RFC 4033 : DNS Security Introduction and Requirements

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 21 novembre 2005. Dernière mise à jour le 15 février 2010

Date de publication du RFC : Mars 2005

https://www.bortzmeyer.org/4033.html

Le DNS n'est pas sûr, on le sait (le RFC 3833 [1] décrit les vulnérabilités du DNS en détail). Comme la quasi-totalité des applications de l'Internet dépendent du DNS à un moment ou à un autre, cette vulnérabilité inquiète beaucoup de gens. Notre RFC, avec ses compagnons RFC 4034 et RFC 4035, propose une solution à certains de ces problèmes, solution nommée **DNSSEC**.

Ce RFC décrit le deuxième DNSSEC, le premier était présenté dans le RFC 2535 mais n'a jamais été un succès.

DNSSEC-bis propose une sécurisation des **données**, pas du **canal**. La cryptographie permet en effet de sécuriser un canal de communication entre deux machines (ce que fait SSL) ou bien de sécuriser le message lui-même, qui peut alors être authentifié et protégé quels que soient les messagers intermédiaires : c'est ce que fait DNSSEC.

Pour cela, DNSSEC **signe** cryptographiquement, par la cryptographie asymétrique, les enregistrements DNS. Plus rigoureusement, il signe des "*RRsets*" ou "*Resource Record Sets*", c'est-à-dire un ensemble d'enregistrements ayant même nom et meme type (en pratique, rassurez-vous, la différence entre enregistrement et "*RRset*" est peu importante). Cette signature est portée par un nouveau type d'enregistrement, le RRSIG. Le RRSIG authentifie un enregistrement qui figure en partie gauche tandis que la partie droite contient la signature, ainsi que quelques informations utiles comme l'algorithme utilisé (c'est typiquement RSA + SHA1).

Les parties **publiques** des clés utilisées par le registre de la zone pour la signature sont publiées dans des enregistrements de type DNSKEY. Pour "amorcer la pompe" de DNSSEC, on récupère typiquement ces clés via un canal non-DNS, par exemple un accès à un serveur Web authentifié.

Voici par exemple un enregistrement RRSIG tel que l'affiche dig :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3833.txt

```
host.example.com. 86400 IN RRSIG A 5 3 86400 20030322173103 (
                                20030220173103 2642 example.com.
                                oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTr
                                PYGv07h108dUKGMeDPKijVCHX3DDKdfb+v6o
                                B9wfuh3DTJXUAfI/M0zmO/zz8bW0Rznl8O3t
                                GNazPwQKkRN20XPXV6nwwfoXmJQbsLNrLfkG
                                J5D6fwFm8nN+6pBzeDQfsS3Ap3o= )
```

Ici, le signataire `example.com` (ce nom apparait en partie droite) a authentifié l'adresse IPv4 (la signature couvre un enregistrement de type A, c'est indiqué juste après le type RRSIG) de `host.example.com`. La signature est le texte en base64 (RFC 4648) qui figure à la fin. S'il possède la clé de `example.com` (on peut la récupérer, par exemple, dans l'enregistrement de type DNSKEY ou "hors bande", par exemple sur une clé USB remise en mains propres), un résolveur DNS pourra alors être sûr de l'authenticité de cette adresse et ceci **même si** l'enregistrement a été transmis via des relais inconnus.

Enfin, DNSSEC spécifie également deux autres types d'enregistrement, plus subtils : NSEC et DS.

Les NSEC servent à répondre à une question délicate : authentifier la non-existence d'un domaine. En effet, DNSSEC marche par la signature d'enregistrements (ce choix a été fait pour éviter de modifier trop fortement le protocole DNS). Mais, si un nom n'existe pas, il n'y a pas d'enregistrement à signer ! Le problème est résolu avec les enregistrements de type NSEC ("*Next Secure*"). Ces enregistrements (qui sont signés comme les autres) indiquent quel est le prochain domaine de la zone. Ainsi, lorsqu'on demande `nimportequoi.example.org`, le serveur de `example.org` peut répondre par un NSEC pointant vers le domaine `suivant.example.org` qui, lui, existe. Voici un exemple d'enregistrement NSEC qui dit que `alfa.example.com` a des enregistrements de quatre types et que le nom suivant est `host.example.com` (donc, on peut être certain que `beta.example.com` n'existe pas) :

```
alfa.example.com. 86400 IN NSEC host.example.com. (
                                A MX RRSIG NSEC)
```

