TP pour l'atelier DNSSEC-IDsA Rennes, 09-10/12/2003

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 : 1.0

Date : 2004-03-25

Table of Contents

1. TP1. Pratique du logiciel BIND

1.1. Compilation du logiciel BIND

Le but de ce premier exercice est d'installer le logiciel qui nous permettra de pratiquer DNSSEC. Actuellement le seul logiciel offrant un support complet de DNSSEC est le logiciel BIND en version `9.3` (la branche de développement/snapshot). La version courante est pour le moment la version bind-9.3s20021217. Ce logiciel est disponible sur le site de l'ISC à l'URL suivante : ftp://ftp.isc.org/isc/bind9/snapshots/bind-9.3.0s20021217.tar.gz

- Télécharger et installer l'arborescence du source dans votre répertoire `$HOME`.

- Lancer la commande **./configure** avec les options appropriées. En particulier, pour les fonctionnalitées DNSSEC, il faut passer avec l'option `--with-openssl`. De plus, l'option `--prefix` permet d'installer l'arborescence binaire du programme BIND à l'endroit de son choix. Pour ce TP, ça sera : `/usr/local/bind-9.3`.

1.2. Test du logiciel

Afin de vérifier le bon fonctionnement du logiciel BIND, effectuons quelques tests. L'outil dig permet d'interroger un serveur DNS :

- interroger le serveur cache récursif de l'atelier (`nscache.atelier.idsa.prd.fr`)
- interroger le aussi avec l'option `+trace`

Remarque : la documentation de référence du logiciel se tourve dans le répertoire `./doc/arm/Bv9ARM.html`.

1.3. Création d'une zone DNS

- Chacun des participants va créer un domaine sous la zone atelier.idsa.prd.fr.
- Le nom du domaine devra avoir la syntaxe suivante : `domxx.atelier.idsa.prd.fr.`

Le fichier de configuration (`named.conf`) devrait ressembler à ce qui suit (à déposer dans `/usr/local/bind-9.3/etc/`) :

```
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 "localhost";
};

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

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

Remarque : les fichiers `localhost*` sont à créer.

Voici un exemple de fichier de zone `domxx.atelier.idsa.prd.fr` avec `xx = 12`:

```
$ORIGIN dom12.atelier.idsa.prd.fr.
$TTL 172800

@    IN    SOA    ns root.ns (
                   2003121000        ;serial
                   21600             ;refresh
                   3600              ;retry
                   3600000           ;expire
                   86400             ;minimum
                   )
     IN    NS    ns
ns   IN    A     192.168.0.12

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

Ne pas oublier pas de charger named :

```
#/usr/local/bind-9.3/sbin/named -c fichier_de_conf
```

1.4. Test de la zone

Afin de vérifier si votre zone est bien servie par named, faites un test avec **dig**.

1.5. Monter un service de secondaire

Partir du principe que votre voisin de gauche vous offre ce service, et que vous offrez à votre voisin de droite ce même type de service. Penser à ajouter le nom du serveur de votre voisin de gauche dans la liste des serveurs faisant autorité sur votre domaine, et à lui autoriser de faire des transferts de zone. Configurer aussi votre démon named pour devenir secondaire pour votre voisin de droite. Faites quelques tests de transferts de zone entre vos serveurs DNS et avec l'outil **dig AXFR**. Bravo, vous êtes désormais expert DNS !

2. TP2. TSIG

2.1. Générer un secret

Pour faire des transactions sécurisées avec TSIG, nous avons besoin d'un secret au format `HMAC-MD5`. L'outil **dnssec-keygen** permet de générer ce type de clé à volonté. Voici comment utiliser **dnssec-keygen** :

```
$/usr/local/bind-9.3/sbin/dnssec-keygen -a <alg> -b <bits> \
 -n HOST <name>
```

Remarque : les deux fichiers générés contiennent le secret (-:

2.2. Synchroniser son horloge

TSIG s'appuie sur la date de la transaction. Il est donc nécessaire que votre machine soit à l'heure. Pour cela, synchronisez-la sur le serveur de temps de l'atelier : `ntp.atelier.idsa.prd.fr` via le démon ntpd (pour vérifier si cette synchro fonctionne, vous pouvez utliser la commande **ntptrace**).

