Zone signing with OpenDNSSEC - part 1

1. Initialize the Software "Hardware Security Module"

    Start by becoming root for this session (or use sudo when required)

    ```
    $ sudo -s
    #
    ```

    ```
    # mkdir -p /usr/local/var/lib/softhsm
    ```

    ```
    # softhsm --init-token --slot 0 --label OpenDNSSEC
    ```

    (use '1234' for both questions below):

    ```
    The SO PIN must have a length between 4 and 255 characters.
    Enter SO PIN: ****
    The user PIN must have a length between 4 and 255 characters.
    Enter user PIN: ****
    The token has been initialized.
    ```

    ```
    # softhsm --show-slots
    ```

    Create configuration files for OpenDNSSEC by making a copy
    of the samples distributed with the package:

    ```
      # cd /usr/local/etc/opendnssec
      # cp kasp.xml.sample kasp.xml
      # cp conf.xml.sample conf.xml
      # cp zonefetch.xml.sample zonefetch.xml
      # cp zonelist.xml.sample zonelist.xml
      # chmod 644 *xml
    ```

2. Change the default Policy to use NSEC instead of NSEC3:

    Edit /usr/local/etc/opendnssec/kasp.xml

    Find this section, and remove all the lines from <NSEC3> ... </NSEC3>

    ```
    <NSEC3>
        <!-- <OptOut/> -->
        <Resalt>P100D</Resalt>
        <Hash>
            <Algorithm>1</Algorithm>
            <Iterations>5</Iterations>
            <Salt length="8"/>
        </Hash>
    </NSEC3>
    ```

    ... and replace them with this single line:

    ```
    <NSEC/>
    ```

    Save & exit.

    Also, set the correct path for the libsofthsm.so in the conf.xml:

    Change

<Module>/usr/local/lib/libsofthsm.so</Module>

to

<Module>/usr/local/lib/softhsm/libsofthsm.so</Module>

    Then save & exit the file.

3. Initialize the KSM

    # ods-ksmutil setup

    WARNING This will erase all data in the database; are you sure? [y/N] y
    SQLite database set to: /usr/local/var/opendnssec/kasp.db
    fixing permissions on file /usr/local/var/opendnssec/kasp.db
    zonelist filename set to /usr/local/etc/opendnssec/zonelist.xml.
    kasp filename set to /usr/local/etc/opendnssec/kasp.xml.
    Repository SoftHSM found
    No Maximum Capacity set.
    RequireBackup NOT set; please make sure that you know the potential
        problems of using keys which are not recoverable
    /usr/local/etc/opendnssec/conf.xml validates
    /usr/local/etc/opendnssec/kasp.xml validates
    Policy default found
    Info: converting P1Y to seconds; M interpreted as 31 days, Y interpreted as 365
days

4. Install a copy of the unsigned zone for OpenDNSSEC to sign

    Earlier, we made a backup copy of our zone, before it was signed
    by BIND9. We are going to use that backup copy now and make it
    available to OpenDNSSEC.

          # cd /etc/namedb/master
          # cp mytld.backup /usr/local/var/opendnssec/unsigned/mytld

5. Add the zone to OpenDNSSEC's database:

    # ods-ksmutil zone add --zone mytld

    zonelist filename set to /usr/local/etc/opendnssec/zonelist.xml.
    Imported zone: mytld

6. Start OpenDNSSEC!

    # ods-control start

    Starting enforcer...
    OpenDNSSEC ods-enforcerd started (version 1.3.10), pid 63495
    Starting signer engine...
    Starting signer...
    OpenDNSSEC signer engine version 1.3.10
    Engine running.

    # ps ax | grep ods

    41588 ?? SsJ   0:00.11 /usr/local/sbin/ods-enforcerd
    41593 ?? SsJ   0:00.07 /usr/local/sbin/ods-signerd

7. Check that the zone is signed

    # ls -l /usr/local/var/opendnssec/signed

    -rw-r--r-- 1 root wheel 2621 Feb 19 09:10 mytld

    Take a look at the contents of the zone - note the key ids for
    the KSK and ZSK.

    If for some reason, you don't see a file in this directory
    (/usr/local/var/opendnssec/signed/), then force the signer to sign:

    # ods-signer sign mytld

8. Moment of reflection

    Ok, so now the zone is signed with OpenDNSSEC - do notice that the
    zone was signed, but you didn't issue any commands to generate keys.

    List the keys currently managed by OpenDNSSEC:

    # ods-ksmutil key list
```
Keys:
Zone:                       Keytype:      State:    Date of next transition:
mytld                        KSK           publish   2012-09-14 09:15:09
mytld                        ZSK           active    2012-10-13 19:15:09
```

    Notice that two keys have just been created by OpenDNSSEC, on the fly.

    But BIND is still loading the zone that was signed earlier (either
    manually or using the inline signer) - can we just modify the named.conf
    definition and point to the signed zone instead ?

    Which KSK is currently being used ? And which DS record is published
    in the parent zone ?

    Would the resolvers be able to verify the signatures on the zone
    signed with OpenDNSSEC ? Why not ? What would you have to do for
    it to work (there are several possible answers)

    If you don't care about the validation problem, then you can proceed
    with the rest of this lab.

9. Tell BIND to load the new zone

    Modify /etc/namedb/named.conf, and change the zone definition for "mytld"
    so it looks like this (REMOVE auto-dnssec, etc...)

```
zone "mytld" {
    file "/usr/local/var/opendnssec/signed/mytld";  // <--- Change path
    type master;
    key-directory "/etc/namedb/keys";   // <--- Remove if there
    auto-dnssec maintain;               // <--- Remove if there
    inline-signing yes;                 // <--- Remove if there
};
```

    Now, BIND is back to being a "passive" nameserver that doesn't sign
    the zone - it just serves the zone signed by OpenDNSSEC.

Restart named:

service named restart

Check the logs in /etc/namedb/log/general to make sure that the zone is loading correctly.

Now, validation will probably fail for those trying to look up data in your zone. Wait a few minutes, and try to lookup a record in your zone:

dig www.mytld +dnssec

What do you notice ?

10. OpenDNSSEC reload BIND

Even better, you can have OpenDNSSEC tell BIND to reload the zone when it has been signed - like this, no need to manually reload.

To do this, modify /usr/local/etc/opendnssec/conf.xml

Find the lines:

```
<!--
                <NotifyCommand>/usr/sbin/rndc reload %zone</NotifyCommand>
-->
```

... remove the comments (the lines '<!--' and '-->') before and after.

Save the file, and restart OpenDNSSEC:

```
# ods-control stop
...
# ods-control start
```

11. Export the DS, ready to upload:

Verify the state of the KSK at this stage:

ods-ksmutil key list

Note the state that the KSK is in.

If it is still in publish state (see https://wiki.opendnssec.org/display/DOCS/Key+States#KeyStates-Publish for reference), then the key is, from OpenDNSSEC's point of view, not ready to be used, as it hasn't had time to propagate.

You can still export the DS record, derived from the KSK:

ods-ksmutil key export --zone mytld --ds --keystate publish >/tmp/dsset-mytld.

12. Upload the DS to the server

```
# scp /tmp/dsset-mytld. sysadm@a.root-servers.net:
```

13. Notify the administrator!

    Ask the root operator to add the new DS to the root zone, and see
    how long it takes before validation starts working again for your zone.

    ... or use the RZM web interface https://rzm.dnssek.org/

14. What's with the keystate ?

    Why is the key in Publish state ? Why is OpenDNSSEC reluctant to let us
    use the key right away ?