Configurer DNSSEC sur un serveur CentOS8 faisant authorité avec Bind □□□

Prérequis

Vérifier la version de *Bind* et (re)prendre connaissance des emplacements de fichier par défaut :

```
named -V
```

```
BIND 9.11.26-RedHat-9.11.26-6.el8 (Extended Support Version) <id:3ff8620>
running on Linux x86_64 4.18.0-305.17.1.el8_4.x86_64 #1 SMP Wed Sep 8 14:00:07 UTC 2021
built by make with [...]

default paths:
  named configuration:  /etc/named.conf
  rndc configuration:   /etc/rndc.conf
  DNSSEC root key:      /etc/bind.keys
  nsupdate session key: /var/run/named/session.key
  named PID file:       /var/run/named/named.pid
  named lock file:      /var/run/named/named.lock
  geoip-directory:      /usr/share/GeoIP
```

On voit ainsi que le fichier de configuration principal est `/etc/named.conf`.
Les fichiers de zones sont dans le dossier `/var/named` (l'info est dans le dossier de configuration).

Fichier de configuration

Dans la déclaration des options du fichier de configuration, ajouter les options suivantes :

```
options {
  [...]
  recursion no;
  dnssec-enable yes;
  dnssec-validation yes;
  dnssec-lookaside auto;
};
```

Génération des clés

Je me suis mis en root pour la suite des opérations...

```
sudo su
```

Terminologie

Source : DNSSEC - Signer la zone DNS de l'Active Directory #B_Terminologie

- **RRSIG** - ***R**esource **R**ecord **Sig**nature*
  - Enregistrement signé et associé à un enregistrement d'origine (ce que l'on verra dans la console DNS à la fin de la configuration)
- **NSEC3** - ***N**ext **Sec**ure **3***
  - Mécanisme pour prouver qu'un enregistrement n'existe pas (on ne retourne pas rien, mais on retourne une réponse négative signée)
- **DNSKEY** - ***DNS Key***
  - Stocker la clé publique (SHA256) pour permettre la vérification de la signature
- **DS** - ***D**elegation **S**igner*
  - Créer une délégation sécurisée (chaîne d'authentification sur les zones enfants)

Par ailleurs, il ne faut pas négliger le principe des clés KSK et ZSK :

- **Clé KSK** - ***K**ey **S**igning **K**ey*
  - Clé privée qui sert à signer les clés privées ZSK
- **Clé ZSK** - ***Z**one **S**igning **K**ey*
  - Clé privée pour signer les données d'une zone

Se rendre dans le dossier de zone

```
cd /var/named
```

Créer la clés *Zone Signing Key* (ZSK)

D'après la doc.

```
dnssec-keygen -a ECDSAP256SHA256 -n ZONE <domain.name>
```

Ce qui produit la sortie suivante :

```
Generating key pair.
K<domain.name>.+xxx+xxxxx
```

⚠ À noter la lettre `K`.

La clés publique doit être insérée dans le fichier de zone :

```
vim /var/named/zone.<domain.name>
```

```
[...]


$INCLUDE /var/named/K<domaine.name>.+xxx+xxxxx.key
```

Créer la clés *Key Signing Key* (KSK)

Problème

En continuant à suivre la documentation de *bind9*, à savoir :

```
dnssec-signzone -o <domain.name> zone.<domain.name>
```

Je me suis retrouvé avec une erreur :

```
dnssec-signzone: fatal: No self-signed KSK DNSKEY found.  Supply an active
key with the KSK flag set, or use '-P'.
```

Solution

D'après les instructions trouvées sur le site www.digitalocean.com.

```
dnssec-keygen -f KSK -a ECDSAP256SHA256 -n ZONE <domain.name>
```

```
Generating key pair.
K<domain.name>.+xxx+xxxxx
```

La clés publique doit également être insérée dans le fichier de zone :

```
vim /var/named/zone.<domain.name>
```

```
[...]


[...]
$INCLUDE K<domaine.name>.+xxx+xxxxx.key
```

Signer la zone

D'après la doc.

