Step-by-Step DNSSEC-Tools Operator Guidance Document

Using the DNSSEC-Tools v0.2 distribution

SPARTA, Inc.

Table of Contents

List of Tables

Chapter 1. Introduction

DNS Security (DNSSEC) helps protect against DNS-spoofing attacks by providing origin authentication and integrity protection of DNS information. Proper maintenance of a DNSSEC-enhanced DNS zone is essential to protecting the domain's zone data.

This Step-by-Step DNSSEC-Tools Operator Guidance Document is intended for operations using the DNSSEC-Tools v0.2 distribution. It will assist operators in gaining operational experience with DNSSEC. Some basic understanding of DNSSEC terms and concepts is required. It follows the format laid out by [dnssec-operators-guide].

This document is meant to be a learning aid and is not intended to define policy in any form. Any implicit recommendations for key sizes, signature validity periods, and command line parameters are for illustration purposes ONLY and MUST NOT be used in production environments unless due-diligence has been taken to ensure that these values are acceptable within such environments. See [dnssec-operational-practices] for suggestions on determining appropriate security characteristics.

This document was written as part of the DNSSEC-Tools project. The goal of this project is to create a set of documentation, tools, patches, applications, wrappers, extensions, and plug-ins that will help ease the deployment of DNSSEC-related technologies. For more information about this project and the tools that are being developed and provided, please see the DNSSEC-Tools project web page at: `http://www.dnssec-tools.org`.

Organization of this Document

The following operations are described in this document:

Conventions Used in this Document

One of the goals of this document is to self-contain DNS Security operations within sections and prevent constant cross-referencing between sections. Consequently, certain parts of the text are repeated throughout the document.

Text marked in bold represents text or commands entered by users within a given procedural step.

Underlined text, which can also be bold, is a place-holder for actual run-time values. These values are either automatically generated or are values that are known to the user from some other step.

Additionally, the following typographical conventions are used in this document.

Table 1.1. Typographical Conventions

command	Command names
`filename`	File and path names
`URL`	Web URLs
execution	Simple command executions

Longer sets of command sequences are given in this format:

```
$ cd /tmp [ENTER]
$ ls [ENTER]
$ rm -fr * [ENTER]
$
```

In most cases, output will not be displayed for given command sequences.

Acknowledgements

This document builds upon the procedures laid out in [dnssec-operators-guide].

Comments

Please send any comments and corrections to developers@dnssec-tools.org.

Chapter 2. Configure the Toolset

The following sections must be read before proceeding with the rest of this guide.

The steps in the section called “Check for Randomness” and the section called “Edit the Configuration File” *MUST* be performed prior to following any other steps.

Check for Randomness

Key generation and zone signing require random data to create strong cryptographic material. The **zonesigner** command defaults to using random data from `/dev/random`. Use this test to verify that `/dev/random` will provide data when requested:

```
$ dd if=/dev/random bs=2 count=10 | od -x [ENTER]
```

The above command checks if /dev/random is able to provide random data when queried; it does not check to see that the data provided is truly random.

If this command provides data immediately, the `/dev/random` will provide the data you need. If it hangs, then **zonesigner** won't be able to retrieve random data from `/dev/random`.

If this check for randomness fails, pseudorandom numbers can be used instead. However, using pseudorandom numbers significantly affects the quality of the crypto material. A more appropriate measure would be to run **zonesigner** on a different system that has `/dev/random` and the ability to generate good random data.

Edit the Configuration File

The default location for the DNSSEC-Tools configuration file is `/usr/local/etc/dnssec/dnssec-tools.conf`.

The set of options that you may consider modifying are outlined below. For each option, a recommended setting is provided.