On note que, puisque les NSEC permettent d'obtenir le domaine suivant, puis le suivant du suivant, ils donnent donc potentiellement accès à l'intégralité de la zone, tout comme un transfert de zones. Ce problème, connu sous le nom de "*zone walking*" a contribué à la décision de certains registres (typiquement ceux qui ne veulent pas distribuer leur fichier de zone) de ne pas déployer DNSSEC. C'est ainsi que Simon Josefsson a développé un programme <`http://josefsson.org/walker/`> de "*zone walking*" qui montre bien la vulnérabilité des zones DNSSEC. Ce problème est résolu en remplaçant les NSEC par les NSEC3 du RFC 5155.

Les enregistrements DS ("*Delegation Signer*"), quant à eux, permettent d'authentifier les délégations. En effet, signer les enregistrements de délégation (les NS, pour "*name servers*") dans la zone parente ne suffit pas, puisqu'il faut aussi permettre de trouver la clé de la zone fille. Un enregistrement DS va donc pointer vers l'enregistrement DNSKEY de la zone fille (si celle-ci n'est pas elle-même signée, le DS est inutile).

DNSSEC bis est aujourd'hui mis en œuvre dans plusieurs serveurs de noms dont BIND <`http://www.isc.org/index.pl?/sw/bind/`> (à partir de la version 9.2) et NSD <`http://www.nlnetlabs.nl/nsd/`> (à partir de la version 2.3.0).

L'outil dig permet d'effectuer une requête avec DNSSEC :

```
% dig +dnssec @ns3.nic.fr +dnssec A ripe.net.
...
;; ANSWER SECTION:
ripe.net.               600     IN      A       193.0.0.214
ripe.net.               600     IN      RRSIG   A 5 2 600 20051221061607 20051121061607 49526 ripe.net. 2NePKf/O
```

On voit ici la réponse à la question posée (l'adresse de `ripe.net`) et la signature.

On peut aussi récupérer la partie publique de la clé :

```
% dig  DNSKEY ripe.net.
...
;; ANSWER SECTION:
ripe.net.               3480    IN      DNSKEY  256 3 5 AQPhEMiv80EEjX6gYDc8E7Osfumf4C/pZxBmTRRiOVL3h6lx1CIVCyPl
ripe.net.               3480    IN      DNSKEY  257 3 5 AQOTT7bx7N38sPgDWniKnHnSnTxYxdMpEq7dyrDHDaRQgq7DULPWX6ZY
```

Et si je veux signer ma zone, quelles solutions puis-je utiliser ? Une version récente de BIND inclue les logiciels `dnssec-keygen` et `dnssec-signzone` qui fournissent tous les services nécessaires. Testons-les sur une Debian en version `lenny`. Il faut d'abord générer la clé de la zone :

```
% dnssec-keygen -a DSA  -b 1024 -n  ZONE sources.org
```

Cette commande laissera sur le disque les fichiers `Ksources.org.+003+55957.key` et `Ksources.org.+003+5595`
Il faut évidemment les sauvegarder précieusement (autrement, il faudra changer la clé, une opération complexe qu'on nomme le "*rollover*") et vérifier que la clé privée n'est pas accessible à tous.

Il faut alors inclure la clé publique (ici, le contenu de `Ksources.org.+003+55957.key`) dans le fichier de zone.

Ensuite, à chaque modification de la zone, il faudra signer :

```
% dnssec-signzone sources.org
sources.org.signed
```

Le programme `dnssec-signzone` produit un fichier du même nom que la zone mais suffixé par `.signed`. Un exemple de règle dans un `Makefile` pour automatiser cela est :

```
reload: sources.org.signed example.org.signed foobar.example.signed

%.signed: %
        @echo Signing requires a lot of entropy in /dev/random, do not hesitate to load the machine...
        dnssec-signzone $^
```

L'avertissement sur l'entropie vient du fait que, sur une machine Linux peu chargée, `/dev/random` se bloque s'il n'a pas assez d'entropie pour générer des nombres vraiment aléatoires, et on a l'impression que `dnssec-signzone` est arrêté.

`dnssec-signzone` produit également un fichier avec les enregistrements DS, pour qu'on puisse les transmettre à son registre (uniquement pour une nouvelle clé ou bien une clé qui change).

