65537.
                        [RT #16391]

2087.   [port]          libisc failed to compile on OS's w/o a vsnprintf.
                        [RT #16382]

2086.   [port]          libbind: FreeBSD now has get*by*_r() functions.
                        [RT #16403]

2085.   [doc]           win32: added index.html and README to zip. [RT #16201]

2084.   [contrib]       dbus update for 9.3.3rc2.

2083.   [port]          win32: Visual C++ 2005 support.

2082.   [doc]           Document 'cache-file' as a test only option.

2081.   [port]          libbind: minor 64-bit portability fix in memcluster.c.
                        [RT #16360]

2080.   [port]          libbind: res_init.c did not compile on older versions
                        of Solaris. [RT #16363]

2079.   [bug]           The lame cache was not handling multiple types
                        correctly. [RT #16361]

2078.   [bug]           dnssec-checkzone output style "default" was badly
                        named.  It is now called "relative". [RT #16326]

2077.   [bug]           'dnssec-signzone -O raw' wasn't outputting the
                        complete signed zone. [RT #16326]

2076.   [bug]           Several files were missing #include <config.h>
                        causing build failures on OSF. [RT #16341]

2075.   [bug]           The spillat timer event handler could leak memory.
                        [RT #16357]

2074.   [bug]           dns_request_createvia2(), dns_request_createvia3(),
```

(continues on next page)

(continued from previous page)

```
                        dns_request_createraw2() and dns_request_createraw3()
                        failed to send multiple UDP requests. [RT #16349]

2073.   [bug]           Incorrect semantics check for update policy "wildcard".
                        [RT #16353]

2072.   [bug]           We were not generating valid HMAC SHA digests.
                        [RT #16320]

2071.   [port]          Test whether gcc accepts -fno-strict-aliasing.
                        [RT #16324]

2070.   [bug]           The remote address was not always displayed when
                        reporting dispatch failures. [RT #16315]

2069.   [bug]           Cross compiling was not working. [RT #16330]

2068.   [cleanup]       Lower incremental tuning message to debug 1.
                        [RT #16319]

2067.   [bug]           'rndc' could close the socket too early triggering
                        a INSIST under Windows. [RT #16317]

2066.   [security]      Handle SIG queries gracefully. [RT #16300]

2065.   [bug]           libbind: probe for HPUX prototypes for
                        endprotoent_r() and endservent_r().  [RT 16313]

2064.   [bug]           libbind: silence AIX compiler warnings. [RT #16218]

2063.   [bug]           Change #1955 introduced a bug which caused the first
                        'rndc flush' call to not free memory. [RT #16244]

2062.   [bug]           'dig +nssearch' was reusing a buffer before it had
                        been returned by the socket code. [RT #16307]

2061.   [bug]           Accept expired wildcard message reversed. [RT #16296]

2060.   [bug]           Enabling DLZ support could leave views partially
                        configured. [RT #16295]

2059.   [bug]           Search into cache rbtdb could trigger an INSIST
                        failure while cleaning up a stale rdataset.
                        [RT #16292]

2058.   [bug]           Adjust how we calculate rtt estimates in the presence
                        of authoritative servers that drop EDNS and/or CD
                        requests.  Also fallback to EDNS/512 and plain DNS
                        faster for zones with less than 3 servers.  [RT #16187]

2057.   [bug]           Make setting "ra" dependent on both allow-query-cache
                        and allow-recursion. [RT #16290]
```

(continues on next page)

(continued from previous page)

```
2056.   [bug]           dig: ixfr= was not being treated case insensitively
                        at all times. [RT #15955]

2055.   [bug]           Missing goto after dropping multicast query.
                        [RT #15944]

2054.   [port]          freebsd: do not explicitly link against -lpthread.
                        [RT #16170]

2053.   [port]          netbsd:libbind: silence compiler warnings. [RT #16220]

2052.   [bug]           'rndc' improve connect failed message to report
                        the failing address. [RT #15978]

2051.   [port]          More strtol() fixes. [RT #16249]

2050.   [bug]           Parsing of NSAP records was not case insensitive.
                        [RT #16287]

2049.   [bug]           Restore SOA before AXFR when falling back from
                        a attempted IXFR when transferring in a zone.
                        Allow a initial SOA query before attempting
                        a AXFR to be requested. [RT #16156]

2048.   [bug]           It was possible to loop forever when using
                        avoid-v4-udp-ports / avoid-v6-udp-ports when
                        the OS always returned the same local port.
                        [RT #16182]

2047.   [bug]           Failed to initialize the interface flags to zero.
                        [RT #16245]

2046.   [bug]           rbtdb.c:rdataset_setadditional() could cause duplicate
                        cleanup [RT #16247].

2045.   [func]          Use lock buckets for acache entries to limit memory
                        consumption. [RT #16183]

2044.   [port]          Add support for atomic operations for Itanium.
                        [RT #16179]

2043.   [port]          nsupdate/nslookup: Force the flushing of the prompt
                        for interactive sessions. [RT #16148]

2042.   [bug]           named-checkconf was incorrectly rejecting the
                        logging category "config". [RT #16117]

2041.   [bug]           "configure --with-dlz-bdb=yes" produced a bad
                        set of libraries to be linked. [RT #16129]

2040.   [bug]           rbtdb no_references() could trigger an INSIST
```

(continues on next page)

(continued from previous page)

```
                        failure with --enable-atomic.  [RT #16022]

2039.   [func]          Check that all buffers passed to the socket code
                        have been retrieved when the socket event is freed.
                        [RT #16122]

2038.   [bug]           dig/nslookup/host was unlinking from wrong list
                        when handling errors. [RT #16122]

2037.   [func]          When unlinking the first or last element in a list
                        check that the list head points to the element to
                        be unlinked. [RT #15959]

2036.   [bug]           'rndc recursing' could cause trigger a REQUIRE.
                        [RT #16075]

2035.   [func]          Make falling back to TCP on UDP refresh failure
                        optional. Default "try-tcp-refresh yes;" for BIND 8
                        compatibility. [RT #16123]

2034.   [bug]           gcc: set -fno-strict-aliasing. [RT #16124]

2033.   [bug]           We weren't creating multiple client memory contexts
                        on demand as expected. [RT #16095]

2032.   [bug]           Remove a INSIST in query_addadditional2(). [RT #16074]

2031.   [bug]           Emit a error message when "rndc refresh" is called on
                        a non slave/stub zone. [RT # 16073]

2030.   [bug]           We were being overly conservative when disabling
                        openssl engine support. [RT #16030]

2029.   [bug]           host printed out the server multiple times when
                        specified on the command line. [RT #15992]

2028.   [port]          linux: socket.c compatibility for old systems.
                        [RT #16015]

2027.   [port]          libbind: Solaris x86 support. [RT #16020]

2026.   [bug]           Rate limit the two recursive client exceeded messages.
                        [RT #16044]

2025.   [func]          Update "zone serial unchanged" message. [RT #16026]

2024.   [bug]           named emitted spurious "zone serial unchanged"
                        messages on reload. [RT #16027]

2023.   [bug]           "make install" should create ${localstatedir}/run and
                        ${sysconfdir} if they do not exist. [RT #16033]
```

(continues on next page)

(continued from previous page)

```
2022.   [bug]           If dnssec validation is disabled only assert CD if
                        CD was requested. [RT #16037]

2021.   [bug]           dnssec-enable no; triggered a REQUIRE. [RT #16037]

2020.   [bug]           rdataset_setadditional() could leak memory. [RT #16034]

2019.   [tuning]        Reduce the amount of work performed per quantum
                        when cleaning the cache. [RT #15986]

2018.   [bug]           Checking if the HMAC MD5 private file was broken.
                        [RT #15960]

2017.   [bug]           allow-query default was not correct. [RT #15946]

2016.   [bug]           Return a partial answer if recursion is not
                        allowed but requested and we had the answer
                        to the original qname. [RT #15945]

2015.   [cleanup]       use-additional-cache is now acache-enable for
                        consistency.  Default acache-enable off in BIND 9.4
                        as it requires memory usage to be configured.
                        It may be enabled by default in BIND 9.5 once we
                        have more experience with it.

2014.   [func]          Statistics about acache now recorded and sent
                        to log. [RT #15976]

2013.   [bug]           Handle unexpected TSIGs on unsigned AXFR/IXFR
                        responses more gracefully. [RT #15941]

2012.   [func]          Don't insert new acache entries if acache is full.
                        [RT #15970]

2011.   [func]          dnssec-signzone can now update the SOA record of
                        the signed zone, either as an increment or as the
                        system time(). [RT #15633]

2010.   [placeholder]   rt15958

2009.   [bug]           libbind: Coverity fixes. [RT #15808]

2008.   [func]          It is now possible to enable/disable DNSSEC
                        validation from rndc.  This is useful for the
                        mobile hosts where the current connection point
                        breaks DNSSEC (firewall/proxy).  [RT #15592]

                                rndc validation newstate [view]

2007.   [func]          It is now possible to explicitly enable DNSSEC
                        validation.  default dnssec-validation no; to
                        be changed to yes in 9.5.0.  [RT #15674]
```

(continues on next page)

(continued from previous page)

```
2006.   [security]      Allow-query-cache and allow-recursion now default
                        to the built in acls "localnets" and "localhost".

                        This is being done to make caching servers less
                        attractive as reflective amplifying targets for
                        spoofed traffic.  This still leave authoritative
                        servers exposed.

                        The best fix is for full BCP 38 deployment to
                        remove spoofed traffic.

2005.   [bug]           libbind: Retransmission timeouts should be
                        based on which attempt it is to the nameserver
                        and not the nameserver itself. [RT #13548]

2004.   [bug]           dns_tsig_sign() could pass a NULL pointer to
                        dst_context_destroy() when cleaning up after a
                        error. [RT #15835]

2003.   [bug]           libbind: The DNS name/address lookup functions could
                        occasionally follow a random pointer due to
                        structures not being completely zeroed. [RT #15806]

2002.   [bug]           libbind: tighten the constraints on when
                        struct addrinfo._ai_pad exists.  [RT #15783]

2001.   [func]          Check the KSK flag when updating a secure dynamic zone.
                        New zone option "update-check-ksk yes;".  [RT #15817]

2000.   [bug]           memmove()/strtol() fix was incomplete. [RT #15812]

1999.   [func]          Implement "rrset-order fixed". [RT #13662]

1998.   [bug]           Restrict handling of fifos as sockets to just SunOS.
                        This allows named to connect to entropy gathering
                        daemons that use fifos instead of sockets. [RT #15840]

1997.   [bug]           Named was failing to replace negative cache entries
                        when a positive one for the type was learnt.
                        [RT #15818]

1996.   [bug]           nsupdate: if a zone has been specified it should
                        appear in the output of 'show'. [RT #15797]

1995.   [bug]           'host' was reporting multiple "is an alias" messages.
                        [RT #15702]

1994.   [port]          OpenSSL 0.9.8 support. [RT #15694]

1993.   [bug]           Log messages, via syslog, were missing the space
                        after the timestamp if "print-time yes" was specified.
```

(continues on next page)

(continued from previous page)

```
                        [RT #15844]

1992.   [bug]           Not all incoming zone transfer messages included the
                        view.  [RT #15825]

1991.   [cleanup]       The configuration data, once read, should be treated
                        as read only.  Expand the use of const to enforce this
                        at compile time. [RT #15813]

1990.   [bug]           libbind:  isc's override of broken gettimeofday()
                        implementations was not always effective.
                        [RT #15709]

1989.   [bug]           win32: don't check the service password when
                        re-installing. [RT #15882]

1988.   [bug]           Remove a bus error from the SHA256/SHA512 support.
                        [RT #15878]

1987.   [func]          DS/DLV SHA256 digest algorithm support. [RT #15608]

1986.   [func]          Report when a zone is removed. [RT #15849]

1985.   [protocol]      DLV has now been assigned a official type code of
                        32769. [RT #15807]

                        Note: care should be taken to ensure you upgrade
                        both named and dnssec-signzone at the same time for
                        zones with DLV records where named is the master
                        server for the zone.  Also any zones that contain
                        DLV records should be removed when upgrading a slave
                        zone.  You do not however have to upgrade all
                        servers for a zone with DLV records simultaneously.

1984.   [func]          dig, nslookup and host now advertise a 4096 byte
                        EDNS UDP buffer size by default. [RT #15855]

1983.   [func]          Two new update policies.  "selfsub" and "selfwild".
                        [RT #12895]

1982.   [bug]           DNSKEY was being accepted on the parent side of
                        a delegation.  KEY is still accepted there for
                        RFC 3007 validated updates. [RT #15620]

1981.   [bug]           win32: condition.c:wait() could fail to reattain
                        the mutex lock.

1980.   [func]          dnssec-signzone: output the SOA record as the
                        first record in the signed zone. [RT #15758]

1979.   [port]          linux: allow named to drop core after changing
                        user ids. [RT #15753]
```

(continues on next page)

```
1978.   [port]          Handle systems which have a broken recvmsg().
                        [RT #15742]

1977.   [bug]           Silence noisy log message. [RT #15704]

1976.   [bug]           Handle systems with no IPv4 addresses. [RT #15695]

1975.   [bug]           libbind: isc_gethexstring() could misparse multi-line
                        hex strings with comments. [RT #15814]

1974.   [doc]           List each of the zone types and associated zone
                        options separately in the ARM.

1973.   [func]          TSIG HMACSHA1, HMACSHA224, HMACSHA256, HMACSHA384 and
                        HMACSHA512 support. [RT #13606]

1972.   [contrib]       DBUS dynamic forwarders integration from
                        Jason Vas Dias <jvdias@redhat.com>.

1971.   [port]          linux: make detection of missing IF_NAMESIZE more
                        robust. [RT #15443]

1970.   [bug]           nsupdate: adjust UDP timeout when falling back to
                        unsigned SOA query. [RT #15775]

1969.   [bug]           win32: the socket code was freeing the socket
                        structure too early. [RT #15776]

1968.   [bug]           Missing lock in resolver.c:validated(). [RT #15739]

1967.   [func]          dig/nslookup/host: warn about missing "QR". [RT #15779]

1966.   [bug]           Don't set CD when we have fallen back to plain DNS.
                        [RT #15727]

1965.   [func]          Suppress spurious "recursion requested but not
                        available" warning with 'dig +qr'. [RT #15780].

1964.   [func]          Separate out MX and SRV to CNAME checks. [RT #15723]

1963.   [port]          Tru64 4.0E doesn't support send() and recv().
                        [RT #15586]

1962.   [bug]           Named failed to clear old update-policy when it
                        was removed. [RT #15491]

1961.   [bug]           Check the port and address of responses forwarded
                        to dispatch. [RT #15474]

1960.   [bug]           Update code should set NSEC ttls from SOA MINIMUM.
                        [RT #15465]
```

(continues on next page)

(continued from previous page)

```
1959.   [func]          Control the zeroing of the negative response TTL to
                        a soa query.  Defaults "zero-no-soa-ttl yes;" and
                        "zero-no-soa-ttl-cache no;". [RT #15460]

1958.   [bug]           Named failed to update the zone's secure state
                        until the zone was reloaded. [RT #15412]

1957.   [bug]           Dig mishandled responses to class ANY queries.
                        [RT #15402]

1956.   [bug]           Improve cross compile support, 'gen' is now built
                        by native compiler.  See README for additional
                        cross compile support information. [RT #15148]

1955.   [bug]           Pre-allocate the cache cleaning iterator. [RT #14998]

1954.   [func]          Named now falls back to advertising EDNS with a
                        512 byte receive buffer if the initial EDNS queries
                        fail.  [RT #14852]

1953.   [func]          The maximum EDNS UDP response named will send can
                        now be set in named.conf (max-udp-size).  This is
                        independent of the advertised receive buffer
                        (edns-udp-size). [RT #14852]

1952.   [port]          hpux: tell the linker to build a runtime link
                        path "-Wl,+b:". [RT #14816].

1951.   [security]      Drop queries from particular well known ports.
                        Don't return FORMERR to queries from particular
                        well known ports.  [RT #15636]

1950.   [port]          Solaris 2.5.1 and earlier cannot bind() then connect()
                        a TCP socket. This prevents the source address being
                        set for TCP connections. [RT #15628]

1949.   [func]          Addition memory leakage checks. [RT #15544]

1948.   [bug]           If was possible to trigger a REQUIRE failure in
                        xfrin.c:maybe_free() if named ran out of memory.
                        [RT #15568]

1947.   [func]          It is now possible to configure named to accept
                        expired RRSIGs.  Default "dnssec-accept-expired no;".
                        Setting "dnssec-accept-expired yes;" leaves named
                        vulnerable to replay attacks.  [RT #14685]

1946.   [bug]           resume_dslookup() could trigger a REQUIRE failure
                        when using forwarders. [RT #15549]

1945.   [cleanup]       dnssec-keygen: RSA (RSAMD5) is no longer recommended.
                        To generate a RSAMD5 key you must explicitly request
```

(continues on next page)

(continued from previous page)

```
                        RSAMD5. [RT #13780]

1944.   [cleanup]       isc_hash_create() does not need a read/write lock.
                        [RT #15522]

1943.   [bug]           Set the loadtime after rolling forward the journal.
                        [RT #15647]

1942.   [bug]           If the name of a DNSKEY match that of one in
                        trusted-keys do not attempt to validate the DNSKEY
                        using the parents DS RRset. [RT #15649]

1941.   [bug]           ncache_adderesult() should set eresult even if no
                        rdataset is passed to it. [RT #15642]

1940.   [bug]           Fixed a number of error conditions reported by
                        Coverity.

1939.   [bug]           The resolver could dereference a null pointer after
                        validation if all the queries have timed out.
                        [RT #15528]

1938.   [bug]           The validator was not correctly handling unsecure
                        negative responses at or below a SEP. [RT #15528]

1937.   [bug]           sdlz doesn't handle RRSIG records. [RT #15564]

1936.   [bug]           The validator could leak memory. [RT #15544]

1935.   [bug]           'acache' was DO sensitive. [RT #15430]

1934.   [func]          Validate pending NS RRsets, in the authority section,
                        prior to returning them if it can be done without
                        requiring DNSKEYs to be fetched.  [RT #15430]

1933.   [bug]           dump_rdataset_raw() had a incorrect INSIST. [RT #15534]

1932.   [bug]           hpux: LDFLAGS was getting corrupted. [RT #15530]

1931.   [bug]           Per-client mctx could require a huge amount of memory,
                        particularly for a busy caching server. [RT #15519]

1930.   [port]          HPUX: ia64 support. [RT #15473]

1929.   [port]          FreeBSD: extend use of PTHREAD_SCOPE_SYSTEM.

1928.   [bug]           Race in rbtdb.c:currentversion(). [RT #15517]

1927.   [bug]           Access to soanode or nsnode in rbtdb violated the
                        lock order rule and could cause a dead lock.
                        [RT #15518]
```

(continues on next page)

(continued from previous page)

```
1926.   [bug]           The Windows installer did not check for empty
                        passwords.  BINDinstall was being installed in
                        the wrong place. [RT #15483]

1925.   [port]          All outer level AC_TRY_RUNs need cross compiling
                        defaults. [RT #15469]

1924.   [port]          libbind: hpux ia64 support. [RT #15473]

1923.   [bug]           ns_client_detach() called too early. [RT #15499]

1922.   [bug]           check-tool.c:setup_logging() missing call to
                        dns_log_setcontext().

1921.   [bug]           Client memory contexts were not using internal
                        malloc. [RT #15434]

1920.   [bug]           The cache rbtdb lock array was too small to
                        have the desired performance characteristics.
                        [RT #15454]

1919.   [contrib]       queryperf: a set of new features: collecting/printing
                        response delays, printing intermediate results, and
                        adjusting query rate for the "target" qps.

1918.   [bug]           Memory leak when checking acls. [RT #15391]

1917.   [doc]           funcsynopsisinfo wasn't being treated as verbatim
                        when generating man pages. [RT #15385]

1916.   [func]          Integrate contributed IDN code from JPNIC. [RT #15383]

1915.   [bug]           dig +ndots was broken. [RT #15215]

1914.   [protocol]      DS is required to accept mnemonic algorithms
                        (RFC 4034).  Still emit numeric algorithms for
                        compatibility with RFC 3658. [RT #15354]

1913.   [func]          Integrate contributed DLZ code into named. [RT #11382]

1912.   [port]          aix: atomic locking for powerpc. [RT #15020]

1911.   [bug]           Update windows socket code. [RT #14965]

1910.   [bug]           dig's +sigchase code overhauled. [RT #14933]

1909.   [bug]           The DLV code has been re-worked to make no longer
                        query order sensitive. [RT #14933]

1908.   [func]          dig now warns if 'RA' is not set in the answer when
                        'RD' was set in the query.  host/nslookup skip servers
                        that fail to set 'RA' when 'RD' is set unless a server
```

(continues on next page)

(continued from previous page)

```
                        is explicitly set.  [RT #15005]

1907.   [func]          host/nslookup now continue (default)/fail on SERVFAIL.
                        [RT #15006]

1906.   [func]          dig now has a '-q queryname' and '+showsearch' options.
                        [RT #15034]

1905.   [bug]           Strings returned from cfg_obj_asstring() should be
                        treated as read-only.  The prototype for
                        cfg_obj_asstring() has been updated to reflect this.
                        [RT #15256]

1904.   [func]          Automatic empty zone creation for D.F.IP6.ARPA and
                        friends.  Note: RFC 1918 zones are not yet covered by
                        this but are likely to be in a future release.

                        New options: empty-server, empty-contact,
                        empty-zones-enable and disable-empty-zone.

1903.   [func]          ISC string copy API.

1902.   [func]          Attempt to make the amount of work performed in a
                        iteration self tuning.  The covers nodes clean from
                        the cache per iteration, nodes written to disk when
                        rewriting a master file and nodes destroyed per
                        iteration when destroying a zone or a cache.
                        [RT #14996]

1901.   [cleanup]       Don't add DNSKEY records to the additional section.

1900.   [bug]           ixfr-from-differences failed to ensure that the
                        serial number increased. [RT #15036]

1899.   [func]          named-checkconf now validates update-policy entries.
                        [RT #14963]

1898.   [bug]           Extend ISC_SOCKADDR_FORMATSIZE and
                        ISC_NETADDR_FORMATSIZE to allow for scope details.

1897.   [func]          x86 and x86_64 now have separate atomic locking
                        implementations.

1896.   [bug]           Recursive clients soft quota support wasn't working
                        as expected. [RT #15103]

1895.   [bug]           A escaped character is, potentially, converted to
                        the output character set too early. [RT #14666]

1894.   [doc]           Review ARM for BIND 9.4.

1893.   [port]          Use uintptr_t if available. [RT #14606]
```

(continues on next page)

(continued from previous page)

```
1892.   [func]          Support for SPF rdata type. [RT #15033]

1891.   [port]          freebsd: pthread_mutex_init can fail if it runs out
                        of memory. [RT #14995]

1890.   [func]          Raise the UDP receive buffer size to 32k if it is
                        less than 32k. [RT #14953]

1889.   [port]          sunos: non blocking i/o support. [RT #14951]

1888.   [func]          Support for IPSECKEY rdata type. [RT #14967]

1887.   [bug]           The cache could delete expired records too fast for
                        clients with a virtual time in the past. [RT #14991]

1886.   [bug]           fctx_create() could return success even though it
                        failed. [RT #14993]

1885.   [func]          dig: report the number of extra bytes still left in
                        the packet after processing all the records.

1884.   [cleanup]       dighost.c: move external declarations into <dig/dig.h>.

1883.   [bug]           dnssec-signzone, dnssec-keygen: handle negative debug
                        levels. [RT #14962]

1882.   [func]          Limit the number of recursive clients that can be
                        waiting for a single query (<qname,qtype,qclass>) to
                        resolve.  New options clients-per-query and
                        max-clients-per-query.

1881.   [func]          Add a system test for named-checkconf. [RT #14931]

1880.   [func]          The lame cache is now done on a <qname,qclass,qtype>
                        basis as some servers only appear to be lame for
                        certain query types.  [RT #14916]

1879.   [func]          "USE INTERNAL MALLOC" is now runtime selectable.
                        [RT #14892]

1878.   [func]          Detect duplicates of UDP queries we are recursing on
                        and drop them.  New stats category "duplicate".
                        [RT #2471]

1877.   [bug]           Fix unreasonably low quantum on call to
                        dns_rbt_destroy2().  Remove unnecessary unhash_node()
                        call. [RT #14919]

1876.   [func]          Additional memory debugging support to track size
                        and mctx arguments. [RT #14814]
```

(continues on next page)

(continued from previous page)

```
1875.   [bug]           process_dhtkey() was using the wrong memory context
                        to free some memory. [RT #14890]

1874.   [port]          sunos: portability fixes. [RT #14814]

1873.   [port]          win32: isc__errno2result() now reports its caller.
                        [RT #13753]

1872.   [port]          win32: Handle ERROR_NETNAME_DELETED.  [RT #13753]

1871.   [placeholder]

1870.   [func]          Added framework for handling multiple EDNS versions.
                        [RT #14873]

1869.   [func]          dig can now specify the EDNS version when making
                        a query. [RT #14873]

1868.   [func]          edns-udp-size can now be overridden on a per
                        server basis. [RT #14851]

1867.   [bug]           It was possible to trigger a INSIST in
                        dlv_validatezonekey(). [RT #14846]

1866.   [bug]           resolv.conf parse errors were being ignored by
                        dig/host/nslookup. [RT #14841]

1865.   [bug]           Silently ignore nameservers in /etc/resolv.conf with
                        bad addresses. [RT #14841]

1864.   [bug]           Don't try the alternative transfer source if you
                        got a answer / transfer with the main source
                        address. [RT #14802]

1863.   [bug]           rrset-order "fixed" error messages not complete.

1862.   [func]          Add additional zone data constancy checks.
                        named-checkzone has extended checking of NS, MX and
                        SRV record and the hosts they reference.
                        named has extended post zone load checks.
                        New zone options: check-mx and integrity-check.
                        [RT #4940]

1861.   [bug]           dig could trigger a INSIST on certain malformed
                        responses. [RT #14801]

1860.   [port]          solaris 2.8: hack_shutup_pthreadmutexinit was
                        incorrectly set. [RT #14775]

1859.   [func]          Add support for CH A record. [RT #14695]

1858.   [bug]           The flush-zones-on-shutdown option wasn't being
```

(continues on next page)

(continued from previous page)

```
                        parsed. [RT #14686]

1857.   [bug]           named could trigger a INSIST() if reconfigured /
                        reloaded too fast.  [RT #14673]

1856.   [doc]           Switch Docbook toolchain from DSSSL to XSL.
                        [RT #11398]

1855.   [bug]           ixfr-from-differences was failing to detect changes
                        of ttl due to dns_diff_subtract() was ignoring the ttl
                        of records.  [RT #14616]

1854.   [bug]           lwres also needs to know the print format for
                        (long long).  [RT #13754]

1853.   [bug]           Rework how DLV interacts with proveunsecure().
                        [RT #13605]

1852.   [cleanup]       Remove last vestiges of dnssec-signkey and
                        dnssec-makekeyset (removed from Makefile years ago).

1851.   [doc]           Doxygen comment markup. [RT #11398]

1850.   [bug]           Memory leak in lwres_getipnodebyaddr(). [RT #14591]

1849.   [doc]           All forms of the man pages (docbook, man, html) should
                        have consistent copyright dates.

1848.   [bug]           Improve SMF integration. [RT #13238]

1847.   [bug]           isc_ondestroy_init() is called too late in
                        dns_rbtdb_create()/dns_rbtdb64_create().
                        [RT #13661]

1846.   [contrib]       query-loc-0.3.0 from Stephane Bortzmeyer
                        <bortzmeyer@nic.fr>.

1845.   [bug]           Improve error reporting to distinguish between
                        accept()/fcntl() and socket()/fcntl() errors.
                        [RT #13745]

1844.   [bug]           inet_pton() accepted more that 4 hexadecimal digits
                        for each 16 bit piece of the IPv6 address.  The text
                        representation of a IPv6 address has been tightened
                        to disallow this (draft-ietf-ipv6-addr-arch-v4-02.txt).
                        [RT #5662]

1843.   [cleanup]       CINCLUDES takes precedence over CFLAGS.  This helps
                        when CFLAGS contains "-I /usr/local/include"
                        resulting in old header files being used.

1842.   [port]          cmsg_len() could produce incorrect results on
```

(continues on next page)

(continued from previous page)

```
                        some platform. [RT #13744]

1841.   [bug]           "dig +nssearch" now makes a recursive query to
                        find the list of nameservers to query. [RT #13694]

1840.   [func]          dnssec-signzone can now randomize signature end times
                        (dnssec-signzone -j jitter). [RT #13609]

1839.   [bug]           <isc/hash.h> was not being installed.

1838.   [cleanup]       Don't allow Linux capabilities to be inherited.
                        [RT #13707]

1837.   [bug]           Compile time option ISC_FACILITY was not effective
                        for 'named -u <user>'.  [RT #13714]

1836.   [cleanup]       Silence compiler warnings in hash_test.c.

1835.   [bug]           Update dnssec-signzone's usage message. [RT #13657]

1834.   [bug]           Bad memset in rdata_test.c. [RT #13658]

1833.   [bug]           Race condition in isc_mutex_lock_profile(). [RT #13660]

1832.   [bug]           named fails to return BADKEY on unknown TSIG algorithm.
                        [RT #13620]

1831.   [doc]           Update named-checkzone documentation. [RT #13604]

1830.   [bug]           adb lame cache has sense of test reversed. [RT #13600]

1829.   [bug]           win32: "pid-file none;" broken. [RT #13563]

1828.   [bug]           isc_rwlock_init() failed to properly cleanup if it
                        encountered a error. [RT #13549]

1827.   [bug]           host: update usage message for '-a'. [RT #37116]

1826.   [bug]           Missing DESTROYLOCK() in isc_mem_createx() on out
                        of memory error. [RT #13537]

1825.   [bug]           Missing UNLOCK() on out of memory error from in
                        rbtdb.c:subtractrdataset(). [RT #13519]

1824.   [bug]           Memory leak on dns_zone_setdbtype() failure.
                        [RT #13510]

1823.   [bug]           Wrong macro used to check for point to point interface.
                        [RT #13418]

1822.   [bug]           check-names test for RT was reversed. [RT #13382]
```

(continues on next page)

(continued from previous page)

```
1821.   [placeholder]

1820.   [bug]           Gracefully handle acl loops. [RT #13659]

1819.   [bug]           The validator needed to check both the algorithm and
                        digest types of the DS to determine if it could be
                        used to introduce a secure zone. [RT #13593]

1818.   [bug]           'named-checkconf -z' triggered an INSIST. [RT #13599]

1817.   [func]          Add support for additional zone file formats for
                        improving loading performance.  The masterfile-format
                        option in named.conf can be used to specify a
                        non-default format.  A separate command
                        named-compilezone was provided to generate zone files
                        in the new format.  Additionally, the -I and -O options
                        for dnssec-signzone specify the input and output
                        formats.