Table 2.1. DNSSEC-Tools Configuration Options

Option	Description	Recommended Setting
kskdirectory	Directory where the KSK keys are stored.	`keys`
zskdirectory	Directory where the ZSK keys are stored.	`keys`
ksdir	Directory where the `keyset-` and `dsset-` files for both the current zone and any children are stored.	`sets`
algorithm	The cryptographic algorithm to use for the keys.	rsasha1
ksklength	The length of the KSK key.	2048
zsklength	The length of the ZSK key.	1024
random	This specifies the random device to use.	If the random device was determined to work fine in the section called “Check for Randomness”, then this should be set to `/dev/random`. If there was a problem with the device, then this can be set to `/dev/urandom`,

Option	Description	Recommended Setting
		but it should be understood that pseudorandom numbers significantly affect the quality of the crypto material.
endtime	The lifetime of the signatures.	+2592000 (30 days)

Chapter 3. Initially Signing a Zone

A zone needs to be re-signed when *any* change is made to it. Follow Chapter 6, *Resigning a Zone* when resigning a zone.

Sign the Zone with zonesigner

```
$ zonesigner -genkeys -gends -zone zone-name zone-file output-file [ENTER]
```

Key generation and signing may take a few minutes to complete depending on the size of the zone file and size of the keys. If the above operation appears to be unresponsive for an unreasonable time frame, use pseudorandom numbers (see Chapter 2, *Configure the Toolset* for more information).

The output is a set of files outlined below.

Table 3.1. zonesigner Output Files

File	Description
`output-file.signed`	The signed zone file. The .signed is added by **zonesigner**.
`keyset-zone-name`	The keyset for the zone. This is stored in the directory specified by the configuration file and may have to be sent to the parent zone - see Chapter 7, *Creating a Signed Delegation - Child Zone Activity*.
`dsset-zone-name`	The dsset for the zone. This is stored in the directory specified by the configuration file and may have to be sent to the parent zone - see Chapter 7, *Creating a Signed Delegation - Child Zone Activity*.
zone-name.krf	The keyrec file. This is used by **zonesigner** to maintain information about the keys used for the zone.
`Kzone-name.+005+keytag.private`	The private key file. This is stored in the directory specified by the configuration file. The `keytag` is a unique identifier for this key. This document uses `RSASHA1` as the cryptographic algorithm, which is represented by the `005` in the key name's algorithm field.
`Kzone-name.+005+keytag.key`	The public key file. This is stored in the directory specified by the configuration file. The `keytag` is a unique identifier for this key. This document uses `RSASHA1` as the cryptographic algorithm, which is represented by the `005` in the key name's algorithm field.

Chapter 4. Configuring and Serving a Signed Zone

Several configuration files must be modified in order to serve a signed zone. Follow the steps below to configure your name server and have it start serving your signed zone.

`named.conf` is the name of the configuration file used in these examples. The configuration file may vary according to the needs of the administrator.

Add the Signed Zone to the Name Server Configuration File

The name of the signed zone file must be added to the name server's configuration file. For the zone whose name is `zone-name`, do the following:

$ **vi named.conf [ENTER]**

```
... zone "zone-name." { type master; file "zone-file.signed"; }; ...
```

Enable DNSSEC

Add the `dnssec-enable yes;` option to the `named.conf` file.

$ **vi named.conf [ENTER]**

```
... options { ... dnssec-enable yes; }; ...
```

Check the Name Server Configuration File for Errors

You must ensure that the configuration file modifications were performed correctly. The **named-checkconf** command will perform this verification. No output indicates that all is well with the zone.

$ **named-checkconf named.conf [ENTER]**

Reload the Zone

The **rndc** command will reload the name server configuration files and zone contents. The name server process is assumed to be already running.

$ **rndc reload zone-name [ENTER]**

Check that the Zone Loaded Properly

Confirm that the SOA serial number of the zone corresponds to the most recent value.

$ **dig @server-IP-address SOA zone-name [ENTER]**

```
; <<>> DiG 9.3.0 <<>> ... ... ;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDI-
TIONAL: 0 ... ;; ANSWER SECTION zone-name 3600 IN SOA servername contact ( 2005092101 ; This
should be the most recent value. ; This value will most likely be different in your zone
file. ... ) ...
```

Chapter 5. Checking Signature Expiration

It is important to regularly check your zone for signatures that are nearing expiration. If the signatures are close to expiring, or already have expiried, see Chapter 6, *Resigning a Zone* for how to resign the zone.

