Practical exercises

OpenDNSSEC training

Lab introduction

[Wait for the presentations: Opening and Uploading the DS RR]

You as a student will have two servers. One will be configured to be the resolver and the other will be the name server hosting your zone.

- resolverX.odslab.se
- nsX.odslab.se

X is the group number given to you by the teacher. The IP-addresses of the resolver and the name server are configured directly in the *odslab.se* zone, and can be used from the beginning.

You will also be working with the domain *groupX.odslab.se*. This domain is however not present in DNS yet, because that will happen later in the lab.

Connecting to a server

User SSH or PuTTY to connect to your server. The private key will be given to you by the teacher.

SSH:

```
chmod 400 student.odslab.se.pem
ssh -i student.odslab.se.pem ubuntu@<ADDRESS>
```

PuTTY:

Enter address to server.
Use the student.odslab.se.ppk for authentication.
Login as “ubuntu”

Documentation

Documentation can be found online: www.opendnssec.org

Lab 1 – Setup of a secure resolver

[Wait for the presentation: DNSSEC introduction]

The purpose of the lab is to setup the resolver on the first server.

1. Connect to the server (resolverX.odslab.se) by using SSH or PuTTY.
2. Change the host name.

```
sudo hostname resolverX.odslab.se
```

3. Login and logout to get an updated command prompt.
4. Install BIND as the resolver.

```
sudo apt-get update
sudo apt-get install bind9
```

5. Change the configuration in BIND so that it only listens on the localhost. Also enable DNSSEC validation.

```
sudo pico /etc/bind/named.conf.options

    options {
              ...
              listen-on-v6 { ::1; };
              listen-on { 127.0.0.1; };
              dnssec-enable yes;
              dnssec-validation yes;
    };

sudo service bind9 restart
```

6. Change the information in the resolv.conf file in order to redirect the queries to the internal resolver.

```
sudo pico /etc/resolv.conf

    nameserver 127.0.0.1
```

7. Write protect the resolv.conf file.

```
sudo chattr +i /etc/resolv.conf
```

8. Test the resolver by using dig.

```
dig www.opendnssec.org
dig www.trasigdnssec.se
```

9. Test the resolver by using dig and DNSSEC. Notice that the DO-flag is set but not the AD-flag.

```
dig +dnssec www.opendnssec.org
```

10. To actually do DNSSEC validation, we need to have the trust anchor to the DNS root. First we start by fetching the key by using DNS.

```
dig +multi +noall +answer . DNSKEY > root.key
```

11. The root key must be verified from a trusted source. You can find the DS on https://www.iana.org/dnssec/ and compare it with the one below.

```
dnssec-dsfromkey -f root.key . > root.ds
cat root.ds
```

12. Add the key marked with 257 to BIND.

```
sudo pico /etc/bind/named.conf.options
```

```
    managed-keys {
              "." initial-key 257 3 8
              "<Insert base64 public key from file root.key>";
    };
```

```
sudo service bind9 restart
```

13. Verify by using dig. Notice that the AD-flag is now set.

```
dig +dnssec www.opendnssec.org
```

14. Also try resolving a domain where DNSSEC is broken.

```
dig www.trasigdnssec.se
```

Lab 2 – Install the DNSSEC server

[Wait for the presentations: About OpenDNSSEC, Installing OpenDNSSEC]

This lab will prepare the second server by installing the latest software.

1. Connect to the server (nsX.odslab.se) by using SSH or PuTTY.
2. Change the host name.

```
sudo hostname nsX.odslab.se
```

3. Login and logout to get an updated command prompt.
4. Start by installing the build tools and BIND (will be used as a name server)

```
sudo apt-get update
sudo apt-get install automake autoconf libtool \
                     build-essential bind9
```

5. Continue to install the DNS-library used for parsing and handling the DNS data. Your teacher will provide you with the version number to use.

```
sudo apt-get install libssl-dev
wget http://www.nlnetlabs.nl/downloads/ldns/ldns-<version>.tar.gz
tar -xzf ldns-<version>.tar.gz
cd ldns-<version>/
./configure --disable-gost
make
sudo make install
cd ..
```

6. We also want to us an HSM and we are going to use SoftHSM. Your teacher will provide you with the version number to use.

```
sudo apt-get install libbotan-1.8.2 libbotan1.8-dev \
                     sqlite3 libsqlite3-dev
wget http://www.opendnssec.org/files/source/softhsm-<version>.tar.gz
tar -xzf softhsm-<version>.tar.gz
cd softhsm-<version>/
./configure
make
make check
sudo make install
cd ..
```