1816.   [port]          UnixWare: failed to compile lib/isc/unix/net.c.
                        [RT #13597]

1815.   [bug]           nsupdate triggered a REQUIRE if the server was set
                        without also setting the zone and it encountered
                        a CNAME and was using TSIG.  [RT #13086]

1814.   [func]          UNIX domain controls are now supported.

1813.   [func]          Restructured the data locking framework using
                        architecture dependent atomic operations (when
                        available), improving response performance on
                        multi-processor machines significantly.
                        x86, x86_64, alpha, powerpc, and mips are currently
                        supported.

1812.   [port]          win32: IN6_IS_ADDR_UNSPECIFIED macro is incorrect.
                        [RT #13453]

1811.   [func]          Preserve the case of domain names in rdata during
                        zone transfers. [RT #13547]

1810.   [bug]           configure, lib/bind/configure make different default
                        decisions about whether to do a threaded build.
                        [RT #13212]

1809.   [bug]           "make distclean" failed for libbind if the platform
                        is not supported.

1808.   [bug]           zone.c:notify_zone() contained a race condition,
                        zone->db could change underneath it.  [RT #13511]

1807.   [bug]           When forwarding (forward only) set the active domain
```

(continues on next page)

(continued from previous page)

```
                        from the forward zone name. [RT #13526]

1806.   [bug]           The resolver returned the wrong result when a CNAME /
                        DNAME was encountered when fetching glue from a
                        secure namespace. [RT #13501]

1805.   [bug]           Pending status was not being cleared when DLV was
                        active. [RT #13501]

1804.   [bug]           Ensure that if we are queried for glue that it fits
                        in the additional section or TC is set to tell the
                        client to retry using TCP. [RT #10114]

1803.   [bug]           dnssec-signzone sometimes failed to remove old
                        RRSIGs. [RT #13483]

1802.   [bug]           Handle connection resets better. [RT #11280]

1801.   [func]          Report differences between hints and real NS rrset
                        and associated address records.

1800.   [bug]           Changes #1719 allowed a INSIST to be triggered.
                        [RT #13428]

1799.   [bug]           'rndc flushname' failed to flush negative cache
                        entries. [RT #13438]

1798.   [func]          The server syntax has been extended to support a
                        range of servers.  [RT #11132]

1797.   [func]          named-checkconf now check acls to verify that they
                        only refer to existing acls. [RT #13101]

1796.   [func]          "rndc freeze/thaw" now freezes/thaws all zones.

1795.   [bug]           "rndc dumpdb" was not fully documented.  Minor
                        formatting issues with "rndc dumpdb -all".  [RT #13396]

1794.   [func]          Named and named-checkzone can now both check for
                        non-terminal wildcard records.

1793.   [func]          Extend adjusting TTL warning messages. [RT #13378]

1792.   [func]          New zone option "notify-delay".  Specify a minimum
                        delay between sets of NOTIFY messages.

1791.   [bug]           'host -t a' still printed out AAAA and MX records.
                        [RT #13230]

1790.   [cleanup]       Move lib/dns/sec/dst up into lib/dns.  This should
                        allow parallel make to succeed.
```

(continues on next page)

(continued from previous page)

```
1789.   [bug]           Prerequisite test for tkey and dnssec could fail
                        with "configure --with-libtool".

1788.   [bug]           libbind9.la/libbind9.so needs to link against
                        libisccfg.la/libisccfg.so.

1787.   [port]          HPUX: both "cc" and "gcc" need -Wl,+vnocompatwarnings.

1786.   [port]          AIX: libt_api needs to be taught to look for
                        T_testlist in the main executable (--with-libtool).
                        [RT #13239]

1785.   [bug]           libbind9.la/libbind9.so needs to link against
                        libisc.la/libisc.so.

1784.   [cleanup]       "libtool -allow-undefined" is the default.
                        Leave hooks in configure to allow it to be set
                        if needed in the future.

1783.   [cleanup]       We only need one copy of libtool.m4, ltmain.sh in the
                        source tree.

1782.   [port]          OSX: --with-libtool + --enable-libbind broke on
                        __evOptMonoTime.  [RT #13219]

1781.   [port]          FreeBSD 5.3: set PTHREAD_SCOPE_SYSTEM. [RT #12810]

1780.   [bug]           Update libtool to 1.5.10.

1779.   [port]          OSF 5.1: libtool didn't handle -pthread correctly.

1778.   [port]          HUX 11.11: fix broken IN6ADDR_ANY_INIT and
                        IN6ADDR_LOOPBACK_INIT macros.

1777.   [port]          OSF 5.1: fix broken IN6ADDR_ANY_INIT and
                        IN6ADDR_LOOPBACK_INIT macros.

1776.   [port]          Solaris 2.9: fix broken IN6ADDR_ANY_INIT and
                        IN6ADDR_LOOPBACK_INIT macros.

1775.   [bug]           Only compile getnetent_r.c when threaded. [RT #13205]

1774.   [port]          Aix: Silence compiler warnings / build failures.
                        [RT #13154]

1773.   [bug]           Fast retry on host / net unreachable. [RT #13153]

1772.   [placeholder]

1771.   [placeholder]

1770.   [bug]           named-checkconf failed to report missing a missing
```

(continues on next page)

(continued from previous page)

```
                        file clause for rbt{64} master/hint zones. [RT #13009]

1769.   [port]          win32: change compiler flags /MTd ==> /MDd,
                        /MT ==> /MD.

1768.   [bug]           nsecnoexistnodata() could be called with a non-NSEC
                        rdataset. [RT #12907]

1767.   [port]          Builds on IPv6 platforms without IPv6 Advanced API
                        support for (struct in6_pktinfo) failed.  [RT #13077]

1766.   [bug]           Update the master file timestamp on successful refresh
                        as well as the journal's timestamp. [RT #13062]

1765.   [bug]           configure --with-openssl=auto failed. [RT #12937]

1764.   [bug]           dns_zone_replacedb failed to emit a error message
                        if there was no SOA record in the replacement db.
                        [RT #13016]

1763.   [func]          Perform sanity checks on NS records which refer to
                        'in zone' names. [RT #13002]

1762.   [bug]           isc_interfaceiter_create() could return ISC_R_SUCCESS
                        even when it failed. [RT #12995]

1761.   [bug]           'rndc dumpdb' didn't report unassociated entries.
                        [RT #12971]

1760.   [bug]           Host / net unreachable was not penalising rtt
                        estimates. [RT #12970]

1759.   [bug]           Named failed to startup if the OS supported IPv6
                        but had no IPv6 interfaces configured. [RT #12942]

1758.   [func]          Don't send notify messages to self. [RT #12933]

1757.   [func]          host now can turn on memory debugging flags with '-m'.

1756.   [func]          named-checkconf now checks the logging configuration.
                        [RT #12352]

1755.   [func]          allow-update is now settable at the options / view
                        level. [RT #6636]

1754.   [bug]           We weren't always attempting to query the parent
                        server for the DS records at the zone cut.
                        [RT #12774]

1753.   [bug]           Don't serve a slave zone which has no NS records.
                        [RT #12894]
```

(continues on next page)

(continued from previous page)

```
1752.   [port]          Move isc_app_start() to after ns_os_daemonise()
                        as some fork() implementations unblock the signals
                        that are blocked by isc_app_start(). [RT #12810]

1751.   [bug]           --enable-getifaddrs failed under linux. [RT #12867]

1750.   [port]          lib/bind/make/rules.in:subdirs was not bash friendly.
                        [RT #12864]
```

```
1749.   [bug]           'check-names response ignore;' failed to ignore.
                        [RT #12866]

1748.   [func]          dig now returns the byte count for axfr/ixfr.

1747.   [bug]           BIND 8 compatibility: named/named-checkconf failed
                        to parse "host-statistics-max" in named.conf.

1746.   [func]          Make public the function to read a key file,
                        dst_key_read_public(). [RT #12450]

1745.   [bug]           Dig/host/nslookup accept replies from link locals
                        regardless of scope if no scope was specified when
                        query was sent. [RT #12745]

1744.   [bug]           If tuple2msgname() failed to convert a tuple to
                        a name a REQUIRE could be triggered. [RT #12796]

1743.   [bug]           If isc_taskmgr_create() was not able to create the
                        requested number of worker threads then destruction
                        of the manager would trigger an INSIST() failure.
                        [RT #12790]

1742.   [bug]           Deleting all records at a node then adding a
                        previously existing record, in a single UPDATE
                        transaction, failed to leave / regenerate the
                        associated RRSIG records. [RT #12788]

1741.   [bug]           Deleting all records at a node in a secure zone
                        using a update-policy grant failed. [RT #12787]

1740.   [bug]           Replace rbt's hash algorithm as it performed badly
                        with certain zones. [RT #12729]

                        NOTE: a hash context now needs to be established
                        via isc_hash_create() if the application was not
                        already doing this.

1739.   [bug]           dns_rbt_deletetree() could incorrectly return
                        ISC_R_QUOTA.  [RT #12695]

1738.   [bug]           Enable overrun checking by default. [RT #12695]
```

(continues on next page)

(continued from previous page)

```
1737.   [bug]           named failed if more than 16 masters were specified.
                        [RT #12627]

1736.   [bug]           dst_key_fromnamedfile() could fail to read a
                        public key. [RT #12687]

1735.   [bug]           'dig +sigtrace' could die with a REQUIRE failure.
                        [RE #12688]

1734.   [cleanup]       'rndc-confgen -a -t' remove extra '/' in path.
                        [RT #12588]

1733.   [bug]           Return non-zero exit status on initial load failure.
                        [RT #12658]

1732.   [bug]           'rrset-order name "*"' wasn't being applied to ".".
                        [RT #12467]

1731.   [port]          darwin: relax version test in ifconfig.sh.
                        [RT #12581]

1730.   [port]          Determine the length type used by the socket API.
                        [RT #12581]

1729.   [func]          Improve check-names error messages.

1728.   [doc]           Update check-names documentation.

1727.   [bug]           named-checkzone: check-names support didn't match
                        documentation.

1726.   [port]          aix5: add support for aix5.

1725.   [port]          linux: update error message on interaction of threads,
                        capabilities and setuid support (named -u). [RT #12541]

1724.   [bug]           Look for DNSKEY records with "dig +sigtrace".
                        [RT #12557]

1723.   [cleanup]       Silence compiler warnings from t_tasks.c. [RT #12493]

1722.   [bug]           Don't commit the journal on malformed ixfr streams.
                        [RT #12519]

1721.   [bug]           Error message from the journal processing were not
                        always identifying the relevant journal. [RT #12519]

1720.   [bug]           'dig +chase' did not terminate on a RFC 2308 Type 1
                        negative response. [RT #12506]

1719.   [bug]           named was not correctly caching a RFC 2308 Type 1
                        negative response. [RT #12506]
```

(continues on next page)

(continued from previous page)

```
1718.   [bug]           nsupdate was not handling RFC 2308 Type 3 negative
                        responses when looking for the zone / master server.
                        [RT #12506]

1717.   [port]          solaris: ifconfig.sh did not support Solaris 10.
                        "ifconfig.sh down" didn't work for Solaris 9.

1716.   [doc]           named.conf(5) was being installed in the wrong
                        location.  [RT #12441]

1715.   [func]          'dig +trace' now randomly selects the next servers
                        to try.  Report if there is a bad delegation.

1714.   [bug]           dig/host/nslookup were only trying the first
                        address when a nameserver was specified by name.
                        [RT #12286]

1713.   [port]          linux: extend capset failure message to say:
                        please ensure that the capset kernel module is
                        loaded.  see insmod(8)

1712.   [bug]           Missing FULLCHECK for "trusted-key" in dig.

1711.   [func]          'rndc unfreeze' has been deprecated by 'rndc thaw'.

1710.   [func]          'rndc notify zone [class [view]]' resend the NOTIFY
                        messages for the specified zone. [RT #9479]

1709.   [port]          solaris: add SMF support from Sun.

1708.   [cleanup]       Replaced dns_fullname_hash() with dns_name_fullhash()
                        for conformance to the name space convention.  Binary
                        backward compatibility to the old function name is
                        provided. [RT #12376]

1707.   [contrib]       sdb/ldap updated to version 1.0-beta.

1706.   [bug]           'rndc stop' failed to cause zones to be flushed
                        sometimes. [RT #12328]

1705.   [func]          Allow the journal's name to be changed via named.conf.

1704.   [port]          lwres needed a snprintf() implementation for
                        platforms without snprintf().  Add missing
                        "#include <isc/print.h>". [RT #12321]

1703.   [bug]           named would loop sending NOTIFY messages when it
                        failed to receive a response. [RT #12322]

1702.   [bug]           also-notify should not be applied to built in zones.
                        [RT #12323]
```

(continues on next page)

(continued from previous page)

```
1701.   [doc]           A minimal named.conf man page.

1700.   [func]          nslookup is no longer to be treated as deprecated.
                        Remove "deprecated" warning message.  Add man page.

1699.   [bug]           dnssec-signzone can generate "not exact" errors
                        when resigning. [RT #12281]

1698.   [doc]           Use reserved IPv6 documentation prefix.

1697.   [bug]           xxx-source{,-v6} was not effective when it
                        specified one of listening addresses and a
                        different port than the listening port. [RT #12257]

1696.   [bug]           dnssec-signzone failed to clean out nodes that
                        consisted of only NSEC and RRSIG records.
                        [RT #12154]

1695.   [bug]           DS records when forwarding require special handling.
                        [RT #12133]

1694.   [bug]           Report if the builtin views of "_default" / "_bind"
                        are defined in named.conf. [RT #12023]

1693.   [bug]           max-journal-size was not effective for master zones
                        with ixfr-from-differences set. [RT #12024]

1692.   [bug]           Don't set -I, -L and -R flags when libcrypto is in
                        /usr/lib. [RT #11971]

1691.   [bug]           sdb's attachversion was not complete. [RT #11990]

1690.   [bug]           Delay detaching view from the client until UPDATE
                        processing completes when shutting down. [RT #11714]

1689.   [bug]           DNS_NAME_TOREGION() and DNS_NAME_SPLIT() macros
                        contained gratuitous semicolons. [RT #11707]

1688.   [bug]           LDFLAGS was not supported.

1687.   [bug]           Race condition in dispatch. [RT #10272]

1686.   [bug]           Named sent a extraneous NOTIFY when it received a
                        redundant UPDATE request. [RT #11943]

1685.   [bug]           Change #1679 loop tests weren't quite right.

1684.   [func]          ixfr-from-differences now takes master and slave in
                        addition to yes and no at the options and view levels.

1683.   [bug]           dig +sigchase could leak memory. [RT #11445]
```

(continues on next page)

(continued from previous page)

```
1682.   [port]          Update configure test for (long long) printf format.
                        [RT #5066]

1681.   [bug]           Only set SO_REUSEADDR when a port is specified in
                        isc_socket_bind(). [RT #11742]

1680.   [func]          rndc: the source address can now be specified.

1679.   [bug]           When there was a single nameserver with multiple
                        addresses for a zone not all addresses were tried.
                        [RT #11706]

1678.   [bug]           RRSIG should use TYPEXXXXX for unknown types.

1677.   [bug]           dig: +aaonly didn't work, +aaflag undocumented.

1676.   [func]          New option "allow-query-cache".  This lets
                        allow-query be used to specify the default zone
                        access level rather than having to have every
                        zone override the global value.  allow-query-cache
                        can be set at both the options and view levels.
                        If allow-query-cache is not set allow-query applies.

1675.   [bug]           named would sometimes add extra NSEC records to
                        the authority section.

1674.   [port]          linux: increase buffer size used to scan
                        /proc/net/if_inet6.

1673.   [port]          linux: issue a error messages if IPv6 interface
                        scans fails.

1672.   [cleanup]       Tests which only function in a threaded build
                        now return R:THREADONLY (rather than R:UNTESTED)
                        in a non-threaded build.

1671.   [contrib]       queryperf: add NAPTR to the list of known types.

1670.   [func]          Log UPDATE requests to slave zones without an acl as
                        "disabled" at debug level 3. [RT #11657]

1669.   [placeholder]

1668.   [bug]           DIG_SIGCHASE was making bin/dig/host dump core.

1667.   [port]          linux: not all versions have IF_NAMESIZE.

1666.   [bug]           The optional port on hostnames in dual-stack-servers
                        was being ignored.

1665.   [func]          rndc now allows addresses to be set in the
```

(continues on next page)

(continued from previous page)

```
                        server clauses.

1664.   [bug]           nsupdate needed KEY for SIG(0), not DNSKEY.

1663.   [func]          Look for OpenSSL by default.

1662.   [bug]           Change #1658 failed to change one use of 'type'
                        to 'keytype'.

1661.   [bug]           Restore dns_name_concatenate() call in
                        adb.c:set_target().  [RT #11582]

1660.   [bug]           win32: connection_reset_fix() was being called
                        unconditionally.  [RT #11595]

1659.   [cleanup]       Cleanup some messages that were referring to KEY vs
                        DNSKEY, NXT vs NSEC and SIG vs RRSIG.

1658.   [func]          Update dnssec-keygen to default to KEY for HMAC-MD5
                        and DH.  Tighten which options apply to KEY and
                        DNSKEY records.

1657.   [doc]           ARM: document query log output.

1656.   [doc]           Update DNSSEC description in ARM to cover DS, NSEC
                        DNSKEY and RRSIG.  [RT #11542]

1655.   [bug]           Logging multiple versions w/o a size was broken.
                        [RT #11446]

1654.   [bug]           isc_result_totext() contained array bounds read
                        error.

1653.   [func]          Add key type checking to dst_key_fromfilename(),
                        DST_TYPE_KEY should be used to read TSIG, TKEY and
                        SIG(0) keys.

1652.   [bug]           TKEY still uses KEY.

1651.   [bug]           dig: process multiple dash options.

1650.   [bug]           dig, nslookup: flush standard out after each command.

1649.   [bug]           Silence "unexpected non-minimal diff" message.
                        [RT #11206]

1648.   [func]          Update dnssec-lookaside named.conf syntax to support
                        multiple dnssec-lookaside namespaces (not yet
                        implemented).

1647.   [bug]           It was possible trigger a INSIST when chasing a DS
                        record that required walking back over a empty node.
```

(continues on next page)

(continued from previous page)

```
                        [RT #11445]

1646.   [bug]           win32: logging file versions didn't work with
                        non-UNC filenames.  [RT #11486]

1645.   [bug]           named could trigger a REQUIRE failure if multiple
                        masters with keys are specified.

1644.   [bug]           Update the journal modification time after a
                        successful refresh query. [RT #11436]

1643.   [bug]           dns_db_closeversion() could leak memory / node
                        references. [RT #11163]

1642.   [port]          Support OpenSSL implementations which don't have
                        DSA support. [RT #11360]

1641.   [bug]           Update the check-names description in ARM. [RT #11389]

1640.   [bug]           win32: isc_socket_cancel(ISC_SOCKCANCEL_ACCEPT) was
                        incorrectly closing the socket.  [RT #11291]

1639.   [func]          Initial dlv system test.

1638.   [bug]           "ixfr-from-differences" could generate a REQUIRE
                        failure if the journal open failed. [RT #11347]

1637.   [bug]           Node reference leak on error in addnoqname().

1636.   [bug]           The dump done callback could get ISC_R_SUCCESS even if
                        a error had occurred.  The database version no longer
                        matched the version of the database that was dumped.

1635.   [bug]           Memory leak on error in query_addds().

1634.   [bug]           named didn't supply a useful error message when it
                        detected duplicate views.  [RT #11208]

1633.   [bug]           named should return NOTIMP to update requests to a
                        slaves without a allow-update-forwarding acl specified.
                        [RT #11331]

1632.   [bug]           nsupdate failed to send prerequisite only UPDATE
                        messages. [RT #11288]

1631.   [bug]           dns_journal_compact() could sometimes corrupt the
                        journal. [RT #11124]

1630.   [contrib]       queryperf: add support for IPv6 transport.

1629.   [func]          dig now supports IPv6 scoped addresses with the
                        extended format in the local-server part. [RT #8753]
```

(continues on next page)

(continued from previous page)

```
1628.   [bug]           Typo in Compaq Trucluster support. [RT #11264]

1627.   [bug]           win32: sockets were not being closed when the
                        last external reference was removed. [RT #11179]

1626.   [bug]           --enable-getifaddrs was broken. [RT #11259]

1625.   [bug]           named failed to load/transfer RFC2535 signed zones
                        which contained CNAMES. [RT #11237]

1624.   [bug]           zonemgr_putio() call should be locked. [RT #11163]

1623.   [bug]           A serial number of zero was being displayed in the
                        "sending notifies" log message when also-notify was
                        used. [RT #11177]

1622.   [func]          probe the system to see if IPV6_(RECV)PKTINFO is
                        available, and suppress wildcard binding if not.

1621.   [bug]           match-destinations did not work for IPv6 TCP queries.
                        [RT #11156]

1620.   [func]          When loading a zone report if it is signed. [RT #11149]

1619.   [bug]           Missing ISC_LIST_UNLINK in end_reserved_dispatches().
                        [RT #11118]

1618.   [bug]           Fencepost errors in dns_name_ishostname() and
                        dns_name_ismailbox() could trigger a INSIST().

1617.   [port]          win32: VC++ 6.0 support.

1616.   [compat]        Ensure that named's version is visible in the core
                        dump. [RT #11127]

1615.   [port]          Define ISC_SOCKADDR_LEN_T based on _BSD_SOCKLEN_T_ if
                        it is defined.

1614.   [port]          win32: silence resource limit messages. [RT #11101]

1613.   [bug]           Builds would fail on machines w/o a if_nametoindex().
                        Missing #ifdef ISC_PLATFORM_HAVEIFNAMETOINDEX/#endif.
                        [RT #11119]

1612.   [bug]           check-names at the option/view level could trigger
                        an INSIST. [RT #11116]

1611.   [bug]           solaris: IPv6 interface scanning failed to cope with
                        no active IPv6 interfaces.

1610.   [bug]           On dual stack machines "dig -b" failed to set the
```

(continues on next page)

(continued from previous page)

```
                        address type to be looked up with "@server".
                        [RT #11069]

1609.   [func]          dig now has support to chase DNSSEC signature chains.
                        Requires -DDIG_SIGCHASE=1 to be set in STD_CDEFINES.

                        DNSSEC validation code in dig coded by Olivier Courtay
                        (olivier.courtay@irisa.fr) for the IDsA project
                        (http://idsa.irisa.fr).

1608.   [func]          dig and host now accept -4/-6 to select IP transport
                        to use when making queries.

1607.   [bug]           dig, host and nslookup were still using random()
                        to generate query ids. [RT #11013]

1606.   [bug]           DLV insecurity proof was failing.

1605.   [func]          New dns_db_find() option DNS_DBFIND_COVERINGNSEC.

1604.   [bug]           A xfrout_ctx_create() failure would result in
                        xfrout_ctx_destroy() being called with a
                        partially initialized structure.

1603.   [bug]           nsupdate: set interactive based on isatty().
                        [RT #10929]

1602.   [bug]           Logging to a file failed unless a size was specified.
                        [RT #10925]

1601.   [bug]           Silence spurious warning 'both "recursion no;" and
                        "allow-recursion" active' warning from view "_bind".
                        [RT #10920]

1600.   [bug]           Duplicate zone pre-load checks were not case
                        insensitive.

1599.   [bug]           Fix memory leak on error path when checking named.conf.

1598.   [func]          Specify that certain parts of the namespace must
                        be secure (dnssec-must-be-secure).

1597.   [func]          Allow notify-source and query-source to be specified
                        on a per server basis similar to transfer-source.
                        [RT #6496]

1596.   [func]          Accept 'notify-source' style syntax for query-source.

1595.   [func]          New notify type 'master-only'.  Enable notify for
                        master zones only.

1594.   [bug]           'rndc dumpdb' could prevent named from answering
```

(continues on next page)

```
                        queries while the dump was in progress.  [RT #10565]

1593.   [bug]           rndc should return "unknown command" to unknown
                        commands. [RT #10642]

1592.   [bug]           configure_view() could leak a dispatch. [RT #10675]

1591.   [bug]           libbind: updated to BIND 8.4.5.

1590.   [port]          netbsd: update thread support.

1589.   [func]          DNSSEC lookaside validation.

1588.   [bug]           win32: TCP sockets could become blocked. [RT #10115]

1587.   [bug]           dns_message_settsigkey() failed to clear existing key.
                        [RT #10590]

1586.   [func]          "check-names" is now implemented.

1585.   [placeholder]

1584.   [bug]           "make test" failed with a read only source tree.
                        [RT #10461]

1583.   [bug]           Records add via UPDATE failed to get the correct trust
                        level. [RT #10452]

1582.   [bug]           rrset-order failed to work on RRsets with more
                        than 32 elements. [RT #10381]

1581.   [func]          Disable DNSSEC support by default.  To enable
                        DNSSEC specify "dnssec-enable yes;" in named.conf.

1580.   [bug]           Zone destruction on final detach takes a long time.
                        [RT #3746]

1579.   [bug]           Multiple task managers could not be created.

1578.   [bug]           Don't use CLASS E IPv4 addresses when resolving.
                        [RT #10346]

1577.   [bug]           Use isc_uint32_t in ultrasparc optimizer bug
                        workaround code. [RT #10331]

1576.   [bug]           Race condition in dns_dispatch_addresponse().
                        [RT #10272]

1575.   [func]          Log TSIG name on TSIG verify failure. [RT #4404]

1574.   [bug]           Don't attempt to open the controls socket(s) when
                        running tests. [RT #9091]
```

(continues on next page)

(continued from previous page)

```
1573.   [port]          linux: update to libtool 1.5.2 so that
                        "make install DESTDIR=/xx" works with
                        "configure --with-libtool".  [RT #9941]

1572.   [bug]           nsupdate: sign the soa query to find the enclosing
                        zone if the server is specified. [RT #10148]

1571.   [bug]           rbt:hash_node() could fail leaving the hash table
                        in an inconsistent state.  [RT #10208]

1570.   [bug]           nsupdate failed to handle classes other than IN.
                        New keyword 'class' which sets the default class.
                        [RT #10202]

1569.   [func]          nsupdate new command 'answer' which displays the
                        complete answer message to the last update.

1568.   [bug]           nsupdate now reports that the update failed in
                        interactive mode. [RT #10236]

1567.   [maint]         B.ROOT-SERVERS.NET is now 192.228.79.201.

1566.   [port]          Support for the cmsg framework on Solaris and HP/UX.
                        This also solved the problem that match-destinations
                        for IPv6 addresses did not work on these systems.
                        [RT #10221]

1565.   [bug]           CD flag should be copied to outgoing queries unless
                        the query is under a secure entry point in which case
                        CD should be set.

1564.   [func]          Attempt to provide a fallback entropy source to be
                        used if named is running chrooted and named is unable
                        to open entropy source within the chroot area.
                        [RT #10133]

1563.   [bug]           Gracefully fail when unable to obtain neither an IPv4
                        nor an IPv6 dispatch. [RT #10230]

1562.   [bug]           isc_socket_create() and isc_socket_accept() could
                        leak memory under error conditions. [RT #10230]

1561.   [bug]           It was possible to release the same name twice if
                        named ran out of memory. [RT #10197]

1560.   [port]          FreeBSD: work around FreeBSD 5.2 mapping EAI_NODATA
                        and EAI_NONAME to the same value.

1559.   [port]          named should ignore SIGFSZ.

1558.   [func]          New DNSSEC 'disable-algorithms'.  Support entry into
                        child zones for which we don't have a supported
```

(continues on next page)

(continued from previous page)

```
                        algorithm.  Such child zones are treated as unsigned.

1557.   [func]          Implement missing DNSSEC tests for
                        * NOQNAME proof with wildcard answers.
                        * NOWILDARD proof with NXDOMAIN.
                        Cache and return NOQNAME with wildcard answers.

1556.   [bug]           nsupdate now treats all names as fully qualified.
                        [RT #6427]

1555.   [func]          'rrset-order cyclic' no longer has a random starting
                        point per query. [RT #7572]

1554.   [bug]           dig, host, nslookup failed when no nameservers
                        were specified in /etc/resolv.conf. [RT #8232]

1553.   [bug]           The windows socket code could stop accepting
                        connections. [RT #10115]

1552.   [bug]           Accept NOTIFY requests from mapped masters if
                        matched-mapped is set. [RT #10049]

1551.   [port]          Open "/dev/null" before calling chroot().

1550.   [port]          Call tzset(), if available, before calling chroot().

1549.   [func]          named-checkzone can now write out the zone contents
                        in a easily parsable format (-D and -o).

1548.   [bug]           When parsing APL records it was possible to silently
                        accept out of range ADDRESSFAMILY values. [RT #9979]

1547.   [bug]           Named wasted memory recording duplicate lame zone
                        entries. [RT #9341]

1546.   [bug]           We were rejecting valid secure CNAME to negative
                        answers.

1545.   [bug]           It was possible to leak memory if named was unable to
                        bind to the specified transfer source and TSIG was
                        being used. [RT #10120]

1544.   [bug]           Named would logged a single entry to a file despite it
                        being over the specified size limit.

1543.   [bug]           Logging using "versions unlimited" did not work.

1542.   [placeholder]

1541.   [func]          NSEC now uses new bitmap format.

