Lab Exercise 10 - DNSSEC

Objective:
Deploy DNSSEC-signed zones.

Background
DNSSEC (or DNS Security Extensions) provide security to the zone files.

Note:
In the steps below, we are using
myzone.net - our domain
db.myzone.net – zonefile for the domain
Kmyzone.net.+005+12345.key/private = ZSK generated
Kmyzone.net.+005+67890.key/private = KSK generated

Steps:

A. **DNSSEC Validation.** To allow your recursive DNS servers to validate DNSSEC-signed zones.

1. Update the DNS configuration. Add options in the configuration file named.conf to allow DNSSEC.

   These options must be enabled:

   ```
   dnssec-enable yes;
   dnssec-validation yes|auto;
   ```

   `dnssec-enable` allows named to respond to DNS requests from DNSSEC-aware clients. The default is yes, but is best added in the named.conf so you know how to turn it off.

   If `dnssec-validation` is set to auto, it defaults to the DNS root zone as the trust anchor.

   If set to yes, a trust anchor must be explicitly configured using the managed-keys or trusted-keys option.

   ```
   managed-keys {
      // root key
      ". Initial-key 257 3 3 "<key-here>"
   };

   trusted-keys {
      // parent zone
      <myzone.net> 257 3 5 "<key-here>";
   };
   ```

2. Using the dig command, do a lookup for a dnssec-enabled domain. The output should show an RRSIG next to the record you asked.

```
dig @nameserver +dnssec +multiline www.apnic.net
```

Also check for the AD bit in the message header flags. It should look something like:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40679
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

B. **Signing the zone**.

1. Generate the key pair.

This command generates the ZSK.

```
dnssec-keygen -r /dev/random -a <algorithm> -b <keysize> \
-n ZONE <myzone>
```

ex:

```
dnssec-keygen -r /dev/random -a rsasha1 -b1024 -n ZONE myzone.net
```

The defaults are RSASHA1 for the algorithm, with 1024 bits for ZSK and 2048 bits for KSK. Since these are all defaults, we can just issue the command:

```
dnssec-keygen -r /dev/random <myzone>
```

This command generates the KSK

```
dnssec-keygen -a <algorithm> -b <keysize> -f KSK -n ZONE <myzone>
```

Or simply

```
dnssec-keygen -f KSK <myzone>
```

This generates 4 files.

2. Include the public DNSKEYs in the zone file.

You can either copy the entire file or reference to it using the $INCLUDE directive. To do the latter, simply add the lines below. Note that you are including only the public portion (.key) into the zone file. The private portion (.private) must be kept secure.

```
$INCLUDE "K<myzone>.+005+<id_of_zsk>.key"
```

```
$INCLUDE "K<myzone>.+005+<id_of_ksk>.key"
```

3. Sign the zone using the secret keys. The syntax is:

   dnssec-signzone [options] {zonefile} [key…]

   ```
   dnssec-signzone –o <zonename> –N INCREMENT -f <output-file> -t \
   –k <KSKfile> <zonefile> <ZSKfile>
   ```

   ex:

   ```
   dnssec-signzone –o myzone.net –N INCREMENT –f <output-file> -t -k \
   Kmyzone.net.+005+12345 db.myzone.net Kmyzone.net.+005+67890
   ```

   If **–f** is not specified, the output file will append a `.signed` in the zonefile.

   <myzone>.signed

   Smart Signing

   You can use the **–S** option so that keys will be imported into the zone automatically. Specify a keys repository in named.conf, which will be checked by named when executed.

   ```
   options {
      keys-directory { "/path/to/keys";
      };
   ```

   Then issue the command:

   ```
   dnssec-signzone –S db.myzone.net
   ```

   The output file is bigger than the original zone file. Check with the commands `ls –al` or `wc`.

C. NSEC and NSEC3.

NSEC records are created to prove the non-existence of a record. It builds a linked list of all the records in the zone file. The problem with this is it allows anyone to list the zone content. This is called "zone walking." Some tools, like the `ldns-walk` (included in the LDNS library), can be used to do exactly this.

NSEC3 can be used to provide more security. It uses a hashing algorithm to output a "hash" to replace the real domain names. This makes it difficult for an attacker, but not totally impossible.

