Corrigé des TP DNSSEC de l'atelier IDsA

Rahim Djaffar, Bertrand Léonard, Jean-Philippe Pick, Mohsen Souissi, AFNIC <Rahim.Djaffar@nic.fr, Bertrand.Leonard@nic.fr, Jean-Philippe.Pick@nic.fr, Mohsen.Souissi@nic.fr>

Version : 1.6

Date : 2004-03-30

Table of Contents

1. Abbréviations

Voici la liste des principales abbréviations utilisées tout au long de ce corrigé :

DNSSEC	Domain Name System SE-Curity Extensions		
TSIG	Transaction SIGnature		
PKI	Public Key Infrastructure		
RR	Resource Record		
	[DNS]	A	IPv4 Address
		AAAA	IPv6 Address
		CNAME	Canonical NAME (Utilisé pour les alias)
		NS	Name Server
		SOA	Start Of Authority
		TXT	TeXT
		MX	Mail
	[DNSSEC]	NXT	NeXT
		KEY	KEY
		SIG	SIGnature
		DS	Delegation Signer
KSK	Key Signing Key		
ZSK	Zone Signing Key		
RRset	n RR avec Nom, Classe et Type identiques		
Keyset	C'est le fichier qui contient la clé publique de la zone au format RR [Resource Record]. Le Keyset est envoyé à la zone parente pour signature.		

2. Objectifs

Mise en place de DNSSEC sur une plate-forme expérimentale en 4 étapes à travers 6 TP :

1. TP1 : monter un service DNS ;
2. TP2 : sécuriser les transferts de zones entre serveurs DNS en utilisant TSIG ;
3. TP[3,4,5] :
    - monter un service DNSSEC ;
    - sécuriser les fichiers de zones DNS et créer des délégations sécurisées.
4. TP6 :
    - gestion des différentes clés ;
    - procédures de roulement des clés ZSK et KSK.

3. TP1 : pratique du logiciel BIND

3.1. Compilation du logiciel BIND

L'ISC [Internet Software Consortium] met à disposition des snapshots de BIND destinés à l'experimentation : ce sont les seules versions dans lesquelles est implémenté un support complet et tenant compte des dernières mises à jour de DNSSEC. Ce TP se base sur l'utilisation de la branche `9.3` de BIND 9, dont le dernier snapshot est bind-9.3.0s20021217. On télécharge la distribution et on l'installe comme suit :

```
[]$ wget \
    ftp://ftp.isc.org/isc/BIND9/snapshots/bind-9.3.0s20021217.tgz

[]$ tar xvfz bind-9.3.0s20021217.tgz
[]$ ./configure --with-openssl --prefix=/usr/local/bind-9.3
[]$ make
[]# make install
```

Pour expérimenter DNSSEC, il est nécessaire de préciser à BIND d'utiliser les librairies d'OpenSSL (option `--with-openssl`) pour l'utilisation d'outils cryptographiques : génération de clés et signature des zones.

L'option `--prefix` permet de spécifier le répertoire d'installation (dans la commande précédente : `/usr/local/bind-9.3`). Dans la suite de ce document, on note `$PREFIX=/usr/local/bind-9.3`. On considérera que le répertoire courant est `$PREFIX`.

Une organisation possible de l'arborescence sous `/usr/local/bind-9.3` est la suivante :

- `.dnssec/` : contient les clés symétriques et asymétriques ;
- `etc/` : contient les fichiers de configuration, notamment `'named.conf'` et `'named.root'` ;
- `master/` : contient les fichiers de zone pour lesquels le serveur est primaire ;
- `run/` : contient le pid du démon named : `'named.pid'` ;
- `slave/` : contient les fichiers de zone pour lesquels le serveur est secondaire.

