Migrating to OpenDNSSEC

It is possible to migrate a DNSSEC signed zone over to OpenDNSSEC. How to migrate your DNSSEC signed zone over to OpenDNSSEC really depends on how your current solution looks like.

The zone data is no problem. Just place a copy of the unsigned zone in the directory for unsigned zones. But the trick is to maintain the private and public keys.

When moving from one system to another, you need to exchange public keys between them in order to always have a valid DNSSEC state. There are three possible solutions:

- Export the keys
- Prepublish DNSKEY record
- Start fresh

On this Page

Export the keys

One solution is to move the key pairs and make them accessible by OpenDNSSEC. The goal is to have the key pairs available to the system using PKCS#11.

The key pairs can e.g. be stored:

- on disc (e.g. BIND .private-key format)
- on a smartcard with no PKCS#11 interface
- in an HSM

On disc

When the key pairs are stored on disc, it means that you have access to files containing the key pairs. The key pairs can be imported into your new HSM using the PKCS#11 API or any tool available from your HSM vendor.

The BIND .private-key file can be convert into the PKCS#8 file format using the tool available with SoftHSM. If you have another file format, then OpenSSL probably can help you to convert it into the PKCS#8 file format.

```
softhsm-keyconv --topkcs8 --in Kexample.com.+005+42952.private --out key.pem
```

- **--topkcs8**, To indicate that you want to convert from BIND .private-key format to PKCS#8.
- **--in <path>**, The path to the BIND .private-key file.
- **--out <path>**, A path to the temporary PKCS#8 file.

The PKCS#8 file can then be imported into the SoftHSM token (if you are using SoftHSM as your HSM).

```
softhsm --import key.pem --slot 1 --pin 123456 --label A2 --id A2
```

- **--import <path>**, The path to the PKCS#8 file that you want to import. This should point to the temporary file that you created in the previous step.
- **--slot <number>**, The key should be imported to a token. Indicate which slot it is connected to.
- **--pin <PIN>**, Provide the PIN so that we can login to the token.
- **--label <text>**, Choose an arbitrary text string. Not used by OpenDNSSEC.
- **--id <hex>**, Choose an ID of the new key pair. The ID is in hexadecimal with a variable length. It must not collide with an existing key pair.

On a smartcard with no PKCS#11 interface

Just connect a smartcard reader to your system and insert your smartcard. Then use **opensc** and **pcscd** to give it a PKCS#11 interface. Remember to protect the location where you have your smartcard reader, since the smartcard needs to be online.

On an HSM

You can either move the HSM to the new server and install it there. Or some vendors may have some functionality to export/transfer the key pairs.

Add the keys to OpenDNSSEC

Once you have the key pairs available on the system via PKCS#11, then you must add them to OpenDNSSEC. Give this command before you start OpenDNSSEC. Also make sure that the zone is properly configured with OpenDNSSEC.

```
ods-ksmutil key import --cka_id <CKA_ID> --repository <repository> --zone <zone> --bits <size> --algorithm
<algorithm> --keystate <state> --keytype <type> --time <time>
```

- **--cka_id <CKA_ID>**, Each key object in the HSM has an ID, the CKA_ID attribute. The private and public key object must have the same ID inorder for OpenDNSESC to find them. The CKA_ID of the key pair to import is indicated, in hexadecimal, by using this option. E.g. *A2*
- **--repository <repository>**, The name of the repository, from conf.xml. E.g. *softHSM1*
- **--zone <zone>**, The name of the zone. E.g. *example.com*
- **--bits <size>**, The key length, E.g. *1024*
- **--algorithm <algorithm>**, The algorithm. E.g. *5* or *7*
- **--keystate <state>**, The key state. E.g. *active* or *ready*
- **--keytype <type>**, The key type. *KSK* or *ZSK*
- **--time <time>**, The time stamp when the key entered the given state. So that OpenDNSSEC know when to change the state. E.g. *200910301000*
- **--retire <retire>**, Optional. If you set the state to active, then you may set the time when the key should be retired. E.g. *201001010000*. Otherwise will OpenDNSSEC use the key lifetime from the KASP.

The difference between active and ready is:

- **active:** Will make the key active, thus used for signing. If there already is an active key, then you will have two of them. If this is not desired, then make sure to give this command right after setup and before you start the system.
- **ready:** The key will only be published in the zone. It will become active in a future rollover, if the key parameters match the policy.

Prepublish DNSKEY record

A second alternative, when migrating a signed zone to OpenDNSSEC, is to do a manual key rollover. When moving from one system to another, you need to exchange public keys between them in order to always have a valid DNSSEC state.

The steps below will perform a manual Double-DS KSK rollover and a manual Pre-Publication ZSK rollover. There must be a period of time between each step and the system rollover; otherwise there will not be sufficient time for the information to propagate out on the Internet. The exact time depends on your setup, but it is typically between two and four weeks. Read more in the DNSSEC Key Timing draft from IETF.

1. Before the system rollover you need to:
    - Extract the DS corresponding to the KSK in the new system and publish it in the parent zone.
    - Publish the new ZSK in the old system.
    - Publish the old ZSK in the new system.
2. System rollover:
    - Re-delegate the zone in the parent zone.
3. After the system rollover you need to:
    - Remove the old DS from the parent zone.
    - Remove the new ZSK from the old system.
    - Remove the old ZSK from the new system.

Start fresh

A third solution is to start fresh. Remove any DS records from the parent zone. Stop signing your zone when the DS records are removed from the DNS caches. It is safe to remove the public keys from your zone when the signatures are not present in any DNS caches. Then transfer the zone over to OpenDNSSEC. And let OpenDNSSEC start signing it again.

Your zone will not be secured by DNSSEC during this transfer.