7. Install OpenDNSSEC.

```
sudo apt-get install libxml2 libxml2-dev libxml2-utils rubygems
sudo gem install dnsruby
wget http://www.opendnssec.org/files/source/opendnssec-<version>.tar.gz
tar -xzf opendnssec-<version>.tar.gz
cd opendnssec-<version>/
./configure
make
sudo make install
```

8. Rebuild the dynamic linker cache

```
sudo ldconfig
```

Lab 3 – Publishing an unsigned zone

An unsigned zone will be prepared and published on the name server.

1. Create some directories.

```
sudo mkdir /var/cache/bind/zones
sudo mkdir /var/cache/bind/zones/unsigned
sudo mkdir /var/cache/bind/zones/signed
```

2. Create a zone named after your group.

```
sudo pico /var/cache/bind/zones/unsigned/groupX.odslab.se
```

```
$TTL 60
@       IN SOA nsX.odslab.se. test.odslab.se. (
                  2011062100 ; serial
                  360        ; refresh (6 minutes)
                  360        ; retry (6 minutes)
                  1800       ; expire (30 minutes)
                  60         ; minimum (1 minute)
                )
        IN NS nsX.odslab.se.
www     IN CNAME nsX.odslab.se.
```

3. Add the zone to BIND.

```
sudo pico /etc/bind/named.conf.local
```

```
zone "groupX.odslab.se" {
        type master;
        file "zones/unsigned/groupX.odslab.se";
};
```

```
sudo rndc reload
```

4. Try resolving the information from the resolver.

```
dig groupX.odslab.se SOA
```

Lab 4 – SoftHSM

[Wait for the presentation: Hardware Security Modules]

HSM:s are used for storing the keys securely and for acceleration. SoftHSM is a generic software-only HSM, which is perfectly suitable for us.

1. First configure the tokens in SoftHSM. Try adding a second token to the configuration.

   ```
   sudo pico /etc/softhsm.conf
   ```

2. Enumerate the slots by using the command below. The command will say that no tokens are present in the slots. This is because we do not have R/W privileges to the tokens.

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

3. You can also verify this in the syslog.

   ```
   sudo tail /var/log/syslog
   ```

4. Try again with correct privileges.

   ```
   sudo softhsm --show-slots
   ```

5. Next task is to initialize the tokens. The label should be unique for each token (e.g., “KSK” and “ZSK”).

   ```
   sudo softhsm --init-token --slot X --label <LABEL1>
   sudo softhsm --init-token --slot Y --label <LABEL2>
   ```

6. Verify the initialization.

   ```
   sudo softhsm --show-slots
   ```

Lab 5 – HSM testing and benchmarking

It is always good to verify the functionality of the HSM before starting to sign your zone. There are some tools available with OpenDNSSEC which will verify the interoperability.

1. OpenDNSSEC cannot access your tokens, which you will see with this command.

   ```
   sudo ods-hsmutil info
   ```

2. You need to configure your tokens in OpenDNSSEC. The repository name is what OpenDNSSEC uses to identify the tokens internally.

   ```
   sudo pico /etc/opendnssec/conf.xml
   ```

   ```
   <Repository name="SoftHSM-KSK">
        <Module>/usr/local/lib/softhsm/libsofthsm.so</Module>
        <TokenLabel><LABEL1></TokenLabel>
        <PIN><User PIN></PIN>
        <SkipPublicKey/>
   </Repository>
   <Repository name="SoftHSM-ZSK">
        <Module>/usr/local/lib/softhsm/libsofthsm.so</Module>
        <TokenLabel><LABEL2></TokenLabel>
        <PIN><User PIN></PIN>
        <SkipPublicKey/>
   </Repository>
   ```

3. Now it is possible for OpenDNSSEC to access the tokens.

   ```
   sudo ods-hsmutil info
   ```

4. Try generating keys and signatures.

   ```
   sudo ods-hsmutil test SoftHSM-KSK
   ```

5. There is also one tool which will perform speed tests on the HSM. It has been noted that using multiple threads on these virtual machines will decrease the performance dramatically.

   ```
   sudo ods-hsmspeed -r SoftHSM-KSK -i 10000 -s 1024 -t 1
   ```

Lab 6 – Editing the OpenDNSSEC configuration

[Wait for the presentation: OpenDNSSEC configuration]

The configuration needs to be adjusted to better fit this setup.

