Root KSK Rollover in BIND

Author: **Stephen Morris** Reference Number: **AA-01525** Views: **4591** Created: **2017-07-01 15:35** **0 Rating**/ Voters
Last Updated: **2017-09-29 00:12**

In late 2017/early 2018 the root zone key-signing key will be changed. A new key will be introduced and used to sign the root zone's DNSKEY RRset, then the old key will be removed. This article describes how BIND will cope with that transition and what action you need to take to ensure a trouble-free key roll.

This article assumes that you are familiar with DNSSEC. If not, it is suggested that you review ISC Knowledgebase Articles articles such as https://kb.isc.org/article/AA-00820 or read the BIND DNSSEC guide first.

Background

DNSSEC was first introduced into the root zone in 2010. The deployment followed a "standard" DNSSEC model: the records in the zone are signed with a zone-signing key (ZSK) and the root zone's DNSKEY RRset is signed with a key-signing key (KSK). The original intention was to update (or "roll") the KSK after five years. For various reasons, that operation has been delayed, but will be going ahead in the next six months or so.

The root zone KSK is special, in that it forms the root of the DNS chain of trust. The KSK of other zones can be validated by DS records in the zone's parent but since there is no parent to the root zone, the root KSK has to be validated in other ways. For BIND this has been done by comparing it with a copy of the root KSK that is available through the server software. This copy of the root KSK is known as a "trust anchor".

Rolling (i.e. replacing) such a key presents a challenge: how can such a change be communicated to potentially millions of name servers? The solution to the problem is to be found in an Internet standards document called RFC 5011. In essence, this says that to roll a trust anchor:

1. Introduce the new key into the zone and sign it with the existing trust anchor.
2. Leave it a while so that name servers see the new key. Because it is signed with a key they trust, name servers can trust that key.
3. When the new key is trusted, revoke the existing trust anchor. This means setting a bit in the old key and keeping the revoked key in the DNSKEY RRset. By querying the DNSKEY RRset, name servers will learn that the old should no longer be used.
4. Remove the revoked key from the zone.

(There are various intervals between these steps and caveats to the process, all of which are documented in RFC 5011.)

BIND's Handling of Trust Anchors

Being the trust anchor for the whole DNS system, the root KSK is special. However, in the early days of DNSSEC, it was not uncommon for there to be different (independent) trust anchors for different parts of the DNS tree. As DNSSEC developed, BIND was updated to take account of it and as a result has ended up with several different ways of specifying the trust anchor. The following list covers common situations.

1. New installation of DNSSEC
If you are going to enable DNSSEC for the first time then the easiest way to ensure that you have a valid root trust anchor is to download a recent version of BIND. All versions of BIND since April 2017 (i.e. 9.9.10, 9.10.5, 9.11.1 and later) include a hard-coded copy of the current root KSK. When you run one of these versions, the key will be available as soon as you enable DNSSEC validation. If a new root KSK has been introduced between the latest release of BIND and your installation, then on startingBIND with DNSSEC validation enabled, the root KSK trust anchor will be automatically updated.

2. No trust anchors specified in configuration file
This is the simplest case, where you have enabled DNSSEC validation by setting **dnssec-validation auto;** but have not explicitly specified any keys. In this case, there is nothing to do. BIND uses its hard-coded keys (or those in the **bind.keys** file if one exists) as the initial key set and periodically tests for the presence of a new root key. When it detects a new key, it notes its presence then, after 30 days, accepts it as a trust anchor. You are able to monitor the process by using the **rndc secroots** (all versions of BIND) or **rndc managed-keys status** (BIND 9.11 and later) commands.

3. trusted-keys clause in the BIND configuration file
If you configured DNSSEC some time ago, you may have set **dnssec-validation yes;** in the configuration file and defined the trust anchors with a **trusted-keys** clause. In this case, you will need to take action to handle the key roll.

One way of doing this is to edit the **trusted-keys** clause to include a copy of the new key. This can be obtained by looking at a default copy of **bind.keys** that ISC provides on its web site: https://ftp.isc.org/isc/bind9/keys/9.11/bind.keys.v9_11. Just copy the entry for the new key (20326) into your **trusted-keys** clause and issue an **rndc reconfig** command (or restart BIND). This causes BIND to trust the new key immediately.

A better way is to convert the **trusted-keys** clause to the newer **managed-keys** clause (see below).