Check the Zone for Expiring Signatures

```
$ expchk -all -warn 10 keyrec-file [ENTER]
```

This checks the keyrec file to see if the zone has signatures expiring in the next 10 days.

Chapter 6. Resigning a Zone

A zone needs to be re-signed when *any* change is made to it.

Resign the Zone with zonesigner

```
$ zonesigner -gends -zone zone-name zone-name zone-file [ENTER]
```

Signing may take a few minutes to complete depending on the size of the zone file. If the above operation appears to be unresponsive for an unreasonable time frame, use pseudorandom numbers (see Chapter 2, *Configure the Toolset* for more information).

The output is a set of files outlined below.

Table 6.1. zonesigner Output Files

File	Description
`output-file.signed`	The signed zone file. The .signed is added by **zonesigner**.
`keyset-zone-name`	The keyset for the zone. This is stored in the directory specified by the configuration file and may have to be sent to the parent zone - see Chapter 7, *Creating a Signed Delegation - Child Zone Activity*.
`dsset-zone-name`	The dsset for the zone. This is stored in the directory specified by the configuration file and may have to be sent to the parent zone - see Chapter 7, *Creating a Signed Delegation - Child Zone Activity*.

Chapter 7. Creating a Signed Delegation - Child Zone Activity

This section describes the steps required by a child for creating a signed delegation.

Securely Transfer the Keyset to the Parent

If any of the zone's KSKs have changed since the last time this file was sent to the parent, then they keyset must also be transferred to the parent. If none of the zone's KSKs have changed, this step may be skipped.

Secure communication between the parent and child zone is done out-of-band.

Wait for the Parent to Publish the DS Record

Before proceeding, wait for the parent zone the publish the DS record. This may be found by using the **dig** command to retrieve the zone's DS record. The `aa` flag in the result must be set and the `ANSWER` section must not be empty.

You may continue if the DS record is the same as the value in the file generated in Chapter 3, *Initially Signing a Zone* or Chapter 6, *Resigning a Zone*.

`$` **dig `@server-IP-address SOA zone-name [ENTER]`**

```
; <<>> DiG 9.3.0 <<>> ... ... ;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDI-
TIONAL: 0 ... ;;ANSWER SECTION zone-name 3600 IN SOA servername contact ( 2005092101 ; This
should be the most recent value. ; This value will most likely be different in your zone
file. ... ) ...
```

Chapter 8. Creating a Signed Delegation - Parent Zone Activity

This section describes the steps required by a parent for creating a signed delegation.

Ensure that the Child Keysets were Received Over a Secure Channel

Secure communication between the parent and child zone is done out-of-band.

Ensure that Each Received Keyset is for a Delegated Zone

The owner name for the DNSKEY record in the received keyset must correspond to a valid delegation.

$ **cat keyset-child-zone-file [ENTER]**

```
child-zone-name. 3600 IN DNSKEY 256 3 5 ( ... ); key id = keytag
```

`child-zone-name` must exist in the parent zone-file as a valid delegation.

$ **cat zone-file [ENTER]**

```
... child-zone-name NS server A ... ...
```

Re-sign the Zone

Re-sign the zone using steps described in Chapter 6, *Resigning a Zone*.

Reload the Zone

The **rndc** command will reload the name server configuration files and zone contents. The name server process is assumed to be already running.

$ **rndc reload zone-name [ENTER]**

Chapter 9. "Current" ZSK Roll-over (Pre-publish Scheme)

This section will be written when the key rollover tool is completed.

Chapter 10. KSK Roll-over (Double Signature Scheme)

This section will be written when the key rollover tool is completed.

Chapter 11. Emergency ZSK Roll-over (Current ZSK Compromise)

This section will be written when the key rollover tool is completed.

Chapter 12. Emergency ZSK Roll-over (Published ZSK Compromise)

This section will be written when the key rollover tool is completed.

Chapter 13. Emergency ZSK Roll-over (Published and Current ZSK Compromise)

This section will be written when the key rollover tool is completed.

