Introduction to DNSSEC signing: sign a DNS zone

```
Version: 2024110800
```

Introduction

In this lab, you will set up a DNS zone that you will sign. This lab does not intend to drive you throught the complete DNSSEC signing process as it does not cover the chain of trust establishment.

Basically, you will connect to your **grpX-soa** virtual machine, create a zone file, sign and serve it in that server. You may later on set up your **grpX-ns1** and **grpX-ns2** as secondary nameservers to serve the zone to the public.

Many DNS software providers have been doing a lot to make it easier to system administrators to sign their zones with DNSSEC. In this lab, you will use BIND software to sign a zone in **grpX-soa**. There are three mathods to achieve that using BIND:

1. **manual signing**: When DNSSEC was first introduced, the only way to sign DNS data was using the `dnssec-signzone` utility; this would take an unsigned zone file and generate a new zone file containing signatures. This file would be loaded by **named** and served the same as any other zone file. Because DNSSEC signatures expire, the zone would have to be periodically resigned and reloaded.
2. **inline signing** with **auto-dnssec**: signes zones completely transparently. A server can load or transfer an unsigned zone, and create a signed version of it which answers all queries and transfer requests, without altering the original unsigned version. As the unsigned zone is updated, **named** will detect the changes that are made to it, and apply those changes to the signed version. This was introduced in BIND 9.9 and has been removed since BIND 9.19.16[1].
3. **dnssec-policy**: it replaces auto-dnssec by specifying a Key and Signing Policy (KASP) and group all KASP related configurations together, making your `named` configuration more intuitive when it comes to DNSSEC and making it easier to enable DNSSEC for your zones. Available from BIND 9.16 and later versions only[2].

In this lab, we will use the second method.

> ⚠ **Warning**
>
> In all this lab, be carefull to **always replace *X*** by your Group number in IP addresses, server name and any other place where required.
>
> Same for ***<lab_domain>*** to be replace by the domain name registered for the class.

1. Configure the zone in your grpX-soa

You will connect to your **grpX-soa** virtual machine and create a zone file for your assigned domain name.

```
$ sudo mkdir -p /etc/bind/zones
$ cd /etc/bind/zones
$ sudo touch db.grpX
```

Copy the below and paste it into your db.grpX file by and update it accordingly.

```
; grpX

$TTL    60
@       IN      SOA     soa.grpX.<lab_domain>.te-labs.training. dnsadmin.grpX.
<lab_domain>.te-labs.training. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
```

```
                          86400 )       ; Negative Cache TTL
;

; grpX
@             NS            ns1.grpX.<lab_domain>.te-labs.training.
@             NS            ns2.grpX.<lab_domain>.te-labs.training.

ns1         A           100.100.X.130
ns1         AAAA        fd--:----:X:128::130
ns2         A           100.100.X.131
ns2         AAAA        fd--:----:X:128::131
www         A           100.100.X.130
```

> **Important**
>
> Once done, it is important to checks the syntax and integrity of the zone file using the `named-check zone` command as below by replacing ZONE_NAME and ZONE_FILE with their appropriate values in your case.

```
$ named-checkzone ZONE_NAME ZONE_FILE
zone ZONE_NAME/IN: loaded serial 1
OK
$
```

Now, you will instruct BIND to load this zone file as a primary server. First, open the ***/etc/bind/named.conf.local*** configuration file.

```
$ sudo nano /etc/bind/named.conf.local
```

Then, add and update accordingly the below to the end of the configuration you are editing :

```
zone "grpX.<lab_domain>.te-labs.training" {
    type primary;
    file "/etc/bind/zones/db.grpX";
    allow-transfer { 100.100.X.130; 100.100.X.131;  };
    also-notify {100.100.X.130; 100.100.X.131; };
};
```

> **Important**
>
> Once done, use ***named-checkconf*** to verify that your BIND config is correct. If the prompt returns without any complaint, your configuration is correct. Else, you should read the output and fix the error before you move on.