1540.   [bug]           "rndc reload <dynamiczone>" was silently accepted.
```

(continues on next page)

(continued from previous page)

```
                        [RT #8934]

1539.   [bug]           Open UDP sockets for notify-source and transfer-source
                        that use reserved ports at startup. [RT #9475]

1538.   [placeholder]   rt9997

1537.   [func]          New option "querylog".  If set specify whether query
                        logging is to be enabled or disabled at startup.

1536.   [bug]           Windows socket code failed to log a error description
                        when returning ISC_R_UNEXPECTED. [RT #9998]

1535.   [placeholder]

1534.   [bug]           Race condition when priming cache. [RT #9940]

1533.   [func]          Warn if both "recursion no;" and "allow-recursion"
                        are active. [RT #4389]

1532.   [port]          netbsd: the configure test for <sys/sysctl.h>
                        requires <sys/param.h>.

1531.   [port]          AIX more libtool fixes.

1530.   [bug]           It was possible to trigger a INSIST() failure if a
                        slave master file was removed at just the correct
                        moment. [RT #9462]

1529.   [bug]           "notify explicit;" failed to log that NOTIFY messages
                        were being sent for the zone. [RT #9442]

1528.   [cleanup]       Simplify some dns_name_ functions based on the
                        deprecation of bitstring labels.

1527.   [cleanup]       Reduce the number of gettimeofday() calls without
                        losing necessary timer granularity.

1526.   [func]          Implemented "additional section caching (or acache)",
                        an internal cache framework for additional section
                        content to improve response performance.  Several
                        configuration options were provided to control the
                        behavior.

1525.   [bug]           dns_cache_create() could trigger a REQUIRE
                        failure in isc_mem_put() during error cleanup.
                        [RT #9360]

1524.   [port]          AIX needs to be able to resolve all symbols when
                        creating shared libraries (--with-libtool).

1523.   [bug]           Fix race condition in rbtdb. [RT #9189]
```

(continues on next page)

(continued from previous page)

```
1522.   [bug]           dns_db_findnode() relax the requirements on 'name'.
                        [RT #9286]

1521.   [bug]           dns_view_createresolver() failed to check the
                        result from isc_mem_create(). [RT #9294]

1520.   [protocol]      Add SSHFP (SSH Finger Print) type.

1519.   [bug]           dnssec-signzone:nsec_setbit() computed the wrong
                        length of the new bitmap.

1518.   [bug]           dns_nsec_buildrdata(), and hence dns_nsec_build(),
                        contained a off-by-one error when working out the
                        number of octets in the bitmap.

1517.   [port]          Support for IPv6 interface scanning on HP/UX and
                        TrueUNIX 5.1.

1516.   [func]          Roll the DNSSEC types to RRSIG, NSEC and DNSKEY.

1515.   [func]          Allow transfer source to be set in a server statement.
                        [RT #6496]

1514.   [bug]           named: isc_hash_destroy() was being called too early.
                        [RT #9160]

1513.   [doc]           Add "US" to root-delegation-only exclude list.

1512.   [bug]           Extend the delegation-only logging to return query
                        type, class and responding nameserver.

1511.   [bug]           delegation-only was generating false positives
                        on negative answers from sub-zones.

1510.   [func]          New view option "root-delegation-only".  Apply
                        delegation-only check to all TLDs and root.
                        Note there are some TLDs that are NOT delegation
                        only (e.g. DE, LV, US and MUSEUM) these can be excluded
                        from the checks by using exclude.

                        root-delegation-only exclude {
                                "DE"; "LV"; "US"; "MUSEUM";
                        };

1509.   [bug]           Hint zones should accept delegation-only.  Forward
                        zone should not accept delegation-only.

1508.   [bug]           Don't apply delegation-only checks to answers from
                        forwarders.

1507.   [bug]           Handle BIND 8 style returns to NS queries to parents
```

(continues on next page)

(continued from previous page)

```
                        when making delegation-only checks.

1506.   [bug]           Wrong return type for dns_view_isdelegationonly().

1505.   [bug]           Uninitialized rdataset in sdb. [RT #8750]

1504.   [func]          New zone type "delegation-only".

1503.   [port]          win32: install libeay32.dll outside of system32.

1502.   [bug]           nsupdate: adjust timeouts for UPDATE requests over TCP.

1501.   [func]          Allow TCP queue length to be specified via
                        named.conf, tcp-listen-queue.

1500.   [bug]           host failed to lookup MX records.  Also look up
                        AAAA records.
```

```
1499.   [bug]           isc_random need to be seeded better if arc4random()
                        is not used.

1498.   [port]          bsdos: 5.x support.

1497.   [placeholder]

1496.   [port]          test for pthread_attr_setstacksize().

1495.   [cleanup]       Replace hash functions with universal hash.

1494.   [security]      Turn on RSA BLINDING as a precaution.

1493.   [placeholder]

1492.   [cleanup]       Preserve rwlock quota context when upgrading /
                        downgrading. [RT #5599]

1491.   [bug]           dns_master_dump*() would produce extraneous $ORIGIN
                        lines. [RT #6206]

1490.   [bug]           Accept reading state as well as working state in
                        ns_client_next(). [RT #6813]

1489.   [compat]        Treat 'allow-update' on slave zones as a warning.
                        [RT #3469]

1488.   [bug]           Don't override trust levels for glue addresses.
                        [RT #5764]

1487.   [bug]           A REQUIRE() failure could be triggered if a zone was
                        queued for transfer and the zone was then removed.
                        [RT #6189]
```

(continues on next page)

(continued from previous page)

```
1486.   [bug]           isc_print_snprintf() '%%' consumed one too many format
                        characters. [RT #8230]

1485.   [bug]           gen failed to handle high type values. [RT #6225]

1484.   [bug]           The number of records reported after a AXFR was wrong.
                        [RT #6229]

1483.   [bug]           dig axfr failed if the message id in the answer failed
                        to match that in the request.  Only the id in the first
                        message is required to match. [RT #8138]

1482.   [bug]           named could fail to start if the kernel supports
                        IPv6 but no interfaces are configured.  Similarly
                        for IPv4. [RT #6229]

1481.   [bug]           Refresh and stub queries failed to use masters keys
                        if specified. [RT #7391]

1480.   [bug]           Provide replay protection for rndc commands.  Full
                        replay protection requires both rndc and named to
                        be updated.  Partial replay protection (limited
                        exposure after restart) is provided if just named
                        is updated.

1479.   [bug]           cfg_create_tuple() failed to handle out of
                        memory cleanup.  parse_list() would leak memory
                        on syntax errors.

1478.   [port]          ifconfig.sh didn't account for other virtual
                        interfaces.  It now takes a optional argument
                        to specify the first interface number. [RT #3907]

1477.   [bug]           memory leak using stub zones and TSIG.

1476.   [placeholder]

1475.   [port]          Probe for old sprintf().

1474.   [port]          Provide strtoul() and memmove() for platforms
                        without them.

1473.   [bug]           create_map() and create_string() failed to handle out
                        of memory cleanup.  [RT #6813]

1472.   [contrib]       idnkit-1.0 from JPNIC, replaces mdnkit.

1471.   [bug]           libbind: updated to BIND 8.4.0.

1470.   [bug]           Incorrect length passed to snprintf. [RT #5966]

1469.   [func]          Log end of outgoing zone transfer at same level
```

(continues on next page)

(continued from previous page)

```
                        as the start of transfer is logged. [RT #4441]

1468.   [func]          Internal zones are no longer counted for
                        'rndc status'.  [RT #4706]

1467.   [func]          $GENERATES now supports optional class and ttl.

1466.   [bug]           lwresd configuration errors resulted in memory
                        and lock leaks.  [RT #5228]

1465.   [bug]           isc_base64_decodestring() and isc_base64_tobuffer()
                        failed to check that trailing bits were zero allowing
                        some invalid base64 strings to be accepted.  [RT #5397]

1464.   [bug]           Preserve "out of zone" data for outgoing zone
                        transfers. [RT #5192]

1463.   [bug]           dns_rdata_from{wire,struct}() failed to catch bad
                        NXT bit maps. [RT #5577]

1462.   [bug]           parse_sizeval() failed to check the token type.
                        [RT #5586]

1461.   [bug]           Remove deadlock from rbtdb code. [RT #5599]

1460.   [bug]           inet_pton() failed to reject certain malformed
                        IPv6 literals.

1459.   [placeholder]

1458.   [cleanup]       sprintf() -> snprintf().

1457.   [port]          Provide strlcat() and strlcpy() for platforms without
                        them.

1456.   [contrib]       gen-data-queryperf.py from Stephane Bortzmeyer.

1455.   [bug]           <netaddr> missing from server grammar in
                        doc/misc/options. [RT #5616]

1454.   [port]          Use getifaddrs() if available for interface scanning.
                        --disable-getifaddrs to override.  Glibc currently
                        has a getifaddrs() that does not support IPv6.
                        Use --enable-getifaddrs=glibc to force the use of
                        this version under linux machines.

1453.   [doc]           ARM: $GENERATE example wasn't accurate. [RT #5298]

1452.   [placeholder]

1451.   [bug]           rndc-confgen didn't exit with a error code for all
                        failures. [RT #5209]
```

(continues on next page)

(continued from previous page)

```
1450.   [bug]           Fetching expired glue failed under certain
                        circumstances.  [RT #5124]

1449.   [bug]           query_addbestns() didn't handle running out of memory
                        gracefully.

1448.   [bug]           Handle empty wildcards labels.

1447.   [bug]           We were casting (unsigned int) to and from (void *).
                        rdataset->private4 is now rdataset->privateuint4
                        to reflect a type change.

1446.   [func]          Implemented undocumented alternate transfer sources
                        from BIND 8.  See use-alt-transfer-source,
                        alt-transfer-source and alt-transfer-source-v6.

                        SECURITY: use-alt-transfer-source is ENABLED unless
                        you are using views.  This may cause a security risk
                        resulting in accidental disclosure of wrong zone
                        content if the master supplying different source
                        content based on IP address.  If you are not certain
                        ISC recommends setting use-alt-transfer-source no;

1445.   [bug]           DNS_ADBFIND_STARTATROOT broke stub zones.  This has
                        been replaced with DNS_ADBFIND_STARTATZONE which
                        causes the search to start using the closest zone.

1444.   [func]          dns_view_findzonecut2() allows you to specify if the
                        cache should be searched for zone cuts.

1443.   [func]          Masters lists can now be specified and referenced
                        in zone masters clauses and other masters lists.

1442.   [func]          New functions for manipulating port lists:
                        dns_portlist_create(), dns_portlist_add(),
                        dns_portlist_remove(), dns_portlist_match(),
                        dns_portlist_attach() and dns_portlist_detach().

1441.   [func]          It is now possible to tell dig to bind to a specific
                        source port.

1440.   [func]          It is now possible to tell named to avoid using
                        certain source ports (avoid-v4-udp-ports,
                        avoid-v6-udp-ports).

1439.   [bug]           Named could return NOERROR with certain NOTIFY
                        failures.  Return NOTAUTH if the NOTIFY zone is
                        not being served.

1438.   [func]          Log TSIG (if any) when logging NOTIFY requests.
```

(continues on next page)

(continued from previous page)

```
1437.   [bug]           Leave space for stdio to work in. [RT #5033]

1436.   [func]          dns_zonemgr_resumexfrs() can be used to restart
                        stalled transfers.

1435.   [bug]           zmgr_resume_xfrs() was being called read locked
                        rather than write locked.  zmgr_resume_xfrs()
                        was not being called if the zone was being
                        shutdown.

1434.   [bug]           "rndc reconfig" failed to initiate the initial
                        zone transfer of new slave zones.

1433.   [bug]           named could trigger a REQUIRE failure if it could
                        not get a file descriptor when attempting to write
                        a master file. [RT #4347]

1432.   [func]          The advertised EDNS UDP buffer size can now be set
                        via named.conf (edns-udp-size).

1431.   [bug]           isc_print_snprintf() "%s" with precision could walk off
                        end of argument. [RT #5191]

1430.   [port]          linux: IPv6 interface scanning support.

1429.   [bug]           Prevent the cache getting locked to old servers.

1428.   [placeholder]

1427.   [bug]           Race condition in adb with threaded build.

1426.   [placeholder]

1425.   [port]          linux/libbind: define __USE_MISC when testing *_r()
                        function prototypes in netdb.h.  [RT #4921]

1424.   [bug]           EDNS version not being correctly printed.

1423.   [contrib]       queryperf: added A6 and SRV.

1422.   [func]          Log name/type/class when denying a query.  [RT #4663]

1421.   [func]          Differentiate updates that don't succeed due to
                        prerequisites (unsuccessful) vs other reasons
                        (failed).

1420.   [port]          solaris: work around gcc optimizer bug.

1419.   [port]          openbsd: use /dev/arandom. [RT #4950]

1418.   [bug]           'rndc reconfig' did not cause new slaves to load.
```

(continues on next page)

(continued from previous page)

```
1417.   [func]          ID.SERVER/CHAOS is now a built in zone.
                        See "server-id" for how to configure.

1416.   [bug]           Empty node should return NOERROR NODATA, not NXDOMAIN.
                        [RT #4715]

1415.   [func]          DS TTL now derived from NS ttl.  NXT TTL now derived
                        from SOA MINIMUM.

1414.   [func]          Support for KSK flag.

1413.   [func]          Explicitly request the (re-)generation of DS records
                        from keysets (dnssec-signzone -g).

1412.   [func]          You can now specify servers to be tried if a nameserver
                        has IPv6 address and you only support IPv4 or the
                        reverse. See dual-stack-servers.

1411.   [bug]           empty nodes should stop wildcard matches. [RT #4802]

1410.   [func]          Handle records that live in the parent zone, e.g. DS.

1409.   [bug]           DS should have attribute DNS_RDATATYPEATTR_DNSSEC.

1408.   [bug]           "make distclean" was not complete. [RT #4700]

1407.   [bug]           lfsr incorrectly implements the shift register.
                        [RT #4617]

1406.   [bug]           dispatch initializes one of the LFSR's with a incorrect
                        polynomial.  [RT #4617]

1405.   [func]          Use arc4random() if available.

1404.   [bug]           libbind: ns_name_ntol() could overwrite a zero length
                        buffer.

1403.   [func]          dnssec-signzone, dnssec-keygen, dnssec-makekeyset
                        dnssec-signkey now report their version in the
                        usage message.

1402.   [cleanup]       A6 has been moved to experimental and is no longer
                        fully supported.

1401.   [bug]           adb wasn't clearing state when the timer expired.

1400.   [bug]           Block the addition of wildcard NS records by IXFR
                        or UPDATE. [RT #3502]

1399.   [bug]           Use serial number arithmetic when testing SIG
                        timestamps. [RT #4268]
```

(continues on next page)

(continued from previous page)

```
1398.   [doc]           ARM: notify-also should have been also-notify.
                        [RT #4345]

1397.   [maint]         J.ROOT-SERVERS.NET is now 192.58.128.30.

1396.   [func]          dnssec-signzone: adjust the default signing time by
                        1 hour to allow for clock skew.

1395.   [port]          OpenSSL 0.9.7 defines CRYPTO_LOCK_ENGINE but doesn't
                        have a working implementation.  [RT #4079]

1394.   [func]          It is now possible to check if a particular element is
                        in a acl.  Remove duplicate entries from the localnets
                        acl.

1393.   [port]          Bind to individual IPv6 interfaces if IPV6_IPV6ONLY
                        is not available in the kernel to prevent accidentally
                        listening on IPv4 interfaces.

1392.   [bug]           named-checkzone: update usage.

1391.   [func]          Add support for IPv6 scoped addresses in named.

1390.   [func]          host now supports ixfr.

1389.   [bug]           named could fail to rotate long log files.  [RT #3666]

1388.   [port]          irix: check for sys/sysctl.h and NET_RT_IFLIST before
                        defining HAVE_IFLIST_SYSCTL. [RT #3770]

1387.   [bug]           named could crash due to an access to invalid memory
                        space (which caused an assertion failure) in
                        incremental cleaning.  [RT #3588]

1386.   [bug]           named-checkzone -z stopped on errors in a zone.
                        [RT #3653]

1385.   [bug]           Setting serial-query-rate to 10 would trigger a
                        REQUIRE failure.

1384.   [bug]           host was incompatible with BIND 8 in its exit code and
                        in the output with the -l option.  [RT #3536]

1383.   [func]          Track the serial number in a IXFR response and log if
                        a mismatch occurs.  This is a more specific error than
                        "not exact". [RT #3445]

1382.   [bug]           make install failed with --enable-libbind. [RT #3656]

1381.   [bug]           named failed to correctly process answers that
                        contained DNAME records where the resulting CNAME
                        resulted in a negative answer.
```

(continues on next page)

(continued from previous page)

```
1380.   [func]          'rndc recursing' dump recursing queries to
                        'recursing-file = "named.recursing";'.

1379.   [func]          'rndc status' now reports tcp and recursion quota
                        states.

1378.   [func]          Improved positive feedback for 'rndc {reload|refresh}.

1377.   [func]          dns_zone_load{new}() now reports if the zone was
                        loaded, queued for loading to up to date.

1376.   [func]          New function dns_zone_logc() to log to specified
                        category.

1375.   [func]          'rndc dumpdb' now dumps the adb cache along with the
                        data cache.

1374.   [func]          dns_adb_dump() now logs the lame zones associated
                        with each server.

1373.   [bug]           Recovery from expired glue failed under certain
                        circumstances.

1372.   [bug]           named crashes with an assertion failure on exit when
                        sharing the same port for listening and querying, and
                        changing listening addresses several times. [RT #3509]

1371.   [bug]           notify-source-v6, transfer-source-v6 and
                        query-source-v6 with explicit addresses and using the
                        same ports as named was listening on could interfere
                        with named's ability to answer queries sent to those
                        addresses.

1370.   [bug]           dig '+[no]recurse' was incorrectly documented.

1369.   [bug]           Adding an NS record as the lexicographically last
                        record in a secure zone didn't work.

1368.   [func]          remove support for bitstring labels.

1367.   [func]          Use response times to select forwarders.

1366.   [contrib]       queryperf usage was incomplete.  Add '-h' for help.

1365.   [func]          "localhost" and "localnets" acls now include IPv6
                        addresses / prefixes.

1364.   [func]          Log file name when unable to open memory statistics
                        and dump database files. [RT #3437]

1363.   [func]          Listen-on-v6 now supports specific addresses.
```

(continues on next page)

(continued from previous page)

```
1362.   [bug]           remove IFF_RUNNING test when scanning interfaces.

1361.   [func]          log the reason for rejecting a server when resolving
                        queries.

1360.   [bug]           --enable-libbind would fail when not built in the
                        source tree for certain OS's.

1359.   [security]      Support patches OpenSSL libraries.
                        http://www.cert.org/advisories/CA-2002-23.html

1358.   [bug]           It was possible to trigger a INSIST when debugging
                        large dynamic updates. [RT #3390]

1357.   [bug]           nsupdate was extremely wasteful of memory.

1356.   [tuning]        Reduce the number of events / quantum for zone tasks.

1355.   [bug]           Fix DNSSEC wildcard proof for CNAME/DNAME.

1354.   [doc]           lwres man pages had illegal nroff.

1353.   [contrib]       sdb/ldap to version 0.9.

1352.   [bug]           dig, host, nslookup when falling back to TCP use the
                        current search entry (if any). [RT #3374]

1351.   [bug]           lwres_getipnodebyname() returned the wrong name
                        when given a IPv4 literal, af=AF_INET6 and AI_MAPPED
                        was set.

1350.   [bug]           dns_name_fromtext() failed to handle too many labels
                        gracefully.

1349.   [security]      Minimum OpenSSL version now 0.9.6e (was 0.9.5a).
                        http://www.cert.org/advisories/CA-2002-23.html

1348.   [port]          win32: Rewrote code to use I/O Completion Ports
                        in socket.c and eliminating a host of socket
                        errors. Performance is enhanced.

1347.   [placeholder]

1346.   [placeholder]

1345.   [port]          Use a explicit -Wformat with gcc.  Not all versions
                        include it in -Wall.

1344.   [func]          Log if the serial number on the master has gone
                        backwards.
                        If you have multiple machines specified in the masters
```

(continues on next page)

```
                        clause you may want to set 'multi-master yes;' to
                        suppress this warning.

1343.   [func]          Log successful notifies received (info).  Adjust log
                        level for failed notifies to notice.

1342.   [func]          Log remote address with TCP dispatch failures.

1341.   [func]          Allow a rate limiter to be stalled.

1340.   [bug]           Delay and spread out the startup refresh load.

1339.   [func]          dig, host and nslookup now use IP6.ARPA for nibble
                        lookups.  Bit string lookups are no longer attempted.

1338.   [placeholder]

1337.   [placeholder]

1336.   [func]          Nibble lookups under IP6.ARPA are now supported by
                        dns_byaddr_create().  dns_byaddr_createptrname() is
                        deprecated, use dns_byaddr_createptrname2() instead.

1335.   [bug]           When performing a nonexistence proof, the validator
                        should discard parent NXTs from higher in the DNS.

1334.   [bug]           When signing/verifying rdatasets, duplicate rdatas
                        need to be suppressed.

1333.   [contrib]       queryperf now reports a summary of returned
                        rcodes (-c), rcodes are printed in mnemonic form (-v).

1332.   [func]          Report the current serial with periodic commits when
                        rolling forward the journal.

1331.   [func]          Generate DNSSEC wildcard proofs.

1330.   [bug]           When processing events (non-threaded) only allow
                        the task one chance to use to use its quantum.

1329.   [func]          named-checkzone will now check if nameservers that
                        appear to be IP addresses.  Available modes "fail",
                        "warn" (default) and "ignore" the results of the
                        check.

1328.   [bug]           The validator could incorrectly verify an invalid
                        negative proof.

1327.   [bug]           The validator would incorrectly mark data as insecure
                        when seeing a bogus signature before a correct
                        signature.

1326.   [bug]           DNAME/CNAME signatures were not being cached when
```

(continues on next page)

(continued from previous page)

```
                        validation was not being performed. [RT #3284]

1325.   [bug]           If the tcpquota was exhausted it was possible to
                        to trigger a INSIST() failure.

1324.   [port]          darwin: ifconfig.sh now supports darwin.

1323.   [port]          linux: Slackware 4.0 needs <asm/unistd.h>. [RT #3205]

1322.   [bug]           dnssec-signzone usage message was misleading.

1321.   [bug]           If the last RRset in a zone is glue, dnssec-signzone
                        would incorrectly duplicate its output and sign it.

1320.   [doc]           query-source-v6 was missing from options section.
                        [RT #3218]

1319.   [func]          libbind: log attempts to exploit #1318.

1318.   [bug]           libbind: Remote buffer overrun.

1317.   [port]          libbind: TrueUNIX 5.1 does not like __align as a
                        element name.

1316.   [bug]           libbind: gethostans() could get out of sync parsing
                        the response if there was a very long CNAME chain.

1315.   [bug]           Options should apply to the internal _bind view.

1314.   [port]          Handle ECONNRESET from sendmsg() [unix].

1313.   [func]          Query log now says if the query was signed (S) or
                        if EDNS was used (E).

1312.   [func]          Log TSIG key used w/ outgoing zone transfers.

1311.   [bug]           lwres_getrrsetbyname leaked memory.  [RT #3159]

1310.   [bug]           'rndc stop' failed to cause zones to be flushed
                        sometimes. [RT #3157]

1309.   [func]          Log that a zone transfer was covered by a TSIG.

1308.   [func]          DS (delegation signer) support.

1307.   [bug]           nsupdate: allow white space base64 key data.

1306.   [bug]           Badly encoded LOC record when the size, horizontal
                        precision or vertical precision was 0.1m.

1305.   [bug]           Document that internal zones are included in the
                        rndc status results.
```

(continues on next page)

(continued from previous page)

```
1304.   [func]          New function: dns_zone_name().

1303.   [func]          Option 'flush-zones-on-shutdown <boolean>;'.

1302.   [func]          Extended rndc dumpdb to support dumping of zones and
                       view selection: 'dumpdb [-all|-zones|-cache] [view]'.

1301.   [func]          New category 'update-security'.

1300.   [port]          Compaq Trucluster support.

1299.   [bug]           Set AI_ADDRCONFIG when looking up addresses
                       via getaddrinfo() (affects dig, host, nslookup, rndc
                       and nsupdate).

1298.   [bug]           The CINCLUDES macro in lib/dns/sec/dst/Makefile
                       could be left with a trailing "\" after configure
                       has been run.

1297.   [port]          linux: make handling EINVAL from socket() no longer
                       conditional on #ifdef LINUX.

1296.   [bug]           isc_log_closefilelogs() needed to lock the log
                       context.

1295.   [bug]           isc_log_setdebuglevel() needed to lock the log
                       context.

1294.   [func]          libbind: no longer attempts bit string labels for
                       IPv6 reverse resolution.  Try IP6.ARPA then IP6.INT
                       for nibble style resolution.

1293.   [func]          Entropy can now be retrieved from EGDs. [RT #2438]

1292.   [func]          Enable IPv6 support when using ioctl style interface
                       scanning and OS supports SIOCGLIFADDR using struct
                       if_laddrreq.

1291.   [func]          Enable IPv6 support when using sysctl style interface
                       scanning.

1290.   [func]          "dig axfr" now reports the number of messages
                       as well as the number of records.

1289.   [port]          See if -ldl is required for OpenSSL? [RT #2672]

1288.   [bug]           Adjusted REQUIRE's in lib/dns/name.c to better
                       reflect written requirements.

1287.   [bug]           REQUIRE that DNS_DBADD_MERGE only be set when adding
                       a rdataset to a zone db in the rbtdb implementation of
```

(continues on next page)

(continued from previous page)

```
                        addrdataset.

1286.   [bug]           dns_name_downcase() enforce requirement that
                        target != NULL or name->buffer != NULL.

1285.   [func]          lwres: probe the system to see what address families
                        are currently in use.

1284.   [bug]           The RTT estimate on unused servers was not aged.
                        [RT #2569]

1283.   [func]          Use "dataready" accept filter if available.

1282.   [port]          libbind: hpux 11.11 interface scanning.

1281.   [func]          Log zone when unable to get private keys to update
                        zone.  Log zone when NXT records are missing from
                        secure zone.

1280.   [bug]           libbind: escape '(' and ')' when converting to
                        presentation form.

1279.   [port]          Darwin uses (unsigned long) for size_t. [RT #2590]

1278.   [func]          dig: now supports +[no]cl +[no]ttlid.

1277.   [func]          You can now create your own customized printing
                        styles: dns_master_stylecreate() and
                        dns_master_styledestroy().

1276.   [bug]           libbind: const pointer conflicts in res_debug.c.

1275.   [port]          libbind: hpux: treat all hpux systems as BIG_ENDIAN.

1274.   [bug]           Memory leak in lwres_gnbarequest_parse().

1273.   [port]          libbind: solaris: 64 bit binary compatibility.

1272.   [contrib]       Berkeley DB 4.0 sdb implementation from
                        Nuno Miguel Rodrigues <nmr@co.sapo.pt>.

1271.   [bug]           "recursion available: {denied,approved}" was too
                        confusing.

1270.   [bug]           Check that system inet_pton() and inet_ntop() support
                        AF_INET6.

1269.   [port]          Openserver: ifconfig.sh support.

1268.   [port]          Openserver: the value FD_SETSIZE depends on whether
                        <sys/param.h> is included or not.  Be consistent.
```

(continues on next page)

(continued from previous page)

```
1267.   [func]          isc_file_openunique() now creates file using mode
                        0666 rather than 0600.

1266.   [bug]           ISC_LINK_INIT, ISC_LINK_UNLINK, ISC_LIST_DEQUEUE,
                        __ISC_LINK_UNLINKUNSAFE and __ISC_LIST_DEQUEUEUNSAFE
                        are not C++ compatible, use *_TYPE versions instead.

1265.   [bug]           libbind: LINK_INIT and UNLINK were not compatible with
                        C++, use LINK_INIT_TYPE and UNLINK_TYPE instead.

1264.   [placeholder]

1263.   [bug]           Reference after free error if dns_dispatchmgr_create()
                        failed.

1262.   [bug]           ns_server_destroy() failed to set *serverp to NULL.

1261.   [func]          libbind: ns_sign2() and ns_sign_tcp() now provide
                        support for compressed TSIG owner names.

1260.   [func]          libbind: res_update can now update IPv6 servers,
                        new function res_findzonecut2().

1259.   [bug]           libbind: get_salen() IPv6 support was broken for OSs
                        w/o sa_len.

1258.   [bug]           libbind: res_nametotype() and res_nametoclass() were
                        broken.

1257.   [bug]           Failure to write pid-file should not be fatal on
                        reload. [RT #2861]

1256.   [contrib]       'queryperf' now has EDNS (-e) + DNSSEC DO (-D) support.

1255.   [bug]           When verifying that an NXT proves nonexistence, check
                        the rcode of the message and only do the matching NXT
                        check.  That is, for NXDOMAIN responses, check that
                        the name is in the range between the NXT owner and
                        next name, and for NOERROR NODATA responses, check
                        that the type is not present in the NXT bitmap.

1254.   [func]          preferred-glue option from BIND 8.3.

1253.   [bug]           The dnssec system test failed to remove the correct
                        files.

1252.   [bug]           Dig, host and nslookup were not checking the address
                        the answer was coming from against the address it was
                        sent to. [RT #2692]

1251.   [port]          win32: a make file contained absolute version specific
                        references.
```

(continues on next page)

(continued from previous page)

```
1250.   [func]          Nsupdate will report the address the update was
                        sent to.
```

```
1249.   [bug]           Missing masters clause was not handled gracefully.
                        [RT #2703]

1248.   [bug]           DESTDIR was not being propagated between makes.

1247.   [bug]           Don't reset the interface index for link/site local
                        addresses. [RT #2576]

1246.   [func]          New functions isc_sockaddr_issitelocal(),
                        isc_sockaddr_islinklocal(), isc_netaddr_issitelocal()
                        and isc_netaddr_islinklocal().

1245.   [bug]           Treat ENOBUFS, ENOMEM and ENFILE as soft errors for
                        accept().

1244.   [bug]           Receiving a TCP message from a blackhole address would
                        prevent further messages being received over that
                        interface.