1. Open the configuration for editing.

   ```
   sudo pico /etc/opendnssec/conf.xml
   ```

2. The repository list was adjusted in a previous lab.
3. We will be rolling keys in a rapid pace. Thus need to lower the Enforcer Interval.

   ```
   <Interval>PT360S</Interval>
   ```

4. There is only one core on this machine and the performance was not increased by using multiple threads. We will then only use one thread in OpenDNSSEC.

   ```
   <WorkerThreads>1</WorkerThreads>
   <SignerThreads>1</SignerThreads>
   ```

5. Save the file and exit.

Lab 7 – Creating a policy

We will now create a KASP name "LAB". It will use very low values on the timing parameters, just so that key rollovers will go faster in this lab environment.

1. Open the kasp.xml file.

   ```
   sudo pico /etc/opendnssec/kasp.xml
   ```

2. Enter the new repository names for the KSK and ZSK.
3. Copy the default policy.

   ```
   Press ctrl+k until you have cut out the default policy.

   Press ctrl+u twice to paste it back.
   ```

4. Rename the policy to "LAB"

   ```
   <Policy name="LAB">
   ```

5. Set the signatures to have a short lifetime.

   ```
   <Signatures>
        <Resign>PT5M</Resign>
        <Refresh>PT45M</Refresh>
        <Validity>
             <Default>PT1H</Default>
             <Denial>PT1H</Denial>
        </Validity>
        <Jitter>PT2M</Jitter>
        <InceptionOffset>PT3600S</InceptionOffset>
   </Signatures>
   ```

6. Also lower the TTL and safety margins for the keys.

   ```
   <TTL>PT1M</TTL>
   <RetireSafety>PT5M</RetireSafety>
   <PublishSafety>PT5M</PublishSafety>
   ```

7. Set the KSK lifetime to 3 hours and the ZSK to 2 hours.

8. The values for the SOA can be found in the zone we created earlier.

```
<Zone>
     <PropagationDelay>PT1M</PropagationDelay>
     <SOA>
          <TTL>PT1M</TTL>
          <Minimum>PT1M</Minimum>
          <Serial>unixtime</Serial>
     </SOA>
</Zone>
```

9. And these values are from the parent zone.

```
<Parent>
     <PropagationDelay>PT5M</PropagationDelay>
     <DS>
          <TTL>PT1M</TTL>
     </DS>
     <SOA>
          <TTL>PT1M</TTL>
          <Minimum>PT1M</Minimum>
     </SOA>
</Parent>
```

10. Save and exit.
11. Verify that the KASP looks ok. The database has not been created yet, you will thus get a warning about this.

```
sudo ods-kaspcheck
```

12. Setup the KASP database.

```
sudo ods-ksmutil setup
```

Lab 8 – Adding the zone

Zones can be added in two ways, either by command line or by editing the zonelist.xml. We will edit the zone list in this lab.

1. Open the zonelist.xml file.

   ```
   sudo pico /etc/opendnssec/zonelist.xml
   ```

2. Uncomment the example zone.
3. Change the name of the zone and the paths to the zone files.

```
<Zone name="groupX.odslab.se">
      <Policy>LAB</Policy>
      <SignerConfiguration>/var/opendnssec/signconf/groupX.odslab.se.xml</SignerConfig
uration>
      <Adapters>
            <Input>
                  <File>/var/cache/bind/zones/unsigned/groupX.odslab.se</File>
            </Input>
            <Output>
                  <File>/var/cache/bind/zones/signed/groupX.odslab.se</File>
            </Output>
      </Adapters>
</Zone>
```

4. Save and exit.
5. Update the Enforcer database. You will get a warning that the Enforcer could not be notified. But we have not started it yet, will do that later.

   ```
   sudo ods-ksmutil update zonelist
   ```

Lab 9 – Start signing the zone

It is now time to sign the zone once the system has been configured.

1. Start OpenDNSSEC

   ```
   sudo ods-control start
   ```

2. Check the syslog to see that the two daemons started, that the signconf was generated, and that Signer Engine signed the zone.

   ```
   sudo tail -n 100 /var/log/syslog
   ```

3. Have a look on the signconf.

   ```
   less /var/opendnssec/signconf/groupX.odslab.se.xml
   ```

4. Have a look on the signed zone file.

   ```
   less /var/cache/bind/zones/signed/groupX.odslab.se
   ```

Lab 10 – Publish the signed zone

The signed zone is now just a file on disc. We have to tell BIND to use this one instead of the unsigned zone file.