Restart **named** and verify its status. You should see an output similar to the below.

```
$ sudo systemctl restart bind9
$ sudo systemctl status bind9

● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
    Drop-In: /run/systemd/system/service.d
             └─zzz-lxc-service.conf
     Active: active (running) since Fri 2024-11-08 19:50:35 UTC; 5s ago
       Docs: man:named(8)
   Main PID: 21090 (named)
      Tasks: 18 (limit: 38066)
     Memory: 8.0M
     CGroup: /system.slice/named.service
             └─21090 /usr/sbin/named -f -u bind

Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: zone localhost/IN:
loaded serial 2
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: zone grpX.
<lab_domain>.te-labs.training/IN: loaded serial 1
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: zone grpX.
<lab_domain>.te-labs.training/IN: sending notifies (serial 1)
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: zone 255.in-
addr.arpa/IN: loaded serial 1
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: zone 127.in-
addr.arpa/IN: loaded serial 1
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: all zones loaded
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: running
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: managed-keys-zone:
Key 20326 for zone . is now trusted (acceptance timer complete)
Nov 08 19:50:35 soa.grpX.<*lab_domain*>.te-labs.training named[21090]: resolver priming
query complete: success
```

Then, query your zone on the local server:

```
$ dig @localhost soa grpX.<lab_domain>.te-labs.training.

; <<>> DiG 9.18.28-0ubuntu0.20.04.1-Ubuntu <<>> @localhost soa grpX.<lab_domain>.te-
labs.training.
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59944
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 3faa151d32fe9a1601000000672e72d560885528eb4d41b0 (good)
;; QUESTION SECTION:
;grpX.<*lab_domain*>.te-labs.training.     IN      SOA
```

```
;; ANSWER SECTION:
grpX.<lab_domain>.te-labs.training. 60   IN      SOA     soa.grpX.<lab_domain>.te-labs.training. dnsadmin.grpX.<lab_domain>.te-labs.training. 1 604800 86400 2419200 86400

;; Query time: 4 msec
;; SERVER: ::1#53(localhost) (UDP)
;; WHEN: Fri Nov 08 20:21:41 UTC 2024
;; MSG SIZE  rcvd: 131
```

At this stage, your zone is being served by your **grpX-soa** virtual machine. Next step will be to sign that zone.

2. Sign the zone: inline signing with auto-dnssec method

2.1 Generate ZSK and KSK key pairs

To sign the zone using the **inline signing** with **auto-dnssec** method, you will follow the below steps.

Firstly, you will generate two pairs of key: one ZSK pair and one KSK pair in a dedicated directory.

```
$ sudo mkdir -p /etc/bind/keys
$ cd /etc/bind/keys
```

- Generate **ZSK**

```
$ sudo dnssec-keygen -a ECDSAP256SHA256 -n ZONE grpX.<lab_domain>.te-labs.training
```

- Generate **KSK**

```
$ sudo dnssec-keygen -f KSK -a ECDSAP256SHA256 -n ZONE grpX.<lab_domain>.te-labs.training
```

Change ownership for the zone and keys folders

```
$ sudo chown -R bind:bind /etc/bind/keys
$ sudo chown -R bind:bind /etc/bind/zones
```

Update your zone configuration statement in `/etc/bind/named.conf.local`, to look like the below :

```
zone "grpX.<lab_domain>.te-labs.training" {
  type primary;
  file "/etc/bind/zones/db.grpX";
  allow-transfer { 100.100.X.130; 100.100.X.131; };
  also-notify {100.100.X.130; 100.100.X.131; };
  key-directory "/etc/bind/keys";
  auto-dnssec maintain;
  inline-signing yes;
};
```

Restart BIND service and check its status to confirm it loaded and runs properly or not. You should now know how to do that (done above in this lab).

Some new files should appear in the *zones* directory.

2.2 Verify that your zone is signed.

You will first check if your keys have signed the zone as expected.

```
$ sudo rndc signing -list grpX.<lab_domain>.te-labs.training
Done signing with key 52159/RSASHA256
Done signing with key 51333/RSASHA256
```

2.3 Query your zone to confirm presence of signatures and public keys.

You can now use *dig* utility to query the zone and confirm it is signed or not.