Chapter 14. Emergency KSK Roll-over (KSK Compromise)

This section will be written when the key rollover tool is completed.

Chapter 15. Parent Action During Child KSK Compromise

During a KSK compromise the secure status of the child zone is dropped. This is done by deleting the DS record in the parent zone.

Ensure that the KSK Compromise Notification Came Over a Secure Channel

Authentication and communication between parent and child occurs out-of-band.

Delete the Child's Keyset File at the Parent

The DS record for the child should not be created. This can simply be achieved by removing the keyset file from the system.

Re-sign the Zone

Re-sign the zone using steps described in Chapter 6, *Resigning a Zone*.

Reload the Zone

The **rndc** command will reload the name server configuration files and zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone-name [ENTER]
```

Chapter 16. Migrate to the Toolset

The zonesigner tool simplifies the maintenance of a signed zone. It automates many of the routine tasks required for signing a zone. Given this, an operator already using BIND tools to maintain a signed zone may want to transition over to zonesigner, while still retaining existing keys that are being used to sign a zone.

This section provides step-by-step instructions to transition from using BIND tools for maintaining a signed zone to using zonesigner. In the examples given below, the zone example.com is currently signed, signed zone file is maintained using **dnssec-signzone** command from BIND 9.3.1, and the following files are present:

Table 16.1. Example Files

File	Description
db-in.example.com.	Unsigned zone file
db-in.example.com..signed	Signed zone file
Kexample.com.+005+47670	KSK files prefix
Kexample.com.+005+48926	ZSK files prefix

Generate the keyrec file

```
$ genkrf -zone=example.com -ksk=Kexample.com.+005+47670 -zskcur=Kexample.com.+005+48926db-in.example.com. db-in.example.com..signed
```

The **genkrf** command generates a keyrec file from existing key files. It also generates any additional keys that zonesigner uses. In the above example, **genkrf** will generate a new key `zskpub` along with the keyrec file named `example.com.krf`. It will display the following message if successful:

```
genkrf: file example.com.krf created successfully.
```

Verify the keyrec file

Examine the contents of the keyrec file and ensure that the original KSK and ZSK files are being used.

```
$ grep Kexample.com.+005+47670 example.com.krf [ENTER]

kskdir "Kexample.com.+005+47670"

$ grep Kexample.com.+005+48296 example.com.krf [ENTER]

zskcur "Kexample.com.+005+48296"
```

Resign the zone with zonesigner

See Chapter 6, *Resigning a Zone* for how to resign the zone.

Chapter 17. Configure a Secure Resolver

Introduction

This document has described how to configure and maintain a secure nameserver which supplies signed zones and delegations. All the signed zones and delegations within the scope of the server form an island of security from which nameserver data can be retrieved in a authenticated and verifiable way by a security aware resolver.

But there are times operationally when a recursing secure name server may need to refer to, and retrieve, data from servers outside this island of security. If the referral is to a non-secure name server there is no secure recourse and the chain of authentication is broken and this data can not then be trusted.

To extend the scope of security, a secure nameserver may be configured with public key data from other remote secure zones so that the chain of trust is expanded. The `trusted-keys` directive in the `named.conf` configuration file provides this capability.

The mechanism described below for extending the chain of trust should be used judiciously and comes with the added operational burden of verifying and maintaining key validity and timeliness.

The following is an example of a `trusted-keys` directive in a `named.conf` which provides verification of data retrieved from the `se.` and `dnssec-tools.org.` zones. Note: key data may be different from that shown and should be obtained as described below.