1. Edit the BIND configuration. Enable DNSSEC and change the path to the zone file.

```
sudo pico /etc/bind/named.conf.options

    options {
       …
       dnssec-enable yes;
    };

sudo pico /etc/bind/named.conf.local

    zone "groupX.odslab.se" {
       type master;
       file "zones/signed/groupX.odslab.se";
    };
```

2. Reload the configuration.

```
sudo rndc reload
```

3. Tell OpenDNSSEC to notify BIND every time the zone has been signed.

```
sudo pico /etc/opendnssec/conf.xml

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

4. Restart the Signer Engine.

```
sudo ods-signer stop
sudo ods-signer start
```

5. Verify that the zone is signed on the resolver machine. Notice that the AD-flag is not set.

```
dig +dnssec www.groupX.odslab.se
```

6. Verify that DNSSEC works for this zone on the resolver machine.

```
dig +noall +answer groupX.odslab.se DNSKEY | grep "257 3" > published.key
dig +sigchase +trusted-key=published.key groupX.odslab.se SOA
```

Lab 11 – Publishing the DS RR

[Wait for the presentation: Key states]

The zone is now signed and we have verified that DNSSEC is working. It is then time to publish the DS RR.

1. Wait until the KSK is ready to be published in the parent zone.

   ```
   sudo ods-ksmutil key list -v
   ```

2. Show the DS RRs that we are about to publish. Notice that they share the key tag with the KSK.

   ```
   sudo ods-ksmutil key export --zone groupX.odslab.se --ds
   ```

3. Save the DS RRs to a file in your home directory. Ask your teacher to get them via SSH and add them in the parent zone.
4. Wait until the DS has been uploaded.

   ```
   dig @ns.odslab.se groupX.odslab.se DS
   ```

5. It is now safe to tell the Enforcer that it has been seen.

   ```
   sudo ods-ksmutil key ds-seen --zone groupX.odslab.se --keytag <TAG>
   ```

6. The KSK is now considered as active.

   ```
   sudo ods-ksmutil key list
   ```

7. Verify that we can query the zone from the resolver machine. Notice that the AD-flag is set.

   ```
   dig +dnssec www.groupX.odslab.se
   ```

Lab 12 - KSK rollover

[Wait for the presentation: Key rollovers]

The KSK rollover is usually done at the end of its lifetime. But a key rollover can be enforced before that by issuing the rollover command.

1. Check how long time it is left before the KSK should be rolled.

   ```
   sudo ods-ksmutil key list
   ```

2. We will now enforce a key rollover. If a key rollover has been initiated then this command will be ignored.

   ```
   sudo ods-ksmutil key rollover --zone groupX.odslab.se --keytype KSK
   ```

3. Wait until the new KSK is ready. It should be maximum 10 minutes. If it is longer than that, then you probably missed to adjust a value in your KASP. Update the KASP to match the LAB policy given here in the document.

   ```
   sudo ods-ksmutil key list -v
   ```

4. The DS RRs can be exported to the teacher once the new KSK is ready. Ask the teacher to upload it.

   ```
   sudo ods-ksmutil key export --zone groupX.odslab.se --ds \
                               --keystate ready > groupX.ds
   ```

5. Wait until the DS has been uploaded.

   ```
   dig @ns.odslab.se groupX.odslab.se DS
   ```

6. It is now safe to tell the Enforcer that it has been seen.

   ```
   sudo ods-ksmutil key ds-seen --zone groupX.odslab.se --keytag <TAG>
   ```

7. The new KSK is now considered as active.

   ```
   sudo ods-ksmutil key list
   ```

8. Verify that we can query the zone from the resolver machine.

   ```
   sudo rndc flush
   dig +dnssec www.groupX.odslab.se
   ```

Lab 13 – Adding a new policy

We will add a third policy named “LAB2”. It will use NSEC and RSASHA512 instead of NSEC3 and RSASHA256.

1. Open the kasp.xml file.

   ```
   sudo pico /etc/opendnssec/kasp.xml
   ```

2. Copy the policy named “LAB”.

   ```
   Press ctrl+k until you have cut out the policy.