Remarque : (`run/` et `slave/` doivent être accessibles en écriture pour l'utilisateur sous l'identité duquel tourne le démon named : on choisit souvent, pour des raisons de sécurité, l'utilisateur `bind`)

3.2. Test du logiciel

On utilise l'outil **dig** pour effectuer les tests (requêtes DNS). Les différentes options sont données par les commandes suivantes :

```
     []$ $PREFIX/bin/dig -h
ou   []$ man -M $PREFIX/man dig
```

Exemples d'utilisation :

```
[]$ $PREFIX/bin/dig @nscache.atelier.idsa.prd.fr \
    atelier.idsa.prd.fr SOA

[]$ $PREFIX/bin/dig @nscache.atelier.idsa.prd.fr \
    atelier.idsa.prd.fr SOA +trace
```

L'option `+trace` permet de visualiser tous les serveurs interrogés depuis la racine lors d'une requête récursive.

3.3. Création d'une zone DNS

3.3.1. Configuration de la zone

Conventions de noms pour chaque participant : on désigne par `HOSTxx`, la machine du participant `xx`. Cette machine va héberger un serveur de nom (`ns.domxx.atelier.idsa.prd.fr`, d'adresse `192.168.0.xx`) qui sera primaire pour la zone `domxx.atelier.idsa.prd.fr`

Par exemple, pour le premier participant :

```
HOST                :    01
ZONE                :    dom01.atelier.idsa.prd.fr.
NS                  :    ns.dom01.atelier.idsa.prd.fr.
Adresse IPv4 (ns)   :    192.168.0.1
```

Il s'agit de créer la zone `dom01.atelier.idsa.prd.fr` correspondant au premier participant (`HOST01`).

```
$ORIGIN dom01.atelier.idsa.prd.fr.
$TTL   172800
@      IN      SOA      ns      hostmaster.nic.fr. (
                                2003120500      ; Serial
                                3600            ; Refresh
                                900             ; Retry
                                3600000         ; Expire
                                3600            ; Minimum caching
                                )
       IN      NS       ns
ns     IN      A        192.168.0.1

$GENERATE 0-999 host${0,3,d} CNAME cname${0,3,d}
```

3.3.2. Configuration du serveur

Ci-après le fichier `'named.conf'` du serveur `ns.dom01.atelier.idsa.prd.fr` :

```
options {
        recursion no;
        notify yes;     //by default
        directory "/usr/local/bind-9.3"
        pid-file "run/named.pid";
        listen-on { any;};
        listen-on-v6 { any; };
};

zone  "." {
                type hint;
                file "etc/named.root";
};

zone "localhost" {
                 type master;
                 file "master/localhost";
};

zone "0.0.127.IN-ADDR.ARPA" {
                type master;
                file "master/localhost.rev";
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0\
      .0.0.0.0.ip6.int" {
                type master;
```

```
                file "master/localhost.rev";
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0\
      .0.0.0.0.ip6.arpa" {
                type master;
                file "master/localhost.rev";
};

zone "dom01.atelier.idsa.prd.fr"  {
                type master;
                file "master/dom01.atelier.idsa.prd.fr";
                allow-transfer { none; };
};
```

Contenu du fichier `localhost.rev` :

```
$TTL    86400
@       IN      SOA     localhost.      root.localhost. (
                        1               ; serial
                        3600            ; refresh
                        900             ; retry
                        3600000         ; expire
                        3600            ; negative cache ttl
                        )

        IN      NS      localhost.
1       IN      PTR     localhost.
```

Contenu du fichier `localhost` :

```
$ORIGIN localhost.
$TTL    86400
$TTL    86400
@       IN      SOA     localhost.      root.localhost. (
                        1               ; serial
                        3600            ; refresh
                        900             ; retry
                        3600000         ; expire
                        3600            ; negative cache ttl
                        )

        IN      NS      localhost.
        IN      A       127.0.0.1
        IN      AAAA    ::1
```

Pour finir, redémarrer le démon BIND pour charger `'named.conf'`. Sous FreeBSD, on lance la commande suivante qui démarre le démon BIND sous l'identité `bind` :

```
[]# $PREFIX/sbin/named -u bind -c $PREFIX/etc/named. conf
```

3.4. Monter un service de secondaire

Dans cette partie, on considère le cas de deux voisins : `HOST01` et `HOST02`

Condition de départ :

`ns.dom01.atelier.idsa.prd.fr` est configuré en primaire pour la zone `dom01.atelier.idsa.prd.fr` et `ns.dom02.atelier.idsa.prd.fr` est configuré en primaire pour la zone `dom02.atelier.idsa.prd.fr`.