1243.   [bug]           It was possible to trigger a REQUIRE() in
                        dns_message_findtype(). [RT #2659]

1242.   [bug]           named-checkzone failed if a journal existed. [RT #2657]

1241.   [bug]           Drop received UDP messages with a zero source port
                        as these are invariably forged. [RT #2621]

1240.   [bug]           It was possible to leak zone references by
                        specifying an incorrect zone to rndc.

1239.   [bug]           Under certain circumstances named could continue to
                        use a name after it had been freed triggering
                        INSIST() failures.  [RT #2614]

1238.   [bug]           It is possible to lockup the server when shutting down
                        if notifies were being processed. [RT #2591]

1237.   [bug]           nslookup: "set q=type" failed.

1236.   [bug]           dns_rdata{class,type}_fromtext() didn't handle non
                        NULL terminated text regions. [RT #2588]

1235.   [func]          Report 'out of memory' errors from openssl.

1234.   [bug]           contrib/sdb: 'zonetodb' failed to call
                        dns_result_register().  DNS_R_SEENINCLUDE should not
                        be fatal.
```

(continues on next page)

(continued from previous page)

```
1233.   [bug]           The flags field of a KEY record can be expressed in
                        hex as well as decimal.

1232.   [bug]           unix/errno2result() didn't handle EADDRNOTAVAIL.

1231.   [port]          HPUX 11.11 recvmsg() can return spurious EADDRNOTAVAIL.

1230.   [bug]           isccc_cc_isreply() and isccc_cc_isack() were broken.

1229.   [bug]           named would crash if it received a TSIG signed
                        query as part of an AXFR response. [RT #2570]

1228.   [bug]           'make install' did not depend on 'make all'. [RT #2559]

1227.   [bug]           dns_lex_getmastertoken() now returns ISC_R_BADNUMBER
                        if a number was expected and some other token was
                        found. [RT #2532]

1226.   [func]          Use EDNS for zone refresh queries. [RT #2551]

1225.   [func]          dns_message_setopt() no longer requires that
                        dns_message_renderbegin() to have been called.

1224.   [bug]           'rrset-order' and 'sortlist' should be additive
                        not exclusive.

1223.   [func]          'rrset-order' partially works 'cyclic' and 'random'
                        are supported.

1222.   [bug]           Specifying 'port *' did not always result in a system
                        selected (non-reserved) port being used. [RT #2537]

1221.   [bug]           Zone types 'master', 'slave' and 'stub' were not being
                        compared case insensitively. [RT #2542]

1220.   [func]          Support for APL rdata type.

1219.   [func]          Named now reports the TSIG extended error code when
                        signature verification fails. [RT #1651]

1218.   [bug]           Named incorrectly returned SERVFAIL rather than
                        NOTAUTH when there was a TSIG BADTIME error. [RT #2519]

1217.   [func]          Report locations of previous key definition when a
                        duplicate is detected.

1216.   [bug]           Multiple server clauses for the same server were not
                        reported.  [RT #2514]

1215.   [port]          solaris: add support to ifconfig.sh for x86 2.5.1

1214.   [bug]           Win32: isc_file_renameunique() could leave zero length
```

(continues on next page)

(continued from previous page)

```
                        files behind.

1213.   [func]          Report view associated with client if it is not a
                        standard view (_default or _bind).

1212.   [port]          libbind: 64k answer buffers were causing stack space
                        to be exceeded for certain OS.  Use heap space instead.

1211.   [bug]           dns_name_fromtext() incorrectly handled certain
                        valid octal bitlabels. [RT #2483]

1210.   [bug]           libbind: getnameinfo() failed to lookup IPv4 mapped /
                        compatible addresses. [RT #2461]

1209.   [bug]           Dig, host, nslookup were not checking the message ids
                        on the responses. [RT #2454]

1208.   [bug]           dns_master_load*() failed to log a error message if
                        an error was detected when parsing the owner name of
                        a record.  [RT #2448]

1207.   [bug]           libbind: getaddrinfo() could call freeaddrinfo() with
                        an invalid pointer.

1206.   [bug]           SERVFAIL and NOTIMP responses to an EDNS query should
                        trigger a non-EDNS retry.

1205.   [bug]           OPT, TSIG and TKEY cannot be used to set the "class"
                        of the message. [RT #2449]

1204.   [bug]           libbind: res_nupdate() failed to update the name
                        server addresses before sending the update.

1203.   [func]          Report locations of previous acl and zone definitions
                        when a duplicate is detected.

1202.   [func]          New functions: cfg_obj_line() and cfg_obj_file().

1201.   [bug]           Require that if 'callbacks' is passed to
                        dns_rdata_fromtext(), callbacks->error and
                        callbacks->warn are initialized.

1200.   [bug]           Log 'errno' that we are unable to convert to
                        isc_result_t. [RT #2404]

1199.   [doc]           ARM reference to RFC 2157 should have been RFC 1918.
                        [RT #2436]

1198.   [bug]           OPT printing style was not consistent with the way the
                        header fields are printed.  The DO bit was not reported
                        if set.  Report if any of the MBZ bits are set.
```

(continues on next page)

(continued from previous page)

```
1197.   [bug]           Attempts to define the same acl multiple times were not
                        detected.

1196.   [contrib]       update mdnkit to 2.2.3.

1195.   [bug]           Attempts to redefine builtin acls should be caught.
                        [RT #2403]

1194.   [bug]           Not all duplicate zone definitions were being detected
                        at the named.conf checking stage. [RT #2431]

1193.   [bug]           dig +besteffort parsing didn't handle packet
                        truncation.  dns_message_parse() has new flag
                        DNS_MESSAGE_IGNORETRUNCATION.

1192.   [bug]           The seconds fields in LOC records were restricted
                        to three decimal places.  More decimal places should
                        be allowed but warned about.

1191.   [bug]           A dynamic update removing the last non-apex name in
                        a secure zone would fail. [RT #2399]

1190.   [func]          Add the "rndc freeze" and "rndc unfreeze" commands.
                        [RT #2394]

1189.   [bug]           On some systems, malloc(0) returns NULL, which
                        could cause the caller to report an out of memory
                        error. [RT #2398]

1188.   [bug]           Dynamic updates of a signed zone would fail if
                        some of the zone private keys were unavailable.

1187.   [bug]           named was incorrectly returning DNSSEC records
                        in negative responses when the DO bit was not set.

1186.   [bug]           isc_hex_tobuffer(,,length = 0) failed to unget the
                        EOL token when reading to end of line.

1185.   [bug]           libbind: don't assume statp->_u._ext.ext is valid
                        unless RES_INIT is set when calling res_*init().

1184.   [bug]           libbind: call res_ndestroy() if RES_INIT is set
                        when res_*init() is called.

1183.   [bug]           Handle ENOSR error when writing to the internal
                        control pipe. [RT #2395]

1182.   [bug]           The server could throw an assertion failure when
                        constructing a negative response packet.

1181.   [func]          Add the "key-directory" configuration statement,
                        which allows the server to look for online signing
```

(continues on next page)

(continued from previous page)

```
                        keys in alternate directories.

1180.   [func]          dnssec-keygen should always generate keys with
                        protocol 3 (DNSSEC), since it's less confusing
                        that way.

1179.   [func]          Add SIG(0) support to nsupdate.

1178.   [bug]           Follow and cache (if appropriate) A6 and other
                        data chains to completion in the additional section.

1177.   [func]          Report view when loading zones if it is not a
                        standard view (_default or _bind). [RT #2270]

1176.   [doc]           Document that allow-v6-synthesis is only performed
                        for clients that are supplied recursive service.
                        [RT #2260]

1175.   [bug]           named-checkzone and named-checkconf failed to call
                        dns_result_register() at startup which could
                        result in runtime exceptions when printing
                        "out of memory" errors. [RT #2335]

1174.   [bug]           Win32: add WSAECONNRESET to the expected errors
                        from connect(). [RT #2308]

1173.   [bug]           Potential memory leaks in isc_log_create() and
                        isc_log_settag(). [RT #2336]

1172.   [doc]           Add CERT, GPOS, KX, NAPTR, NSAP, PX and TXT to
                        table of RR types in ARM.

1171.   [func]          Added function isc_region_compare(), updated files in
                        lib/dns to use this function instead of local one.

1170.   [bug]           Don't attempt to print the token when a I/O error
                        occurs when parsing named.conf. [RT #2275]

1169.   [func]          Identify recursive queries in the query log.

1168.   [bug]           Empty also-notify clauses were not handled. [RT #2309]

1167.   [contrib]       nslint-2.1a3 (from author).

1166.   [bug]           "Not Implemented" should be reported as NOTIMP,
                        not NOTIMPL. [RT #2281]

1165.   [bug]           We were rejecting notify-source{-v6} in zone clauses.

1164.   [bug]           Empty masters clauses in slave / stub zones were not
                        handled gracefully. [RT #2262]
```

(continues on next page)

(continued from previous page)

```
1163.   [func]          isc_time_formattimestamp() now includes the year.

1162.   [bug]           The allow-notify option was not accepted in slave
                       zone statements.

1161.   [bug]           named-checkzone looped on unbalanced brackets.
                       [RT #2248]

1160.   [bug]           Generating Diffie-Hellman keys longer than 1024
                       bits could fail. [RT #2241]

1159.   [bug]           MD and MF are not permitted to be loaded by RFC1123.

1158.   [func]          Report the client's address when logging notify
                       messages.

1157.   [func]          match-clients and match-destinations now accept
                       keys. [RT #2045]

1156.   [port]          The configure test for strsep() incorrectly
                       succeeded on certain patched versions of
                       AIX 4.3.3. [RT #2190]

1155.   [func]          Recover from master files being removed from under
                       us.

1154.   [bug]           Don't attempt to obtain the netmask of a interface
                       if there is no address configured. [RT #2176]

1153.   [func]          'rndc {stop|halt} -p' now reports the process id
                       of the instance of named being shutdown.

1152.   [bug]           libbind: read buffer overflows.

1151.   [bug]           nslookup failed to check that the arguments to
                       the port, timeout, and retry options were
                       valid integers and in range. [RT #2099]

1150.   [bug]           named incorrectly accepted TTL values
                       containing plus or minus signs, such as
                       1d+1h-1s.

1149.   [func]          New function isc_parse_uint32().

1148.   [func]          'rndc-confgen -a' now provides positive feedback.

1147.   [func]          Set IPV6_V6ONLY on IPv6 sockets if supported by
                       the OS.  listen-on-v6 { any; }; should no longer
                       result in IPv4 queries be accepted.  Similarly
                       control { inet :: ... }; should no longer result
                       in IPv4 connections being accepted.  This can be
                       overridden at compile time by defining
```

(continues on next page)

(continued from previous page)

```
                        ISC_ALLOW_MAPPED=1.

1146.   [func]          Allow IPV6_IPV6ONLY to be set/cleared on a socket if
                        supported by the OS by a new function
                        isc_socket_ipv6only().

1145.   [func]          "host" no longer reports a NOERROR/NODATA response
                        by printing nothing. [RT #2065]

1144.   [bug]           rndc-confgen would crash if both the -a and -t
                        options were specified. [RT #2159]

1143.   [bug]           When a trusted-keys statement was present and named
                        was built without crypto support, it would leak memory.

1142.   [bug]           dnssec-signzone would fail to delete temporary files
                        in some failure cases. [RT #2144]

1141.   [bug]           When named rejected a control message, it would
                        leak a file descriptor and memory.  It would also
                        fail to respond, causing rndc to hang.
                        [RT #2139, #2164]

1140.   [bug]           rndc-confgen did not accept IPv6 addresses as arguments
                        to the -s option. [RT #2138]

1139.   [func]          It is now possible to flush a given name from the
                        cache(s) via 'rndc flushname name [view]'. [RT #2051]

1138.   [func]          It is now possible to flush a given name from the
                        cache by calling the new function
                        dns_cache_flushname().

1137.   [func]          It is now possible to flush a given name from the
                        ADB by calling the new function dns_adb_flushname().

1136.   [bug]           CNAME records synthesized from DNAMEs did not
                        have a TTL of zero as required by RFC2672.
                        [RT #2129]

1135.   [func]          You can now override the default syslog() facility for
                        named/lwresd at compile time. [RT #1982]

1134.   [bug]           Multi-threaded servers could deadlock in ferror()
                        when reloading zone files. [RT #1951, #1998]

1133.   [bug]           IN6_IS_ADDR_LOOPBACK was not portably defined on
                        platforms without IN6_IS_ADDR_LOOPBACK. [RT #2106]

1132.   [func]          Improve UPDATE prerequisite failure diagnostic messages.

1131.   [bug]           The match-destinations view option did not work with
```

(continues on next page)

(continued from previous page)

```
                        IPv6 destinations. [RT #2073, #2074]

1130.   [bug]           Log messages reporting an out-of-range serial number
                        did not include the out-of-range number but the
                        following token. [RT #2076]

1129.   [bug]           Multi-threaded servers could crash under heavy
                        resolution load due to a race condition. [RT #2018]

1128.   [func]          sdb drivers can now provide RR data in either text
                        or wire format, the latter using the new functions
                        dns_sdb_putrdata() and dns_sdb_putnamedrdata().

1127.   [func]          rndc: If the server to contact has multiple addresses,
                        try all of them.

1126.   [bug]           The server could access a freed event if shut
                        down while a client start event was pending
                        delivery. [RT #2061]

1125.   [bug]           rndc: -k option was missing from usage message.
                        [RT #2057]

1124.   [doc]           dig: +[no]dnssec, +[no]besteffort and +[no]fail
                        are now documented. [RT #2052]

1123.   [bug]           dig +[no]fail did not match description. [RT #2052]

1122.   [tuning]        Resolution timeout reduced from 90 to 30 seconds.
                        [RT #2046]

1121.   [bug]           The server could attempt to access a NULL zone
                        table if shut down while resolving.
                        [RT #1587, #2054]

1120.   [bug]           Errors in options were not fatal. [RT #2002]

1119.   [func]          Added support in Win32 for NTFS file/directory ACL's
                        for access control.

1118.   [bug]           On multi-threaded servers, a race condition
                        could cause an assertion failure in resolver.c
                        during resolver shutdown. [RT #2029]

1117.   [port]          The configure check for in6addr_loopback incorrectly
                        succeeded on AIX 4.3 when compiling with -O2
                        because the test code was optimized away.
                        [RT #2016]

1116.   [bug]           Setting transfers in a server clause, transfers-in,
                        or transfers-per-ns to a value greater than
                        2147483647 disabled transfers. [RT #2002]
```

(continues on next page)

(continued from previous page)

```
1115.   [func]          Set maximum values for cleaning-interval,
                        heartbeat-interval, interface-interval,
                        max-transfer-idle-in, max-transfer-idle-out,
                        max-transfer-time-in, max-transfer-time-out,
                        statistics-interval of 28 days and
                        sig-validity-interval of 3660 days. [RT #2002]

1114.   [port]          Ignore more accept() errors. [RT #2021]

1113.   [bug]           The allow-update-forwarding option was ignored
                        when specified in a view. [RT #2014]

1112.   [placeholder]

1111.   [bug]           Multi-threaded servers could deadlock processing
                        recursive queries due to a locking hierarchy
                        violation in adb.c. [RT #2017]

1110.   [bug]           dig should only accept valid abbreviations of +options.
                        [RT #2003]

1109.   [bug]           nsupdate accepted illegal ttl values.

1108.   [bug]           On Win32, rndc was hanging when named was not running
                        due to failure to select for exceptional conditions
                        in select(). [RT #1870]

1107.   [bug]           nsupdate could catch an assertion failure if an
                        invalid domain name was given as the argument to
                        the "zone" command.

1106.   [bug]           After seeing an out of range TTL, nsupdate would
                        treat all TTLs as out of range. [RT #2001]

1105.   [port]          OpenUNIX 8 enable threads by default. [RT #1970]

1104.   [bug]           Invalid arguments to the transfer-format option
                        could cause an assertion failure. [RT #1995]

1103.   [port]          OpenUNIX 8 support (ifconfig.sh). [RT #1970]

1102.   [doc]           Note that query logging is enabled by directing the
                        queries category to a channel.

1101.   [bug]           Array bounds read error in lwres_gai_strerror.

1100.   [bug]           libbind: DNSSEC key ids were computed incorrectly.

1099.   [cleanup]       libbind: defining REPORT_ERRORS in lib/bind/dst caused
                        compile time errors.
```

(continues on next page)

```
1098.   [bug]           libbind: HMAC-MD5 key files are now mode 0600.

1097.   [func]          libbind: RES_PRF_TRUNC for dig.

1096.   [func]          libbind: "DNSSEC OK" (DO) support.

1095.   [func]          libbind: resolver option: no-tld-query.  disables
                        trying unqualified as a tld.  no_tld_query is also
                        supported for FreeBSD compatibility.

1094.   [func]          libbind: add support gcc's format string checking.

1093.   [doc]           libbind: miscellaneous nroff fixes.

1092.   [bug]           libbind: get*by*() failed to check if res_init() had
                        been called.

1091.   [bug]           libbind: misplaced va_end().

1090.   [bug]           libbind: dns_ho.c:add_hostent() was not returning
                        the amount of memory consumed resulting in garbage
                        address being returned.  Alignment calculations were
                        wasting space.  We weren't suppressing duplicate
                        addresses.

1089.   [func]          libbind: inet_{cidr,net}_{pton,ntop}() now have IPv6
                        support.

1088.   [port]          libbind: MPE/iX C.70 (incomplete)

1087.   [bug]           libbind: struct __res_state too large on 64 bit arch.

1086.   [port]          libbind: sunos: old sprintf.

1085.   [port]          libbind: solaris: sys_nerr and sys_errlist do not
                        exist when compiling in 64 bit mode.

1084.   [cleanup]       libbind: gai_strerror() rewritten.

1083.   [bug]           The default control channel listened on the
                        wildcard address, not the loopback as documented.
                        [RT #1975]

1082.   [bug]           The -g option to named incorrectly caused logging
                        to be sent to syslog in addition to stderr.
                        [RT #1974]

1081.   [bug]           Multicast queries were incorrectly identified
                        based on the source address, not the destination
                        address.

1080.   [bug]           BIND 8 compatibility: accept bare IP prefixes
                        as the second element of a two-element top level
```

(continues on next page)

(continued from previous page)

```
                        sort list statement. [RT #1964]

1079.   [bug]           BIND 8 compatibility: accept bare elements at top
                        level of sort list treating them as if they were
                        a single element list. [RT #1963]

1078.   [bug]           We failed to correct bad tv_usec values in one case.
                        [RT #1966]

1077.   [func]          Do not accept further recursive clients when
                        the total number of recursive lookups being
                        processed exceeds max-recursive-clients, even
                        if some of the lookups are internally generated.
                        [RT #1915, #1938]

1076.   [bug]           A badly defined global key could trigger an assertion
                        on load/reload if views were used. [RT #1947]

1075.   [bug]           Out-of-range network prefix lengths were not
                        reported. [RT #1954]

1074.   [bug]           Running out of memory in dump_rdataset() could
                        cause an assertion failure. [RT #1946]

1073.   [bug]           The ADB cache cleaning should also be space driven.
                        [RT #1915, #1938]

1072.   [bug]           The TCP client quota could be exceeded when
                        recursion occurred. [RT #1937]

1071.   [bug]           Sockets listening for TCP DNS connections
                        specified an excessive listen backlog. [RT #1937]

1070.   [bug]           Copy DNSSEC OK (DO) to response as specified by
                        draft-ietf-dnsext-dnssec-okbit-03.txt.

1069.   [placeholder]

1068.   [bug]           errno could be overwritten by catgets(). [RT #1921]

1067.   [func]          Allow quotas to be soft, isc_quota_soft().

1066.   [bug]           Provide a thread safe wrapper for strerror().
                        [RT #1689]

1065.   [func]          Runtime support to select new / old style interface
                        scanning using ioctls.

1064.   [bug]           Do not shut down active network interfaces if we
                        are unable to scan the interface list. [RT #1921]

1063.   [bug]           libbind: "make install" was failing on IRIX.
```

(continues on next page)

(continued from previous page)

```
                        [RT #1919]

1062.   [bug]           If the control channel listener socket was shut
                        down before server exit, the listener object could
                        be freed twice. [RT #1916]

1061.   [bug]           If periodic cache cleaning happened to start
                        while cleaning due to reaching the configured
                        maximum cache size was in progress, the server
                        could catch an assertion failure. [RT #1912]

1060.   [func]          Move refresh, stub and notify UDP retry processing
                        into dns_request.

1059.   [func]          dns_request now support will now retry UDP queries,
                        dns_request_createvia2() and dns_request_createraw2().

1058.   [func]          Limited lifetime ticker timers are now available,
                        isc_timertype_limited.

1057.   [bug]           Reloading the server after adding a "file" clause
                        to a zone statement could cause the server to
                        crash due to a typo in change 1016.

1056.   [bug]           Rndc could catch an assertion failure on SIGINT due
                        to an uninitialized variable. [RT #1908]

1055.   [func]          Version and hostname queries can now be disabled
                        using "version none;" and "hostname none;",
                        respectively.

1054.   [bug]           On Win32, cfg_categories and cfg_modules need to be
                        exported from the libisccfg DLL.

1053.   [bug]           Dig did not increase its timeout when receiving
                        AXFRs unless the +time option was used. [RT #1904]

1052.   [bug]           Journals were not being created in binary mode
                        resulting in "journal format not recognized" error
                        under Win32. [RT #1889]

1051.   [bug]           Do not ignore a network interface completely just
                        because it has a noncontiguous netmask.  Instead,
                        omit it from the localnets ACL and issue a warning.
                        [RT #1891]

1050.   [bug]           Log messages reporting malformed IP addresses in
                        address lists such as that of the forwarders option
                        failed to include the correct error code, file
                        name, and line number. [RT #1890]

1049.   [func]          "pid-file none;" will disable writing a pid file.
```

(continues on next page)

(continued from previous page)

```
                        [RT #1848]

1048.   [bug]           Servers built with -DISC_MEM_USE_INTERNAL_MALLOC=1
                        didn't work.

1047.   [bug]           named was incorrectly refusing all requests signed
                        with a TSIG key derived from an unsigned TKEY
                        negotiation with a NOERROR response. [RT #1886]

1046.   [bug]           The help message for the --with-openssl configure
                        option was inaccurate. [RT #1880]

1045.   [bug]           It was possible to skip saving glue for a nameserver
                        for a stub zone.

1044.   [bug]           Specifying allow-transfer, notify-source, or
                        notify-source-v6 in a stub zone was not treated
                        as an error.

1043.   [bug]           Specifying a transfer-source or transfer-source-v6
                        option in the zone statement for a master zone was
                        not treated as an error. [RT #1876]

1042.   [bug]           The "config" logging category did not work properly.
                        [RT #1873]

1041.   [bug]           Dig/host/nslookup could catch an assertion failure
                        on SIGINT due to an uninitialized variable. [RT #1867]

1040.   [bug]           Multiple listen-on-v6 options with different ports
                        were not accepted. [RT #1875]

1039.   [bug]           Negative responses with CNAMEs in the answer section
                        were cached incorrectly. [RT #1862]

1038.   [bug]           In servers configured with a tkey-domain option,
                        TKEY queries with an owner name other than the root
                        could cause an assertion failure. [RT #1866, #1869]

1037.   [bug]           Negative responses whose authority section contain
                        SOA or NS records whose owner names are not equal
                        equal to or parents of the query name should be
                        rejected. [RT #1862]

1036.   [func]          Silently drop requests received via multicast as
                        long as there is no final multicast DNS standard.

1035.   [bug]           If we respond to multicast queries (which we
                        currently do not), respond from a unicast address
                        as specified in RFC 1123. [RT #137]

1034.   [bug]           Ignore the RD bit on multicast queries as specified
```

(continues on next page)

(continued from previous page)

```
                        in RFC 1123. [RT #137]

1033.   [bug]           Always respond to requests with an unsupported opcode
                        with NOTIMP, even if we don't have a matching view
                        or cannot determine the class.

1032.   [func]          hostname.bind/txt/chaos now returns the name of
                        the machine hosting the nameserver.  This is useful
                        in diagnosing problems with anycast servers.

1031.   [bug]           libbind.a: isc__gettimeofday() infinite recursion.
                        [RT #1858]

1030.   [bug]           On systems with no resolv.conf file, nsupdate
                        exited with an error rather than defaulting
                        to using the loopback address. [RT #1836]

1029.   [bug]           Some named.conf errors did not cause the loading
                        of the configuration file to return a failure
                        status even though they were logged. [RT #1847]

1028.   [bug]           On Win32, dig/host/nslookup looked for resolv.conf
                        in the wrong directory. [RT #1833]

1027.   [bug]           RRs having the reserved type 0 should be rejected.
                        [RT #1471]

1026.   [placeholder]

1025.   [bug]           Don't use multicast addresses to resolve iterative
                        queries. [RT #101]

1024.   [port]          Compilation failed on HP-UX 11.11 due to
                        incompatible use of the SIOCGLIFCONF macro
                        name. [RT #1831]

1023.   [func]          Accept hints without TTLs.

1022.   [bug]           Don't report empty root hints as "extra data".
                        [RT #1802]

1021.   [bug]           On Win32, log message timestamps were one month
                        later than they should have been, and the server
                        would exhibit unspecified behavior in December.

1020.   [bug]           IXFR log messages did not distinguish between
                        true IXFRs, AXFR-style IXFRs, and mere version
                        polls. [RT #1811]

1019.   [bug]           The value of the lame-ttl option was limited to 18000
                        seconds, not 1800 seconds as documented. [RT #1803]
```

(continues on next page)

(continued from previous page)

```
1018.   [bug]           The default log channel was not always initialized
                        correctly. [RT #1813]

1017.   [bug]           When specifying TSIG keys to dig and nsupdate using
                        the -k option, they must be HMAC-MD5 keys. [RT #1810]

1016.   [bug]           Slave zones with no backup file were re-transferred
                        on every server reload.

1015.   [bug]           Log channels that had a "versions" option but no
                        "size" option failed to create numbered log
                        files. [RT #1783]

1014.   [bug]           Some queries would cause statistics counters to
                        increment more than once or not at all. [RT #1321]

1013.   [bug]           It was possible to cancel a query twice when marking
                        a server as bogus or by having a blackhole acl.
                        [RT #1776]

1012.   [bug]           The -p option to named did not behave as documented.

1011.   [cleanup]       Removed isc_dir_current().

1010.   [bug]           The server could attempt to execute a command channel
                        command after initiating server shutdown, causing
                        an assertion failure. [RT #1766]

1009.   [port]          OpenUNIX 8 support. [RT #1728]

1008.   [port]          libtool.m4, ltmain.sh from libtool-1.4.2.

1007.   [port]          config.guess, config.sub from autoconf-2.52.

1006.   [bug]           If a KEY RR was found missing during DNSSEC validation,
                        an assertion failure could subsequently be triggered
                        in the resolver. [RT #1763]

1005.   [bug]           Don't copy nonzero RCODEs from request to response.
                        [RT #1765]

1004.   [port]          Deal with recvfrom() returning EHOSTDOWN. [RT #1770]

1003.   [func]          Add the +retry option to dig.

1002.   [bug]           When reporting an unknown class name in named.conf,
                        including the file name and line number. [RT #1759]

1001.   [bug]           win32 socket code doio_recv was not catching a
                        WSACONNRESET error when a client was timing out
                        the request and closing its socket. [RT #1745]
```

(continues on next page)

(continued from previous page)

```
1000.   [bug]           BIND 8 compatibility: accept "HESIOD" as an alias
                        for class "HS". [RT #1759]

 999.   [func]          "rndc retransfer zone [class [view]]" added.
                        [RT #1752]

 998.   [func]          named-checkzone now has arguments to specify the
                        chroot directory (-t) and working directory (-w).
                        [RT #1755]

 997.   [func]          Add support for RSA-SHA1 keys (RFC3110).

 996.   [func]          Issue warning if the configuration filename contains
                        the chroot path.

 995.   [bug]           dig, host, nslookup: using a raw IPv6 address as a
                        target address should be fatal on a IPv4 only system.

 994.   [func]          Treat non-authoritative responses to queries for type
                        NS as referrals even if the NS records are in the
                        answer section, because BIND 8 servers incorrectly
                        send them that way.  This is necessary for DNSSEC
                        validation of the NS records of a secure zone to
                        succeed when the parent is a BIND 8 server. [RT #1706]

 993.   [func]          dig: -v now reports the version.

 992.   [doc]           dig: ~/.digrc is now documented.

 991.   [func]          Lower UDP refresh timeout messages to level
                        debug 1.

 990.   [bug]           The rndc-confgen man page was not installed.

 989.   [bug]           Report filename if $INCLUDE fails for file related
                        errors. [RT #1736]

 988.   [bug]           'additional-from-auth no;' did not work reliably
                        in the case of queries answered from the cache.
                        [RT #1436]

 987.   [bug]           "dig -help" didn't show "+[no]stats".

 986.   [bug]           "dig +noall" failed to clear stats and command
                        printing.

 985.   [func]          Consider network interfaces to be up iff they have
                        a nonzero IP address rather than based on the
                        IFF_UP flag. [RT #1160]

 984.   [bug]           Multi-threading should be enabled by default on
                        Solaris 2.7 and newer, but it wasn't.
```

(continues on next page)

(continued from previous page)

```
983.   [func]          The server now supports generating IXFR difference
                       sequences for non-dynamic zones by comparing zone
                       versions, when enabled using the new config
                       option "ixfr-from-differences". [RT #1727]

982.   [func]          If "memstatistics-file" is set in options the memory
                       statistics will be written to it.

981.   [func]          The dnssec tools can now take multiple '-r randomfile'
                       arguments.

980.   [bug]           Incoming zone transfers restarting after an error
                       could trigger an assertion failure. [RT #1692]

979.   [func]          Incremental master file dumping.  dns_master_dumpinc(),
                       dns_master_dumptostreaminc(), dns_dumpctx_attach(),
                       dns_dumpctx_detach(), dns_dumpctx_cancel(),
                       dns_dumpctx_db() and dns_dumpctx_version().

978.   [bug]           dns_db_attachversion() had an invalid REQUIRE()
                       condition.

977.   [bug]           Improve "not at top of zone" error message.