   Press ctrl+u twice to paste it back.
   ```

3. Rename the policy to “LAB2”

   ```
   <Policy name="LAB2">
   ```

4. Change to NSEC.

   ```
   <Denial>
         <NSEC/>
   </Denial>
   ```

5. Change KSK and ZSK algorithm to RSASHA512. You can find the algorithm number on http://www.iana.org/assignments/dns-sec-alg-numbers/
6. Save and exit.
7. Verify that the KASP looks ok.

   ```
   sudo ods-kaspcheck
   ```

8. Load the new policy into OpenDNSSEC.

   ```
   sudo ods-ksmutil update kasp
   ```

Lab 14 – Adding a new zone

A second zone will be added by using the command line interface.

1. Make a copy of your current zone.

   ```
   cd /var/cache/bind/zones/unsigned/
   sudo cp groupX.odslab.se sub.groupX.odslab.se
   ```

2. Add it to OpenDNSSEC. You will get an error from *rndc*, because we have not configured BIND to know about the sub-zone. This will be done later.

   ```
   sudo ods-ksmutil zone add --zone sub.groupX.odslab.se \
        --policy LAB2 \
        --input /var/cache/bind/zones/unsigned/sub.groupX.odslab.se \
        --output /var/cache/bind/zones/signed/sub.groupX.odslab.se
   ```

3. Have a look in syslog and see that the zone gets signed and that BIND does not know about the zone.

   ```
   sudo tail /var/log/syslog
   ```

4. Add the zone BIND.

   ```
   sudo pico /etc/bind/named.conf.local
   sudo rndc reload
   ```

This zone is present on the same server as your original parent zone. Any DNS query on it will thus get an answer, but a delegation to it should be added in the parent zone. This will be fixed in the next lab.

Lab 15 – Updating the zone content

We need to create a delegation to the zone that we just created. And also make sure that include the DS RRs.

1. Wait until the KSK is ready in the new zone.

   ```
   sudo ods-ksmutil key list
   ```

2. Export the DS for the zone.

   ```
   sudo ods-ksmutil key export --zone sub.groupX.odslab.se --ds
   ```

3. Add these DS records and create a delegation to the zone. Remember to remove the TTL from the DS RRs.

   ```
   sudo pico /var/cache/bind/zones/unsigned/groupX.odslab.se

       sub IN NS nsX.odslab.se.
   ```

4. Signal OpenDNSSEC to read the zone content again.

   ```
   sudo ods-signer sign groupX.odslab.se
   ```

5. Go the resolver and query for the DS.

   ```
   dig sub.groupX.odslab.se DS
   ```

6. Tell OpenDNSSEC that the DS has been seen.

   ```
   sudo ods-ksmutil key ds-seen --zone sub.groupX.odslab.se \
                                --keytag <TAG>
   ```

Lab 16 – Migrating a signed zone

When migrating a signed zone you could either move the keys from one system to another or roll the keys between the systems. From a security perspective, it is better to roll the keys but it can be a little bit more complicated since you have to have two cooperating systems. This is described in more details in the slides. This lab will migrate the keys within one system, thus not exposing the keys with the outside world.

1. Generate some BIND keys that we will migrate from.

```
dnssec-keygen -r /dev/urandom –a NSEC3RSASHA1 –b 1024 \
    sub2.groupX.odslab.se
dnssec-keygen -r /dev/urandom –a NSEC3RSASHA1 –b 2048 -f KSK \
    sub2.groupX.odslab.se
```

2. Convert the BIND .private-files to PKCS#8 key file format.

   The first private key (ZSK) should be the result of the first dnssec-keygen command, and the second private key (KSK) should be the result of the second dnssec-keygen command.

```
softhsm-keyconv --topkcs8 --out zsk.p8 --in \
                Ksub2.groupx.odslab.se.+007+<TAG>.private
softhsm-keyconv --topkcs8 --out ksk.p8 --in \
                Ksub2.groupx.odslab.se.+007+<TAG>.private
```

3. Import the keys into SoftHSM.

```
sudo softhsm --import zsk.p8 --slot 0 --label 4321 \
             --id 4321 --pin 1234
sudo softhsm --import ksk.p8 --slot 1 --label 8765 \
             --id 8765 --pin 1234
```

4. List the keys in the HSM.

```
sudo ods-hsmutil list
```

5. Stop the Enforcer so that it will not create its own keys for this new zone.

```
sudo ods-ksmutil stop
```

6. Copy the zone data.