1. Using NSEC3 to generate keys.

   To do this, you may use NSEC3RSASHA1 as your algorithm. The easier way to do this is to use -3 option instead. This option allows a few other algorithms such as RSASHA256 and RSAHSHA512 but sets NSEC3RSASHA1 as default.

```
dnssec-keygen –r /dev/random -3 <myzone>
dnssec-keygen –f ksk –r /dev/random -3 <myzone>
```

2. Sign the zone with a salt.

```
dnssec-signzone –A -3 <salt> –o <zonename> –N INCREMENT -f <output-file> -t –k <KSKfile> <zonefile> <ZSKfile>
```

   The salt is a random hexadecimal number appended to the domain before hashing. It's a public data that is part of the NSEC3PARAM record. It must be changed once in a while or on regular intervals.

   To generate the salt, you can use either of these:

```
date | sha1sum | cut –b 1-16
head –c 1000 /dev/random | sha1sum | cut –b 1-16
```

   Example:

```
dnssec-signzone –A -3 $(head –c 1000 /dev/random | sha1sum | cut –b \
1-16) –o myzone.net –N INCREMENT –f <output-file> -t -k \
Kmyzone.net.+005+12345 db.myzone.net Kmyzone.net.+005+67890
```

   To use NSEC3 without a salt, simply use a single dash.

D. Publishing the zone.

1. Reconfigure to load the signed zone. Edit named.conf and point to the signed zone. For example:

```
zone "<myzone>" {
      type master;
      # file "db.myzone.net";
      file "db.myzone.net.signed";
};
```

   Change the file to point to the signed zone.

2. Push the DS record up to your parent domain. Another output of the dnssec-signzone command is the file dsset-<yourdomain> (ex: dsset-myzone.net) which contains the DS records.

   The contents of the file look something like this:

```
myzone.net.          IN DS 4297 5 1 C5A8C518B2208463F87CB30E35F247DD7EACCDB1
myzone.net.          IN DS 4297 5 2 27E89E4A769F6C6BC889BB6F2E98374CA835D2B8C750D5505F32144E 1E79B881
```

   where:

   Keytag = 4297
   Algorithm = 5
   Digest type = 1 (for the first line), 2 (for the second line)
   Digest = C5A8C518B2208463F87CB30E35F247DD7EACCDB1

   You must contact the parent zone to communicate these values to them. This can be done by sending this file, or filling up an online form provided by your parent zone (or domain registrar). In the class, send/copy the file using SCP.

   The parent zone will then include the DS record in their zonefile. The $INCLUDE statement can be used at this stage.

```
$INCLUDE "dsset-myzone.net."
```

   You may check if it has been successfully added using dig.

```
dig @nameserver +noadditional DS myzone.net | grep DS
```

3. For slave servers, the configuration is simple.

   In the configuration file, add in the options section:

```
dnssec-enable yes;

dnssec-validation auto | yes;
```

   Then edit the zone section to point to the new signed zonefile. After reload, verify that this file exists in the folder specified in the config.

```
zone "<myzone>" {
      type slave;
      masters { X.X.X.X; };
       # file "db.myzone.net";
      file "db.myzone.net.signed";
};
```

4. Check if DNSSEC is working using the dig command.

```
dig @localhost +dnssec +multiline myzone.net
```

E. **Key rollover**. This involves performing scheduled zone maintenance. There are typically two commonly used means to do this.

1. KSK rollover using Double Signing.

Double signing is the easiest way to do key rollover, but it's primarily used only for KSK rollover.

Note: when you change KSK keys, the DS record in your parent zone must be updated. Otherwise, your zone will not validate.

a. Generate a new KSK using `dnssec-keygen` (see B.1 above).
b. Insert the new key into the zone's DNSKEY RRset and use both keys for signing.

```
dnssec-signzone –o myzone.net –N increment –f <output-file> \ -k
Kmyzone.net.+005+11111 db.myzone.net Kmyzone.net.+005+67890
```

where Kmyzone.net.+005+11111 is the new KSK generated. New DS records will be added to the file dsset-<myzone.net>.

c. Send the new DS record(s) to the parent. You may need to wait until the DS is introduced and propagated and then for the TTL of the old DS to pass.
d. Remove old key and re-sign.

