LAB: securing nameserver (DNSSEC)

Lab Environment:

The workshop wifi:

SSID: workshop

PASS: iij/2497

Hosts - Virtual machines (Ubuntu16.04LTS/LXC):

Hostname: nsXX.local

IPv6: fd00:2497:1::X

IPv4: 10.0.0.X

Note: XX is your group id

For group1, it's ns01.local, fd00:2497:1::1 and 10.0.0.1

For group10, its' ns10.local, fd00:2497:1::10 and 10.0.0.10

username: workshop

password: iij/2497

Domain names:

groupXX.local

Note: XX is your gourp id

group01.local for group1, group10.local for group10, and so on

Renew ZSK (Zone signing key):

1. Create and publish a new ZSK, but still using the old ZSK to sign your zone
2. Wait for a while (at least the TTL of your DNSKEY records)
3. Start to sign your zone by using the new ZSK

4. Wait for a while (at least the maximum TTL of your records)
5. Delete the old ZSK

On bind9 dnssec-signzone's smart signing function is smart enough to handle such a task based on the various date information of keys.
Set inactivate timing and delete timing to your old ZSK file.

$ cd /etc/bind/keys

$ dnssec-settime -I +15mi -D +20mi <your old ZSK file>

Note: -I to indicate 'inactive timing', and -D for 'delete timing'. In this example, we set it as inactivate the key after 15min, and delete it after 20min from now. You can check the syntax by

$ man dnssec-settime

Note: dnssec-settime -p all <key file> will show all the timing parameters of the key file.

Then, generate a new ZSK by having publication timing as 10min before its activation timing, and getting needed parameters such as activation timing and zone name from your old ZSK by -S.

$ cd /etc/bind/keys

$ sudo dnssec-keygen -r /dev/urandom-i 10min -S < your old ZSK file>

Increase the serial value on your zone file, sign and reload it. Check for a while. The dnssec-signzone command will automatically renew your ZSK based on the timing information in the keys.

$ cd /etc/bind

$ sudo vi /etc/bind/groupXX.local

$ sudo dnssec-signzone -S -K keys groupXX.local

$ sudo rndc reload

$ dig +dnssec www.groupXX.local

Renew KSK (Key signing key):

1. Create a new KSK
2. Sign your DNSKEY by using both the new and the old KSK
3. Wait for a while (at least the maximum TTL of your records)
4. Register the new DS records to your parent zone, and delete the old DS at the same time
5. Wait for a while (at least the TTL of DS records)
6. Delete the old KSK

$ cd /etc/bind/keys

$ sudo dnssec-keygen -r /dev/urandom -a RSASHA256 -b 2048 -f ksk groupXX.local

Note: remember your KSK filename you just generated. It's needed later.

Sign your zone with your keys, then you'll see 2 active KSKs in the command result. Make sure www.groupXX.local is still resolvable.

$ cd /etc/bind

$ sudo dnssec-signzone -S -K keys groupXX.local

$ sudo rndc reload

$ dig +dnssec www.groupXX.local

Get new DS records by a dnssec-fromkey command.

$ cd /etc/bind/keys

$ sudo dnssec-dsfromkey <the new KSK file>

ssh to your web server and replace the DS records in the dsset-groupXX.local. The file now should have your new DS records only. Then ask instructors to fetch and register it.

$ ssh www.groupXX.local

$ sudo vi /var/www/html/dsset-groupXX.local.

After waiting for a while (at least the TTL of DS records), set inactivate timing and deletion timing to your old KSK file.

$ cd /etc/bind/keys

$ dnssec-settime -I +1 -D +20mi <your old KSK file>

Note: This command immediately inactivate the old KSK, and delete it 20min later from now.

Increase the serial value on your zone file, sign and reload it and check for a while. The dnssec-signzone command will automatically renew your KSK based on the timing information in the keys.

$ cd /etc/bind
$ sudo vi /etc/bind/groupXX.local
$ sudo dnssec-signzone -S -K keys groupXX.local
$ sudo rndc reload
$ dig +dnssec www.groupXX.local