BIND 9 Administrator Reference Manual

BIND Version 9.10.2-P2

Chapter 1

Introduction

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

Chapter 3

Name Server Configuration

In this chapter we provide some suggested configurations along with guidelines for their use. We suggest reasonable values for certain option settings.

3.1 Sample Configurations

```
     // This is the default
     allow-query { any; };
     // Do not provide recursive service
     recursion no;
};

// Provide a reverse mapping for the loopback
// address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
     type master;
     file "localhost.rev";
     notify no;
};
// We are the master server for example.com
zone "example.com" {
     type master;
     file "example.com.db";
     // IP addresses of slave servers allowed to
     // transfer example.com
     allow-transfer {
          192.168.4.14;
          192.168.5.53;
     };
};
// We are a slave server for eng.example.com
zone "eng.example.com" {
     type slave;
     file "eng.example.com.bk";
     // IP address of eng.example.com master server
     masters { 192.168.4.12; };
};
```

3.2 Load Balancing

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

For example, if you have three WWW servers with network addresses of 10.0.0.1, 10.0.0.2 and 10.0.0.3,

3.3 Name Server Operations

3.3.1 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.

3.3.1.1 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 The domain information groper (**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.

```
};
options {
     default-server 127.0.0.1;
     default-key    rndc_key;
};
```

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

```
$rndc reload
```

Chapter 4

Advanced DNS Features

4.1 Notify

DNS NOTIFY is a mechanism that allows master servers to notify their slave servers of changes to a zone's data. In response to a **NOTIFY** from a master server, the slave will check to see that its version of the zone is the current version and, if not, initiate a zone transfer.

For more information about DNS **NOTIFY**, see the description of the **notify** option in Section 6.2.16.1 and the description of the zone option **also-notify** in Section 6.2.16.7. The **NOTIFY** protocol is specified in RFC 1996.

NOTE

At this point, the key is recognized. This means that if the server receives a message signed by this key,

4.9.2 Dynamic DNS update method

To insert the keys via dynamic update:

```
% nsupdate
> ttl 3600
```

4.9.13 NSEC3 and OPTOUT

named only supports creating new NSEC3 chains where all the NSEC3 records in the zone have the same OPTOUT state. **named** supports UPDATES to zones where the NSEC3 records in the chain have mixed OPTOUT state. **named** does not support changing the OPTOUT state of an individual NSEC3 record, the entire chain needs to be changed if the OPTOUT state of an individual NSEC3 needs to be changed.

4.10 Dynamic Trust Anchor Management

BIND 9.7.0 introduces support for RFC 5011, dynamic trust anchor management. Using this feature

4.11.3.1 Patching OpenSSL

```
(pkcs11) PKCS #11 engine support (crypto accelerator)
```

Next, run "**apps/openssl engine pkcs11 -t**". This will attempt to initialize the PKCS#11 engine. If it is

If you wish to generate a second key in the HSM for use as a zone-signing key, follow the same procedure above, using a different keylabel, a smaller key size, and omitting "-f KSK" from the dnssec-keyfromlabel arguments:

(Note: When using OpenSSL-based PKCS#11 the label is an arbitrary string which identifies the key. With native PKCS#11, the label is a PKCS#11 URI string which may include other details about the key and the HSM, including its PIN. See dnssec-keyfromlabel(8) for details.)

```
$ pkcs11-keygen -b 1024 -l sample-zsk
$ dnssec-keyfromlabel -l sample-zsk example.net
```

Alternatively, you may prefer to generate a conventional on-disk key, using dnssec-keygen:

```
$ dnssec-keygen example.net
```

This provides less security than an HSM key, but since HSMs can be slow or cumbersome to use for security reasons, it may be more efficient to reserve HSM keys for use in the less frequent key-signing operation. The zone-signing key can be rolled more frequently, if you wish, to compensate for a reduction in key security. (Note: When using native PKCS#11, there is no speed advantage to using on-disk keys, as cryptographic operations will be done by the HSM regardless.)

Now you can sign the zone. (Note: If not using the -S option to **dnssec-signzone**

accomplished by placing the PIN into the openssl.cnf file (in the above examples, `/opt/pkcs11/usr/ssl/openssl.cnf`).