Dans un deuxième temps :

On va configurer `ns.dom02.atelier.idsa.prd.fr` en secondaire pour `dom01.atelier.idsa.prd.fr`. Et réciproquement, on va configurer `ns.dom01.atelier.idsa.prd.fr` en secondaire pour `dom02.atelier.idsa.prd.fr`.

Pour la suite de la correction, on va considérer les modifications à apporter concernant la zone `dom01.atelier.idsa.prd.fr`.

On rajoute les lignes suivantes dans le fichier `named.conf` de `ns.dom02.atelier.idsa.prd.fr` :

```
zone "dom01.atelier.idsa.prd.fr" {
                        type slave;
                        file "slave/dom01.atelier.idsa.prd.fr";
                        masters { 192.168.0.1; };
};
```

Et pour autoriser le transfert de zone à partir du serveur primaire `ns.dom01.atelier.idsa.prd.fr`, on remplace, dans la configuration de celui-ci,

```
allow-transfer { none; };
```

par

```
allow-transfer { 192.168.0.2; };
```

Pour tester si le transfert de zone a bien fonctionné, on interroge le serveur `ns.dom02.atelier.idsa.prd.fr` :

```
[]$ $PREFIX/bin/dig @ns.dom02.atelier.idsa.prd.fr \
    dom01.atelier.afnic.idsa.prd.fr SOA
```

On vérifie dans l'en-tête de la réponse que le drapeau AA est bien positionné, ce qui prouve que le serveur est bien autoritaire pour la zone interrogée.

4. TP2 : TSIG

4.1. Générer un secret

On utilise l'outil de génération de clés fourni avec la distribution bind, **dnssec-keygen**, dont les options sont données par :

```
[]$ $PREFIX/sbin/dnssec-keygen -h
```

On va générer une clé symétrique relative à un serveur. L'utilisation de cette clé pour sécuriser les transferts de zone entre deux serveurs nécessite la transmission de cette clé secrète au deuxième serveur. Dans le cas de la sécurisation des transferts de zone entre `HOST01` et `HOST02`, `HOST01` va créer une clé secrète pour sécuriser le transfert de la zone `dom01.atelier.afnic.idsa.prd.fr` ; il va ensuite transmettre cette clé à `HOST02`.

Et réciproquement `HOST02` va générer une clé secrète qu'il va transmettre à `HOST01`.

```
[]$ $PREFIX/sbin/dnssec-keygen -a HMAC-MD5 -b 512 \
    -n HOST tsig-HOST01-HOST02
```

Cette commande crée les fichiers `Ktsig-host01-host02.+157+33556.[key, private]` contenant le secret. On remarquera que `Ktsig-host01-host02.+157+33556.key` et `Ktsig-host01-host02.+157+33556.private` sont deux fichiers contenant le même secret. Néanmoins, dans certains cas d'utilisation avec **dig**, ces deux fichiers sont nécessaires. On les placera donc tous les deux dans le répertoire `.dnssec` avec des droits d'accès minimaux `(-r--------)`.

Sur `HOST01`, on va créer un fichier `tsig-HOST01-HOST02` contenant la clé symétrique `tsig-HOST01-HOST02.` suivant ce format :

```
key tsig-HOST01-HOST02. {
                algorithm hmac-md5;
                secret "6v/jfxti ... 3noe9g==";
};
```

Enfin `HOST01` va transmettre ce fichier de manière sécurisée à `HOST02` (par exemple en utilisant **scp**).