976.   [func]          named-checkconf can now test load master zones
                       (named-checkconf -z). [RT #1468]

975.   [bug]           "max-cache-size default;" as a view option
                       caused an assertion failure.

974.   [bug]           "max-cache-size unlimited;" as a global option
                       was not accepted.

973.   [bug]           Failed to log the question name when logging:
                       "bad zone transfer request: non-authoritative zone
                       (NOTAUTH)".

972.   [bug]           The file modification time code in zone.c was using the
                       wrong epoch. [RT #1667]

971.   [placeholder]

970.   [func]          'max-journal-size' can now be used to set a target
                       size for a journal.

969.   [func]          dig now supports the undocumented dig 8 feature
                       of allowing arbitrary labels, not just dotted
                       decimal quads, with the -x option.  This can be
                       used to conveniently look up RFC2317 names as in
                       "dig -x 10.0.0.0-127". [RT #827, #1576, #1598]
```

(continues on next page)

(continued from previous page)

```
968.   [bug]           On win32, the isc_time_now() function was unnecessarily
                       calling strtime(). [RT #1671]

967.   [bug]           On win32, the link for bindevt was not including the
                       required resource file to enable the event viewer
                       to interpret the error messages in the event log,
                       [RT #1668]

966.   [placeholder]

965.   [bug]           Including data other than root server NS and A
                       records in the root hint file could cause a rbtdb
                       node reference leak. [RT #1581, #1618]

964.   [func]          Warn if data other than root server NS and A records
                       are found in the root hint file. [RT #1581, #1618]

963.   [bug]           Bad ISC_LANG_ENDDECLS. [RT #1645]

962.   [bug]           libbind: bad "#undef", don't attempt to install
                       non-existent nlist.h. [RT #1640]

961.   [bug]           Tried to use a IPV6 feature when ISC_PLATFORM_HAVEIPV6
                       was not defined. [RT #1482]

960.   [port]          liblwres failed to build on systems with support for
                       getrrsetbyname() in the OS. [RT #1592]

959.   [port]          On FreeBSD, determine the number of CPUs by calling
                       sysctlbyname(). [RT #1584]

958.   [port]          ssize_t is not available on all platforms. [RT #1607]

957.   [bug]           sys/select.h inclusion was broken on older platforms.
                       [RT #1607]

956.   [bug]           ns_g_autorndcfile changed to ns_g_keyfile
                       in named/win32/os.c due to code changes in
                       change #953. win32 .make file for rndc-confgen
                       updated to add include path for os.h header.
```

```
      --- 9.2.0rc1 released ---

955.   [bug]           When using views, the zone's class was not being
                       inherited from the view's class. [RT #1583]

954.   [bug]           When requesting AXFRs or IXFRs using dig, host, or
                       nslookup, the RD bit should not be set as zone
                       transfers are inherently non-recursive. [RT #1575]

953.   [func]          The /var/run/named.key file from change #843
                       has been replaced by /etc/rndc.key.  Both
```

(continues on next page)

(continued from previous page)

```
                        named and rndc will look for this file and use
                        it to configure a default control channel key
                        if not already configured using a different
                        method (rndc.conf / controls).  Unlike
                        named.key, rndc.key is not created automatically;
                        it must be created by manually running
                        "rndc-confgen -a".

952.    [bug]           The server required manual intervention to serve the
                        affected zones if it died between creating a journal
                        and committing the first change to it.

951.    [bug]           CFLAGS was not passed to the linker when
                        linking some of the test programs under
                        bin/tests. [RT #1555].

950.    [bug]           Explicit TTLs did not properly override $TTL
                        due to a bug in change 834. [RT #1558]

949.    [bug]           host was unable to print records larger than 512
                        bytes. [RT #1557]
```

```
      --- 9.2.0b2 released ---

948.    [port]          Integrated support for building on Windows NT /
                        Windows 2000.

947.    [bug]           dns_rdata_soa_t had a badly named element "mname" which
                        was really the RNAME field from RFC1035.  To avoid
                        confusion and silent errors that would occur it the
                        "origin" and "mname" elements were given their correct
                        names "mname" and "rname" respectively, the "mname"
                        element is renamed to "contact".

946.    [cleanup]       doc/misc/options is now machine-generated from the
                        configuration parser syntax tables, and therefore
                        more likely to be correct.

945.    [func]          Add the new view-specific options
                        "match-destinations" and "match-recursive-only".

944.    [func]          Check for expired signatures on load.

943.    [bug]           The server could crash when receiving a command
                        via rndc if the configuration file listed only
                        nonexistent keys in the controls statement. [RT #1530]

942.    [port]          libbind: GETNETBYADDR_ADDR_T was not correctly
                        defined on some platforms.

941.    [bug]           The configuration checker crashed if a slave
                        zone didn't contain a masters statement. [RT #1514]
```

(continues on next page)

(continued from previous page)

```
940.   [bug]           Double zone locking failure on error path. [RT #1510]
```

```
       --- 9.2.0b1 released ---

939.   [port]          Add the --disable-linux-caps option to configure for
                       systems that manage capabilities outside of named.
                       [RT #1503]

938.   [placeholder]

937.   [bug]           A race when shutting down a zone could trigger a
                       INSIST() failure. [RT #1034]

936.   [func]          Warn about IPv4 addresses that are not complete
                       dotted quads. [RT #1084]

935.   [bug]           inet_pton failed to reject leading zeros.

934.   [port]          Deal with systems where accept() spuriously returns
                       ECONNRESET.

933.   [bug]           configure failed doing libbind on platforms not
                       supported by BIND 8. [RT #1496]
```

```
       --- 9.2.0a3 released ---

932.   [bug]           Use INSTALL_SCRIPT, not INSTALL_PROGRAM,
                       when installing isc-config.sh.
                       [RT #198, #1466]

931.   [bug]           The controls statement only attempted to verify
                       messages using the first key in the key list.
                       (9.2.0a1/a2 only).

930.   [func]          Query performance testing tool added as
                       contrib/queryperf.

929.   [placeholder]

928.   [bug]           nsupdate would send empty update packets if the
                       send (or empty line) command was run after
                       another send but before any new updates or
                       prerequisites were specified.  It should simply
                       ignore this command.

927.   [bug]           Don't hold the zone lock for the entire dump to disk.
                       [RT #1423]

926.   [bug]           The resolver could deadlock with the ADB when
                       shutting down (multi-threaded builds only).
```

(continues on next page)

(continued from previous page)

```
                      [RT #1324]

925.  [cleanup]       Remove openssl from the distribution; require that
                      --with-openssl be specified if DNSSEC is needed.

924.  [port]          Extend support for pre-RFC2133 IPv6 implementation.
                      [RT #987]

923.  [bug]           Multiline TSIG secrets (and other multiline strings)
                      were not accepted in named.conf. [RT #1469]

922.  [func]          Added two new lwres_getrrsetbyname() result codes,
                      ERR_NONAME and ERR_NODATA.

921.  [bug]           lwres returned an incorrect error code if it received
                      a truncated message.

920.  [func]          Increase the lwres receive buffer size to 16K.
                      [RT #1451]

919.  [placeholder]

918.  [func]          In nsupdate, TSIG errors are no longer treated as
                      fatal errors.

917.  [func]          New nsupdate command 'key', allowing TSIG keys to
                      be specified in the nsupdate command stream rather
                      than the command line.

916.  [bug]           Specifying type ixfr to dig without specifying
                      a serial number failed in unexpected ways.

915.  [func]          The named-checkconf and named-checkzone programs
                      now have a '-v' option for printing their version.
                      [RT #1151]

914.  [bug]           Global 'server' statements were rejected when
                      using views, even though they were accepted
                      in 9.1. [RT #1368]

913.  [bug]           Cache cleaning was not sufficiently aggressive.
                      [RT #1441, #1444]

912.  [bug]           Attempts to set the 'additional-from-cache' or
                      'additional-from-auth' option to 'no' in a
                      server with recursion enabled will now
                      be ignored and cause a warning message.
                      [RT #1145]

911.  [placeholder]

910.  [port]          Some pre-RFC2133 IPv6 implementations do not define
```

(continues on next page)

(continued from previous page)

```
                        IN6ADDR_ANY_INIT. [RT #1416]

909.    [placeholder]

908.    [func]          New program, rndc-confgen, to simplify setting up rndc.

907.    [func]          The ability to get entropy from either the
                        random device, a user-provided file or from
                        the keyboard was migrated from the DNSSEC tools
                        to libisc as isc_entropy_usebestsource().

906.    [port]          Separated the system independent portion of
                        lib/isc/unix/entropy.c into lib/isc/entropy.c
                        and added lib/isc/win32/entropy.c.

905.    [bug]           Configuring a forward "zone" for the root domain
                        did not work. [RT #1418]

904.    [bug]           The server would leak memory if attempting to use
                        an expired TSIG key. [RT #1406]

903.    [bug]           dig should not crash when receiving a TCP packet
                        of length 0.

902.    [bug]           The -d option was ignored if both -t and -g were also
                        specified.

901.    [placeholder]

900.    [bug]           A config.guess update changed the system identification
                        string of FreeBSD systems; configure and
                        bin/tests/system/ifconfig.sh now recognize the new
                        string.
```

```
      --- 9.2.0a2 released ---

899.    [bug]           lib/dns/soa.c failed to compile on many platforms
                        due to inappropriate use of a void value.
                        [RT #1372, #1373, #1386, #1387, #1395]

898.    [bug]           "dig" failed to set a nonzero exit status
                        on UDP query timeout. [RT #1323]

897.    [bug]           A config.guess update changed the system identification
                        string of UnixWare systems; configure now recognizes
                        the new string.

896.    [bug]           If a configuration file is set on named's command line
                        and it has a relative pathname, the current directory
                        (after any possible jailing resulting from named -t)
                        will be prepended to it so that reloading works
                        properly even when a directory option is present.
```

(continues on next page)

(continued from previous page)

```
895.   [func]          New function, isc_dir_current(), akin to POSIX's
                       getcwd().

894.   [bug]           When using the DNSSEC tools, a message intended to warn
                       when the keyboard was being used because of the lack
                       of a suitable random device was not being printed.

893.   [func]          Removed isc_file_test() and added isc_file_exists()
                       for the basic functionality that was being added
                       with isc_file_test().

892.   [placeholder]

891.   [bug]           Return an error when a SIG(0) signed response to
                       an unsigned query is seen.  This should actually
                       do the verification, but it's not currently
                       possible. [RT #1391]

890.   [cleanup]       The man pages no longer require the mandoc macros
                       and should now format cleanly using most versions of
                       nroff, and HTML versions of the man pages have been
                       added.  Both are generated from DocBook source.

889.   [port]          Eliminated blank lines before .TH in nroff man
                       pages since they cause problems with some versions
                       of nroff. [RT #1390]

888.   [bug]           Don't die when using TKEY to delete a nonexistent
                       TSIG key. [RT #1392]

887.   [port]          Detect broken compilers that can't call static
                       functions from inline functions. [RT #1212]

886.   [placeholder]

885.   [placeholder]

884.   [placeholder]

883.   [placeholder]

882.   [placeholder]

881.   [placeholder]

880.   [placeholder]

879.   [placeholder]

878.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
877.   [placeholder]

876.   [placeholder]

875.   [placeholder]

874.   [placeholder]

873.   [placeholder]

872.   [placeholder]

871.   [placeholder]

870.   [placeholder]

869.   [placeholder]

868.   [placeholder]

867.   [placeholder]

866.   [func]          Close debug only file channels when debug is set to
                       zero. [RT #1246]

865.   [bug]           The new configuration parser did not allow
                       the optional debug level in a "severity debug"
                       clause of a logging channel to be omitted.
                       This is now allowed and treated as "severity
                       debug 1;" like it does in BIND 8.2.4, not as
                       "severity debug 0;" like it did in BIND 9.1.
                       [RT #1367]

864.   [cleanup]       Multi-threading is now enabled by default on
                       OSF1, Solaris 2.7 and newer, AIX, IRIX, and HP-UX.

863.   [bug]           If an error occurred while an outgoing zone transfer
                       was starting up, the server could access a domain
                       name that had already been freed when logging a
                       message saying that the transfer was starting.
                       [RT #1383]

862.   [bug]           Use after realloc(), non portable pointer arithmetic in
                       grmerge().

861.   [port]          Add support for Mac OS X, by making it equivalent
                       to Darwin.  This was derived from the config.guess
                       file shipped with Mac OS X. [RT #1355]

860.   [func]          Drop cross class glue in zone transfers.

859.   [bug]           Cache cleaning now won't swamp the CPU if there
```

(continues on next page)

(continued from previous page)

```
                        is a persistent over limit condition.

858.    [func]          isc_mem_setwater() no longer requires that when the
                        callback function is non-NULL then its hi_water
                        argument must be greater than its lo_water argument
                        (they can now be equal) or that they be non-zero.

857.    [cleanup]       Use ISC_MAGIC() to define all magic numbers for
                        structs, for our friends in EBCDIC-land.

856.    [func]          Allow partial rdatasets to be returned in answer and
                        authority sections to help non-TCP capable clients
                        recover from truncation. [RT #1301]

855.    [bug]           Stop spurious "using RFC 1035 TTL semantics" warnings.

854.    [bug]           The config parser didn't properly handle config
                        options that were specified in units of time other
                        than seconds. [RT #1372]

853.    [bug]           configure_view_acl() failed to detach existing acls.
                        [RT #1374]

852.    [bug]           Handle responses from servers which do not know
                        about IXFR.

851.    [cleanup]       The obsolete support-ixfr option was not properly
                        ignored.
```

```
       --- 9.2.0a1 released ---

850.    [bug]           dns_rbt_findnode() would not find nodes that were
                        split on a bitstring label somewhere other than in
                        the last label of the node. [RT #1351]

849.    [func]          <isc/net.h> will ensure INADDR_LOOPBACK is defined.

848.    [func]          A minimum max-cache-size of two megabytes is enforced
                        by the cache cleaner.

847.    [func]          Added isc_file_test(), which currently only has
                        some very basic functionality to test for the
                        existence of a file, whether a pathname is absolute,
                        or whether a pathname is the fundamental representation
                        of the current directory.  It is intended that this
                        function can be expanded to test other things a
                        programmer might want to know about a file.

846.    [func]          A non-zero 'param' to dst_key_generate() when making an
                        hmac-md5 key means that good entropy is not required.

845.    [bug]           The access rights on the public file of a symmetric
```

(continues on next page)

(continued from previous page)

```
                        key are now restricted as soon as the file is opened,
                        rather than after it has been written and closed.

844.    [func]          <isc/net.h> will ensure INADDR_LOOPBACK is defined,
                        just as <lwres/net.h> does.

843.    [func]          If no controls statement is present in named.conf,
                        or if any inet phrase of a controls statement is
                        lacking a keys clause, then a key will be automatically
                        generated by named and an rndc.conf-style file
                        named named.key will be written that uses it.  rndc
                        will use this file only if its normal configuration
                        file, or one provided on the command line, does not
                        exist.

842.    [func]          'rndc flush' now takes an optional view.

841.    [bug]           When sdb modules were not declared threadsafe, their
                        create and destroy functions were not serialized.

840.    [bug]           The config file parser could print the wrong file
                        name if an error was detected after an included file
                        was parsed. [RT #1353]

839.    [func]          Dump packets for which there was no view or that the
                        class could not be determined to category "unmatched".

838.    [port]          UnixWare 7.x.x is now supported by
                        bin/tests/system/ifconfig.sh.

837.    [cleanup]       Multi-threading is now enabled by default only on
                        OSF1, Solaris 2.7 and newer, and AIX.

836.    [func]          Upgraded libtool to 1.4.

835.    [bug]           The dispatcher could enter a busy loop if
                        it got an I/O error receiving on a UDP socket.
                        [RT #1293]

834.    [func]          Accept (but warn about) master files beginning with
                        an SOA record without an explicit TTL field and
                        lacking a $TTL directive, by using the SOA MINTTL
                        as a default TTL.  This is for backwards compatibility
                        with old versions of BIND 8, which accepted such
                        files without warning although they are illegal
                        according to RFC1035.

833.    [cleanup]       Moved dns_soa_*() from <dns/journal.h> to
                        <dns/soa.h>, and extended them to support
                        all the integer-valued fields of the SOA RR.

832.    [bug]           The default location for named.conf in named-checkconf
```

(continues on next page)

(continued from previous page)

```
                      should depend on --sysconfdir like it does in named.
                      [RT #1258]

831.  [placeholder]

830.  [func]          Implement 'rndc status'.

829.  [bug]           The DNS_R_ZONECUT result code should only be returned
                      when an ANY query is made with DNS_DBFIND_GLUEOK set.
                      In all other ANY query cases, returning the delegation
                      is better.

828.  [bug]           The errno value from recvfrom() could be overwritten
                      by logging code. [RT #1293]

827.  [bug]           When an IXFR protocol error occurs, the slave
                      should retry with AXFR.

826.  [bug]           Some IXFR protocol errors were not detected.

825.  [bug]           zone.c:ns_query() detached from the wrong zone
                      reference. [RT #1264]

824.  [bug]           Correct line numbers reported by dns_master_load().
                      [RT #1263]

823.  [func]          The output of "dig -h" now goes to stdout so that it
                      can easily be piped through "more". [RT #1254]

822.  [bug]           Sending nxrrset prerequisites would crash nsupdate.
                      [RT #1248]

821.  [bug]           The program name used when logging to syslog should
                      be stripped of leading path components.
                      [RT #1178, #1232]

820.  [bug]           Name server address lookups failed to follow
                      A6 chains into the glue of local authoritative
                      zones.

819.  [bug]           In certain cases, the resolver's attempts to
                      restart an address lookup at the root could cause
                      the fetch to deadlock (with itself) instead of
                      restarting. [RT #1225]

818.  [bug]           Certain pathological responses to ANY queries could
                      cause an assertion failure. [RT #1218]

817.  [func]          Adjust timeouts for dialup zone queries.

816.  [bug]           Report potential problems with log file accessibility
                      at configuration time, since such problems can't
```

(continues on next page)

(continued from previous page)

```
                      reliably be reported at the time they actually occur.

815.   [bug]          If a log file was specified with a path separator
                      character (i.e. "/") in its name and the directory
                      did not exist, the log file's name was treated as
                      though it were the directory name. [RT #1189]

814.   [bug]          Socket objects left over from accept() failures
                      were incorrectly destroyed, causing corruption
                      of socket manager data structures.

813.   [bug]          File descriptors exceeding FD_SETSIZE were handled
                      badly. [RT #1192]

812.   [bug]          dig sometimes printed incomplete IXFR responses
                      due to an uninitialized variable. [RT #1188]

811.   [bug]          Parentheses were not quoted in zone dumps. [RT #1194]

810.   [bug]          The signer name in SIG records was not properly
                      down-cased when signing/verifying records. [RT #1186]

809.   [bug]          Configuring a non-local address as a transfer-source
                      could cause an assertion failure during load.

808.   [func]         Add 'rndc flush' to flush the server's cache.

807.   [bug]          When setting up TCP connections for incoming zone
                      transfers, the transfer-source port was not
                      ignored like it should be.

806.   [bug]          DNS_R_SEENINCLUDE was failing to propagate back up
                      the calling stack to the zone maintenance level,
                      causing zones to not reload when an included file was
                      touched but the top-level zone file was not.

805.   [bug]          When using "forward only", missing root hints should
                      not cause queries to fail. [RT #1143]

804.   [bug]          Attempting to obtain entropy could fail in some
                      situations.  This would be most common on systems
                      with user-space threads. [RT #1131]

803.   [bug]          Treat all SIG queries as if they have the CD bit set,
                      otherwise no data will be returned [RT #749]

802.   [bug]          DNSSEC key tags were computed incorrectly in almost
                      all cases. [RT #1146]

801.   [bug]          nsupdate should treat lines beginning with ';' as
                      comments. [RT #1139]
```

(continues on next page)

(continued from previous page)

```
800.   [bug]           dnssec-signzone produced incorrect statistics for
                       large zones. [RT #1133]

799.   [bug]           The ADB didn't find AAAA glue in a zone unless A6
                       glue was also present.

798.   [bug]           nsupdate should be able to reject bad input lines
                       and continue. [RT #1130]

797.   [func]          Issue a warning if the 'directory' option contains
                       a relative path. [RT #269]

796.   [func]          When a size limit is associated with a log file,
                       only roll it when the size is reached, not every
                       time the log file is opened. [RT #1096]

795.   [func]          Add the +multiline option to dig. [RT #1095]

794.   [func]          Implement the "port" and "default-port" statements
                       in rndc.conf.

793.   [cleanup]       The DNSSEC tools could create filenames that were
                       illegal or contained shell meta-characters.  They
                       now use a different text encoding of names that
                       doesn't have these problems. [RT #1101]

792.   [cleanup]       Replace the OMAPI command channel protocol with a
                       simpler one.

791.   [bug]           The command channel now works over IPv6.

790.   [bug]           Wildcards created using dynamic update or IXFR
                       could fail to match. [RT #1111]

789.   [bug]           The "localhost" and "localnets" ACLs did not match
                       when used as the second element of a two-element
                       sortlist item.

788.   [func]          Add the "match-mapped-addresses" option, which
                       causes IPv6 v4mapped addresses to be treated as
                       IPv4 addresses for the purpose of acl matching.

787.   [bug]           The DNSSEC tools failed to downcase domain
                       names when mapping them into file names.

786.   [bug]           When DNSSEC signing/verifying data, owner names were
                       not properly down-cased.

785.   [bug]           A race condition in the resolver could cause
                       an assertion failure. [RT #673, #872, #1048]

784.   [bug]           nsupdate and other programs would not quit properly
```

(continues on next page)

(continued from previous page)

```
                        if some signals were blocked by the caller. [RT #1081]

783.    [bug]           Following CNAMEs could cause an assertion failure
                        when either using an sdb database or under very
                        rare conditions.

782.    [func]          Implement the "serial-query-rate" option.

781.    [func]          Avoid error packet loops by dropping duplicate FORMERR
                        responses. [RT #1006]

780.    [bug]           Error handling code dealing with out of memory or
                        other rare errors could lead to assertion failures
                        by calling functions on uninitialized names. [RT #1065]

779.    [func]          Added the "minimal-responses" option.

778.    [bug]           When starting cache cleaning, cleaning_timer_action()
                        returned without first pausing the iterator, which
                        could cause deadlock. [RT #998]

777.    [bug]           An empty forwarders list in a zone failed to override
                        global forwarders. [RT #995]

776.    [func]          Improved error reporting in denied messages. [RT #252]

775.    [placeholder]

774.    [func]          max-cache-size is implemented.

773.    [func]          Added isc_rwlock_trylock() to attempt to lock without
                        blocking.

772.    [bug]           Owner names could be incorrectly omitted from cache
                        dumps in the presence of negative caching entries.
                        [RT #991]

771.    [cleanup]       TSIG errors related to unsynchronized clocks
                        are logged better. [RT #919]

770.    [func]          Add the "edns yes_or_no" statement to the server
                        clause. [RT #524]

769.    [func]          Improved error reporting when parsing rdata. [RT #740]

768.    [bug]           The server did not emit an SOA when a CNAME
                        or DNAME chain ended in NXDOMAIN in an
                        authoritative zone.

767.    [placeholder]

766.    [bug]           A few cases in query_find() could leak fname.
```

(continues on next page)

(continued from previous page)

```
                        This would trigger the mpctx->allocated == 0
                        assertion when the server exited.
                        [RT #739, #776, #798, #812, #818, #821, #845,
                        #892, #935, #966]

765.    [func]          ACL names are once again case insensitive, like
                        in BIND 8. [RT #252]

764.    [func]          Configuration files now allow "include" directives
                        in more places, such as inside the "view" statement.
                        [RT #377, #728, #860]

763.    [func]          Configuration files no longer have reserved words.
                        [RT #731, #753]

762.    [cleanup]       The named.conf and rndc.conf file parsers have
                        been completely rewritten.

761.    [bug]           _REENTRANT was still defined when building with
                        --disable-threads.

760.    [contrib]       Significant enhancements to the pgsql sdb driver.

759.    [bug]           The resolver didn't turn off "avoid fetches" mode
                        when restarting, possibly causing resolution
                        to fail when it should not.  This bug only affected
                        platforms which support both IPv4 and IPv6. [RT #927]

758.    [bug]           The "avoid fetches" code did not treat negative
                        cache entries correctly, causing fetches that would
                        be useful to be avoided.  This bug only affected
                        platforms which support both IPv4 and IPv6. [RT #927]

757.    [func]          Log zone transfers.

756.    [bug]           dns_zone_load() could "return" success when no master
                        file was configured.

755.    [bug]           Fix incorrectly formatted log messages in zone.c.

754.    [bug]           Certain failure conditions sending UDP packets
                        could cause the server to retry the transmission
                        indefinitely. [RT #902]

753.    [bug]           dig, host, and nslookup would fail to contact a
                        remote server if getaddrinfo() returned an IPv6
                        address on a system that doesn't support IPv6.
                        [RT #917]

752.    [func]          Correct bad tv_usec elements returned by
                        gettimeofday().
```

(continues on next page)

(continued from previous page)

```
751.   [func]          Log successful zone loads / transfers.  [RT #898]

750.   [bug]           A query should not match a DNAME whose trust level
                       is pending. [RT #916]

749.   [bug]           When a query matched a DNAME in a secure zone, the
                       server did not return the signature of the DNAME.
                       [RT #915]

748.   [doc]           List supported RFCs in doc/misc/rfc-compliance.
                       [RT #781]

747.   [bug]           The code to determine whether an IXFR was possible
                       did not properly check for a database that could
                       not have a journal. [RT #865, #908]

746.   [bug]           The sdb didn't clone rdatasets properly, causing
                       a crash when the server followed delegations. [RT #905]

745.   [func]          Report the owner name of records that fail
                       semantic checks while loading.

744.   [bug]           When returning DNS_R_CNAME or DNS_R_DNAME as the
                       result of an ANY or SIG query, the resolver failed
                       to setup the return event's rdatasets, causing an
                       assertion failure in the query code. [RT #881]

743.   [bug]           Receiving a large number of certain malformed
                       answers could cause named to stop responding.
                       [RT #861]

742.   [placeholder]

741.   [port]          Support openssl-engine. [RT #709]

740.   [port]          Handle openssl library mismatches slightly better.

739.   [port]          Look for /dev/random in configure, rather than
                       assuming it will be there for only a predefined
                       set of OSes.

738.   [bug]           If a non-threadsafe sdb driver supported AXFR and
                       received an AXFR request, it would deadlock or die
                       with an assertion failure. [RT #852]

737.   [port]          stdtime.c failed to compile on certain platforms.

736.   [func]          New functions isc_task_{begin,end}exclusive().

735.   [doc]           Add BIND 4 migration notes.

734.   [bug]           An attempt to re-lock the zone lock could occur if
```

(continues on next page)

(continued from previous page)

```
                        the server was shutdown during a zone transfer.
                        [RT #830]

733.    [bug]           Reference counts of dns_acl_t objects need to be
                        locked but were not. [RT #801, #821]

732.    [bug]           Glue with 0 TTL could also cause SERVFAIL. [RT #828]

731.    [bug]           Certain zone errors could cause named-checkzone to
                        fail ungracefully. [RT #819]

730.    [bug]           lwres_getaddrinfo() returns the correct result when
                        it fails to contact a server. [RT #768]

729.    [port]          pthread_setconcurrency() needs to be called on Solaris.

728.    [bug]           Fix comment processing on master file directives.
                        [RT #757]

727.    [port]          Work around OS bug where accept() succeeds but
                        fails to fill in the peer address of the accepted
                        connection, by treating it as an error rather than
                        an assertion failure. [RT #809]

726.    [func]          Implement the "trace" and "notrace" commands in rndc.

725.    [bug]           Installing man pages could fail.

724.    [func]          New libisc functions isc_netaddr_any(),
                        isc_netaddr_any6().

723.    [bug]           Referrals whose NS RRs had a 0 TTL caused the resolver
                        to return DNS_R_SERVFAIL. [RT #783]

722.    [func]          Allow incremental loads to be canceled.

721.    [cleanup]       Load manager and dns_master_loadfilequota() are no
                        more.

720.    [bug]           Server could enter infinite loop in
                        dispatch.c:do_cancel(). [RT #733]

719.    [bug]           Rapid reloads could trigger an assertion failure.
                        [RT #743, #763]

718.    [cleanup]       "internal" is no longer a reserved word in named.conf.
                        [RT #753, #731]

717.    [bug]           Certain TKEY processing failure modes could
                        reference an uninitialized variable, causing the
                        server to crash. [RT #750]
```

(continues on next page)

(continued from previous page)

```
716.   [bug]           The first line of a $INCLUDE master file was lost if
                       an origin was specified. [RT #744]

715.   [bug]           Resolving some A6 chains could cause an assertion
                       failure in adb.c. [RT #738]

714.   [bug]           Preserve interval timers across reloads unless changed.
                       [RT #729]

713.   [func]          named-checkconf takes '-t directory' similar to named.
                       [RT #726]

712.   [bug]           Sending a large signed update message caused an
                       assertion failure. [RT #718]

711.   [bug]           The libisc and liblwres implementations of
                       inet_ntop contained an off by one error.

710.   [func]          The forwarders statement now takes an optional
                       port. [RT #418]

709.   [bug]           ANY or SIG queries for data with a TTL of 0
                       would return SERVFAIL. [RT #620]

708.   [bug]           When building with --with-openssl, the openssl headers
                       included with BIND 9 should not be used. [RT #702]

707.   [func]          The "filename" argument to named-checkzone is no
                       longer optional, to reduce confusion. [RT #612]

706.   [bug]           Zones with an explicit "allow-update { none; };"
                       were considered dynamic and therefore not reloaded
                       on SIGHUP or "rndc reload".

705.   [port]          Work out resource limit type for use where rlim_t is
                       not available. [RT #695]

704.   [port]          RLIMIT_NOFILE is not available on all platforms.
                       [RT #695]

703.   [port]          sys/select.h is needed on older platforms. [RT #695]

702.   [func]          If the address 0.0.0.0 is seen in resolv.conf,
                       use 127.0.0.1 instead. [RT #693]

701.   [func]          Root hints are now fully optional.  Class IN
                       views use compiled-in hints by default, as
                       before.  Non-IN views with no root hints now
                       provide authoritative service but not recursion.
                       A warning is logged if a view has neither root
                       hints nor authoritative data for the root. [RT #696]
```

(continues on next page)

```
700.   [bug]           $GENERATE range check was wrong. [RT #688]

699.   [bug]           The lexer mishandled empty quoted strings. [RT #694]

698.   [bug]           Aborting nsupdate with ^C would lead to several
                       race conditions.