4.12.1 Configuring DLZ

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; when answering a query, all DLZ modules with `search` set to `yes` will be queried to find out if they contain

Chapter 5

The BIND 9 Lightweight Resolver

5.1 The Lightweight Resolver Library

Traditionally applications have been linked with a stub resolver library that sends recursive DNS queries to a local caching name server.

```
ip
```

6.1.2.1 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
```

6.1.2.2 Definition and Usage

Comments may appear 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.

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.
include	

6.2.8 key Statement Definition and Usage

The **key** statement defines a shared secret key for use with TSIG (see Section 4.5) or the command channel (see Section 6.2.4).

The **key**

update	Dynamic updates.
update-security	Approval and denial of update requests.
queries	Specify where queries should be logged to. At startup, specifying the category **queries** will also enable query logging unless **querylog** option has been specified. The query log entry reports the client's IP address and

`qrysent`	The number of queries the resolver sent at the `domain` zone.
`timeout`	The number of timeouts since the resolver received the last response.
`lame`	

The **listen-on** statement specifies a list of IPv4 addresses (and ports) that this instance of a lightweight

```
    [ try-tcp-refresh yes_or_no; ]
```

dns64-server and **dns64-contact** can be used to specify the name of the server and contact for the zones. These are settable at the view / options level. These are not settable on a per-prefix basis.

Each **dns64** supports an optional **clients** ACL that determines which clients are affected by this directive. If not defined, it defaults to `any;`.

Each **dns64** 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;`.

Normally, DNS64 won't apply to a domain name that owns one or more AAAA records; these records will simply be returned. The optional **exclude**

recursion If yes, and a DNS query requests recursion, then the server will attempt to do all the work required to answer the query. If recursion is off and the server does not already know the answer, it

DNAME chains.

When both of these options are set to yes (the default) and a query is being answered from authoritative data (a zone configured into the server), the additional data section of the reply will be filled in using data from other authoritative zones and from the cache. In some situations this is undesirable, such as when there is concern over the correctness of the cache, or in servers where

check-wildcard

dnssec-loadkeys-interval When a zone is configured with **auto-dnssec maintain;**

result in the default being used.

6.2.16.5 Interfaces

The interfaces and ports that the server will answer queries from may be specified using the **listen-on** option. **listen-on** takes an optional port and an `address_match_list` of IPv4 addresses. (IPv6 addresses are ignored, with a logged warning.) The server will listen on all interfaces allowed by the

```
query-source address * port *;
query-source-v6 address * port *;
```

If **use-v4-udp-ports** or **use-v6-udp-ports** is unspecified, **named** will check if the operating system provides a programming interface to retrieve the system's default range for ephemeral ports. If such an interface is available, **named** will address

6.2.16.15 Tuning

lame-ttl

Note that when a zone file in a different format than `text` is loaded, **named** may omit some of the checks which would be performed for a file in the

- 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
-

For example, if you own a domain named "example.net" and your internal network uses an IPv4 prefix 192.0.2.0/24, you 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 lets a web browser in your 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 will be ignored.

On the other hand, if the browser looks up a legitimate internal web server "www.example.net" and the following response is returned to the BIND 9 server

```
www.example.net. A 192.0.2.2
```

it will be accepted since the owner name "www.example.net" matches the

IPv6 addresses are encoded in a format similar to the standard IPv6 text representation, `prefixlength.W8.W7.W6.W5.W4.W3.W2.W1.rpz-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 ad-

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 CNAME whose target is the wildcard top-level domain (*.). It rewrites the response to NODATA or ANCOUNT=1.

Local Data A set of ordinary DNS records can be used to answer queries. Queries for record types not 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

The TTL of a record modified by RPZ policies is set from the TTL of the relevant record in policy zone. It is then limited to a maximum value. The **max-policy-ttl** clause changes that maximum from its default of 5.

For example, you might use this option statement

```
response-policy { zone "badlist"; };
```

and this zone statement

```
zone "badlist" {type master; file "master/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
```

6.2.16.21 Response Rate Limiting

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

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 will be internally configured.

```
example.com. NS example.com.
example.com. A 192.0.2.1
example.com. AAAA 2001:db8::1234
```

These records are internally used 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 will initiate recursive resolution and send queries to 192.0.2.1 and / or 2001:db8::1234.

`self`	This rule matches when the name being updated matches the contents of the *`identity`* field. The *`name`* field is ignored, but should be the same as the

DNAME	Replaces the domain name specified with another name to be looked up, effectively aliasing an entire subtree of the domain name space rather than a single record as in the case of the CNAME RR. Described in RFC 2672.
DNSKEY	Stores a public key associated with a signed DNS zone. Described in RFC 4034.
DS	Stores the hash of a public key associated with a signed DNS zone. Described in RFC 4034.
GPOS	Specifies the global position. Superseded by LOC.

6.3.3 Setting TTLs

The time-to-live of the RR field is a 32-bit integer represented in units of seconds, and is primarily used