4.2. Synchroniser son horloge

Les transactions utilisant TSIG sont horodatées. Il est donc nécessaire que les horloges des entités communicantes soient synchronisées. Il faut synchroniser les serveurs sur un serveur de temps commun (pour cet atelier il s'agit de `ntp.atelier.idsa.prd.fr`).

On modifie dans un premier temps le fichier de configuration `ntp.conf` en rajoutant la ligne :

```
server ntp.atelier.idsa.prd.fr
```

Puis dans un deuxième temps, il faut lancer le démon ntpd avec l'option `-c /etc/ntp.conf` pour charger le fichier de configuration.

Pour vérifier que la synchronisation fonctionne bien, on peut utiliser la commande suivante :

```
[]$ /usr/sbin/ntptrace ntp.atelier.idsa.prd.fr
```

4.3. Faire un transfert de zone signé entre serveurs

Dans le cadre d'un transfert de la zone `dom01.atelier.idsa.prd.fr` : `ns.dom01.atelier.idsa.prd.fr` est primaire pour la zone et `ns.dom02.atelier.idsa.prd.fr` est secondaire.

Il faut modifier le fichier `named.conf` sur `ns.dom01.atelier.idsa.prd.fr` pour y inclure le chemin d'accès à la clé symétrique. Pour ce faire, on rajoute la directive :

```
include "$PREFIX/.dnssec/tsig-HOST01-HOST02";
```

Sur HOST 01 on trouvera la configuration suivante :

```
include "/usr/local/bind-9.3/.dnssec/tsig-HOST01-HOST02";

zone "dom01.atelier.idsa.prd.fr" {
                type master;
                file "master/dom01.atelier.idsa.prd.fr";
                allow-transfer {
                        key tsig-HOST01-HOST02.;
                };
};
```

Sur HOST 02 on trouvera la configuration suivante :

```
include "/usr/local/bind-9.3/.dnssec/tsig-HOST01-HOST02";

server 192.168.0.1  {
                        keys { tsig-HOST01-HOST02.; };
};

zone "dom01.atelier.idsa.prd.fr" {
                type slave;
                file "slave/dom01.atelier.idsa.prd.fr";
                masters { 192.168.0.1; };
};
```

4.4. Avec dig

Pour tester le transfer de zone, on interroge le serveur `ns.dom01.atelier.idsa.prd.fr` :

```
[]$ $PREFIX/bin/dig @ns.dom01.atelier.idsa.prd.fr \
    -y tsig-HOST01-HOST02:6v/jfxti...gsf dom01.atelier.idsa.prd.fr AXFR

[]$ $PREFIX/bin/dig @ns.dom01.atelier.idsa.prd.fr \
    -k Ktsig-host01-host02.+157+33556.private \
    dom01.atelier.idsa.prd.fr AXFR
```

Pour une erreur sur la signature, on obtient le message :

```
Transfer failed. BADSIG
```

Pour une erreur sur le nom de la clé :

```
Transfer failed. BADKEY
```

S'il n'y a pas d'erreur, on reçoit :

```
NOERROR
```

5. TP3 : sécurisation locale

5.1. Générer une paire de clés

Sur le serveur primaire d'abord générer une clé ZSK de longueur `1024 bits` pour signer `dom01.atelier.afnic.idsa.prd.fr`, et placer le couple de clés dans `.dnssec`.

```
[]$ $PREFIX/sbin/dnssec-keygen -a RSASHA1 -b 1024 \
    -n ZONE dom01.atelier.idsa.prd.fr
```

qui retourne `Kdom01.atelier.idsa.prd.fr.+005+18920`

Après avoir généré le couple de clés asymétriques, on inclut la partie publique dans le fichier de zone (`dom01.atelier.idsa.prd.fr`) et on incrémente le sérial avant de signer la zone pour la prise en compte des modifications. La partie privée est à conserver par l'utilisateur pour la signature de données.

```
[]$ cat Kdom01.atelier.idsa.prd.fr.+005+18920.key \
    >> dom01.atelier.idsa.prd.fr
```

5.2. Signer sa zone

Pour signer une zone DNS, on utilise l'outil **dnssec-signzone** dans un répertoire contenant le fichier de zone ainsi que le couple de clés ZSK.