697.   [bug]           nsupdate was not compatible with the undocumented
                       BIND 8 behavior of ignoring TTLs in "update delete"
                       commands. [RT #693]

696.   [bug]           lwresd would die with an assertion failure when passed
                       a zero-length name. [RT #692]

695.   [bug]           If the resolver attempted to query a blackholed or
                       bogus server, the resolution would fail immediately.

694.   [bug]           $GENERATE did not produce the last entry.
                       [RT #682, #683]

693.   [bug]           An empty lwres statement in named.conf caused
                       the server to crash while loading.

692.   [bug]           Deal with systems that have getaddrinfo() but not
                       gai_strerror(). [RT #679]

691.   [bug]           Configuring per-view forwarders caused an assertion
                       failure. [RT #675, #734]

690.   [func]          $GENERATE now supports DNAME. [RT #654]

689.   [doc]           man pages are now installed. [RT #210]

688.   [func]          "make tags" now works on systems with the
                       "Exuberant Ctags" etags.

687.   [bug]           Only say we have IPv6, with sufficient functionality,
                       if it has actually been tested. [RT #586]

686.   [bug]           dig and nslookup can now be properly aborted during
                       blocking operations. [RT #568]

685.   [bug]           nslookup should use the search list/domain options
                       from resolv.conf by default. [RT #405, #630]

684.   [bug]           Memory leak with view forwarders. [RT #656]

683.   [bug]           File descriptor leak in isc_lex_openfile().

682.   [bug]           nslookup displayed SOA records incorrectly. [RT #665]

681.   [bug]           $GENERATE specifying output format was broken. [RT #653]
```

(continues on next page)

(continued from previous page)

```
680.   [bug]           dns_rdata_fromstruct() mishandled options bigger
                       than 255 octets.

679.   [bug]           $INCLUDE could leak memory and file descriptors on
                       reload. [RT #639]

678.   [bug]           "transfer-format one-answer;" could trigger an assertion
                       failure. [RT #646]

677.   [bug]           dnssec-signzone would occasionally use the wrong ttl
                       for database operations and fail. [RT #643]

676.   [bug]           Log messages about lame servers to category
                       'lame-servers' rather than 'resolver', so as not
                       to be gratuitously incompatible with BIND 8.

675.   [bug]           TKEY queries could cause the server to leak
                       memory.

674.   [func]          Allow messages to be TSIG signed / verified using
                       a offset from the current time.

673.   [func]          The server can now convert RFC1886-style recursive
                       lookup requests into RFC2874-style lookups, when
                       enabled using the new option "allow-v6-synthesis".

672.   [bug]           The wrong time was in the "time signed" field when
                       replying with BADTIME error.

671.   [bug]           The message code was failing to parse a message with
                       no question section and a TSIG record. [RT #628]

670.   [bug]           The lwres replacements for getaddrinfo and
                       getipnodebyname didn't properly check for the
                       existence of the sockaddr sa_len field.

669.   [bug]           dnssec-keygen now makes the public key file
                       non-world-readable for symmetric keys. [RT #403]

668.   [func]          named-checkzone now reports multiple errors in master
                       files.

667.   [bug]           On Linux, running named with the -u option and a
                       non-world-readable configuration file didn't work.
                       [RT #626]

666.   [bug]           If a request sent by dig is longer than 512 bytes,
                       use TCP.

665.   [bug]           Signed responses were not sent when the size of the
                       TSIG + question exceeded the maximum message size.
                       [RT #628]
```

(continues on next page)

(continued from previous page)

```
664.   [bug]           The t_tasks and t_timers module tests are now skipped
                       when building without threads, since they require
                       threads.

663.   [func]          Accept a size_spec, not just an integer, in the
                       (unimplemented and ignored) max-ixfr-log-size option
                       for compatibility with recent versions of BIND 8.
                       [RT #613]

662.   [bug]           dns_rdata_fromtext() failed to log certain errors.

661.   [bug]           Certain UDP IXFR requests caused an assertion failure
                       (mpctx->allocated == 0). [RT #355, #394, #623]

660.   [port]          Detect multiple CPUs on HP-UX and IRIX.

659.   [performance]   Rewrite the name compression code to be much faster.

658.   [cleanup]       Remove all vestiges of 16 bit global compression.

657.   [bug]           When a listen-on statement in an lwres block does not
                       specify a port, use 921, not 53.  Also update the
                       listen-on documentation. [RT #616]

656.   [func]          Treat an unescaped newline in a quoted string as
                       an error.  This means that TXT records with missing
                       close quotes should have meaningful errors printed.

655.   [bug]           Improve error reporting on unexpected eof when loading
                       zones. [RT #611]

654.   [bug]           Origin was being forgotten in TCP retries in dig.
                       [RT #574]

653.   [bug]           +defname option in dig was reversed in sense.
                       [RT #549]

652.   [bug]           zone_saveunique() did not report the new name.

651.   [func]          The AD bit in responses now has the meaning
                       specified in <draft-ietf-dnsext-ad-is-secure>.

650.   [bug]           SIG(0) records were being generated and verified
                       incorrectly. [RT #606]

649.   [bug]           It was possible to join to an already running fctx
                       after it had "cloned" its events, but before it sent
                       them.  In this case, the event of the newly joined
                       fetch would not contain the answer, and would
                       trigger the INSIST() in fctx_sendevents().  In
                       BIND 9.0, this bug did not trigger an INSIST(), but
```

(continues on next page)

(continued from previous page)

```
                      caused the fetch to fail with a SERVFAIL result.
                      [RT #588, #597, #605, #607]

648.  [port]          Add support for pre-RFC2133 IPv6 implementations.

647.  [bug]           Resolver queries sent after following multiple
                      referrals had excessively long retransmission
                      timeouts due to incorrectly counting the referrals
                      as "restarts".

646.  [bug]           The UnixWare ISC_PLATFORM_FIXIN6INADDR fix in isc/net.h
                      didn't _cleanly_ fix the problem it was trying to fix.

645.  [port]          BSD/OS 3.0 needs pthread_init(). [RT #603]

644.  [bug]           #622 needed more work. [RT #562]

643.  [bug]           xfrin error messages made more verbose, added class
                      of the zone. [RT #599]

642.  [bug]           Break the exit_check() race in the zone module.
                      [RT #598]
```

```
      --- 9.1.0b2 released ---

641.  [bug]           $GENERATE caused a uninitialized link to be used.
                      [RT #595]

640.  [bug]           Memory leak in error path could cause
                      "mpctx->allocated == 0" failure. [RT #584]

639.  [bug]           Reading entropy from the keyboard would sometimes fail.
                      [RT #591]

638.  [port]          lib/isc/random.c needed to explicitly include time.h
                      to get a prototype for time() when pthreads was not
                      being used. [RT #592]

637.  [port]          Use isc_u?int64_t instead of (unsigned) long long in
                      lib/isc/print.c.  Also allow lib/isc/print.c to
                      be compiled even if the platform does not need it.
                      [RT #592]

636.  [port]          Shut up MSVC++ about a possible loss of precision
                      in the ISC__BUFFER_PUTUINT*() macros. [RT #592]

635.  [bug]           Reloading a server with a configured blackhole list
                      would cause an assertion. [RT #590]

634.  [bug]           A log file will completely stop being written when
                      it reaches the maximum size in all cases, not just
                      when versioning is also enabled. [RT #570]
```

(continues on next page)

(continued from previous page)

```
633.   [port]          Cope with rlim_t missing on BSD/OS systems. [RT #575]

632.   [bug]           The index array of the journal file was
                       corrupted as it was written to disk.

631.   [port]          Build without thread support on systems without
                       pthreads.

630.   [bug]           Locking failure in zone code. [RT #582]

629.   [bug]           9.1.0b1 dereferenced a null pointer and crashed
                       when responding to a UDP IXFR request.

628.   [bug]           If the root hints contained only AAAA addresses,
                       named would be unable to perform resolution.

627.   [bug]           The EDNS0 blackhole detection code of change 324
                       waited for three retransmissions to each server,
                       which takes much too long when a domain has many
                       name servers and all of them drop EDNS0 queries.
                       Now we retry without EDNS0 after three consecutive
                       timeouts, even if they are all from different
                       servers. [RT #143]

626.   [bug]           The lightweight resolver daemon no longer crashes
                       when asked for a SIG rrset. [RT #558]

625.   [func]          Zones now inherit their class from the enclosing view.

624.   [bug]           The zone object could get timer events after it had
                       been destroyed, causing a server crash. [RT #571]

623.   [func]          Added "named-checkconf" and "named-checkzone" program
                       for syntax checking named.conf files and zone files,
                       respectively.

622.   [bug]           A canceled request could be destroyed before
                       dns_request_destroy() was called. [RT #562]

621.   [port]          Disable IPv6 at runtime if IPv6 sockets are unusable.
                       This mostly affects Red Hat Linux 7.0, which has
                       conflicts between libc and the kernel.

620.   [bug]           dns_master_load*inc() now require 'task' and 'load'
                       to be non-null.  Also 'done' will not be called if
                       dns_master_load*inc() fails immediately. [RT #565]

619.   [placeholder]

618.   [bug]           Queries to a signed zone could sometimes cause
                       an assertion failure.
```

(continues on next page)

(continued from previous page)

```
617.   [bug]           When using dynamic update to add a new RR to an
                       existing RRset with a different TTL, the journal
                       entries generated from the update did not include
                       explicit deletions and re-additions of the existing
                       RRs to update their TTL to the new value.

616.   [func]          dnssec-signzone -t output now includes performance
                       statistics.

615.   [bug]           dnssec-signzone did not like child keysets signed
                       by multiple keys.

614.   [bug]           Checks for uninitialized link fields were prone
                       to false positives, causing assertion failures.
                       The checks are now disabled by default and may
                       be re-enabled by defining ISC_LIST_CHECKINIT.

613.   [bug]           "rndc reload zone" now reloads primary zones.
                       It previously only updated slave and stub zones,
                       if an SOA query indicated an out of date serial.

612.   [cleanup]       Shutup a ridiculously noisy HP-UX compiler that
                       complains relentlessly about how its treatment
                       of 'const' has changed as well as how casting
                       sometimes tightens alignment constraints.

611.   [func]          allow-notify can be used to permit processing of
                       notify messages from hosts other than a slave's
                       masters.

610.   [func]          rndc dumpdb is now supported.

609.   [bug]           getrrsetbyname() would crash lwresd if the server
                       found more SIGs than answers. [RT #554]

608.   [func]          dnssec-signzone now adds a comment to the zone
                       with the time the file was signed.

607.   [bug]           nsupdate would fail if it encountered a CNAME or
                       DNAME in a response to an SOA query. [RT #515]

606.   [bug]           Compiling with --disable-threads failed due
                       to isc_thread_self() being incorrectly defined
                       as an integer rather than a function.

605.   [func]          New function isc_lex_getlasttokentext().

604.   [bug]           The named.conf parser could print incorrect line
                       numbers when long comments were present.

603.   [bug]           Make dig handle multiple types or classes on the same
```

(continues on next page)

(continued from previous page)

```
                        query more correctly.

602.    [func]          Cope automatically with UnixWare's broken
                        IN6_IS_ADDR_* macros. [RT #539]

601.    [func]          Return a non-zero exit code if an update fails
                        in nsupdate.

600.    [bug]           Reverse lookups sometimes failed in dig, etc...

599.    [func]          Added four new functions to the libisc log API to
                        support i18n messages.  isc_log_iwrite(),
                        isc_log_ivwrite(), isc_log_iwrite1() and
                        isc_log_ivwrite1() were added.

598.    [bug]           An update-policy statement would cause the server
                        to assert while loading. [RT #536]

597.    [func]          dnssec-signzone is now multi-threaded.

596.    [bug]           DNS_RDATASLAB_FORCE and DNS_RDATASLAB_EXACT are
                        not mutually exclusive.

595.    [port]          On Linux 2.2, socket() returns EINVAL when it
                        should return EAFNOSUPPORT.  Work around this.
                        [RT #531]

594.    [func]          sdb drivers are now assumed to not be thread-safe
                        unless the DNS_SDBFLAG_THREADSAFE flag is supplied.

593.    [bug]           If a secure zone was missing all its NXTs and
                        a dynamic update was attempted, the server entered
                        an infinite loop.

592.    [bug]           The sig-validity-interval option now specifies a
                        number of days, not seconds.  This matches the
                        documentation. [RT #529]
```

```
      --- 9.1.0b1 released ---

591.    [bug]           Work around non-reentrancy in openssl by disabling
                        pre-computation in keys.

590.    [doc]           There are now man pages for the lwres library in
                        doc/man/lwres.

589.    [bug]           The server could deadlock if a zone was updated
                        while being transferred out.

588.    [bug]           ctx->in_use was not being correctly initialized when
                        when pushing a file for $INCLUDE. [RT #523]
```

(continues on next page)

(continued from previous page)

```
587.   [func]          A warning is now printed if the "allow-update"
                       option allows updates based on the source IP
                       address, to alert users to the fact that this
                       is insecure and becoming increasingly so as
                       servers capable of update forwarding are being
                       deployed.

586.   [bug]           multiple views with the same name were fatal. [RT #516]

585.   [func]          dns_db_addrdataset() and dns_rdataslab_merge()
                       now support 'exact' additions in a similar manner to
                       dns_db_subtractrdataset() and dns_rdataslab_subtract().

584.   [func]          You can now say 'notify explicit'; to suppress
                       notification of the servers listed in NS records
                       and notify only those servers listed in the
                       'also-notify' option.

583.   [func]          "rndc querylog" will now toggle logging of
                       queries, like "ndc querylog" in BIND 8.

582.   [bug]           dns_zone_idetach() failed to lock the zone.
                       [RT #199, #463]

581.   [bug]           log severity was not being correctly processed.
                       [RT #485]

580.   [func]          Ignore trailing garbage on incoming DNS packets,
                       for interoperability with broken server
                       implementations. [RT #491]

579.   [bug]           nsupdate did not take a filename to read update from.
                       [RT #492]

578.   [func]          New config option "notify-source", to specify the
                       source address for notify messages.

577.   [func]          Log illegal RDATA combinations. e.g. multiple
                       singleton types, cname and other data.

576.   [doc]           isc_log_create() description did not match reality.

575.   [bug]           isc_log_create() was not setting internal state
                       correctly to reflect the default channels created.

574.   [bug]           TSIG signed queries sent by the resolver would fail to
                       have their responses validated and would leak memory.

573.   [bug]           The journal files of IXFRed slave zones were
                       inadvertently discarded on server reload, causing
                       "journal out of sync with zone" errors on subsequent
                       reloads. [RT #482]
```

(continues on next page)

(continued from previous page)

```
572.   [bug]           Quoted strings were not accepted as key names in
                       address match lists.

571.   [bug]           It was possible to create an rdataset of singleton
                       type which had more than one rdata. [RT #154]
                       [RT #279]

570.   [bug]           rbtdb.c allowed zones containing nodes which had
                       both a CNAME and "other data". [RT #154]

569.   [func]          The DNSSEC AD bit will not be set on queries which
                       have not requested a DNSSEC response.

568.   [func]          Add sample simple database drivers in contrib/sdb.

567.   [bug]           Setting the zone transfer timeout to zero caused an
                       assertion failure. [RT #302]

566.   [func]          New public function dns_timer_setidle().

565.   [func]          Log queries more like BIND 8: query logging is now
                       done to category "queries", level "info". [RT #169]

564.   [func]          Add sortlist support to lwresd.

563.   [func]          New public functions dns_rdatatype_format() and
                       dns_rdataclass_format(), for convenient formatting
                       of rdata type/class mnemonics in log messages.

562.   [cleanup]       Moved lib/dns/*conf.c to bin/named where they belong.

561.   [func]          The 'datasize', 'stacksize', 'coresize' and 'files'
                       clauses of the options{} statement are now implemented.

560.   [bug]           dns_name_split did not properly the resulting prefix
                       when a maximal length bitstring label was split which
                       was preceded by another bitstring label. [RT #429]

559.   [bug]           dns_name_split did not properly create the suffix
                       when splitting within a maximal length bitstring label.

558.   [func]          New functions, isc_resource_getlimit and
                       isc_resource_setlimit.

557.   [func]          Symbolic constants for libisc integral types.

556.   [func]          The DNSSEC OK bit in the EDNS extended flags
                       is now implemented.  Responses to queries without
                       this bit set will not contain any DNSSEC records.

555.   [bug]           A slave server attempting a zone transfer could
```

(continues on next page)

(continued from previous page)

```
                        crash with an assertion failure on certain
                        malformed responses from the master. [RT #457]

554.    [bug]           In some cases, not all of the dnssec tools were
                        properly installed.

553.    [bug]           Incoming zone transfers deferred due to quota
                        were not started when quota was increased but
                        only when a transfer in progress finished. [RT #456]

552.    [bug]           We were not correctly detecting the end of all c-style
                        comments. [RT #455]

551.    [func]          Implemented the 'sortlist' option.

550.    [func]          Support unknown rdata types and classes.

549.    [bug]           "make" did not immediately abort the build when a
                        subdirectory make failed [RT #450].

548.    [func]          The lexer now ungets tokens more correctly.

547.    [placeholder]

546.    [func]          Option 'lame-ttl' is now implemented.

545.    [func]          Name limit and counting options removed from dig;
                        they didn't work properly, and cannot be correctly
                        implemented without significant changes.

544.    [func]          Add statistics option, enable statistics-file option,
                        add RNDC option "dump-statistics" to write out a
                        query statistics file.

543.    [doc]           The 'port' option is now documented.

542.    [func]          Add support for update forwarding as required for
                        full compliance with RFC2136.  It is turned off
                        by default and can be enabled using the
                        'allow-update-forwarding' option.

541.    [func]          Add bogus server support.

540.    [func]          Add dialup support.

539.    [func]          Support the blackhole option.

538.    [bug]           fix buffer overruns by 1 in lwres_getnameinfo().

537.    [placeholder]

536.    [func]          Use transfer-source{-v6} when sending refresh queries.
```

(continues on next page)

(continued from previous page)

```
                      Transfer-source{-v6} now take a optional port
                      parameter for setting the UDP source port.  The port
                      parameter is ignored for TCP.

535.  [func]          Use transfer-source{-v6} when forwarding update
                      requests.

534.  [func]          Ancestors have been removed from RBT chains.  Ancestor
                      information can be discerned via node parent pointers.

533.  [func]          Incorporated name hashing into the RBT database to
                      improve search speed.

532.  [func]          Implement DNS UPDATE pseudo records using
                      DNS_RDATA_UPDATE flag.

531.  [func]          Rdata really should be initialized before being assigned
                      to (dns_rdata_fromwire(), dns_rdata_fromtext(),
                      dns_rdata_clone(), dns_rdata_fromregion()),
                      check that it is.

530.  [func]          New function dns_rdata_invalidate().

529.  [bug]           521 contained a bug which caused zones to always
                      reload.  [RT #410]

528.  [func]          The ISC_LIST_XXXX macros now perform sanity checks
                      on their arguments.  ISC_LIST_XXXXUNSAFE can be use
                      to skip the checks however use with caution.

527.  [func]          New function dns_rdata_clone().

526.  [bug]           nsupdate incorrectly refused to add RRs with a TTL
                      of 0.

525.  [func]          New arguments 'options' for dns_db_subtractrdataset(),
                      and 'flags' for dns_rdataslab_subtract() allowing you
                      to request that the RR's must exist prior to deletion.
                      DNS_R_NOTEXACT is returned if the condition is not met.

524.  [func]          The 'forward' and 'forwarders' statement in
                      non-forward zones should work now.

523.  [doc]           The source to the Administrator Reference Manual is
                      now an XML file using the DocBook DTD, and is included
                      in the distribution.  The plain text version of the
                      ARM is temporarily unavailable while we figure out
                      how to generate readable plain text from the XML.

522.  [func]          The lightweight resolver daemon can now use
                      a real configuration file, and its functionality
                      can be provided by a name server.  Also, the -p and -P
```

(continues on next page)

(continued from previous page)

```
                      options to lwresd have been reversed.

521.   [bug]          Detect master files which contain $INCLUDE and always
                      reload. [RT #196]

520.   [bug]          Upgraded libtool to 1.3.5, which makes shared
                      library builds almost work on AIX (and possibly
                      others).

519.   [bug]          dns_name_split() would improperly split some bitstring
                      labels, zeroing a few of the least significant bits in
                      the prefix part.  When such an improperly created
                      prefix was returned to the RBT database, the bogus
                      label was dutifully stored, corrupting the tree.
                      [RT #369]

518.   [bug]          The resolver did not realize that a DNAME which was
                      "the answer" to the client's query was "the answer",
                      and such queries would fail. [RT #399]

517.   [bug]          The resolver's DNAME code would trigger an assertion
                      if there was more than one DNAME in the chain.
                      [RT #399]

516.   [bug]          Cache lookups which had a NULL node pointer, e.g.
                      those by dns_view_find(), and which would match a
                      DNAME, would trigger an INSIST(!search.need_cleanup)
                      assertion. [RT #399]

515.   [bug]          The ssu table was not being attached / detached
                      by dns_zone_[sg]etssutable. [RT #397]

514.   [func]         Retry refresh and notify queries if they timeout.
                      [RT #388]

513.   [func]         New functionality added to rdnc and server to allow
                      individual zones to be refreshed or reloaded.

512.   [bug]          The zone transfer code could throw an exception with
                      an invalid IXFR stream.

511.   [bug]          The message code could throw an assertion on an
                      out of memory failure. [RT #392]

510.   [bug]          Remove spurious view notify warning. [RT #376]

509.   [func]         Add support for write of zone files on shutdown.

508.   [func]         dns_message_parse() can now do a best-effort
                      attempt, which should allow dig to print more invalid
                      messages.
```

(continues on next page)

(continued from previous page)

```
507.   [func]          New functions dns_zone_flush(), dns_zt_flushanddetach()
                       and dns_view_flushanddetach().

506.   [func]          Do not fail to start on errors in zone files.

505.   [bug]           nsupdate was printing "unknown result code". [RT #373]

504.   [bug]           The zone was not being marked as dirty when updated via
                       IXFR.

503.   [bug]           dumptime was not being set along with
                       DNS_ZONEFLG_NEEDDUMP.

502.   [func]          On a SERVFAIL reply, DiG will now try the next server
                       in the list, unless the +fail option is specified.

501.   [bug]           Incorrect port numbers were being displayed by
                       nslookup. [RT #352]

500.   [func]          Nearly useless +details option removed from DiG.

499.   [func]          In DiG, specifying a class with -c or type with -t
                       changes command-line parsing so that classes and
                       types are only recognized if following -c or -t.
                       This allows hosts with the same name as a class or
                       type to be looked up.

498.   [doc]           There is now a man page for "dig"
                       in doc/man/bin/dig.1.

497.   [bug]           The error messages printed when an IP match list
                       contained a network address with a nonzero host
                       part where not sufficiently detailed. [RT #365]

496.   [bug]           named didn't sanity check numeric parameters. [RT #361]

495.   [bug]           nsupdate was unable to handle large records. [RT #368]

494.   [func]          Do not cache NXDOMAIN responses for SOA queries.

493.   [func]          Return non-cachable (ttl = 0) NXDOMAIN responses
                       for SOA queries.  This makes it easier to locate
                       the containing zone without polluting intermediate
                       caches.

492.   [bug]           attempting to reload a zone caused the server fail
                       to shutdown cleanly. [RT #360]

491.   [bug]           nsupdate would segfault when sending certain
                       prerequisites with empty RDATA. [RT #356]

490.   [func]          When a slave/stub zone has not yet successfully
```

(continues on next page)

(continued from previous page)

```
                      obtained an SOA containing the zone's configured
                      retry time, perform the SOA query retries using
                      exponential backoff. [RT #337]

489.  [func]          The zone manager now has a "i/o" queue.

488.  [bug]           Locks weren't properly destroyed in some cases.

487.  [port]          flockfile() is not defined on all systems.

486.  [bug]           nslookup: "set all" and "server" commands showed
                      the incorrect port number if a port other than 53
                      was specified. [RT #352]

485.  [func]          When dig had more than one server to query, it would
                      send all of the messages at the same time.  Add
                      rate limiting of the transmitted messages.

484.  [bug]           When the server was reloaded after removing addresses
                      from the named.conf "listen-on" statement, sockets
                      were still listening on the removed addresses due
                      to reference count loops. [RT #325]

483.  [bug]           nslookup: "set all" showed a "search" option but it
                      was not settable.

482.  [bug]           nslookup: a plain "server" or "lserver" should be
                      treated as a lookup.

481.  [bug]           nslookup:get_next_command() stack size could exceed
                      per thread limit.

480.  [bug]           strtok() is not thread safe. [RT #349]

479.  [func]          The test suite can now be run by typing "make check"
                      or "make test" at the top level.

478.  [bug]           "make install" failed if the directory specified with
                      --prefix did not already exist.

477.  [bug]           The the isc-config.sh script could be installed before
                      its directory was created. [RT #324]

476.  [bug]           A zone could expire while a zone transfer was in
                      progress triggering a INSIST failure. [RT #329]

475.  [bug]           query_getzonedb() sometimes returned a non-null version
                      on failure.  This caused assertion failures when
                      generating query responses where names subject to
                      additional section processing pointed to a zone
                      to which access had been denied by means of the
                      allow-query option. [RT #336]
```

(continues on next page)

(continued from previous page)

```
474.   [bug]           The mnemonic of the CHAOS class is CH according to
                       RFC1035, but it was printed and read only as CHAOS.
                       We now accept both forms as input, and print it
                       as CH. [RT #305]

473.   [bug]           nsupdate overran the end of the list of name servers
                       when no servers could be reached, typically causing
                       it to print the error message "dns_request_create:
                       not implemented".

472.   [bug]           Off-by-one error caused isc_time_add() to sometimes
                       produce invalid time values.

471.   [bug]           nsupdate didn't compile on HP/UX 10.20

470.   [func]          $GENERATE is now supported.  See also
                       doc/misc/migration.

469.   [bug]           "query-source address * port 53;" now works.

468.   [bug]           dns_master_load*() failed to report file and line
                       number in certain error conditions.

467.   [bug]           dns_master_load*() failed to log an error if
                       pushfile() failed.

466.   [bug]           dns_master_load*() could return success when it failed.

465.   [cleanup]       Allow 0 to be set as an omapi_value_t value by
                       omapi_value_storeint().

464.   [cleanup]       Build with openssl's RSA code instead of dnssafe.

463.   [bug]           nsupdate sent malformed SOA queries to the second
                       and subsequent name servers in resolv.conf if the
                       query sent to the first one failed.

462.   [bug]           --disable-ipv6 should work now.

461.   [bug]           Specifying an unknown key in the "keys" clause of the
                       "controls" statement caused a NULL pointer dereference.
                       [RT #316]

460.   [bug]           Much of the DNSSEC code only worked with class IN.

459.   [bug]           Nslookup processed the "set" command incorrectly.

458.   [bug]           Nslookup didn't properly check class and type values.
                       [RT #305]

457.   [bug]           Dig/host/hslookup didn't properly handle connect
```

(continues on next page)

(continued from previous page)

```
                        timeouts in certain situations, causing an
                        unnecessary warning message to be printed.

456.    [bug]           Stub zones were not resetting the refresh and expire
                        counters, loadtime or clearing the DNS_ZONE_REFRESH
                        (refresh in progress) flag upon successful update.
                        This disabled further refreshing of the stub zone,
                        causing it to eventually expire. [RT #300]

455.    [doc]           Document IPv4 prefix notation does not require a
                        dotted decimal quad but may be just dotted decimal.

454.    [bug]           Enforce dotted decimal and dotted decimal quad where
                        documented as such in named.conf. [RT #304, RT #311]

453.    [bug]           Warn if the obsolete option "maintain-ixfr-base"
                        is specified in named.conf. [RT #306]

452.    [bug]           Warn if the unimplemented option "statistics-file"
                        is specified in named.conf. [RT #301]

451.    [func]          Update forwarding implemented.

450.    [func]          New function ns_client_sendraw().

449.    [bug]           isc_bitstring_copy() only works correctly if the
                        two bitstrings have the same lsb0 value, but this
                        requirement was not documented, nor was there a
                        REQUIRE for it.

448.    [bug]           Host output formatting change, to match v8. [RT #255]

447.    [bug]           Dig didn't properly retry in TCP mode after
                        a truncated reply. [RT #277]

446.    [bug]           Confusing notify log message. [RT #298]

445.    [bug]           Doing a 0 bit isc_bitstring_copy() of an lsb0
                        bitstring triggered a REQUIRE statement.  The REQUIRE
                        statement was incorrect. [RT #297]

444.    [func]          "recursion denied" messages are always logged at
                        debug level 1, now, rather than sometimes at ERROR.
                        This silences these warnings in the usual case, where
                        some clients set the RD bit in all queries.

443.    [bug]           When loading a master file failed because of an
                        unrecognized RR type name, the error message
                        did not include the file name and line number.
                        [RT #285]

442.    [bug]           TSIG signed messages that did not match any view
```

(continues on next page)

```
                      crashed the server. [RT #290]

441.   [bug]          Nodes obscured by a DNAME were inaccessible even
                      when DNS_DBFIND_GLUEOK was set.

440.   [func]         New function dns_zone_forwardupdate().

439.   [func]         New function dns_request_createraw().

438.   [func]         New function dns_message_getrawmessage().

437.   [func]         Log NOTIFY activity to the notify channel.

436.   [bug]          If recvmsg() returned EHOSTUNREACH or ENETUNREACH,
                      which sometimes happens on Linux, named would enter
                      a busy loop.  Also, unexpected socket errors were
                      not logged at a high enough logging level to be
                      useful in diagnosing this situation. [RT #275]

435.   [bug]          dns_zone_dump() overwrote existing zone files
                      rather than writing to a temporary file and
                      renaming.  This could lead to empty or partial
                      zone files being left around in certain error
                      conditions involving the initial transfer of a
                      slave zone, interfering with subsequent server
                      startup. [RT #282]

434.   [func]         New function isc_file_isabsolute().