```
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.
..R2.EXAMPLE.
```

BIND 8 does not support the optional TTL and CLASS fields.

6.3.7 Additional File Formats

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

The `raw`

Symbol	BIND8 Symbol	Description

<**TYPE**>**RecvErr**	Errors in socket receive operations. This includes errors of send operations on a connected UDP socket notified by an ICMP error message.

6.4.1.5 Compatibility with *BIND*

Chapter 7

BIND 9 Security Considerations

7.1 Access Control Lists

Access Control Lists (ACLs) are address match lists that you can set up and nickname for future use in

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

Appendix A

Release Notes

A.1 Release Notes for BIND Version 9.10.2-P2

A.1.1 Introduction

This document summarizes changes since BIND 9.10.2:

BIND 9.10.2-P2 addresses a security issue described in CVE-2015-4620.

BIND 9.10.2-P1 addressed several bugs that have been identified in the BIND 9.10 implementation of

A.1.6 Bug Fixes

-

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 several corporations, and by the tireless work efforts of numerous individuals.

References

Standards

[RFC1034] *Domain Names — Concepts and Facilities*, P.V. Mockapetris, November 1987.

[RFC1035] *Domain Names — Implementation and Specification*, P. V. Mockapetris, November 1987.

[RFC974] *Mail Routing and the Domain System*, C. Partridge, January 1986.

Proposed Standards

[RFC4074] *Common Misbehaviour Against DNS Queries for IPv6 Addresses*, Y. Morishita and T. Jinmei, May 2005.

Resource Record Types

[RFC1183] *New DNS RR Definitions*, C.F. Everhart, L. A. Mamakos, R. Ullmann, and P. Mockapetris, October 1990.

[RFC1706]

DNS Operations

[RFC1033] *Domain administrators operations guide.*, M. Lottor, November 1987.

[RFC1537] *Common DNS Data File Configuration Errors*, P. Beertema, October 1993.

[RFC1912] *Common DNS Operational and Configuration Errors*, D. Barr, February 1996.

[RFC2010] *Operational Criteria for Root Name Servers.*, B. Manning and P. Vixie, October 1996.

[RFC2219] *Use of DNS Aliases for Network Services.*, M. Hamilton and R. Wright, October 1997.

Internationalized Domain Names

[RFC2825] *A Tangled Web: Issues of I18N, Domain Names, and the Other Internet protocols*,

Appendix D

BIND 9 DNS Library Support

D.1 BIND 9 DNS Library Support

This version of BIND 9 "exports" its internal libraries so that they can be used by third-party applications more easily (we call them "export" libraries in this document). In addition to all major DNS-related APIs

D.1.3 Installation

```
$ cd lib/export
$ make install
```

This will install library object files under the directory specified by the –with-export-libdir configure option (default: EPREFIX/lib/bind9), and header files under the directory specified by the –with-export-includedir configure option (default: PREFIX/include/bind9). Root privilege is normally required. "**make install**" at the top directory will do the same.

D.1.6.1 sample: a simple stub resolver utility

It sends a query of a given name (of a given optional RR type) to a specified recursive server, and prints the result as a list of RRs. It can also act as a validating stub resolver if a trust anchor is given via a set of command line options.

Usage: sample [options] server_address hostname

Options and Arguments:

-t RRtype specify the RR type of the query. The default is the A RR.

[-a algorithm] [-e] -k keyname -K keystring specify a command-line DNS key to validate the answer. For example, to specify the followingt RRtype

D.1.7 Library References

Appendix E

Manual pages

E.1 dig

Name

dig — DNS lookup utility

Synopsis

SIMPLE USAGE

A typical invocation of **dig** looks like:

```
dig @server name type
```

where:

`server`

IDN SUPPORT

If **dig** has been built with IDN (internationalized domain name) support, it can accept and display non-ASCII domain names. **dig** appropriately converts character encoding of domain name before sending a request to DNS server or displaying a reply from the server. If you'd like to turn off the IDN support for some reason, defines the `IDN_DISABLE` environment variable. The IDN support is disabled if the variable is set when **dig** runs.

FILES

`/etc/resolv.conf`

`${HOME}/.digrc`

SEE ALSO

host(1), named(8), dnssec-keygen(8), *RFC1035*.

BUGS

There are probably too many query options.

626Tf-Sdot-250(many)-250(query)-2508ptions.

SEE ALSO

dig(1), named(8).

E.3 delv

Name

delv — DNS lookup and validation utility

Synopsis

`delv [@server] [-4] [-6] [-a` *`anchor-file`*

-p *port#* Specifies a destination port to use for queries instead of the standard DNS port number 53.

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 will product the same output (but will affect other logging categories as well).

`+[no]mtrace` Toggle message logging. This produces a detailed dump of the responses received by **delv**