Pour la zone `dom01.atelier.idsa.prd.fr.` :

```
[]$ $PREFIX/sbin/dnssec-signzone -r /dev/random \
    -o dom01.atelier.idsa.prd.fr \                 [src]
    -f dom01.atelier.idsa.prd.fr.signed \         [dest]
    -e +1296000 \                [expire dans +1296000 secondes]
    dom01.atelier.idsa.prd.fr \                          [fichier de zone]
    Kdom01.atelier.idsa.prd.fr.+005+18920.private [ZSK]
```

Placer le fichier `dom01.atelier.idsa.prd.fr.signed` généré dans le répertoire `$PREFIX/master`

5.3. Charger sa zone

On modifie le fichier `named.conf` en remplacant le fichier de zone `dom01.atelier.idsa.prd.fr` par `dom01.atelier.prd.fr.signed`.

```
zone "dom01.atelier.idsa.prd.fr" {
                type master;
                file "master/dom01.atelier.idsa.prd.fr.signed";
                allow-transfer {
                                key tsig-HOST01-HOST02.;
                };
};
```

Puis on recharge le fichier named.conf sous l'identité `bind` comme vu précédemment. On peut recharger la zone après avoir récupéré le PID sous lequel tourne le démon BIND en utilisant la commande :

```
[]# kill -HUP `cat $PREFIX/run/named.pid`
```

5.4. Tester sa zone signée

Pour tester sa zone signée, on interroge le serveur : `ns.dom01.atelier.idsa.prd.fr` et on positionne le bit DO [Dnssec OK] (option `+dnssec`). Pour rendre l'affichage plus lisible, on utilise l'option `+multiline`.

```
[]$ $PREFIX/bin/dig @ns.dom01.atelier.idsa.prd.fr \
    dom01.atelier.idsa.prd.fr. ns +dnssec +multiline
```

Faire avec des requêtes sur des Resources Records (RR) qui existent et d'autres qui n'existent pas (mise en évidence des enregistrements NXT)

```
[]$ $PREFIX/bin/dig @ns.dom01.atelier.idsa.prd.fr \
    inexistant.dom01.atelier.idsa.prd.fr a +dnssec +multiline
```

6. TP4 : délégation sécurisée

6.1. Générer une paire de clé

En utilisant la commande **dnssec-keygen**, on génère une nouvelle clé appelée KSK pour `dom01.atelier.idsa.prd.fr` (la longueur de la clé est de `2048 bits`).

```
[]$ $PREFIX/sbin/dnssec-keygen -a RSASHA1 -b 2048 -n ZONE \
    dom01.atelier.idsa.prd.fr Kdom01.atelier.idsa.prd.fr.+005+12178
```

6.2. Signer sa zone

On insère cette deuxième clé dans le fichier de zone et on incrémente le sérial. On re-signe sur le modèle précédent la zone `dom01.atelier.idsa.prd.fr` ; on peut préciser l'option `-d $PREFIX/.dnssec/keysets` qui spécifie le répertoire de destination du keyset qui sera généré lors de la signature. D'autre part on doit utiliser l'option `-k` suivie du nom du fichier contenant la KSK.

```
[]$ $PREFIX/sbin/dnssec-signzone -r /dev/random \
    -o dom01.atelier.idsa.prd.fr \                  [src]
    -f dom01.atelier.idsa.prd.fr.signed \         [dest]
    -e +1296000 \                 [expire dans +1296000 secondes]
    -d $PREFIX/.dnssec/keysets \
    -k  Kdom01.atelier.idsa.prd.fr.+005+12178.private \
    dom01.atelier.idsa.prd.fr.\                          [fichier de zone]
    Kdom01.atelier.idsa.prd.fr.+005+18920.private [ZSK]
```