Finally, there is the option of removing the clause altogether and changing **dnssec-validation yes;** to **dnssec-validation auto;**. Unless you need a trust anchor in addition to the root key, there is no reason to specify any key in the configuration file. All modern versions of BIND will use the list of root trust anchors hard-coded into them as the initial key set, and use the "managed-keys mechanism" to update them.

> **dnssec-validation yes; or dnssec-validation auto;**
>
> The difference between **dnssec-validation yes;** and **dnssec-validation auto;** is explained in chapter 6 of the BIND Administrator Reference Manual. Note that if you have used the former then you will need to take manual action to handle the root key rollover.

4. managed-keys clause in the BIND configuration file
Recent versions of BIND have introduced the **managed-keys** clause, which should nowadays always be used in preference to the **trusted-keys** clause. The difference between them is that **trusted-keys** requires manual maintenance of key changes, whereas the update of trust anchors defined by **managed-keys** will happen automatically: BIND queries the zones associated with the trust anchors on a regular basis and uses the algorithm described in RFC 5011 to update both the keys and their status.

Progress of the Rollover

As BIND updates the keys, the configuration file is not changed. Instead, BIND writes updated key information to files in BIND's working directories (specified with the **directory** and/or **managed-keys-directory** options in your configuration file). You can check the progress of the rollover by looking at these files or, more conveniently, executing the **rndc secroots** (all versions of BIND) or **rndc managed-keys status** (BIND 9.11 and later) commands.

Initially, assuming that you don't have any trust anchors other than a current root key configured, executing the **rndc managed-keys status** output should result in something like:

```
    name: .
    keyid: 19036
        algorithm: RSASHA256
        flags: SEP
        next refresh: Tue, 23 May 2017 16:21:37 GMT
        trusted since: Mon, 22 May 2017 16:21:37 GMT
```

(The exact dates and times in the output will be different from those in this example.) The "trusted since" date is an indication of how long BIND has trusted that this key is the root key. When the new key is introduced BIND will see it and, within two days, add it to the list of managed keys. At this point, the listing looks something like:

```
    name: .
    keyid: 19036
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon, 29 May 2017 10:07:39 GMT
        trusted since: Mon, 22 May 2017 16:21:37 GMT
    keyid: 20236
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon, 29 May 2017 10:07:39 GMT
        trust pending: Tue, 27 Jun 2017 10:07:39 GMT
```

A read-only working directory can cause problems

If you do NOT see the new key above, and you are using **managed-keys** (or you have selected **dnssec-validation auto;**), the first thing to check is that BIND's working directories (specified by the **directory** and/or **managed-keys-directory** clauses in the configuration file) are writable by the user ID under which BIND is running.

The standard that BIND is following for this operation - described in the document RFC 5011 - states that for the key to be fully trusted, it must be present in the zone for a period of 30 days. For this reason, BIND does not fully trust the key and so marks it as "trust pending". During this 30-day period, BIND periodically checks the zone. If ever the new key is not present, BIND will forget about it - should it reappear, the 30-day timer starts again.

Assuming however that the key remains in the zone for the full "trust pending" period, BIND will eventually accept it as a trusted key:

```
    name: .
    keyid: 19036
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon,  3 Jul 2017 10:07:39 GMT
        trusted since: Mon, 22 May 2017 16:21:37 GMT
    keyid: 20236
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon,  3 Jul 2017 10:07:39 GMT
        trusted since: Tue, 27 Jun 2017 10:07:39 GMT
```

When this occurs, BIND can use either key to validate DNSSEC signatures in the root zone and so the new key is also regarded as the root trust anchor for the world's DNS system. Some time after this, the root zone will be signed by the new key and three months after that, the old KSK is revoked:

```
    name: .
    keyid: 19036
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon, 24 Jul 2017 10:07:39 GMT
        remove at: Mon, 14 Aug 2017 16:21:37 GMT
        trust revoked
    keyid: 20236
        algorithm: RSASHA256
        flags: SEP
        next refresh: Mon, 24 Jul 2017 10:07:39 GMT
        trusted since: Tue, 27 Jun 2017 10:07:39 GMT
```

Once the "remove at" date has passed the key is removed.

For assistance with problems and questions for which you have not been able to find an answer in our Knowledge Base, we recommend searching our community mailing list archives and/or posting your question there (you will need to register there first for your posts to be accepted). The bind-users and the dhcp-users lists particularly have a long-standing and active membership.

ISC relies on the financial support of the community to fund the development of its open source software products. If you would like to support future product evolution and maintenance as well having peace of mind knowing that our team of experts are poised to provide you with individual technical assistance whenever you call upon them, then please consider our Professional Subscription Support services - details can be found on our main website.