2.3. Faire une transfert de zone signé

2.3.1. Entre seveurs

- Ajouter votre secret dans vos configurations DNS :

```
key "host1-host2" {
        algorithm hmac-md5;
        secret "sARtfkSRFds56hj132lldFg=";
};
```

- Configurer le primaire :

```
allow-transfer { key transfert-key; };
```

- Et le secondaire,

```
server 192.134.4.120 {
        keys { transfert-key; };
};
```

- Inspecter les logs de BIND.

2.3.2. Avec dig

Vous pouvez aussi vérifier votre configuration avec l'outil **dig**. Celui-ci permet de faire des transferts de zones signés avec TSIG via les options `-k` et `-y`. Essayez quelques transferts avec la bonne clé mais aussi avec une mauvaise clé et une mauvaise synchronisation. Inspectez les résultats que produit **dig** à l'écran.

3. TP3. Sécurisation locale

3.1. Générer un paire de clés

La commande magique pour générer une paire de clé est toujours la même : **dnssec-keygen**. Par contre l'algorithme est `RSASH1` (ou `DSA`).

```
$/usr/local/bind-9.3/sbin/dnssec-keygen -a  <alg> -b <bits> \
 -n ZONE <name>
```

Nous appellerons cette clé, la `ZSK`.

3.2. Signer sa zone

La commande pour signer son fichier de zone s'appelle : **dnssec-signzone** :

```
$/usr/local/bind-9.3/sbin/dnssec-signzone domxx.atelier.idsa.prd.fr ZSK
```

Il y a quelques options intéressantes :

- `-a` : verification des signatures générées
- `-t` : affiche quelques statistiques
- `-o` : origine de la zone (par défaut nom du fichier de zone)
- `-f` : nom du fichier de sortie (par défaut `zonefile+.signed`)
- `-e` : date d'expiration des signatures (par défaut `now+30d`)
- `-u` : générateur d'aléa

Remarque : n'oubliez pas d'inclure dans le fichier source la partie publique de votre clé !

3.3. Charger sa zone

Charger le fichier généré (`.signed`) dans named.

Remarque : ne pas oublier pas d'augmenter le `serial` si vous re-signez !

3.4. Tester la zone signée

Faites quelques tests avec **dig** `+dnssec`. Cette option permet d'interroger des serveurs DNSSEC en positionnant le bit `DO` dans la requête. L'option `+multiline` peut être utile pour améliorer l'affichage. Testez les enregistrements NXT. L'option `+cdflag` de **dig** pourra vous être utile en cas de problème.

4. TP4. Délégation sécurisée

4.1. Générer une paire de clés

Toujours avec la même commande : **dnssec-keygen**. Mais cette fois, nous appellerons cette nouvelle clé, la KSK.

4.2. Signer sa zone

Pour signer son fichier de zone avec ses deux clés, **dnssec-signzone** requiert une nouvelle option `-k` KSK, pour préciser le nom de la clé qui ne signera que le `RRset KEY`.

```
$/usr/local/bind-9.3/sbin/dnssec-signzone -k <KSK> \
 domxx.atelier.idsa.prd.fr <ZSK>
```

4.3. Génération du DS

dnssec-signzone a produit au passage un fichier *keyset*. C'est ce fichier qu'il faut transmettre à son parent pour qu'il sécurise la délégation, via l'enregistrement DS. Transmettre son keyset à son parent. Attendre qu'il sécurise la délégation.

4.4. Monter un cache

Il s'agit de monter un nouveau serveur DNS sur votre machine qui ferait office de serveur récursif. Celui-ci vous permettra de faire suivre une *chaîne de confiance* et de vérifier les signatures reçues. Démarrez une nouvelle instance de named sur une seconde addresse IP (cette nouvelle adresse IP est obtenue en ajoutant `100` au dernier nombre décimal de la première adresse), en activant l'option `recursion` :

```
options {
          recursion yes;
          directory "/usr/local/bind-9.3"
          pid-file "run/named.pid";
          listen-on { 192.168.0.112;};
          listen-on-v6 { any; };
};
```

Et activez les logs `"dnssec"` en mode verbeux :

```
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";
          };
};
```

4.5. Vérifier les SIGs

Lancez quelques récursions DNSSEC via votre nouveau cache et regardez ce qui se passe dans les logs de named.

5. TP5. Sécurisation globale du DNS

5.1. Configurer son cache pour interroger RS.NET

Modifiez votre cache pour lui faire interroger les serveurs racines expérimentaux.

Remarque importante : l'utilisation des serveurs racines expérimentaux est assez restreinte et soumise à accréditation de la part de Bill Manning. Merci de ne pas les utiliser hors du cardre de cet atelier.

