Automated zone INLINE signing with BIND

Remember that if you see '#' before a command, it means
you need to run this command as root, either via:

a) sudo -s

b) sudo command

We'll build on the previous labs and enable inline signing on BIND (9.9+)

When doing inline signing, the original zone is never modified: this
allows the operator to make, for example, a dump of a DB containing the
zone, and BIND will just sign it.

When the unsigned zone is updated, named detects the changes, and re-signs.

*** ON YOUR MASTER (auth1) SERVER ***

1. We're going to add a couple of statements to the BIND named.conf
   configuration file to enable inline dnssec signing.

   First, edit named.conf under /etc/namedb/, and make the following changes:

```
zone "mytld" {
  file "/etc/namedb/master/mytld";    // <--- remove ".signed", if there

  type master;
  allow-transfer { key mydomain-key; };

  key-directory "/etc/namedb/keys";   // <--- Add this if not done
  auto-dnssec maintain;               // <--- Add this if not yet done
  inline-signing yes;                 // <--- Add this

  // update-policy local;             // <--- Remove if it's there
};
```

   Save and exit.

2. Preparing the keys

   If you've done the manual signing lab from before, you have already
   generated keys, and we can reuse those. Otherwise, we'll generate
   a new set of keys.

   a) If you already have keys (otherwise go to step b)

   We need to make sure the directory has the right permissions - since BIND
   will be managing this, it needs access to the files and the directory:

```
  $ sudo chown -R bind /etc/namedb/keys
```

   Let's look at the keys, listed by time (oldest to newest)

```
  $ cd /etc/namedb/keys/
$ ls -ltr Kmytld*
-rw-r--r--  1 bind  wheel   591 Feb 18 15:52 Kmytld.+008+52159.key
```

```
-rw-------  1 bind  wheel  1774 Feb 18 15:52 Kmytld.+008+52159.private
-rw-r--r--  1 bind  wheel   417 Feb 18 15:52 Kmytld.+008+51333.key
-rw-------  1 bind  wheel  1010 Feb 18 15:52 Kmytld.+008+51333.private
```

If you have extra ZSK and KSK from manual key rollover exercizes, delete the oldest ZSK and KSK. Make sure to leave just one KSK and one ZSK. If you delete the wrong ones, reconfig with the web interface (or submit a new DS via scp!)

b) If you don't have keys yet:

```
$ sudo mkdir -p /etc/namedb/keys
$ sudo chown -R bind /etc/namedb/keys
$ cd /etc/namedb/keys
```

- Generate first key pair (Zone Signing Key)

```
$ sudo dnssec-keygen -a RSASHA256 mytld
```

... will output something like:

```
Generating key pair.....................+++++ + ....
Kmytld.+005+51333)
```

- Generate second key pair (Key Signing Key)

```
$ sudo dnssec-keygen -f KSK -a RSASHA256 mytld
Kmytld.+005+52159
```

(once again, some output will show)

Check that the keys are there:

```
$ ls -l Kmytld*
```

Notice that we don't specify any flags such as algorithm, key size, etc... We're using the defaults for now.

3. Now let's take care of the zone file

If you have made a backup of your zone file, let's copy it back over our zone, to start fresh:

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

Note the serial number in "mytld.signed"

```
$ sudo cp mytld.backup mytld
```

Increment the serial in "mytld" (which we just restored from the backup) to be higher than what we noted above.

Remove the old .signed zone - BIND will create that automatically!

```
$ sudo rm mytld.signed
```

Again, remember to check in named.conf, that you are loading "mytld", and *NOT* "mytld.signed".

We also need to make sure BIND can write in the master directory:

```
$ sudo chown bind /etc/namedb/master
```

4. Now reconfig the nameserver

```
$ sudo rndc reconfig
```

At this point you should see some new files appear in the master/ dir:

```
$ cd /etc/namedb/master
$ ls -l
```

```
...
-rw-r--r--  1 root  wheel   497 Sep 13 14:56 mytld
-rw-r--r--  1 root  wheel   497 Sep 12 09:49 mytld.backup
-rw-r--r--  1 bind  wheel   512 Sep 13 15:04 mytld.jbk
-rw-r--r--  1 bind  wheel  1331 Sep 13 15:04 mytld.signed
-rw-r--r--  1 bind  wheel  3581 Sep 13 15:04 mytld.signed.jnl
...
```