> ⚠ **Warning**
>
> Remember that at this stage, you have only signed the zone and have not yet established the chain of trust.

QUESTION: Will you get the "ad" flag ? Why ?

1. dig DNSKEY *grpX*.<*lab_domain*>.te-labs.training @100.100.X.66
2. dig DNSKEY *grpX*.<*lab_domain*>.te-labs.training @100.100.X.66 +dnssec +multi
3. dig SOA *grpX*.<*lab_domain*>.te-labs.training @100.100.X.66 +dnssec +multi

QUESTIONS: did you get answers to your queries ? Did you receive the signatures (RRSIG) ? Did you get the "ad" flag ? Why ?

3. Extract and share the Delegation Signer (DS) with your parent

3.1 Generate the DS

Here, you will extract the DS of your zone and send it to your parent for signing and publication in his zone (chain of trust)

Execute the following command to get the DS record and save it in the required file:

```
$ sudo dnssec-dsfromkey /etc/bind/keys/KgrpX.<lab_domain>.te-
labs.training.+_XYZ+YOUR_KSK_key_tag.key > /tmp/DS_YOUR_KSK_key_tag.grpX
```

For instance, if your KSK public key file is KgrpX.<lab_domain>.te-labs.training.+013+62384.key, your command will be

```
$ sudo dnssec-dsfromkey /etc/bind/keys/KgrpX.<lab_domain>.te-labs.training.+_013+62384.key
> /tmp/DS_62384.grpX
```

Verify the content of the generated file:

```
$ cat /tmp/DS_YOUR_KSK_key_tag.grpX
```

Which should contain something similar to the below:

```
grpX.<lab_domain>.te-labs.training. IN DS DS_YOUR_KSK_key_tag 13 2
018A86C0139BA5500AC87A5BAD8FB5D8D4F9672C319B34DB5A7F3BC10A424D6E
```

3.2 Push the DS to the parent

It is now time to push the DS file to your parent. In this lab, you will use **scp** to do that.

```
$ scp /tmp/DS_YOUR-KSK-key-tag.grpX dsuser@100.64.0.53:DS/
```

Type in the "dsuser" account password if prompted. Password is the username. You should see transfer status as below:

```
DS_YOUR-KSK-key-tag.grpX        100%   106    123.6KB/s    00:00
```

> ⓘ **Note**
>
> Once you have pushed your DS to the parent, a script operating every 15 min at your parent zone will publish it automatically. You should wait the next quarter hour and rund a dig query to confirm if your DS is published.

3.3 Verify that your DS is published

Query your parent zone and confirm that they have published your DS.

```
sysadm@cli:~$ dig DS grpX.<*lab_domain*>.te-labs.training

; <<>> DiG 9.16.1-Ubuntu <<>> DS grpX.<*lab_domain*>.te-labs.training
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57805
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;grpX.<*lab_domain*>.te-labs.training.      IN      DS

;; ANSWER SECTION:
grpX.<*lab_domain*>.te-labs.training. 60    IN      DS      2404 8 2
8A4D8024E59D115331C8ECAF715E1168A429282646E6861420BEF8D1 7F9676E7

;; Query time: 320 msec
;; SERVER: 9.9.9.9#53(9.9.9.9)
;; WHEN: Fri Nov 08 00:23:17 UTC 2024
;; MSG SIZE  rcvd: 101

sysadm@cli:~$
```

At this stage, you are telling the world that your zone is signed and any validator (recursive resolver with **DNSSEC validation enabled**) should validate DNS answers from your zone. But in the context of this lab, your public nameservers have not been set up (**grpX-ns1** and **grpX-ns2**). Unless you configure them, you will not be able to see the effect of validation on your domain. Ready to configure NS1 and NS2 ?

[1] Inline Signing in ISC BIND 9.9.0: https://kb.isc.org/docs/aa-00626 ↩

[2] BIND DNSSEC Key and Signing Policy: https://kb.isc.org/docs/dnssec-key-and-signing-policy ↩