Pour le coup, il devient possible de signer la zone :

```
dnssec-signzone -o <domain.name> zone.<domain.name>
```

```
Verifying the zone using the following algorithms: ECDSAP256SHA256.
Zone fully signed:
Algorithm: ECDSAP256SHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                            ZSKs: 1 active, 0 stand-by, 0 revoked
zone.<domain.name>.signed
```

Ce qui crée un nouveau fichier qu'il faut renseigner dans le fichier de configuration local des zones :

```
vim /etc/named.conf.local
```

```
zone "<domain.name>" {
  type master;
  file "zone.<domain.name>.signed";
};
```

En fait, il suffit de rajouter le `.signed` au nom de zone précédemment renseigné.

Récap des fichiers créés

Le dossier de zone contient donc 6 nouveaux fichiers :

1. `K<domaine.name>.+xxx+xxxxx.key` pour la clés ZSK
2. `K<domaine.name>.+xxx+xxxxx.private` pour la clés ZSK
3. `K<domaine.name>.+xxx+xxxxx.key` pour la clés KSK
4. `K<domaine.name>.+xxx+xxxxx.private` pour la clés KSK
5. `zone.<domain.name>.signed` pour la signature de zone
6. `dsset-<domain.name>.` contenant des *DS Record*

Récupérer le `DS Record`

Deux façons de récupérer le `DS Record` pour procéder à l'enregistrement auprès de son fournisseur de zone de nom de domaine supérieur (*désolé, je ne sais pas mieux m'expliquer à l'instant*) :

dnssec-dsfromkey

```
dnssec-dsfromkey -2 K<domain.name>.+xxx+xxxxx.key
```

Où la clés correspond à la dernière clés créée, c'est à dire KSK.

cat

```
cat dsset-<domain.name>.
```

Pour ma part, j'ai choisi la première méthode afin de ne pas avoir de tabulation dans ma sortie.

Enfin

On relance `bind` :

```
systemctl restart named
```

Quitter l'utilisateur `root`...

On peut vérifier avec les sites https://dnssec-analyzer.verisignlabs.com, https://dnschecker.org, https://www.nslookup.io/...

Ou en ligne de commande :

```
dig @9.9.9.9 <domain.name>. DNSKEY +dnssec +cd +multiline
```

```
; <<>> DiG 9.10.6 <<>> @9.9.9.9 <domaine.name>. DNSKEY +dnssec +cd +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: <number>
;; flags: qr rd ra ad cd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;<domaine.name>.□IN DNSKEY

;; ANSWER SECTION:
<domaine.name>.□43200 IN DNSKEY□257 3 13 (
□□□□1WclJbuqBHhaiqL0W9EO7ZaLd9aaBe1BGXNbE4cjVqutjH1T
□□□□XZP9kFzcQopEZjwIbvv2LT18tB6GdvBfyyXxV3cA==
□□□□) ; KSK; alg = ECDSAP256SHA256 ; key id = <KSK_Id>
<domaine.name>.□43200 IN DNSKEY□256 3 13 (
□□□□RcFlnT/aqL+UNcrLlSe5DFRuX5Srhie5UYkNbnm+5J4rZE
□□□□3qRTKSQqqpKrxb4XwtdsSZ1wl5zixer3XR2ngXqPSw==
□□□□) ; ZSK; alg = ECDSAP256SHA256 ; key id = <ZSK_Id>
<domaine.name>.□43200 IN RRSIG DNSKEY 13 4 86400 (
□□□□<expiration_date> <creation_date> 20998 <domaine.name>.
□□□□XPTzd2+tTxVfhVrz2+haiqLKdfXx5QyWPwFfi+utUKTK5oPGf
□□□□1WwdRdwDy3Pu8P7Wc55ZBEaqyayM+BJsUGPp2XBnovAXEpqvQ== )
<domaine.name>.□43200 IN RRSIG DNSKEY 13 4 86400 (
□□□□<expiration_date> <creation_date> 61619 <domaine.name>.
□□□□3fac0v2znWPzv777Wc55ZBi6yTZTf7Wc55ZB4N6ckds8VjZqIlq
□□□□3kpRF3texF4bZi2zh4Bw2o1CuWPwFfi+5kJfh0lzfg== )

;; Query time: 359 msec
;; SERVER: 9.9.9.9#53(9.9.9.9)
;; WHEN: Thu Jan 20 22:08:13 CET 2022
```

```
;; MSG SIZE  rcvd: 435
```

Le *fournisseur de zone de nom de domaine supérieur* peut avoir un peu de délai avant de procéder à l'enregistrement.