```
trusted-keys {

se.       257       3       5       "AwEAAaxPMcR2x0HbQV4WeZB6oEDX+r0QM65KbhTjrW1ZaARmPhEZZe3Y
9ifgEuq7vZ/zGZUdEGNWy+JZzus0lUptwgjGwhUS1558Hb4JKUbbOTcM
8pwXlj0EiX3oDFVmjHO444gLkBOUKUf/mC7HvfwYH/Be22GnClrinKJp
1Og4ywzO9WglMk7jbfW33gUKvirTHr25GL7STQUzBb5Usxt8lgnyTUHs
1t3JwCY5hKZ6CqFxmAVZP20igTixin/1LcrgX/KMEGd/buvF4qJCydui
eHukuY3H4XMAcR+xia2nIUPvm/oyWR8BW/hWdzOvnSCThlHf3xiYleDb t/o1OTQ09A0=";

dnssec-tools.org.  257  3  5  "AQOoEFn3VnV1qDwnNX9GlukAsbL7buCk6Wmt3VG9BOVae84VVc/yWghg  tFM/
WKw/5243XoBEeNyaahRIrlAJEnErLUWlKO/YuWkasRN4jkS2dDjS
MWgjdGxzux+e0UV2UZfpjyygYvaD9U8xTwwzLYLDkamr1SCaHWCWUOO+                              QMa/
WY//r3ObbOFOYCvyqvsLRwofSFnQnsbihKbcP9HQSDQ4iRqbCTMV
B+yq5NXiFoZT05Sqm/ijOrjLznZkUqIal9EXqyhNT0dTa9Gdn8+tfn+l                              YAp-
wK91NA2YG/3t8ZKTYjDLe1YlwKg8OBTTN4ARap+265EtE87BhE6ZK fp+DUx4N";

};
```

The format of the directive is:

```
trusted-keys { <zone> <flags> <protocol> <algorithm> <quoted-key-string>; };
```

The *`flags`*, *`protocol`*, *`algorithm`* and *`quoted-key-string`* data may be obtained using the following **dig** command, but the content of the string should be verified in a secure out-of-band way to ensure its validity.

```
$ dig se. DNSKEY

;; Truncated, retrying in TCP mode. ; <<>> DiG 9.3.1 <<>> se. DNSKEY ;; global options:
printcmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31166 ;; flags:
qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 8, ADDITIONAL: 0 ;; QUESTION SECTION: ;se. IN DNS-
KEY    ;;    ANSWER    SECTION:    se.    3600    IN    DNSKEY    257    3    5    AwEAAaxPM-
```

```
cR2x0HbQV4WeZB6oEDX+r0QM65KbhTjrW1ZaARmPhEZZe3Y
9ifgEuq7vZ/zGZUdEGNWy+JZzus0lUptwgjGwhUS1558Hb4JKUbbOTcM
8pwXlj0EiX3oDFVmjHO444gLkBOUKUf/mC7HvfwYH/Be22GnClrinKJp
1Og4ywzO9WglMk7jbfW33gUKvirTHr25GL7STQUzBb5Usxt8lgnyTUHs
1t3JwCY5hKZ6CqFxmAVZP20igTixin/1LcrgX/KMEGd/buvF4qJCydui
eHukuY3H4XMAcR+xia2nIUPvm/oyWR8BW/hWdzOvnSCThlHf3xiYleDb t/o1OTQ09A0= ...
```

Note: from the output select the DNSKEY whose flags have the zone signing key bit set (257).

Once the 'named.conf' is edited as above, the configuration can be reloaded with:

$ **rndc `reload`**

It may also be necessary to flush the cache data before retrieving authenticated results:

$ **rndc `flush`**

To verify that the `trusted-keys` directive is working properly perform a secure **dig** at the configured server for the remote signed zone data and observe that the `ad` flag is set in the response. For example:

$ **dig `@localhost se. ANY +dnssec`**

```
;; Truncated, retrying in TCP mode. ;; Connection to ::1#53(::1) for se. failed: connection refused. ; <<>> DiG 9.3.1 <<>> @localhost ANY se. +dnssec ; (2 servers found) ;; global options: printcmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56473 ;; flags: qr rd ra ad; QUERY: 1, ANSWER: 23, AUTHORITY: 9, ADDITIONAL: 1 ...
```

References

[dnssec-operators-guide] SPARTA, Inc.. *Step-by-Step DNS Security Operator Guidance Document*. 31 August 2005.

[dnssec-operational-practices] Olaf Kolkman and Miek Gieben. *DNSSEC Operational Practices*. 24 October 2005. draft-ietf-dnsop-dnssec-operational-practices-06.txt (work in progress).