433.   [func]         isc_base64_decodestring() now accepts newlines
                      within the base64 data.  This makes it possible
                      to break up the key data in a "trusted-keys"
                      statement into multiple lines. [RT #284]

432.   [func]         Added refresh/retry jitter.  The actual refresh/
                      retry time is now a random value between 75% and
                      100% of the configured value.

431.   [func]         Log at ISC_LOG_INFO when a zone is successfully
                      loaded.

430.   [bug]          Rewrote the lightweight resolver client management
                      code to handle shutdown correctly and general
                      cleanup.

429.   [bug]          The space reserved for a TSIG record in a response
                      was 2 bytes too short, leading to message
                      generation failures.

428.   [bug]          rbtdb.c:find_closest_nxt() erroneously returned
                      DNS_R_BADDB for nodes which had neither NXT nor SIG NXT
                      (e.g. glue).  This could cause SERVFAILs when
                      generating negative responses in a secure zone.
```

(continues on next page)

(continued from previous page)

```
427.   [bug]           Avoid going into an infinite loop when the validator
                       gets a negative response to a key query where the
                       records are signed by the missing key.

426.   [bug]           Attempting to generate an oversized RSA key could
                       cause dnssec-keygen to dump core.

425.   [bug]           Warn about the auth-nxdomain default value change
                       if there is no auth-nxdomain statement in the
                       config file. [RT #287]

424.   [bug]           notify_createmessage() could trigger an assertion
                       failure when creating the notify message failed,
                       e.g. due to corrupt zones with multiple SOA records.
                       [RT #279]

423.   [bug]           When responding to a recursive query, errors that occur
                       after following a CNAME should cause the query to fail.
                       [RT #274]

422.   [func]          get rid of isc_random_t, and make isc_random_get()
                       and isc_random_jitter() use rand() internally
                       instead of local state.  Note that isc_random_*()
                       functions are only for weak, non-critical "randomness"
                       such as timing jitter and such.

421.   [bug]           nslookup would exit when given a blank line as input.

420.   [bug]           nslookup failed to implement the "exit" command.

419.   [bug]           The certificate type PKIX was misspelled as SKIX.

418.   [bug]           At debug levels >= 10, getting an unexpected
                       socket receive error would crash the server
                       while trying to log the error message.

417.   [func]          Add isc_app_block() and isc_app_unblock(), which
                       allow an application to handle signals while
                       blocking.

416.   [bug]           Slave zones with no master file tried to use a
                       NULL pointer for a journal file name when they
                       received an IXFR. [RT #273]

415.   [bug]           The logging code leaked file descriptors.

414.   [bug]           Server did not shut down until all incoming zone
                       transfers were finished.

413.   [bug]           Notify could attempt to use the zone database after
                       it had been unloaded. [RT #267]
```

(continues on next page)

(continued from previous page)

```
412.   [bug]           named -v didn't print the version.

411.   [bug]           A typo in the HS A code caused an assertion failure.

410.   [bug]           lwres_gethostbyname() and company set lwres_h_errno
                       to a random value on success.

409.   [bug]           If named was shut down early in the startup
                       process, ns_omapi_shutdown() would attempt to lock
                       an uninitialized mutex. [RT #262]

408.   [bug]           stub zones could leak memory and reference counts if
                       all the masters were unreachable.

407.   [bug]           isc_rwlock_lock() would needlessly block
                       readers when it reached the read quota even
                       if no writers were waiting.

406.   [bug]           Log messages were occasionally lost or corrupted
                       due to a race condition in isc_log_doit().

405.   [func]          Add support for selective forwarding (forward zones)

404.   [bug]           The request library didn't completely work with IPv6.

403.   [bug]           "host" did not use the search list.

402.   [bug]           Treat undefined acls as errors, rather than
                       warning and then later throwing an assertion.
                       [RT #252]

401.   [func]          Added simple database API.

400.   [bug]           SIG(0) signing and verifying was done incorrectly.
                       [RT #249]
```

```
399.   [bug]           When reloading the server with a config file
                       containing a syntax error, it could catch an
                       assertion failure trying to perform zone
                       maintenance on, or sending notifies from,
                       tentatively created zones whose views were
                       never fully configured and lacked an address
                       database and request manager.

398.   [bug]           "dig" sometimes caught an assertion failure when
                       using TSIG, depending on the key length.

397.   [func]          Added utility functions dns_view_gettsig() and
                       dns_view_getpeertsig().

396.   [doc]           There is now a man page for "nsupdate"
```

(continues on next page)

(continued from previous page)

```
                        in doc/man/bin/nsupdate.8.

395.    [bug]           nslookup printed incorrect RR type mnemonics
                        for RRs of type >= 21 [RT #237].

394.    [bug]           Current name was not propagated via $INCLUDE.

393.    [func]          Initial answer while loading (awl) support.
                        Entry points: dns_master_loadfileinc(),
                        dns_master_loadstreaminc(), dns_master_loadbufferinc().
                        Note: calls to dns_master_load*inc() should be rate
                        be rate limited so as to not use up all file
                        descriptors.

392.    [func]          Add ISC_R_FAMILYNOSUPPORT.  Returned when OS does
                        not support the given address family requested.

391.    [clarity]       ISC_R_FAMILY -> ISC_R_FAMILYMISMATCH.

390.    [func]          The function dns_zone_setdbtype() now takes
                        an argc/argv style vector of words and sets
                        both the zone database type and its arguments,
                        making the functions dns_zone_adddbarg()
                        and dns_zone_cleardbargs() unnecessary.

389.    [bug]           Attempting to send a request over IPv6 using
                        dns_request_create() on a system without IPv6
                        support caused an assertion failure [RT #235].

388.    [func]          dig and host can now do reverse ipv6 lookups.

387.    [func]          Add dns_byaddr_createptrname(), which converts
                        an address into the name used by a PTR query.

386.    [bug]           Missing strdup() of ACL name caused random
                        ACL matching failures [RT #228].

385.    [cleanup]       Removed functions dns_zone_equal(), dns_zone_print(),
                        and dns_zt_print().

384.    [bug]           nsupdate was incorrectly limiting TTLs to 65535 instead
                        of 2147483647.

383.    [func]          When writing a master file, print the SOA and NS
                        records (and their SIGs) before other records.

382.    [bug]           named -u failed on many Linux systems where the
                        libc provided kernel headers do not match
                        the current kernel.

381.    [bug]           Check for IPV6_RECVPKTINFO and use it instead of
                        IPV6_PKTINFO if found. [RT #229]
```

(continues on next page)

(continued from previous page)

```
380.   [bug]           nsupdate didn't work with IPv6.

379.   [func]          New library function isc_sockaddr_anyofpf().

378.   [func]          named and lwresd will log the command line arguments
                       they were started with in the "starting ..." message.

377.   [bug]           When additional data lookups were refused due to
                       "allow-query", the databases were still being
                       attached causing reference leaks.

376.   [bug]           The server should always use good entropy when
                       performing cryptographic functions needing entropy.

375.   [bug]           Per-zone "allow-query" did not properly override the
                       view/global one for CNAME targets and additional
                       data [RT #220].

374.   [bug]           SOA in authoritative negative responses had wrong TTL.

373.   [func]          nslookup is now installed by "make install".

372.   [bug]           Deal with Microsoft DNS servers appending two bytes of
                       garbage to zone transfer requests.

371.   [bug]           At high debug levels, doing an outgoing zone transfer
                       of a very large RRset could cause an assertion failure
                       during logging.

370.   [bug]           The error messages for roll-forward failures were
                       overly terse.

369.   [func]          Support new named.conf options, view and zone
                       statements:

                               max-retry-time, min-retry-time,
                               max-refresh-time, min-refresh-time.

368.   [func]          Restructure the internal ".bind" view so that more
                       zones can be added to it.

367.   [bug]           Allow proper selection of server on nslookup command
                       line.

366.   [func]          Allow use of '-' batch file in dig for stdin.

365.   [bug]           nsupdate -k leaked memory.

364.   [func]          Added additional-from-{cache,auth}

363.   [placeholder]
```

(continues on next page)

(continued from previous page)

```
362.   [bug]           rndc no longer aborts if the configuration file is
                       missing an options statement. [RT #209]

361.   [func]          When the RBT find or chain functions set the name and
                       origin for a node that stores the root label
                       the name is now set to an empty name, instead of ".",
                       to simplify later use of the name and origin by
                       dns_name_concatenate(), dns_name_totext() or
                       dns_name_format().

360.   [func]          dns_name_totext() and dns_name_format() now allow
                       an empty name to be passed, which is formatted as "@".

359.   [bug]           dnssec-signzone occasionally signed glue records.

358.   [cleanup]       Rename the intermediate files used by the dnssec
                       programs.

357.   [bug]           The zone file parser crashed if the argument
                       to $INCLUDE was a quoted string.

356.   [cleanup]       isc_task_send no longer requires event->sender to
                       be non-null.

355.   [func]          Added isc_dir_createunique(), similar to mkdtemp().

354.   [doc]           Man pages for the dnssec tools are now included in
                       the distribution, in doc/man/dnssec.

353.   [bug]           double increment in lwres/gethost.c:copytobuf().
                       [RT #187]

352.   [bug]           Race condition in dns_client_t startup could cause
                       an assertion failure.

351.   [bug]           Constructing a response with rcode SERVFAIL to a TSIG
                       signed query could crash the server.

350.   [bug]           Also-notify lists specified in the global options
                       block were not correctly reference counted, causing
                       a memory leak.

349.   [bug]           Processing a query with the CD bit set now works
                       as expected.

348.   [func]          New boolean named.conf options 'additional-from-auth'
                       and 'additional-from-cache' now supported in view and
                       global options statement.

347.   [bug]           Don't crash if an argument is left off options in dig.
```

(continues on next page)

(continued from previous page)

```
346.   [placeholder]

345.   [bug]           Large-scale changes/cleanups to dig:
                       * Significantly improve structure handling
                       * Don't pre-load entire batch files
                       * Add name/rr counting/limiting
                       * Fix SIGINT handling
                       * Shorten timeouts to match v8's behavior

344.   [bug]           When shutting down, lwresd sometimes tried
                       to shut down its client tasks twice,
                       triggering an assertion.

343.   [bug]           Although zone maintenance SOA queries and
                       notify requests were signed with TSIG keys
                       when configured for the server in case,
                       the TSIG was not verified on the response.

342.   [bug]           The wrong name was being passed to
                       dns_name_dup() when generating a TSIG
                       key using TKEY.

341.   [func]          Support 'key' clause in named.conf zone masters
                       statement to allow authentication via TSIG keys:

                               masters {
                                       10.0.0.1 port 5353 key "foo";
                                       10.0.0.2 ;
                               };

340.   [bug]           The top-level COPYRIGHT file was missing from
                       the distribution.

339.   [bug]           DNSSEC validation of the response to an ANY
                       query at a name with a CNAME RR in a secure
                       zone triggered an assertion failure.

338.   [bug]           lwresd logged to syslog as named, not lwresd.

337.   [bug]           "dig" did not recognize "nsap-ptr" as an RR type
                       on the command line.

336.   [bug]           "dig -f" used 64 k of memory for each line in
                       the file.  It now uses much less, though still
                       proportionally to the file size.

335.   [bug]           named would occasionally attempt recursion when
                       it was disallowed or undesired.

334.   [func]          Added hmac-md5 to libisc.

333.   [bug]           The resolver incorrectly accepted referrals to
```

(continues on next page)

(continued from previous page)

```
                      domains that were not parents of the query name,
                      causing assertion failures.

332.  [func]          New function dns_name_reset().

331.  [bug]           Only log "recursion denied" if RD is set. [RT #178]

330.  [bug]           Many debugging messages were partially formatted
                      even when debugging was turned off, causing a
                      significant decrease in query performance.

329.  [func]          omapi_auth_register() now takes a size_t argument for
                      the length of a key's secret data.  Previously
                      OMAPI only stored secrets up to the first NUL byte.

328.  [func]          Added isc_base64_decodestring().

327.  [bug]           rndc.conf parser wasn't correctly recognizing an IP
                      address where a host specification was required.

326.  [func]          'keys' in an 'inet' control statement is now
                      required and must have at least one item in it.
                      A "not supported" warning is now issued if a 'unix'
                      control channel is defined.

325.  [bug]           isc_lex_gettoken was processing octal strings when
                      ISC_LEXOPT_CNUMBER was not set.

324.  [func]          In the resolver, turn EDNS0 off if there is no
                      response after a number of retransmissions.
                      This is to allow queries some chance of succeeding
                      even if all the authoritative servers of a zone
                      silently discard EDNS0 requests instead of
                      sending an error response like they ought to.

323.  [bug]           dns_rbt_findname() did not ignore empty rbt nodes.
                      Because of this, servers authoritative for a parent
                      and grandchild zone but not authoritative for the
                      intervening child zone did not correctly issue
                      referrals to the servers of the child zone.

322.  [bug]           Queries for KEY RRs are now sent to the parent
                      server before the authoritative one, making
                      DNSSEC insecurity proofs work in many cases
                      where they previously didn't.

321.  [bug]           When synthesizing a CNAME RR for a DNAME
                      response, query_addcname() failed to initialize
                      the type and class of the CNAME dns_rdata_t,
                      causing random failures.

320.  [func]          Multiple rndc changes: parses an rndc.conf file,
```

(continues on next page)

(continued from previous page)

```
                      uses authentication to talk to named, command
                      line syntax changed.  This will all be described
                      in the ARM.

319.  [func]          The named.conf "controls" statement is now used
                      to configure the OMAPI command channel.

318.  [func]          dns_c_ndcctx_destroy() could never return anything
                      except ISC_R_SUCCESS; made it have void return instead.

317.  [func]          Use callbacks from libomapi to determine if a
                      new connection is valid, and if a key requested
                      to be used with that connection is valid.

316.  [bug]           Generate a warning if we detect an unexpected <eof>
                      but treat as <eol><eof>.

315.  [bug]           Handle non-empty blanks lines. [RT #163]

314.  [func]          The named.conf controls statement can now have
                      more than one key specified for the inet clause.

313.  [bug]           When parsing resolv.conf, don't terminate on an
                      error.  Instead, parse as much as possible, but
                      still return an error if one was found.

312.  [bug]           Increase the number of allowed elements in the
                      resolv.conf search path from 6 to 8.  If there
                      are more than this, ignore the remainder rather
                      than returning a failure in lwres_conf_parse.

311.  [bug]           lwres_conf_parse failed when the first line of
                      resolv.conf was empty or a comment.

310.  [func]          Changes to named.conf "controls" statement (inet
                      subtype only)

                        - support "keys" clause

                              controls {
                                 inet * port 1024
                                      allow { any; } keys { "foo"; }
                              }

                        - allow "port xxx" to be left out of statement,
                          in which case it defaults to omapi's default port
                          of 953.

309.  [bug]           When sending a referral, the server did not look
                      for name server addresses as glue in the zone
                      holding the NS RRset in the case where this zone
                      was not the same as the one where it looked for
```

(continues on next page)

(continued from previous page)

```
                      name server addresses as authoritative data.

308.   [bug]          Treat a SOA record not at top of zone as an error
                      when loading a zone. [RT #154]

307.   [bug]          When canceling a query, the resolver didn't check for
                      isc_socket_sendto() calls that did not yet have their
                      completion events posted, so it could (rarely) end up
                      destroying the query context and then want to use
                      it again when the send event posted, triggering an
                      assertion as it tried to cancel an already-canceled
                      query.  [RT #77]

306.   [bug]          Reading HMAC-MD5 private key files didn't work.

305.   [bug]          When reloading the server with a config file
                      containing a syntax error, it could catch an
                      assertion failure trying to perform zone
                      maintenance on tentatively created zones whose
                      views were never fully configured and lacked
                      an address database.

304.   [bug]          If more than LWRES_CONFMAXNAMESERVERS servers
                      are listed in resolv.conf, silently ignore them
                      instead of returning failure.

303.   [bug]          Add additional sanity checks to differentiate a AXFR
                      response vs a IXFR response. [RT #157]

302.   [bug]          In dig, host, and nslookup, MXNAME should be large
                      enough to hold any legal domain name in presentation
                      format + terminating NULL.

301.   [bug]          Uninitialized pointer in host:printmessage(). [RT #159]

300.   [bug]          Using both <isc/net.h> and <lwres/net.h> didn't work
                      on platforms lacking IPv6 because each included their
                      own ipv6 header file for the missing definitions.  Now
                      each library's ipv6.h defines the wrapper symbol of
                      the other (ISC_IPV6_H and LWRES_IPV6_H).

299.   [cleanup]      Get the user and group information before changing the
                      root directory, so the administrator does not need to
                      keep a copy of the user and group databases in the
                      chroot'ed environment.  Suggested by Hakan Olsson.

298.   [bug]          A mutex deadlock occurred during shutdown of the
                      interface manager under certain conditions.
                      Digital Unix systems were the most affected.

297.   [bug]          Specifying a key name that wasn't fully qualified
                      in certain parts of the config file could cause
```

(continues on next page)

(continued from previous page)

```
                      an assertion failure.

296.   [bug]          "make install" from a separate build directory
                      failed unless configure had been run in the source
                      directory, too.

295.   [bug]          When invoked with type==CNAME and a message
                      not constructed by dns_message_parse(),
                      dns_message_findname() failed to find anything
                      due to checking for attribute bits that are set
                      only in dns_message_parse().  This caused an
                      infinite loop when constructing the response to
                      an ANY query at a CNAME in a secure zone.

294.   [bug]          If we run out of space in while processing glue
                      when reading a master file and commit "current name"
                      reverts to "name_current" instead of staying as
                      "name_glue".

293.   [port]         Add support for FreeBSD 4.0 system tests.

292.   [bug]          Due to problems with the way some operating systems
                      handle simultaneous listening on IPv4 and IPv6
                      addresses, the server no longer listens on IPv6
                      addresses by default.  To revert to the previous
                      behavior, specify "listen-on-v6 { any; };" in
                      the config file.

291.   [func]         Caching servers no longer send outgoing queries
                      over TCP just because the incoming recursive query
                      was a TCP one.

290.   [cleanup]      +twiddle option to dig (for testing only) removed.

289.   [cleanup]      dig is now installed in $bindir instead of $sbindir.
                      host is now installed in $bindir.  (Be sure to remove
                      any $sbindir/dig from a previous release.)

288.   [func]         rndc is now installed by "make install" into $sbindir.

287.   [bug]          rndc now works again as "rndc 127.1 reload" (for
                      only that task).  Parsing its configuration file and
                      using digital signatures for authentication has been
                      disabled until named supports the "controls" statement,
                      post-9.0.0.

286.   [bug]          On Solaris 2, when named inherited a signal state
                      where SIGHUP had the SIG_IGN action, SIGHUP would
                      be ignored rather than causing the server to reload
                      its configuration.

285.   [bug]          A change made to the dst API for beta4 inadvertently
```

(continues on next page)

(continued from previous page)

```
                      broke OMAPI's creation of a dst key from an incoming
                      message, causing an assertion to be triggered.  Fixed.

284.  [func]          The DNSSEC key generation and signing tools now
                      generate randomness from keyboard input on systems
                      that lack /dev/random.

283.  [cleanup]       The 'lwresd' program is now a link to 'named'.

282.  [bug]           The lexer now returns ISC_R_RANGE if parsed integer is
                      too big for an unsigned long.

281.  [bug]           Fixed list of recognized config file category names.

280.  [func]          Add isc-config.sh, which can be used to more
                      easily build applications that link with
                      our libraries.

279.  [bug]           Private omapi function symbols shared between
                      two or more files in libomapi.a were not namespace
                      protected using the ISC convention of starting with
                      the library name and two underscores ("omapi__"...)

278.  [bug]           bin/named/logconf.c:category_fromconf() didn't take
                      note of when isc_log_categorybyname() wasn't able
                      to find the category name and would then apply the
                      channel list of the unknown category to all categories.

277.  [bug]           isc_log_categorybyname() and isc_log_modulebyname()
                      would fail to find the first member of any category
                      or module array apart from the internal defaults.
                      Thus, for example, the "notify" category was improperly
                      configured by named.

276.  [bug]           dig now supports maximum sized TCP messages.

275.  [bug]           The definition of lwres_gai_strerror() was missing
                      the lwres_ prefix.

274.  [bug]           TSIG AXFR verify failed when talking to a BIND 8
                      server.

273.  [func]          The default for the 'transfer-format' option is
                      now 'many-answers'.  This will break zone transfers
                      to BIND 4.9.5 and older unless there is an explicit
                      'one-answer' configuration.

272.  [bug]           The sending of large TCP responses was canceled
                      in mid-transmission due to a race condition
                      caused by the failure to set the client object's
                      "newstate" variable correctly when transitioning
                      to the "working" state.
```

(continues on next page)

```
271.   [func]          Attempt to probe the number of cpus in named
                       if unspecified rather than defaulting to 1.

270.   [func]          Allow maximum sized TCP answers.

269.   [bug]           Failed DNSSEC validations could cause an assertion
                       failure by causing clone_results() to be called with
                       with hevent->node == NULL.

268.   [doc]           A plain text version of the Administrator
                       Reference Manual is now included in the distribution,
                       as doc/arm/Bv9ARM.txt.

267.   [func]          Nsupdate is now provided in the distribution.

266.   [bug]           zone.c:save_nsrrset() node was not initialized.

265.   [bug]           dns_request_create() now works for TCP.

264.   [func]          Dispatch can not take TCP sockets in connecting
                       state.  Set DNS_DISPATCHATTR_CONNECTED when calling
                       dns_dispatch_createtcp() for connected TCP sockets
                       or call dns_dispatch_starttcp() when the socket is
                       connected.

263.   [func]          New logging channel type 'stderr'

                               channel some-name {
                                       stderr;
                                       severity error;
                               }

262.   [bug]           'master' was not initialized in zone.c:stub_callback().

261.   [func]          Add dns_zone_markdirty().

260.   [bug]           Running named as a non-root user failed on Linux
                       kernels new enough to support retaining capabilities
                       after setuid().

259.   [func]          New random-device and random-seed-file statements
                       for global options block of named.conf. Both accept
                       a single string argument.

258.   [bug]           Fixed printing of lwres_addr_t.address field.

257.   [bug]           The server detached the last zone manager reference
                       too early, while it could still be in use by queries.
                       This manifested itself as assertion failures during the
                       shutdown process for busy name servers. [RT #133]

256.   [func]          isc_ratelimiter_t now has attach/detach semantics, and
```

(continues on next page)

(continued from previous page)

```
                      isc_ratelimiter_shutdown guarantees that the rate
                      limiter is detached from its task.

255.  [func]          New function dns_zonemgr_attach().

254.  [bug]           Suppress "query denied" messages on additional data
                      lookups.
```

```
      --- 9.0.0b4 released ---

253.  [func]          resolv.conf parser now recognizes ';' and '#' as
                      comments (anywhere in line, not just as the beginning).

252.  [bug]           resolv.conf parser mishandled masks on sortlists.
                      It also aborted when an unrecognized keyword was seen,
                      now it silently ignores the entire line.

251.  [bug]           lwresd caught an assertion failure on startup.

250.  [bug]           fixed handling of size+unit when value would be too
                      large for internal representation.

249.  [cleanup]       max-cache-size config option now takes a size-spec
                      like 'datasize', except 'default' is not allowed.

248.  [bug]           global lame-ttl option was not being printed when
                      config structures were written out.

247.  [cleanup]       Rename cache-size config option to max-cache-size.

246.  [func]          Rename global option cachesize to cache-size and
                      add corresponding option to view statement.

245.  [bug]           If an uncompressed name will take more than 255
                      bytes and the buffer is sufficiently long,
                      dns_name_fromwire should return DNS_R_FORMERR,
                      not ISC_R_NOSPACE.  This bug caused cause the
                      server to catch an assertion failure when it
                      received a query for a name longer than 255
                      bytes.

244.  [bug]           empty named.conf file and empty options statement are
                      now parsed properly.

243.  [func]          new cachesize option for named.conf

242.  [cleanup]       fixed incorrect warning about auth-nxdomain usage.

241.  [cleanup]       nscount and soacount have been removed from the
                      dns_master_*() argument lists.

240.  [func]          databases now come in three flavours: zone, cache
```

(continues on next page)

(continued from previous page)

```
                        and stub.

239.    [func]          If ISC_MEM_DEBUG is enabled, the variable
                        isc_mem_debugging controls whether messages
                        are printed or not.

238.    [cleanup]       A few more compilation warnings have been quieted:
                        + missing sigwait prototype on BSD/OS 4.0/4.0.1.
                        + PTHREAD_ONCE_INIT unbraced initializer warnings on
                                Solaris 2.8.
                        + IN6ADDR_ANY_INIT unbraced initializer warnings on
                                BSD/OS 4.*, Linux and Solaris 2.8.

237.    [bug]           If connect() returned ENOBUFS when the resolver was
                        initiating a TCP query, the socket didn't get
                        destroyed, and the server did not shut down cleanly.

236.    [func]          Added new listen-on-v6 config file statement.

235.    [func]          Consider it a config file error if a listen-on
                        statement has an IPv6 address in it, or a
                        listen-on-v6 statement has an IPv4 address in it.

234.    [bug]           Allow a trusted-key's first field (domain-name) be
                        either a quoted or an unquoted string, instead of
                        requiring a quoted string.

233.    [cleanup]       Convert all config structure integer values to unsigned
                        integer (isc_uint32_t) to match grammar.

232.    [bug]           Allow slave zones to not have a file.

231.    [func]          Support new 'port' clause in config file options
                        section. Causes 'listen-on', 'masters' and
                        'also-notify' statements to use its value instead of
                        default (53).

230.    [func]          Replace the dst sign/verify API with a cleaner one.

229.    [func]          Support config file sig-validity-interval statement
                        in options, views and zone statements (master
                        zones only).

228.    [cleanup]       Logging messages in config module stripped of
                        trailing period.

227.    [cleanup]       The enumerated identifiers dns_rdataclass_*,
                        dns_rcode_*, dns_opcode_*, and dns_trust_* are
                        also now cast to their appropriate types, as with
                        dns_rdatatype_* in item number 225 below.

226.    [func]          dns_name_totext() now always prints the root name as
```

(continues on next page)

(continued from previous page)

```
                        '.', even when omit_final_dot is true.

225.    [cleanup]       The enumerated dns_rdatatype_* identifiers are now
                        cast to dns_rdatatype_t via macros of their same name
                        so that they are of the proper integral type wherever
                        a dns_rdatatype_t is needed.

224.    [cleanup]       The entire project builds cleanly with gcc's
                        -Wcast-qual and -Wwrite-strings warnings enabled,
                        which is now the default when using gcc.  (Warnings
                        from confparser.c, because of yacc's code, are
                        unfortunately to be expected.)

223.    [func]          Several functions were re-prototyped to qualify one
                        or more of their arguments with "const".  Similarly,
                        several functions that return pointers now have
                        those pointers qualified with const.

222.    [bug]           The global 'also-notify' option was ignored.

221.    [bug]           An uninitialized variable was sometimes passed to
                        dns_rdata_freestruct() when loading a zone, causing
                        an assertion failure.

220.    [cleanup]       Set the default outgoing port in the view, and
                        set it in sockaddrs returned from the ADB.
                        [31-May-2000 explorer]

219.    [bug]           Signed truncated messages more correctly follow
                        the respective specs.

218.    [func]          When an rdataset is signed, its ttl is normalized
                        based on the signature validity period.

217.    [func]          Also-notify and trusted-keys can now be used in
                        the 'view' statement.

216.    [func]          The 'max-cache-ttl' and 'max-ncache-ttl' options
                        now work.

215.    [bug]           Failures at certain points in request processing
                        could cause the assertion INSIST(client->lockview
                        == NULL) to be triggered.

214.    [func]          New public function isc_netaddr_format(), for
                        formatting network addresses in log messages.

213.    [bug]           Don't leak memory when reloading the zone if
                        an update-policy clause was present in the old zone.

212.    [func]          Added dns_message_get/settsigkey, to make TSIG
                        key management reasonable.
```

(continues on next page)

(continued from previous page)

```
211.   [func]          The 'key' and 'server' statements can now occur
                       inside 'view' statements.

210.   [bug]           The 'allow-transfer' option was ignored for slave
                       zones, and the 'transfers-per-ns' option was
                       was ignored for all zones.

209.   [cleanup]       Upgraded openssl files to new version 0.9.5a

208.   [func]          Added ISC_OFFSET_MAXIMUM for the maximum value
                       of an isc_offset_t.

207.   [func]          The dnssec tools properly use the logging subsystem.

206.   [cleanup]       dst now stores the key name as a dns_name_t, not
                       a char *.

205.   [cleanup]       On IRIX, turn off the mostly harmless warnings 1692
                       ("prototyped function redeclared without prototype")
                       and 1552 ("variable ... set but not used") when
                       compiling in the lib/dns/sec/{dnssafe,openssl}
                       directories, which contain code imported from outside
                       sources.

204.   [cleanup]       On HP/UX, pass +vnocompatwarnings to the linker
                       to quiet the warnings that "The linked output may not
                       run on a PA 1.x system."

203.   [func]          notify and zone soa queries are now tsig signed when
                       appropriate.

202.   [func]          isc_lex_getsourceline() changed from returning int
                       to returning unsigned long, the type of its underlying
                       counter.

201.   [cleanup]       Removed the test/sdig program, it has been
                       replaced by bin/dig/dig.
```

```
       --- 9.0.0b3 released ---

200.   [bug]           Failures in sending query responses to clients
                       (e.g., running out of network buffers) were
                       not logged.

199.   [bug]           isc_heap_delete() sometimes violated the heap
                       invariant, causing timer events not to be posted
                       when due.

198.   [func]          Dispatch managers hold memory pools which
                       any managed dispatcher may use.  This allows
                       us to avoid dipping into the memory context for
```

(continues on next page)

(continued from previous page)

```
                      most allocations. [19-May-2000 explorer]

197.   [bug]          When an incoming AXFR or IXFR completes, the
                      zone's internal state is refreshed from the
                      SOA data. [19-May-2000 explorer]

196.   [func]         Dispatchers can be shared easily between views
                      and/or interfaces. [19-May-2000 explorer]

195.   [bug]          Including the NXT record of the root domain
                      in a negative response caused an assertion
                      failure.