```
+[no]dlv[=DLV]
```

Synopsis

EXAMPLE

To build the SHA-256 DS RR from the `Kexample.com.+003+26160` keyfile name, the following command would be issued:

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 will be generated, or it may be read from any other file or from the standard input, in which case both .key and .private files will be 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,

SEE ALSO

dnssec-keygen(8), dnssec-signzone(8), *BIND 9 Administrator Reference Manual*, *RFC 5011*.

AUTHOR

Internet Systems Consortium

E.8 dnssec-keyfromlabel

Name

dnssec-keyfromlabel — DNSSEC key generation tool

Synopsis

```
dnssec-keyfromlabel -l label [-3] [-a algorithm] [-A date/offset] [-c
    class] [-D date/offset] [-E engine] [-f flag] [-G] [-I date/offset] [-i
    interval] [-k] [-K directory] [-L ttl] [-n nametype] [-P date/offset]
    [-p protocol] [-R date/offset] [-S key] [-t type] [-v level] [-V] [-y]
    name
```

DESCRIPTION

dnssec-keyfromlabel generates a key pair of files that referencing a key object stored in a cryptographic

-p *protocol*

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.

When BIND is built with OpenSSL PKCS#11 support, this defaults to the string "pkcs11", which identifies an OpenSSL engine that can drive a cryptographic accelerator or hardware service module. When BIND is built with native PKCS#11 cryptography (–enable-native-pkcs11), it defaults to the path of the PKCS#11 provider library specified via "–with-pkcs11".

-f *flag*

for use with SIG(0).

Using any TSIG algorithm (HMAC-* or DH) forces this option to KEY.

-t *type* Indicates the use of the key. `type`

E.12 dnssec-signzone

Name

dnssec-signzone — DNSSEC zone signing tool

Synopsis

`dnssec-signzone [-a] [-c class] [-d directory] [-D] [-E`

-K *directory*

NSEC or to NSEC3 with different parameters. Without this option, **dnssec-signzone** will retain the existing chain when re-signing.

-v *level* Sets the debugging level.

-x Only sign the DNSKEY RRset with key-signing keys, and omit signatures from zone-signing keys. (This is similar to the **dnssec-dnskey-kskonly yes;** zone option in **named**.)

-z Ignore KSK flag on key when determining what to sign. This causes KSK-flagged keys to sign all records, not just the DNSKEY RRset. (This is similar to the **update-check-ksk no;** zone option in **named**.)

-3 *salt* Generate an NSEC3

-v Print the version of the **named-checkconf** program and exit.

-k *mode* Perform **"check-names"**

-d *debug-level* Set the daemon's debug level to *debug-level*. Debugging traces from **named** become more verbose as the debug level increases.

-D *string* 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, specifies the hardware to use for cryptographic operations, such as a secure key store used for signing.

Synopsis

```
named-journalprint
```

SEE ALSO

RFC 1034, *RFC 1035*, named(8)

E.19 nsupdate

Name

nsupdate — Dynamic DNS update utility

Synopsis

```
nsupdate [-d] [-D] [-g | -o | -l | -y [hmac:]keyname:secret | -k keyfile]
    [-t timeout] [-u udptimeout] [-r udpretries] [-R randomdev] [-v] [-T]
    [-P] [-V] [filename]
```

DESCRIPTION

nsupdate is used to submit Dynamic DNS Update requests as defined in RFC 2136 to a name server.

`/var/run/named/session.key` sets the default TSIG key for use in local-only mode

`K{name}.+157.+{random}.key` base-64 encoding of HMAC-MD5 key created by dnssec-keygen(8).

`K{name}.+157.+{random}.private`

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 **auto-dnssec** zone option be set to `maintain`, and also requires the zone to be configured to allow dynamic DNS. (See "Dynamic Update Policies" in the Administrator Reference Manual for more details.)

```
stop [-p]
```

Synopsis

```
};


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**

E.22 rndc-confgen

Name

rndc-confgen — rndc key generation tool

is keyboard input. `randomdev` specifies the name of a character device or file containing random data to be used instead of the default. The special value `keyboard` indicates that keyboard input should be used.

DESCRIPTION

SEE ALSO

BIND 9 Administrator Reference Manual, *RFC 2104*.

AUTHOR