Manual Key Rollover Exercise

OBJECTIVE

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

PLEASE make note of the KSK/ZSK IDs and write them down on a piece of paper as you work to remember which is which.

REMINDERS

- we are keeping our keys in /etc/namedb/keys/
- we currently have two pairs of keys in that directory, one ZSK and one KSK. Each pair 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

ZSK ROLLOVER

1. Take a look at what keys we have already generated. Make a note of the names of the files containing the current ZSK and KSK.

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

2. Generate a new ZSK, which we will use to replace the old one.

```
$ sudo dnssec-keygen mytld    <---- replace mytld with the name of your zone
```

Make sure all the keyfiles are readable by the named process:

```
$ sudo chown bind K*
$ sudo chmod u+r K*
$ ls
```

You should now have a third key pair in the directory. If you check the DNSKEY RDATA, you should see the flags field is 256 (i.e. this is a ZSK, not a KSK). Make a note of the name of the file containing the new ZSK.

3. Take a look at your current DNSKEY RRSet.

```
$ dig mytld dnskey
```

Your zone should contain one KSK and one ZSK (check the flags to distinguish between them).

We need to add the new key to the zone, so it gets included in the next signing. At the end of the file "mytld", ADD the new key:

```
$include "/etc/namedb/keys/Kmytld.+005+45000.key";
```

Increment the serial number.

Save the file and exit

4. Re-sign your zone to get the new ZSK signed, but we will NOT sign using
   the new ZSK - we only want the new ZSK to be signed by the current ZSK.
   This is called a "pre publish".

```
  $ cd /etc/namedb/keys
  $ sudo dnssec-signzone -o mytld -k Kmytld.+005+46516 ../master/mytld Kmytld.
+005+36390
```

  Notice in the above example that we are only using the current ZSK
  to sign, *NOT* the new one - this is to make sure that dnssec-signzone
  doesn't try to sign with both ZSKs. It wouldn't be "bad", but it would
  mean twice the data in the zone!

  So we tell dnssec-signzone exactly which keys to use when doing a
  rollover, PRECISELY because you want to control the timing of when
  a key is introduced, used to sign, and finally retired.

  The output of the above command should be:

```
Zone signing complete:
Algorithm: RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                    ZSKs: 1 active, 1 stand-by, 0 revoked
mytld.signed
```

  Notice the ZSKs: 1 active, 1 stand-by

5. See what difference this has made to the zone.

```
  $ sudo rndc reload mytld
  $ dig mytld dnskey
  $ dig mytld dnskey +dnssec
  $ dig mytld soa +dnssec
```

Your zone should now contain one KSK and two ZSKs; both ZSKs should be
present in the DNSKEY RRSet, which should be signed by the KSK.

BUT the SOA record (and other RRSets in the zone) should ONLY be signed once,
using the old ZSK. And the DNSKEY RRset should show all 3 keys (1 KSK, 2 ZSKs).
This is called "pre-publish".

At this time, we should in principle wait 2 x TTL for both ZSKs to
show up in everyone's cache (by default it is 120 seconds, or 2 minutes,
in our lab, but this will be different "in real life"). Anyways, let's
wait for at least 2 minutes before we sign with the new ZSK instead of the
old ZSK.

After a few minutes, ask one of your neighbors if they can lookup the DNSKEY
for your domain. They can check the in-class cache (10.20.0.230) and,
if they have configured it, their own cache.

Again, the command to lookup the keys is:

```
  $ dig mytld dnskey
```

Once we are certain that "all the internet" (everyone in the class)
can see both keys, we can sign with the new ZSK.

6. Sign with the new ZSK.

Remember, we have 3 keys - in our zone, we have:

```
$include "/etc/namedb/keys/Kmytld.+005+46516.key"; // KSK
$include "/etc/namedb/keys/Kmytld.+005+36390.key"; // ZSK we retire
$include "/etc/namedb/keys/Kmytld.+005+45000.key"; // new ZSK
```

Increment the serial number. Then:

```
$ cd /etc/namedb/keys
$ sudo dnssec-signzone -o mytld -k Kmytld.+005+46516 ../master/mytld Kmytld.
+005+45000
```

... Notice how we now use 45000 (second ZSK) to sign, not 36390 anymore

Now, reload the zone to propagage the changes

```
$ sudo rndc reload mytld
```

Check with dig like in step 5 that you are seeing only ONE signature for your RRsets - which means we are only signing using ONE ZSK - you still have to wait for the TTL to expire before you can retire the old ZSK.

7. Now you should notice, using dig like in step 5, that we are only signing with one key

```
$ dig www.mytld +dnssec
```

But also verify that the OLD ZSK is still published in the DNSKEY RRset:

```
$ dig mytld dnskey
```

You should still see three keys.

8. Retire the old ZSK.

After waiting at least 2 minutes (120s), retire the old ZSK:

```
$ cd /etc/namedb/master/
```

Edit the zone file and add a comment sign (';') in front of the old ZSK (double check which key!)

```
$ sudo ee mytld
```

```
$include "/etc/namedb/keys/Kmytld.+005+46516.key";  // KSK
;$include "/etc/namedb/keys/Kmytld.+005+36390.key"; // ZSK (commented out)
$include "/etc/namedb/keys/Kmytld.+005+45000.key";  // new ZSK
```

Increment the serial number.

Now resign the zone, but you will notice that we explicitly DON'T specify the ZSK we just commented:

```
$ cd /etc/namedb/keys
$ sudo dnssec-signzone -o mytld -k Kmytld.+005+46516 ../master/mytld Kmytld.
+005+45000
$ sudo rndc reload mytld
$ tail /etc/namedb/log/general
```

9. Like in the step 5, check that signatures still work, and that
   the OLD KZK is no longer in the RRset

   Also, check the RRSIGs (dig +dnssec soa mytld) in your zone show the
   key ID of the new ZSK.

   Does your domain still work ? :)