Le cas présenté ici, avec une seule clé, est le plus simple. Elle est peut-être préférable pour les petites zones, ou pour ceux qui débutent avec DNSSEC. Mais le RFC recommande une autre approche, plus complexe mais plus sûre, la séparation en deux clés, la KSK ("*Key Signing Key*") et la ZSK ("*Zone Signing Key*"). La première, introduite dans le RFC 3757, ne sert qu'à signer la seconde. Elle est typiquement très robuste (à l'heure actuelle, typiquement de 4096 bits) et change relativement rarement car c'est elle qui est utilisée par les autres zones, comme "*trust anchor*". La seconde, la ZSK, est plus légère (par exemple 1024 bits, taille au delà de laquelle la signature des grandes zones devient vraiment pénible) et sert à signer les enregistrements. Du fait de cette utilisation, il y a davantage d'information qui « fuit », ; et qui peut donc aider un attaquant. Elle change donc plus fréquemment, ce qui ne pose pas trop de problèmes opérationnels puisqu'elle n'est connue que localement.

Pour créer la KSK, on doit normalement s'entourer de multiples précautions (si la zone qui l'utilisera est vitale, on s'enfermera dans une cage de Faraday, on se déconnectera du réseau, etc). Ensuite, on tape :

```
% dnssec-keygen -f KSK \
    -a RSASHA1 -b 4096 -n ZONE example.org
```

(Cette opération peut prendre plusieurs heures sur une machine qui a peu d'entropie.) On garde évidemment précieusement la partie privée de cette clé ! On publie la partie publique (par exemple en l'envoyant à la zone parente ou bien en la mettant sur son site Web) et on la met dans son fichier de zone.

La ZSK se génère avec une commande presque identique à part l'option `-f` et une clé plus petite :

```
% dnssec-keygen \
    -a RSASHA1 -b 1024 -n ZONE example.org
```

Les deux clés se mettent ensuite dans le fichier de zone. Au moment de la signature, `dnssec-signzone` ne signera qu'avec la ou les clés qui n'ont pas été générées avec l'option `-f KSK`. Toutefois, BIND ne trouve pas forcément les signatures seul et, selon la façon dont les clés ont été nommées et/ou placées, il peut être utile de dire à `dnssec-signzone` quelle est la KSK (option `-k`) et quelle est la ZSK (nom de fichier après le nom de la zone) :

```
dnssec-signzone -t -k example.org.KSK example.org example.org.ZSK
```

Notez que tout ce processus peut s'automatiser partiellement avec des outils comme OpenDNSSEC <https://www.bortzmeyer.org/opendnssec-debut.html>.

Quelques documentations sur la configuration de DNSSEC :
— "*DNSSEC in 6 minutes*" <http://alan.clegg.com/files/DNSSEC_in_6_minutes.pdf>,
— "*DNSSEC HOWTO*" <http://www.nlnetlabs.nl/dnssec_howto/>.

— *"BIND 9 Administrator Reference Manual"* <http://www.bind9.net/manual/bind/9.3.3/Bv9ARM.ch04.html#DNSSEC>,

Voyons maintenant quelques points important de DNSSEC mais qui ne sont pas mentionnés dans le RFC (celui-ci ne fait que spécifier un protocole, il ne traite pas de tous les problèmes liés à ce protocole).

On notera que le RFC ne spécifie pas l'API du service. Que doit faire une fonction comme `getaddrinfo(3)` lorsqu'une vérification DNSSEC échoue ? L'API actuelle ne permet pas de donner de détails sur la cause de l'échec. En outre, à ma connaissance, aucune mise en œuvre actuelle ne permet à l'administrateur système de définir une politique de sécurité (par exemple dans son fichier de configuration `/etc/resolv.conf`).

Aujourd'hui, plusieurs registres ont commencé à déployer DNSSEC comme le RIPE-NCC <https://www.ripe.net/projects/disi/keys/>, notamment pour les domaines de résolution inverse (`in-addr.arpa`) qu'il gère ou bien le registre suédois <http://www.nic-se.se/english/nyheter/pr/2005-09-14?lang=en>. La racine du DNS est en cours de signature <https://www.bortzmeyer.org/la-racine-commence-signature.html>.

Si on souhaite plus d'informations, on pourra regarder un excellent article <http://ds9a.nl/dnssec/> de Bert Hubert, l'auteur de PowerDNS <http://wiki.powerdns.com/> ou bien le portail de DNSSEC <http://www.dnssec.net/>. Et, sur mon blog, il y a de nombreux articles sur DNSSEC </search?pattern=dnssec>.