6.3. Génération du DS

Le Keyset contenant la partie publique de la KSK de notre zone a été généré dans `$PREFIX/.dnssec/keysets`

```
[]$ cat $PREFIX/bin.dnssec/keysets keyset-dom01.atelier.idsa.prd.fr.

dom01.atelier.idsa.prd.fr. 172800  IN KEY  256 3 5 (
                          AQPTZNSmITKyDO4...I0iNKJOCQukLvI+Ru81d7
                          ) ; key id = 12178
```

Tansmettre le Keyset au parent `atelier.idsa.prd.fr` pour qu'il sécurise la délégation.

La délégation passe de *Verifiable Insecure* à l'état *Verifiable Secure* une fois le DS intégré dans la zone `atelier.idsa.prd.fr`.

Pour le vérifier on peut lancer la commande :

```
[]$ $PREFIX/bin/dig @ns.atelier.idsa.prd.fr dom01.atelier.idsa.prd.fr \
    DS +dnssec +multiline
```

6.4. Monter un cache

Considérons `HOST01` : on va démarrer sur cette machine une deuxième instance de BIND qui va écouter sur une autre adresse de la machine.

Dans le fichier `named.conf` de `ns.dom01.atelier.idsa.prd.fr`, on remplace `" listen-on {any;};"` par `"listen-on {194.168.0.1;};"`

Puis, on associe le nom `nscache.dom01.atelier.idsa.prd.fr` à la deuxième adresse IP de `HOST01` : `192.168.0.101`

On modifie le fichier de zone de `dom01.atelier.idsa.prd.fr` en conséquence en ajoutant la ligne :

```
nscache.dom01.atelier.idsa.prd.fr IN A 192.168.0.101
```

Puis on incrémente le sérial, on re-signe la zone et on recharge le démon BIND.

Ensuite on crée un deuxième fichier de configuration que l'on va appeler `named-nscache.conf` :

```
options {
        recursion yes;
        directory "/usr/local/bind-9.3"
        pid-file "run/named-nscache.pid";
        listen-on { 192.168.0.101;};
};
```

On active les logs dnssec comme indiqué dans l'énoncé du TP :

```
logging {
        channel "dnssec-channel" {
                file "log/dnssec";
                severity debug 3;
                print-category yes;
                print-severity yes;
                print-time yes;
        };
        category "dnssec" {
                "dnssec-channel";
        };

        channel "general-channel" {
                file "log/general";
                severity debug 3;
                print-category yes;
                print-severity yes;
                print-time yes;
        };
        category "general" {
                "general-channel";
        };
};
```

Enfin, on démarre ce deuxième serveur en utilisant le même type de commande que précedemment :

```
[]# $PREFIX/sbin/named -u bind -c $PREFIX/etc/named-nscache.conf
```

7. TP5 : sécurisation globale du DNS

Nous allons voir dans cette partie comment mettre à profit l'existence d'une arborescence DNS sécurisée globale pour la validation des enregistrements DNS.

Dans un premier temps, on souhaite que notre serveur récursif utilise l'arborescence DNS sécurisée du projet `RS.net`.

NB: La participation au projet `RS.net` étant soumise au respect d'une convention de non-divulgation des in-

formations, nous n'exposerons ici ni les adresses des serveurs racines du projet, ni la clé de confiance de cette racine. Les formalités d'adhésion à ce projet sont données sur http://www.rs.net/.

On va donc faire pointer notre serveur récursif vers les serveurs racines expérimentaux, en remplaçant :

```
 file "etc/named.root";
```

par

```
 file "etc/named-rsnet.root";
```

dans :

```
zone  "." {
                        type hint;
                        file "etc/named.root";
};
```

Le fichier `etc/named-rsnet.root` contient les racines expérimentales du projet `RS.net` (cf énoncé).