Check that signing did work:

```
$ sudo rndc signing -list mytld
Done signing with key 52159/RSASHA256
Done signing with key 51333/RSASHA256
```

Also look in the logs:

```
$ less /etc/namedb/log/general
```

```
13-Sep-2012 15:04:27.444 reloading configuration succeeded
13-Sep-2012 15:04:27.450 zone mytld/IN (unsigned): loaded serial 2012022301
13-Sep-2012 15:04:27.451 any newly configured zones are now loaded
13-Sep-2012 15:04:27.471 zone mytld/IN (signed): loaded serial 2012022301
13-Sep-2012 15:04:27.493 zone mytld/IN (signed): receive_secure_serial: unchanged
13-Sep-2012 15:04:27.501 zone mytld/IN (signed): reconfiguring zone keys
13-Sep-2012 15:04:27.544 zone mytld/IN (signed): next key event: 13-Sep-2012
16:04:27.501
```

```
$ dig @localhost mytld NS +dnssec
```

Note that the signed zone is not stored in a human readable format.

To see the contents of the signed zone, one can either do a zone transfer (axfr) or:

```
$ sudo named-checkzone -D -f raw -o - mytld mytld.signed | less
```

5. Changes to the zone

So how do we update the zone and resign it ? Simple!

Let's modify the zone and add a "mail" record with the IP address of the auth1 server:

```
mail        A           10.20.XX.1        ; X is your group
```

So edit the zone file "mytld" and add the line above.

Remember to update the serial!

Now, reload the zone. named will be automatically resign the zone:

```
$ sudo rndc reload mytld
```

Wait a few seconds, then:

```
$ tail /etc/namedb/log/general
```

What do you observe ?

```
$ dig @localhost mail.mytld a
$ dig @localhost mytld soa
```

- Do the above tests using your own resolver (10.20.X.3)
- Also try using the class resolver (10.20.0.230)

You should be able to resolve "mail.mytld" in all cases.

Notice the serial!

6. If you haven't already uploaded the DS record in a previous lab, it's time to communicate it to your parent (the root). Otherwise, you can skip the rest of this lab!

   (DS = digest fingerprint of the Key Signing Key).

   Generate a "DS" from your key:

   Find which key is the key signing key:

   ```
   $ cd /etc/namedb/keys
   $ more Kmytld*key
   ```

   Look at which one has "IN DNSKEY 257". Find the "keyid" and replace the string "+008+52159" below with "+008+keyid" where "keyid" is the number displayed.

   ```
   $ sudo -s         # We need to be root here!
   # dnssec-dsfromkey Kmytld.+008+52159 >dsset-mytld.
   # exit
   $
   ```

   REMEMBER the dot!

7. Upload the dsset for your zone (containing the hash of your zone) to the ROOT server.

   a) If using the RZM:

   Log into the RZM classroom web site at https://rzm.dnssek.org/ using your username (your domain name) and password.

   Check to see under Trust Anchor Details that your DS has automatically appeared AND matches. It is NOT automatically activated - the only thing

the the RZM has done is "grab" the key from you and is waiting for your confirmation to enable the DS in the parent zone.

If not, note that you can always add the DS record manually: cut-and-paste the tag/digest data into the proper fields. Then click "Update" to make the change.

The DS will automatically be included and signed shortly.

b) If not using the RZM:

```
$ scp dsset-mytld. sysadm@a.root-servers.net:
```

The password is the same as in class

Tell the instructor you have done so!

The instructor will include the DS-set in the root and re-sign the zone

8. You should be able to verify this:

```
$ dig @a.root-servers.net DS mytld.
```

And, doing:

```
$ dig @10.20.X.3 +dnssec DNSKEY mytld.
or
$ dig @10.20.0.230 +dnssec DNSKEY mytld.
```

should show the "AD" flag bit set indicating the that the validating resolvers were able to successfuly create a chain of trust to the root.

Optional:

If using the RZM, You may also view the MONITOR clasroom web site in a few minutes to see if it has detected your newly signed TLD:

http://monitor.dnssek.org/