Remarque pour les lecteurs de cet enoncé : pour le besoin de publier le présent énoncé, nous avons choisi de masquer l'identité des 4 serveurs racine expérimentaux en remplaçant leurs noms et adresses par des noms et adresses génériques.

```
.                       3600000      NS    A.EXAMPLE.COM.
.                       360000       NS    B.EXAMPLE.COM.
.                       360000       NS    C.EXAMPLE.COM.
.                       360000       NS    D.EXAMPLE.COM.
A.EXAMPLE.COM.          3600000      A     X.Y.Z.T
A.EXAMPLE.COM.          3600000      AAAA  3ffe:dead:beef::1
B.EXAMPLE.COM.          360000       AAAA  3ffe:dead:beef::2
C.EXAMPLE.COM.          360000       AAAA  3ffe:dead:beef::3
D.EXAMPLE.COM.          360000       AAAA  3ffe:dead:beef::4
```

Ajoutez dans la configuration de votre cache la clé de la racine :

Remarque : seule une partie de cette clé est affichée dans le présent énonce.

```
trusted-keys {
        . 256 3 5
        "AQO
        [...]      // Clé-de-confiance-de-la-racine
        XXQ==";
};
```

Remarque : au cas où la racine expérimentale ne fonctionnerait pas, voici la clé de `.fr` :

```
trusted-keys {
        fr. 256 3 5
                "AQPrTytUGPbQ/4PRRZvsRGGRfaFLxMa5IbUIM+58SMbNCVUhN0uaVK25
                 iSPLBUQbdUurDIZlghsTsPe9kIWyddA5OOfAWHj47zPTxPED58emZaaZ
                 Klbm6evSjaJ1xQ5JTHgu3wtNo5sCUL8/+GIkGJnUjnHfJ7h5hvLe/Ofx
                 zK1RjFhPpM8LUno9WXodI0fdOdGK+YOp6yaTb9thO6Y9/UIQ+rdJpqjN
                 BArur4YQBBqHW/pNazQExUM8ktX2O3G7HgUkT3fVOqypuCCgRdIudwbF
                 BC82VH5rEpjZc1y9a929HC0Y32+I8REVWPKpUScUJzWEByizTjj6rvCj
                 jXRf9VFB";
};
```

5.2. Tester la résolution complète de la chaîne de confiance

Il suffit d'utiliser **dig** en interrogeant votre cache local, et de regarder dans les logs de named. Le serveur cache récursif va parcourir la chaîne de confiance et effectuera les vérifications des SIGs.

6. TP6. Roulement de clés

Partir d'une zone signée avec le parent signé et possédant le DS de la zone fille.

6.1. Roulement de la ZSK

Tous les changements sont effectués localement par l'administrateur.

6.1.1. Pré-publier une nouvelle clé :

- Générez une nouvelle clé (cf. TP3)
- Inclure cette clé dans le fichier de zone (n'oubliez pas d'incrémenter le `serial`)
- Re-signer le fichier de zone avec l'ancienne clé
- Recharger le fichier de zone
- Attendre la propagation dans les caches de la nouvelle clé.

6.1.2. Signer avec la nouvelle clé

Attendre la disparition dans les caches de l'ancienne clé.

6.1.3. Post-supprimer l'ancienne clé

- Supprimer l'ancienne ZSK
- Re-signer le fichier de zone avec la ZSK courante
- Recharger le fichier de zone.

Le roulement est terminé.

6.2. Roulement de KSK

6.2.1. Pré-publier une nouvelle clé

- Générer une nouvelle clé
- Inclure cette clé dans le fichier de zone (n'oubliez pas d'incrémenter le `serial`).
- Re-signer le fichier de zone avec les anciennes clés.
- Recharger le fichier de zone.
- Transmettre le nouveau `keyset` (contenant les deux KSK) au parent.
- Attendre la resignature par le parent pour la prise en compte du second DS
- Notifier le changement de KSK aux resolveurs qui utiliseraient votre clé de confiance
- Attendre la propagation du nouveau DS dans les caches.

6.2.2. Re-signer le fichier de zone avec la nouvelle KSK uniquement

- Attendre que l'ancienne KSK expire des cache.

6.2.3. Post-supprimer l'ancienne clé

- Transmettre le nouveau `keyset` (qui ne contient plus qu'une KSK) au parent
- Attendre que le parent re-signe (et ne publie que le nouveau DS)
- Supprimer l'ancienne KSK
- Re-signer le fichier de zone avec la KSK courante.