On ajoute la(les) clé(s) de confiance dans le fichier `named-nscache.conf` : on donne l'exemple de la clé de confiance de la racine.

```
trusted-keys {
        . 256 3 5
                "Clef-de-confiance-de-la-racine";
};
```

Notre cache est donc prêt à répondre à toute requête en parcourant une chaîne de confiance depuis la zone racine et en effectuant toutes les vérifications adéquates pour la validation des enregistrements des zones `domxx.atelier.idsa.prd.fr`. On consultera les logs pour visualiser les différentes étapes de la validation.

Une deuxième possibilité consiste à continuer à utiliser le fichier `named.root` traditionnel tout en configurant dans le fichier `named-nscache.conf` la zone `fr` en zone *forward*. Cela permet de rediriger les requêtes portant sur `fr` vers les serveurs expérimentaux où est stockée la version signée de `fr`. Dans ce cas, on utilisera la clé de confiance de `fr`. Voici alors ce qu'il faut ajouter au fichier `named-nscache.conf` :

```
trusted-keys {
        fr. 257 3 5
        "AQPrTytUGPbQ/4PRRZvsRGGRfaFLxMa5IbUIM+58SMbNCVUhN0uaVK25
iSPLBUQbdUurDIZlghsTsPe9kIWyddA5OOfAWHj47zPTxPED58emZaaZ
Klbm6evSjaJ1xQ5JTHgu3wtNo5sCUL8/+GIkGJnUjnHfJ7h5hvLe/Ofx
zK1RjFhPpM8LUno9WXodI0fdOdGK+YOp6yaTb9thO6Y9/UIQ+rdJpqjN
BArur4YQBBqHW/pNazQExUM8ktX2O3G7HgUkT3fVOqypuCCgRdIudwbF
BC82VH5rEpjZc1y9a929HC0Y32+I8REVWPKpUScUJzWEByizTjj6rvCj
        jXRf9VFB"
};


zone "fr" {
          type forward;
          forwarders {
          192.134.7.201;
          192.134.7.202;
          };
};
```

`192.134.7.201` et `192.134.7.202` sont les adresses de `ns[1|2].dnssec.nic.fr` qui sont les serveurs expérimentaux de la zone `fr` signée.

Cette manière de procéder permet d'accéder à un enregistrement quelconque du sous-arbre fr de manière sécurisée. On peut alors déterminer avec certitude le statut de ces enregistrements :

- `"verifiable secure"` : il existe une chaîne de confiance reliant la clé de confiance de la zone `fr` à l'enregistrement demandé ;
- `"verifiable insecure"` : entre la zone `fr` et la zone contenant l'enregistrement demandé, il existe au moins une délégation non sécurisée. L'enregistrement NXT signé associé au point de délégation doit prouver l'absence de DS pour cette délégation ;
- `"wrong"` : la vérification des signatures et délégations sécurisées fait apparaître une ou plusieures signatures erronées.

Pour le reste de l'arbre DNS, la résolution s'effectue classiquement (de manière non-sécurisée).

8. TP6 : roulement des clés

La procédure de roulement des clés (ZSK et KSK) est une procédure relativement complexe dont nous avons donné les grandes étapes théoriques dans l'énoncé du TP. Au moment d'écriture de ce corrigé, il n'existe pas d'algorithme unique pour le roulement des clés.

Pour de plus amples informations sur le sujet, on pourra consulter les documents suivants cités dans la section Références ci-dessous.

Références

B. Léonard. *"Documentation détaillée décrivant les procédures pour déployer DNSSEC"*. Livrable L2.4, projet IDsA (SP2). 2003.

O. Kolkman and R. Gieben. *"DNSSEC Operational Practices"*. draft-ietf-dnsop-dnssec-operational-practices-00.txt, work in progress. 2003.