```
sudo cp /var/cache/bind/zones/unsigned/sub.groupX.odslab.se \
        /var/cache/bind/zones/unsigned/sub2.groupX.odslab.se
```

7. Create a policy, “LAB3”, which uses RSASHA1-NSEC3-SHA1.

```
sudo pico /etc/opendnssec/kasp.xml
sudo ods-ksmutil update kasp
```

8. Add the zone to OpenDNSSEC.

```
sudo pico /etc/opendnssec/zonelist.xml
sudo ods-ksmutil update zonelist
```

9. Import the keys. Use the current date and time.

```
sudo ods-ksmutil key import --cka_id 4321 \
     --repository SoftHSM-ZSK --zone sub2.groupX.odslab.se \
     --bits 1024 --algorithm 7 --keystate ACTIVE \
     --keytype ZSK --time 20YYMMDDHHMM00
sudo ods-ksmutil key import --cka_id 8765 \
     --repository SoftHSM-KSK --zone sub2.groupX.odslab.se \
     --bits 2048 --algorithm 7 --keystate ACTIVE \
     --keytype KSK --time 20YYMMDDHHMM00
```

10. List the keys.

```
sudo ods-ksmutil key list
```

11. Start the Enforcer.

```
sudo ods-ksmutil start
```

12. Add the zone to BIND.

```
sudo pico /etc/bind/named.conf.local
sudo rndc reload
```

13. Create a signed delegation and re-sign the parent zone.

```
sudo ods-ksmutil key export --ds --zone sub2.groupX.odslab.se
sudo pico /var/cache/bind/zones/unsigned/groupX.odslab.se
sudo ods-signer sign groupX.odslab.se
```

14. Verify that it works from the resolver.

```
dig +dnssec www.sub2.groupX.odslab.se
```

Lab 17 – Zone transfers

TODO

Lab 18 – Tools to test DNS and DNSSEC

[Wait for the presentation: Testing]

There are various tools where you can test DNS and DNSSEC.

1. DNSCheck is a program that was designed to help people check, measure and hopefully also understand the workings of the Domain Name System, DNS.

   http://dnscheck.iis.se/?setLanguage=en

2. DNSViz is a DNS visualization tool.

   http://dnsviz.net/

3. OARC's DNS Reply Size Test Server

   https://www.dns-oarc.net/oarc/services/replysizetest

4. OARC's source port test

   https://www.dns-oarc.net/oarc/services/porttest

5. A tool which verifies the DNSSEC chain

   http://dnssec-debugger.verisignlabs.com/

Lab 19 – Integration

[Wait for the presentation: Integration]

There are three areas to think of when integrating OpenDNSSEC with your environment.

- Adding / removing zones
- Zone distribution
- Sending the public key to the parent zone

The first two have been covered in previous labs. This lab will cover the third area by using a simple script.

1. Create a simple script.

```
sudo pico /var/opendnssec/dnskey.pl
```

```
#!/usr/bin/perl
my $fh;
open($fh, '>>', "/var/opendnssec/dnskey.txt");
print $fh <STDIN>;
close $fh;
exit(0);
```

2. Make it executable.

```
sudo chmod +x /var/opendnssec/dnskey.pl
```

3. Configure OpenDNSSEC to run this script.

```
sudo pico /etc/opendnssec/conf.xml
```

```
<Enforcer>
  …
  <DelegationSignerSubmitCommand>/var/opendnssec/dnskey.pl
  </DelegationSignerSubmitCommand>
</Enforcer>
```

4. Restart the Enforcer.

```
sudo ods-ksmutil stop
sudo ods-ksmutil start
```

5. Enforce a key rollover.

```
sudo ods-ksmutil key rollover --zone sub2.groupX.odslab.se \
                              --keytype KSK
```

6. You will get a DNSKEY in your file once the new KSK is ready.

```
sudo ods-ksmutil key list --zone sub2.groupX.odslab.se
```

Lab 20 – Useful information

It is always good to follow the latest discussions and development within DNSSEC and OpenDNSSEC.

1. OpenDNSSEC Announce Mailing List

   https://lists.opendnssec.org/mailman/listinfo/opendnssec-announce

2. OpenDNSSEC User Mailing List

   https://lists.opendnssec.org/mailman/listinfo/opendnssec-user

3. DNSSEC Deployment Mailing List

   https://dnssec-deployment.org/mailman/listinfo/dnssec-deployment

4. CircleID

   http://www.circleid.com/topics/dnssec

[Wait for the presentations: Monitoring, DRP, Operational Practices, and Closing]