Renouvellement des dates d'expiration

Manuellement

On renouvelle les clés :

```
dnssec-signzone -o <domain.name> -k /var/named/K<domain.name>.+013+<KSK_Id>.key
/var/named/zone.<domain.name> /var/named/K<domain.name>.+013+<ZSK_Id>.key
```

```
Verifying the zone using the following algorithms: ECDSAP256SHA256.
Zone fully signed:
Algorithm: ECDSAP256SHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                            ZSKs: 1 active, 0 stand-by, 0 revoked
zone.<domain.name>.signed
```

On relance Bind9 :

```
systemctl reload named.service
```

Script

```
#!/bin/bash


declare -r ZONE_KEY_DIRECTORY="/var/named/"


declare -a ZONE_DOMAINS=("<domain1.name>" "<domain2.name>")
declare -a ZONE_NAMES=("zone.<domain1.name>" "zone.<domain2.name>")
declare -a ZONE_KEYS_KSK=("K<domain1.name>.+xxx+xxxxx.key" "K<domain2.name>.+xxx+xxxxx.key")
declare -a ZONE_KEYS_ZSK=("K<domain1.name>.+xxx+xxxxx.key" "K<domain2.name>.+xxx+xxxxx.key")
```

```
cd $ZONE_KEY_DIRECTORY || exit 1

# Loop on domain name zones
for (( i=0; i < "${#ZONE_DOMAINS[@]}"; i++ ));
do
    # Search for serial number
    SERIAL=$(grep -Po '\d{10}(?= {3}; sn)' "${ZONE_NAMES[$i]}")
    if [[ ! "$SERIAL" =~ ^[[:digit:]]+$ ]]; then exit 1; fi
    SERIAL_NEW=$((SERIAL + 1))
    # Increment serial number
    sed -i "s/""$SERIAL""/""$SERIAL_NEW""/" "${ZONE_NAMES[$i]}"

    /usr/sbin/dnssec-signzone -o "${ZONE_DOMAINS[$i]}" -k "${ZONE_KEYS_KSK[$i]}" "${ZONE_NAMES[$i]}"
"${ZONE_KEYS_ZSK[$i]}"
    EXIT_CODE=$?

    if [ $EXIT_CODE -eq 0 ];
    then
        echo "Renew is done correctly for ${ZONE_DOMAINS[$i]}! (SN: $SERIAL_NEW)"
    else
        echo "Something went wrong with ${ZONE_DOMAINS[$i]}... (SN: $SERIAL_NEW)"
        exit 1
    fi
done

systemctl reload named.service
EXIT_CODE=$?
if [ $EXIT_CODE -eq 0 ];
then
    echo "Everything seems correct."
    exit 0
else
    echo "There is an error with Bind!!!"
    exit 1
fi
```

On lance un *cronjob* :

```
0 4 25 * * /bin/sh /root/.scripts/renew-dnskey.sh
```

Différentes sources ou ressources

- DNSSEC Guide — BIND 9 documentation - #Semi automatic signing
- DNSSEC Guide — BIND 9 documentation - #Setting key timing information
- DNSSEC Guide — BIND 9 documentation - #Manual signing
- How To Setup DNSSEC on an Authoritative BIND DNS Server DigitalOcean - #Modifying zone records
- DNSSEC - Signer la zone DNS de l'Active Directory

Révision #12
Créé 13 décembre 2021 22:51:11 par Mickaël G.
Mis à jour 9 août 2023 07:58:21 par Mickaël G.