2. KSK rollover using Pre-Publication. You can also rollover the KSK using this method. In this method, we are publishing the new key but we will not use it for signing.

a. Generate the new KSK using dnssec-keygen (see B.1 above).

```
dnssec-keygen –K keydir –f ksk –A none <myzone.net>
rndc loadkeys example.com
```

-K key directory option in named configuration. Can be configured per zone or at the global config

b. Generate a new ZSK using `dnssec-keygen` (see B.1 above).
c. Publish both keys, but use only the old one for signing.
d. Wait at least propagation time and then the TTL of the DNSKEY RR to expire.
e. Then use dnssec-settime once you are ready to sign the zone. Use the new key for zone signing, leaving the old one published.

```
dnssec-settime –K keydir –A now Kexample.com.+005+12345

rndc loadkeys example.com
```

f. Wait for the propagation and then the maximum TTL in the old zone.
g. Set the old key to no longer sign with the key, but leaves it in the zone.

```
dnssec-settime –K keydir -I now Kexample.com.+005+12345
rndc loadkeys example.com
```

This removes all the associations to Kexample.com.+005+12345.

h. Now remove the old keys. This completely removes the keys.

```
dnssec-settime -K keydir -D now Kexample.com.+005+12345
rndc loadkeys example.com
```

Note: Changing of the ZSK can occur as often as possible without introducing changes to the parent zone.

3. Automating the Signing. Starting at Bind 9.7, meta-data is introduced into the keys.

   a. Use RNDC to sign and load keys to named. Assuming that you have already configured RNDC (see Lab 5), you only need to add the option below into named.conf

   ```
   auto-dnssec allow;
   ```

   The auto-dnssec command is used to automate the signing and key rollover. The complete options are as follows.

   ```
   auto-dnssec off; (default setting)
   auto-dnssec allow; (this enables RNDC signing)
   auto-dnssec maintain; (updates DNSSEC based on key meta-data)
   ```

   Then you can use the following commands in

   ```
   rndc loadkeys
   ```

   ```
   rndc sign
   ```

   b. If you are sure when you want to publish, activate and retire certain keys, you can use the timing options in the dnssec-keygen command.

   ```
   dnssec-keygen -P now -A now+30d -I now+2y -D \
   now+25mo example.com
   ```

   This command inserts into DNSKEY RRset now, use for signing in 30 days, retire in 2 years, deletes in 2 years 1 month.

   Here are some dnssec-keygen timing options

   -P publish

   -A activate

   -I retire

   -D delete

F. **(optional) Domain Look-aside Buffer**. If the parent zone isn't signed yet, you can use DLV. We are not going to do this in class, but this is more for information only.

1. Create an account with ISC DLV at http://dlv.isc.org. Once done, you can add a
2. Sign the zone with the –l option.

```
dnssec-signzone -l dlv.isc.org -r /dev/urandom -o myzone.net -k
Kmyzone.net.+005+67890 myzone.net Kmyzone.net.+005+12345.key
```

3. In the ISC DLV login page, add your zone. Then add the DNSKEY for the particular zone. Instructions will be provided on the page.

4. Download the ISC DLV's key from here. Then add it as a trusted key.

```
trusted-keys {
   dlv.isc.org 257 3 5 "<key-here";
   };
```

5. Enable DLV.

```
options {
   dnssec-lookaside . trust-anchor dlv.isc.org.;
   };
```

6. Test using the dig command.

```
dig +dnssec @localhost myzone.net
```

G. **DNSSEC and Dynamic Zones**.

1. Edit named.conf to point to add the update policy.

```
zone <myzone.net> {
      type master;
      update policy-local;
      auto-dnssec maintain;
      file "dynamic/example.com";
      key-directory "keys";
};
```

The above options in bold means:

auto-dnssec maintain = allows

2. Generate keys for the dynamic zones.

```
dnssec-keygen -K </path/to/keys> -r /dev/urandom <zonename>
```

```
dnssec-keygen -f ksk -K </path/to/keys> -r /dev/urandom \
<zonename>
```

3. Use RNDC to sign the zone.

```
rndc sign
rndc loadkeys
```

4. Add or remove zone contents using nsupdate.

```
nsupdate -l
> update add myzone.net DNSKEY 256 3 7 <key-here>
> send
```