194.   [doc]          The PDF version of the Administrator's Reference
                      Manual is no longer included in the ISC BIND9
                      distribution.

193.   [func]         changed dst_key_free() prototype.

192.   [bug]          Zone configuration validation is now done at end
                      of config file parsing, and before loading
                      callbacks.

191.   [func]         Patched to compile on UnixWare 7.x.  This platform
                      is not directly supported by the ISC.

190.   [cleanup]      The DNSSEC tools have been moved to a separate
                      directory dnssec/ and given the following new,
                      more descriptive names:

                            dnssec-keygen
                            dnssec-signzone
                            dnssec-signkey
                            dnssec-makekeyset

                      Their command line arguments have also been changed to
                      be more consistent.  dnssec-keygen now prints the
                      name of the generated key files (sans extension)
                      on standard output to simplify its use in automated
                      scripts.

189.   [func]         isc_time_secondsastimet(), a new function, will ensure
                      that the number of seconds in an isc_time_t does not
                      exceed the range of a time_t, or return ISC_R_RANGE.
                      Similarly, isc_time_now(), isc_time_nowplusinterval(),
                      isc_time_add() and isc_time_subtract() now check the
                      range for overflow/underflow.  In the case of
                      isc_time_subtract, this changed a calling requirement
                      (ie, something that could generate an assertion)
                      into merely a condition that returns an error result.
                      isc_time_add() and isc_time_subtract() were void-
                      valued before but now return isc_result_t.
```

(continues on next page)

(continued from previous page)

```
188.   [func]          Log a warning message when an incoming zone transfer
                       contains out-of-zone data.

187.   [func]          isc_ratelimiter_enqueue() has an additional argument
                       'task'.

186.   [func]          dns_request_getresponse() has an additional argument
                       'preserve_order'.

185.   [bug]           Fixed up handling of ISC_MEMCLUSTER_LEGACY.  Several
                       public functions did not have an isc__ prefix, and
                       referred to functions that had previously been
                       renamed.

184.   [cleanup]       Variables/functions which began with two leading
                       underscores were made to conform to the ANSI/ISO
                       standard, which says that such names are reserved.

183.   [func]          ISC_LOG_PRINTTAG option for log channels.  Useful
                       for logging the program name or other identifier.

182.   [cleanup]       New command-line parameters for dnssec tools

181.   [func]          Added dst_key_buildfilename and dst_key_parsefilename

180.   [func]          New isc_result_t ISC_R_RANGE.  Supersedes DNS_R_RANGE.

179.   [func]          options named.conf statement *must* now come
                       before any zone or view statements.

178.   [func]          Post-load of named.conf check verifies a slave zone
                       has non-empty list of masters defined.

177.   [func]          New per-zone boolean:

                               enable-zone yes | no ;

                       intended to let a zone be disabled without having
                       to comment out the entire zone statement.

176.   [func]          New global and per-view option:

                               max-cache-ttl number

175.   [func]          New global and per-view option:

                               additional-data internal | minimal | maximal;

174.   [func]          New public function isc_sockaddr_format(), for
                       formatting socket addresses in log messages.
```

(continues on next page)

(continued from previous page)

```
173.   [func]          Keep a queue of zones waiting for zone transfer
                       quota so that a new transfer can be dispatched
                       immediately whenever quota becomes available.

172.   [bug]           $TTL directive was sometimes missing from dumped
                       master files because totext_ctx_init() failed to
                       initialize ctx->current_ttl_valid.

171.   [cleanup]       On NetBSD systems, the mit-pthreads or
                       unproven-pthreads library is now always used
                       unless --with-ptl2 is explicitly specified on
                       the configure command line.  The
                       --with-mit-pthreads option is no longer needed
                       and has been removed.

170.   [cleanup]       Remove inter server consistency checks from zone,
                       these should return as a separate module in 9.1.
                       dns_zone_checkservers(), dns_zone_checkparents(),
                       dns_zone_checkchildren(), dns_zone_checkglue().

                       Remove dns_zone_setadb(), dns_zone_setresolver(),
                       dns_zone_setrequestmgr() these should now be found
                       via the view.

169.   [func]          ratelimiter can now process N events per interval.

168.   [bug]           include statements in named.conf caused syntax errors
                       due to not consuming the semicolon ending the include
                       statement before switching input streams.

167.   [bug]           Make lack of masters for a slave zone a soft error.

166.   [bug]           Keygen was overwriting existing keys if key_id
                       conflicted, now it will retry, and non-null keys
                       with key_id == 0 are not generated anymore.  Key
                       was not able to generate NOAUTHCONF DSA key,
                       increased RSA key size to 2048 bits.

165.   [cleanup]       Silence "end-of-loop condition not reached" warnings
                       from Solaris compiler.

164.   [func]          Added functions isc_stdio_open(), isc_stdio_close(),
                       isc_stdio_seek(), isc_stdio_read(), isc_stdio_write(),
                       isc_stdio_flush(), isc_stdio_sync(), isc_file_remove()
                       to encapsulate nonportable usage of errno and sync.

163.   [func]          Added result codes ISC_R_FILENOTFOUND and
                       ISC_R_FILEEXISTS.

162.   [bug]           Ensure proper range for arguments to ctype.h functions.

161.   [cleanup]       error in yyparse prototype that only HPUX caught.
```

(continues on next page)

(continued from previous page)

```
160.   [cleanup]       getnet*() are not going to be implemented at this
                       stage.

159.   [func]          Redefinition of config file elements is now an
                       error (instead of a warning).

158.   [bug]           Log channel and category list copy routines
                       weren't assigning properly to output parameter.

157.   [port]          Fix missing prototype for getopt().

156.   [func]          Support new 'database' statement in zone.

                               database "quoted-string";

155.   [bug]           ns_notify_start() was not detaching the found zone.

154.   [func]          The signer now logs libdns warnings to stderr even when
                       not verbose, and in a nicer format.

153.   [func]          dns_rdata_tostruct() 'mctx' is now optional.  If 'mctx'
                       is NULL then you need to preserve the 'rdata' until
                       you have finished using the structure as there may be
                       references to the associated memory.  If 'mctx' is
                       non-NULL it is guaranteed that there are no references
                       to memory associated with 'rdata'.

                       dns_rdata_freestruct() must be called if 'mctx' was
                       non-NULL and may safely be called if 'mctx' was NULL.

152.   [bug]           keygen dumped core if domain name argument was omitted
                       from command line.

151.   [func]          Support 'disabled' statement in zone config (causes
                       zone to be parsed and then ignored). Currently must
                       come after the 'type' clause.

150.   [func]          Support optional ports in masters and also-notify
                       statements:

                               masters [ port xxx ] { y.y.y.y [ port zzz ] ; }

149.   [cleanup]       Removed unused argument 'olist' from
                       dns_c_view_unsetordering().

148.   [cleanup]       Stop issuing some warnings about some configuration
                       file statements that were not implemented, but now are.

147.   [bug]           Changed yacc union size to be smaller for yaccs that
                       put yacc-stack on the real stack.
```

(continues on next page)

(continued from previous page)

```
146.   [cleanup]       More general redundant header file cleanup.  Rather
                       than continuing to itemize every header which changed,
                       this changelog entry just notes that if a header file
                       did not need another header file that it was including
                       in order to provide its advertised functionality, the
                       inclusion of the other header file was removed.  See
                       util/check-includes for how this was tested.

145.   [cleanup]       Added <isc/lang.h> and ISC_LANG_BEGINDECLS/
                       ISC_LANG_ENDDECLS to header files that had function
                       prototypes, and removed it from those that did not.

144.   [cleanup]       libdns header files too numerous to name were made
                       to conform to the same style for multiple inclusion
                       protection.

143.   [func]          Added function dns_rdatatype_isknown().

142.   [cleanup]       <isc/stdtime.h> does not need <time.h> or
                       <isc/result.h>.

141.   [bug]           Corrupt requests with multiple questions could
                       cause an assertion failure.

140.   [cleanup]       <isc/time.h> does not need <time.h> or <isc/result.h>.

139.   [cleanup]       <isc/net.h> now includes <isc/types.h> instead of
                       <isc/int.h> and <isc/result.h>.

138.   [cleanup]       isc_strtouq moved from str.[ch] to string.[ch] and
                       renamed isc_string_touint64.  isc_strsep moved from
                       strsep.c to string.c and renamed isc_string_separate.

137.   [cleanup]       <isc/commandline.h>, <isc/mem.h>, <isc/print.h>
                       <isc/serial.h>, <isc/string.h> and <isc/offset.h>
                       made to conform to the same style for multiple
                       inclusion protection.

136.   [cleanup]       <isc/commandline.h>, <isc/interfaceiter.h>,
                       <isc/net.h> and Win32's <isc/thread.h> needed
                       ISC_LANG_BEGINDECLS/ISC_LANG_ENDDECLS.

135.   [cleanup]       Win32's <isc/condition.h> did not need <isc/result.h>
                       or <isc/boolean.h>, now uses <isc/types.h> in place
                       of <isc/time.h>, and needed ISC_LANG_BEGINDECLS
                       and ISC_LANG_ENDDECLS.

134.   [cleanup]       <isc/dir.h> does not need <limits.h>.

133.   [cleanup]       <isc/ipv6.h> needs <isc/platform.h>.

132.   [cleanup]       <isc/app.h> does not need <isc/task.h>, but does
```

(continues on next page)

(continued from previous page)

```
                      need <isc/eventclass.h>.

131.  [cleanup]       <isc/mutex.h> and <isc/util.h> need <isc/result.h>
                      for ISC_R_* codes used in macros.

130.  [cleanup]       <isc/condition.h> does not need <pthread.h> or
                      <isc/boolean.h>, and now includes <isc/types.h>
                      instead of <isc/time.h>.

129.  [bug]           The 'default_debug' log channel was not set up when
                      'category default' was present in the config file

128.  [cleanup]       <isc/dir.h> had ISC_LANG_BEGINDECLS instead of
                      ISC_LANG_ENDDECLS at end of header.

127.  [cleanup]       The contracts for the comparison routines
                      dns_name_fullcompare(), dns_name_compare(),
                      dns_name_rdatacompare(), and dns_rdata_compare() now
                      specify that the order value returned is < 0, 0, or > 0
                      instead of -1, 0, or 1.

126.  [cleanup]       <isc/quota.h> and <isc/taskpool.h> need <isc/lang.h>.

125.  [cleanup]       <isc/eventclass.h>, <isc/ipv6.h>, <isc/magic.h>,
                      <isc/mutex.h>, <isc/once.h>, <isc/region.h>, and
                      <isc/resultclass.h> do not need <isc/lang.h>.

124.  [func]          signer now imports parent's zone key signature
                      and creates null keys/sets zone status bit for
                      children when necessary

123.  [cleanup]       <isc/event.h> does not need <stddef.h>.

122.  [cleanup]       <isc/task.h> does not need <isc/mem.h> or
                      <isc/result.h>.

121.  [cleanup]       <isc/symtab.h> does not need <isc/mem.h> or
                      <isc/result.h>.  Multiple inclusion protection
                      symbol fixed from ISC_SYMBOL_H to ISC_SYMTAB_H.
                      isc_symtab_t moved to <isc/types.h>.

120.  [cleanup]       <isc/socket.h> does not need <isc/boolean.h>,
                      <isc/bufferlist.h>, <isc/task.h>, <isc/mem.h> or
                      <isc/net.h>.

119.  [cleanup]       structure definitions for generic rdata structures do
                      not have _generic_ in their names.

118.  [cleanup]       libdns.a is now namespace-clean, on NetBSD, excepting
                      YACC crust (yyparse, etc) [2000-apr-27 explorer]

117.  [cleanup]       libdns.a changes:
```

(continues on next page)

(continued from previous page)

```
                        dns_zone_clearnotify() and dns_zone_addnotify()
                        are replaced by dns_zone_setnotifyalso().
                        dns_zone_clearmasters() and dns_zone_addmaster()
                        are replaced by dns_zone_setmasters().

116.    [func]          Added <isc/offset.h> for isc_offset_t (aka off_t
                        on Unix systems).

115.    [port]          Shut up the -Wmissing-declarations warning about
                        <stdio.h>'s __sputaux on BSD/OS pre-4.1.

114.    [cleanup]       <isc/sockaddr.h> does not need <isc/buffer.h> or
                        <isc/list.h>.

113.    [func]          Utility programs dig and host added.

112.    [cleanup]       <isc/serial.h> does not need <isc/boolean.h>.

111.    [cleanup]       <isc/rwlock.h> does not need <isc/result.h> or
                        <isc/mutex.h>.

110.    [cleanup]       <isc/result.h> does not need <isc/boolean.h> or
                        <isc/list.h>.

109.    [bug]           "make depend" did nothing for
                        bin/tests/{db,mem,sockaddr,tasks,timers}/.

108.    [cleanup]       DNS_SETBIT/DNS_GETBIT/DNS_CLEARBIT moved from
                        <dns/types.h> to <dns/bit.h> and renamed to
                        DNS_BIT_SET/DNS_BIT_GET/DNS_BIT_CLEAR.

107.    [func]          Add keysigner and keysettool.

106.    [func]          Allow dnssec verifications to ignore the validity
                        period.  Used by several of the dnssec tools.

105.    [doc]           doc/dev/coding.html expanded with other
                        implicit conventions the developers have used.

104.    [bug]           Made compress_add and compress_find static to
                        lib/dns/compress.c.

103.    [func]          libisc buffer API changes for <isc/buffer.h>:
                        Added:
                                isc_buffer_base(b)          (pointer)
                                isc_buffer_current(b)       (pointer)
                                isc_buffer_active(b)        (pointer)
                                isc_buffer_used(b)          (pointer)
                                isc_buffer_length(b)            (int)
                                isc_buffer_usedlength(b)        (int)
                                isc_buffer_consumedlength(b)    (int)
                                isc_buffer_remaininglength(b)   (int)
```

(continues on next page)

(continued from previous page)

```
                                isc_buffer_activelength(b)      (int)
                                isc_buffer_availablelength(b)   (int)
                        Removed:
                                ISC_BUFFER_USEDCOUNT(b)
                                ISC_BUFFER_AVAILABLECOUNT(b)
                                isc_buffer_type(b)
                        Changed names:
                                isc_buffer_used(b, r) ->
                                        isc_buffer_usedregion(b, r)
                                isc_buffer_available(b, r) ->
                                        isc_buffer_available_region(b, r)
                                isc_buffer_consumed(b, r) ->
                                        isc_buffer_consumedregion(b, r)
                                isc_buffer_active(b, r) ->
                                        isc_buffer_activeregion(b, r)
                                isc_buffer_remaining(b, r) ->
                                        isc_buffer_remainingregion(b, r)

                        Buffer types were removed, so the ISC_BUFFERTYPE_*
                        macros are no more, and the type argument to
                        isc_buffer_init and isc_buffer_allocate were removed.
                        isc_buffer_putstr is now void (instead of isc_result_t)
                        and requires that the caller ensure that there
                        is enough available buffer space for the string.

102.    [port]          Correctly detect inet_aton, inet_pton and inet_ptop
                        on BSD/OS 4.1.

101.    [cleanup]       Quieted EGCS warnings from lib/isc/print.c.

100.    [cleanup]       <isc/random.h> does not need <isc/int.h> or
                        <isc/mutex.h>.  isc_random_t moved to <isc/types.h>.

 99.    [cleanup]       Rate limiter now has separate shutdown() and
                        destroy() functions, and it guarantees that all
                        queued events are delivered even in the shutdown case.

 98.    [cleanup]       <isc/print.h> does not need <stdarg.h> or <stddef.h>
                        unless ISC_PLATFORM_NEEDVSNPRINTF is defined.

 97.    [cleanup]       <isc/ondestroy.h> does not need <stddef.h> or
                        <isc/event.h>.

 96.    [cleanup]       <isc/mutex.h> does not need <isc/result.h>.

 95.    [cleanup]       <isc/mutexblock.h> does not need <isc/result.h>.

 94.    [cleanup]       Some installed header files did not compile as C++.

 93.    [cleanup]       <isc/msgcat.h> does not need <isc/result.h>.

 92.    [cleanup]       <isc/mem.h> does not need <stddef.h>, <isc/boolean.h>,
```

(continues on next page)

(continued from previous page)

```
                      or <isc/result.h>.

91.   [cleanup]       <isc/log.h> does not need <sys/types.h> or
                      <isc/result.h>.

90.   [cleanup]       Removed unneeded ISC_LANG_BEGINDECLS/ISC_LANG_ENDDECLS
                      from <named/listenlist.h>.

89.   [cleanup]       <isc/lex.h> does not need <stddef.h>.

88.   [cleanup]       <isc/interfaceiter.h> does not need <isc/result.h> or
                      <isc/mem.h>.  isc_interface_t and isc_interfaceiter_t
                      moved to <isc/types.h>.

87.   [cleanup]       <isc/heap.h> does not need <isc/boolean.h>,
                      <isc/mem.h> or <isc/result.h>.

86.   [cleanup]       isc_bufferlist_t moved from <isc/bufferlist.h> to
                      <isc/types.h>.

85.   [cleanup]       <isc/bufferlist.h> does not need <isc/buffer.h>,
                      <isc/list.h>, <isc/mem.h>, <isc/region.h> or
                      <isc/int.h>.

84.   [func]          allow-query ACL checks now apply to all data
                      added to a response.

83.   [func]          If the server is authoritative for both a
                      delegating zone and its (nonsecure) delegatee, and
                      a query is made for a KEY RR at the top of the
                      delegatee, then the server will look for a KEY
                      in the delegator if it is not found in the delegatee.

82.   [cleanup]       <isc/buffer.h> does not need <isc/list.h>.

81.   [cleanup]       <isc/int.h> and <isc/boolean.h> do not need
                      <isc/lang.h>.

80.   [cleanup]       <isc/print.h> does not need <stdio.h> or <stdlib.h>.

79.   [cleanup]       <dns/callbacks.h> does not need <stdio.h>.

78.   [cleanup]       lwres_conftest renamed to lwresconf_test for
                      consistency with other *_test programs.

77.   [cleanup]       typedef of isc_time_t and isc_interval_t moved from
                      <isc/time.h> to <isc/types.h>.

76.   [cleanup]       Rewrote keygen.

75.   [func]          Don't load a zone if its database file is older
                      than the last time the zone was loaded.
```

(continues on next page)

```
74.   [cleanup]       Removed mktemplate.o and ufile.o from libisc.a,
                      subsumed by file.o.

73.   [func]          New "file" API in libisc, including new function
                      isc_file_getmodtime, isc_mktemplate renamed to
                      isc_file_mktemplate and isc_ufile renamed to
                      isc_file_openunique.  By no means an exhaustive API,
                      it is just what's needed for now.

72.   [func]          DNS_RBTFIND_NOPREDECESSOR and DNS_RBTFIND_NOOPTIONS
                      added for dns_rbt_findnode, the former to disable the
                      setting of the chain to the predecessor, and the
                      latter to make clear when no options are set.

71.   [cleanup]       Made explicit the implicit REQUIREs of
                      isc_time_seconds, isc_time_nanoseconds, and
                      isc_time_subtract.

70.   [func]          isc_time_set() added.

69.   [bug]           The zone object's master and also-notify lists grew
                      longer with each server reload.

68.   [func]          Partial support for SIG(0) on incoming messages.

67.   [performance]   Allow use of alternate (compile-time supplied)
                      OpenSSL libraries/headers.

66.   [func]          Data in authoritative zones should have a trust level
                      beyond secure.

65.   [cleanup]       Removed obsolete typedef of dns_zone_callbackarg_t
                      from <dns/types.h>.

64.   [func]          The RBT, DB, and zone table APIs now allow the
                      caller find the most-enclosing superdomain of
                      a name.

63.   [func]          Generate NOTIFY messages.

62.   [func]          Add UDP refresh support.

61.   [cleanup]       Use single quotes consistently in log messages.

60.   [func]          Catch and disallow singleton types on message
                      parse.

59.   [bug]           Cause net/host unreachable to be a hard error
                      when sending and receiving.

58.   [bug]           bin/named/query.c could sometimes trigger the
                      (client->query.attributes & NS_QUERYATTR_NAMEBUFUSED)
```

(continues on next page)

(continued from previous page)

```
                      == 0 assertion in query_newname().

57.   [func]          Added dns_nxt_typepresent()

56.   [bug]           SIG records were not properly returned in cached
                      negative answers.

55.   [bug]           Responses containing multiple names in the authority
                      section were not negatively cached.

54.   [bug]           If a fetch with sigrdataset==NULL joined one with
                      sigrdataset!=NULL or vice versa, the resolver
                      could catch an assertion or lose signature data,
                      respectively.

53.   [port]          freebsd 4.0: lib/isc/unix/socket.c requires
                      <sys/param.h>.

52.   [bug]           rndc: taskmgr and socketmgr were not initialized
                      to NULL.

51.   [cleanup]       dns/compress.h and dns/zt.h did not need to include
                      dns/rbt.h; it was needed only by compress.c and zt.c.

50.   [func]          RBT deletion no longer requires a valid chain to work,
                      and dns_rbt_deletenode was added.

49.   [func]          Each cache now has its own mctx.

48.   [func]          isc_task_create() no longer takes an mctx.
                      isc_task_mem() has been eliminated.

47.   [func]          A number of modules now use memory context reference
                      counting.

46.   [func]          Memory contexts are now reference counted.
                      Added isc_mem_inuse() and isc_mem_preallocate().
                      Renamed isc_mem_destroy_check() to
                      isc_mem_setdestroycheck().

45.   [bug]           The trusted-key statement incorrectly loaded keys.

44.   [bug]           Don't include authority data if it would force us
                      to unset the AD bit in the message.

43.   [bug]           DNSSEC verification of cached rdatasets was failing.

42.   [cleanup]       Simplified logging of messages with embedded domain
                      names by introducing a new convenience function
                      dns_name_format().

41.   [func]          Use PR_SET_KEEPCAPS on Linux 2.3.99-pre3 and later
```

(continues on next page)

(continued from previous page)

```
                        to allow 'named' to run as a non-root user while
                        retaining the ability to bind() to privileged
                        ports.

40.   [func]            Introduced new logging category "dnssec" and
                        logging module "dns/validator".

39.   [cleanup]         Moved the typedefs for isc_region_t, isc_textregion_t,
                        and isc_lex_t to <isc/types.h>.

38.   [bug]             TSIG signed incoming zone transfers work now.

37.   [bug]             If the first RR in an incoming zone transfer was
                        not an SOA, the server died with an assertion failure
                        instead of just reporting an error.

36.   [cleanup]         Change DNS_R_SUCCESS (and others) to ISC_R_SUCCESS

35.   [performance]     Log messages which are of a level too high to be
                        logged by any channel in the logging configuration
                        will not cause the log mutex to be locked.

34.   [bug]             Recursion was allowed even with 'recursion no'.

33.   [func]            The RBT now maintains a parent pointer at each node.

32.   [cleanup]         bin/lwresd/client.c needs <string.h> for memset()
                        prototype.

31.   [bug]             Use ${LIBTOOL} to compile bin/named/main.@O@.

30.   [func]            config file grammar change to support optional
                        class type for a view.

29.   [func]            support new config file view options:

                                auth-nxdomain recursion query-source
                                query-source-v6 transfer-source
                                transfer-source-v6 max-transfer-time-out
                                max-transfer-idle-out transfer-format
                                request-ixfr provide-ixfr cleaning-interval
                                fetch-glue notify rfc2308-type1 lame-ttl
                                max-ncache-ttl min-roots

28.   [func]            support lame-ttl, min-roots and serial-queries
                        config global options.

27.   [bug]             Only include <netinet6/in6.h> on BSD/OS 4.[01]*.
                        Including it on other platforms (eg, NetBSD) can
                        cause a forced #error from the C preprocessor.

26.   [func]            new match-clients statement in config file view.
```

(continues on next page)

(continued from previous page)

```
25.   [bug]           make install failed to install <isc/log.h> and
                      <isc/ondestroy.h>.

24.   [cleanup]       Eliminate some unnecessary #includes of header
                      files from header files.

23.   [cleanup]       Provide more context in log messages about client
                      requests, using a new function ns_client_log().

22.   [bug]           SIGs weren't returned in the answer section when
                      the query resulted in a fetch.

21.   [port]          Look at STD_CINCLUDES after CINCLUDES during
                      compilation, so additional system include directories
                      can be searched but header files in the bind9 source
                      tree with conflicting names take precedence.  This
                      avoids issues with installed versions of dnssafe and
                      openssl.

20.   [func]          Configuration file post-load validation of zones
                      failed if there were no zones.

19.   [bug]           dns_zone_notifyreceive() failed to unlock the zone
                      lock in certain error cases.

18.   [bug]           Use AC_TRY_LINK rather than AC_TRY_COMPILE in
                      configure.in to check for presence of in6addr_any.

17.   [func]          Do configuration file post-load validation of zones.

16.   [bug]           put quotes around key names on config file
                      output to avoid possible keyword clashes.

15.   [func]          Add dns_name_dupwithoffsets().  This function is
                      improves comparison performance for duped names.

14.   [bug]           free_rbtdb() could have 'put' unallocated memory in
                      an unlikely error path.

13.   [bug]           lib/dns/master.c and lib/dns/xfrin.c didn't ignore
                      out-of-zone data.

12.   [bug]           Fixed possible uninitialized variable error.

11.   [bug]           axfr_rrstream_first() didn't check the result code of
                      db_rr_iterator_first(), possibly causing an assertion
                      to be triggered later.

10.   [bug]           A bug in the code which makes EDNS0 OPT records in
                      bin/named/client.c and lib/dns/resolver.c could
                      trigger an assertion.
```

(continues on next page)

(continued from previous page)

```
9.   [cleanup]       replaced bit-setting code in confctx.c and replaced
                     repeated code with macro calls.

8.   [bug]           Shutdown of incoming zone transfer accessed
                     freed memory.

7.   [cleanup]       removed 'listen-on' from view statement.

6.   [bug]           quote RR names when generating config file to
                     prevent possible clash with config file keywords
                     (such as 'key').

5.   [func]          syntax change to named.conf file: new ssu grant/deny
                     statements must now be enclosed by an 'update-policy'
                     block.

4.   [port]          bin/named/unix/os.c didn't compile on systems with
                     linux 2.3 kernel includes due to conflicts between
                     C library includes and the kernel includes.  We now
                     get only what we need from <linux/capability.h>, and
                     avoid pulling in other linux kernel .h files.

3.   [bug]           TKEYs go in the answer section of responses, not
                     the additional section.

2.   [bug]           Generating cryptographic randomness failed on
                     systems without /dev/random.

1.   [bug]           The installdirs rule in
                     lib/isc/unix/include/isc/Makefile.in had a typo which
                     prevented the isc directory from being created if it
                     didn't exist.
```

```
--- 9.0.0b2 released ---
```

CHAPTER

SIXTEEN

# A BRIEF HISTORY OF THE DNS AND BIND

Although the Domain Name System "officially" began in 1984 with the publication of **RFC 920**, the core of the new system was described in 1983 in **RFC 882** and **RFC 883**. From 1984 to 1987, the ARPAnet (the precursor to today's Internet) became a testbed of experimentation for developing the new naming/addressing scheme in a rapidly expanding, operational network environment. New RFCs were written and published in 1987 that modified the original documents to incorporate improvements based on the working model. **RFC 1034**, "Domain Names-Concepts and Facilities," and **RFC 1035**, "Domain Names-Implementation and Specification," were published and became the standards upon which all DNS implementations are built.

The first working domain name server, called "Jeeves," was written in 1983-84 by Paul Mockapetris for operation on DEC Tops-20 machines located at the University of Southern California's Information Sciences Institute (USC-ISI) and SRI International's Network Information Center (SRI-NIC). A DNS server for Unix machines, the Berkeley Internet Name Domain (BIND) package, was written soon after by a group of graduate students at the University of California at Berkeley under a grant from the US Defense Advanced Research Projects Administration (DARPA).

Versions of BIND through 4.8.3 were maintained by the Computer Systems Research Group (CSRG) at UC Berkeley. Douglas Terry, Mark Painter, David Riggle, and Songnian Zhou made up the initial BIND project team. After that, additional work on the software package was done by Ralph Campbell. Kevin Dunlap, a Digital Equipment Corporation employee on loan to the CSRG, worked on BIND for 2 years, from 1985 to 1987. Many other people also contributed to BIND development during that time: Doug Kingston, Craig Partridge, Smoot Carl-Mitchell, Mike Muuss, Jim Bloom, and Mike Schwartz. BIND maintenance was subsequently handled by Mike Karels and Øivind Kure.

BIND versions 4.9 and 4.9.1 were released by Digital Equipment Corporation (which became Compaq Computer Corporation and eventually merged with Hewlett-Packard). Paul Vixie, then a DEC employee, became BIND's primary caretaker. He was assisted by Phil Almquist, Robert Elz, Alan Barrett, Paul Albitz, Bryan Beecher, Andrew Partan, Andy Cherenson, Tom Limoncelli, Berthold Paffrath, Fuat Baran, Anant Kumar, Art Harkin, Win Treese, Don Lewis, Christophe Wolfhugel, and others.

In 1994, BIND version 4.9.2 was sponsored by Vixie Enterprises. Paul Vixie became BIND's principal architect/programmer.

BIND versions from 4.9.3 onward have been developed and maintained by Internet Systems Consortium and its predecessor, the Internet Software Consortium, with support provided by ISC's sponsors.

As co-architects/programmers, Bob Halley and Paul Vixie released the first production-ready version of BIND version 8 in May 1997.

BIND version 9 was released in September 2000 and is a major rewrite of nearly all aspects of the underlying BIND architecture.

BIND versions 4 and 8 are officially deprecated. No additional development is done on BIND version 4 or BIND version 8.

BIND development work is made possible today by the sponsorship of corporations who purchase professional support services from ISC (https://www.isc.org/contact/) and/or donate to our mission, and by the tireless efforts of numerous individuals.