Manual Key Rollover Exercise

OBJECTIVE

We are going to roll the KSK for the zones we have just signed.

REMINDERS

- we are keeping our keys in /etc/namedb/keys/

- we currently have two or more keys in that directory, one KSK and one or more ZSKs.
  Each key is represented by two files, one ending in ".key" (the public key) and one ending in ".private" (the private key)

- there is a DS RRSet in the "root" zone corresponding to our KSK

KSK ROLLOVER

The process is rather similar to the ZSK rollover:

1. Go to the key dir:

```
$ cd /etc/namedb/keys/
$ ls K*
```

2. Just like in step 2 of the ZSK rollover, generate a new KSK
You will need to use the "-f KSK" parameter to dnssec-keygen:

```
$ dnssec-keygen -f KSK -a RSASHA256 -b 2048 -n ZONE mytld
```

This will output something like:

```
Kmytld.+008+54511
```

3. Calculate a DS RRSet for the new KSK.

```
$ cd /etc/namedb/keys/
$ sudo dnssec-dsfromkey Kmytld.+008+54511.key > dsset-mytld-54511.
```

(here 54511 is just the ID of the new KSK so we know which DS is which).

At this stage, we can decide to do the rollover in one of two ways:

- Double signature

  We introduce a new KSK in to the DNSKEY RR set, and we will sign the ZSK with *both* the current ("old") KSK, and the new KSK. When a sufficient amount of time has elapsed (propagation time, TTL, etc.), we then substitute the DS record in the parent zone with that of the new KSK. Validators will have both KSKs in the cache, and the chain of trust can be validated using the new DS (trust anchor) in the parent.

- Pre-publish

  We submit the DS for the new KSK immediately to the parent zone, and have it published alongside the existing one. After a sufficient amount of time has elapsed, we replace the current ("old") KSK with the new one (and proceed to sign the ZSK with the new KSK). Validators will by then have both DS in the cache, and the chain of trust can be validated.

Of the two methods above, the double signature tends to be preferred as it doesn't require that the parent be able to handle multiple DS records for each child zone. Also, although this is perfectly valid, the extra DS with no (yet) published corresponding KSK in the child zone can cause some tools to issue warnings. And, as pointed out in point 12 below, pre-publishing requires two interactions with the parent (introduce new DS, retire old DS) while the double signature method only requires one (swap).

* Method 1: Double signature KSK rollover

4. Add the new KSK to the zone (edit the file):

   From this:

```
$include "/etc/namedb/keys/Kmytld.+008+52159.key"; // KSK
```

   To this:

```
$include "/etc/namedb/keys/Kmytld.+008+52159.key"; // KSK old
$include "/etc/namedb/keys/Kmytld.+008+54511.key"; // KSK new
```

   Remember to increment the serial number too.

5. Let's sign the zone with the old and new KSK (only the ZSK will be signed by both KSKs)

```
  $ cd /etc/namedb/keys
  $ sudo dnssec-signzone -o mytld -k Kmytld.+008+oldksk -k Kmytld.+008+newksk
../master/mytld Kmytld.+008+zsk

  $ sudo rndc reload mytld
```

6. Check with dig

```
  $ dig @127.0.0.1 dnskey mytld +multi
  $ dig @127.0.0.1 dnskey mytld +dnssec +multi
```

7. Log into RZM and click "Update". You should notice that RZM has discovered your new KSK. Verify that the DS record(s) match the contents of the dsset-mytld-newksk file created above.
   If so, click on SHA256 "eye" to mark as good then mark the old ksk DS record for deletion. Then click "Update".

8. Check with dig - both before and after the TTL expire (e.g., 2 x max TTL of mytld zone and DS record)

```
  $ dig dnskey mytld +multi
  $ dig dnskey mytld +dnssec +multi
```

9. Remove the OLD KSK to the zone (edit the file):

   From this:

```
$include "/etc/namedb/keys/Kmytld.+008+52159.key"; // KSK old
```

```
$include "/etc/namedb/keys/Kmytld.+008+54511.key"; // KSK new
```

To this:

```
$include "/etc/namedb/keys/Kmytld.+008+54511.key"; // KSK new
```

Remember to increment the serial number too.

10. Let's sign the zone with only the new KSK

```
  $ cd /etc/namedb/keys
  $ sudo dnssec-signzone -o mytld -k Kmytld.+008+newksk ./master/mytld Kmytld.
+008+zsk

  $ sudo rndc reload mytld
```

11. Check with dig - both before and after the TTL expire
(e.g., 2 x max TTL of mytld zone and DS record)

```
  $ dig dnskey mytld +multi
  $ dig dnskey mytld +dnssec +multi
```

12. Note that double signing requires only one interaction with the parent
while pre-publishing requires two.

* Method 2: Pre-publish KSK rollover

4. Upload the dsset for your zone, using the web interface or using
SCP as shown by the root instructor

Tell an instructor that you have submitted a new DS RRSet, and that
you would like it to be added to the "root" zone. If you used the
web interface, this should have happened automatically.

If using web interface, login as before.

Under the "Edit Trust Anchor Details" section enter the Key Tag,
Digest, Algorithm, and Digest type from the output of
step 3 above. E.g.,

```
   mytld. IN DS 54511    8          2      983F33D43D1EBB069BF60...
                TAG  Algorithm Digest-Type Digest
                     RSASHA256
```

Make sure to eliminate any spaces from the Digest and note that you
only need one trust anchor.

Click "Update" when done. Wait a minute for update to propagate.

5. Double check that the new DS is published in the parent (root) zone
alongside the existing one (you should wait at least 2 x TTL
until all the caches are updated):

```
   $ dig @10.20.0.230 DS mytld
   ...
   ;; ANSWER SECTION:
   mytld    900 IN  DS 52159 8 2 31F1...
   mytld    900 IN  DS 54511 8 2 983F...  // <-- the new KSK
   ...
```

Since both DS are now present in the cache, we can roll our KSK.

Then we add the new KSK to the zone (edit the file), and we comment out (remove) the old KSK:

From this:

```
$include "/etc/namedb/keys/Kmytld.+008+52159.key"; // KSK
```

To this:

```
;$include "/etc/namedb/keys/Kmytld.+008+52159.key"; // KSK old
$include "/etc/namedb/keys/Kmytld.+008+54511.key"; // KSK new
```

Remember to increment the serial number too.

... notice how we simply get rid of the old KSK - we don't need it - both DS records are there, so it's enough to have only one KSK, since we already "know" about its DS "on the internet".

6. Let's sign the zone with the new KSK

```
  $ cd /etc/namedb/keys
  $ sudo dnssec-signzone -o mytld -k Kmytld.+008+54511 ../master/mytld Kmytld.
+008+45000

  $ sudo rndc reload mytld
```

7. Check with dig - both before and after the TTL expire (or cache flush)

```
  $ dig dnskey mytld +multi
  $ dig dnskey mytld +dnssec +multi
```

8. Tell an instructor that you would like the original DS resource records to be removed from the "root" zone (or remove it yourself using the web interface)

9. Sit back and reflect on what an involved and annoying process this was, and how much better things would be if all your key rollovers were managed automatically.