dnssec-trigger, un outil pour mettre DNSSEC à la disposition de M. Toutlemonde

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 octobre 2011

https://www.bortzmeyer.org/dnssec-trigger.html

Une des faiblesses de la sécurité de l'Internet est que le mécanisme d'annuaire principal, le DNS, n'est pas très sécurisé. Il est trop facile de tromper un serveur DNS en lui injectant de fausses informations (faille dite Kaminsky <https://www.bortzmeyer.org/comment-fonctionne-la-faille-kaminsky.html>). Mais il y a pire : souvent, c'est le serveur DNS mis à la disposition de l'utilisateur, par le FAI ou par le service informatique local, qui le trompe <https://www.bortzmeyer.org/dns-menteur.html>. Une solution technique existe, à ces deux problèmes : DNSSEC. Mais pour que DNSSEC protège M. Toutlemonde, les vérifications que permet ce protocole doivent être faites sur la machine de M. Toutlemonde, la seule en qui il peut avoir un peu confiance. C'est ce que permet le nouveau logiciel dnssec-trigger <http://www.nlnetlabs.nl/projects/dnssec-trigger/>.

Depuis que DNSSEC existe (sa version actuelle a été normalisée dans le RFC 4033 [1]), il y a un débat sur l'endroit où doit se faire la **validation**, c'est-à-dire la vérification, par des calculs cryptographiques, que l'information reçue est bien authentique. Comme cette validation nécessite des calculs complexes et, jusqu'à la signature de la racine et de nombreux TLD, vers 2010-2011, nécessitait une configuration complexe, il semblait logique de faire la validation sur les résolveurs DNS que tout FAI, tout réseau local, met à la disposition de M. Toutlemonde. Le problème est que ces résolveurs sont souvent les premiers à mentir, comme on l'a vu de nombreuses fois, l'une des dernières étant l'affaire Earthlink <https://secure.dslreports.com/forum/r26348218-General-Earthlink-DNS-server-pointing-www.google.com-to-unknow>. Bien que ce soit une violation claire de la neutralité de l'intermédiaire <https://www.bortzmeyer.org/neutralite.html>, d'autres FAI se livrent à ce genre de pratiques.

Il reste donc à faire la validation sur la machine de M. Toutlemonde. Pour un PC moderne, les calculs cryptographiques nécessaires ne représentent qu'une tâche bien légère (cela peut être différent pour un "*smartphone*"). Mais la validation directement sur le PC de l'utilisateur pose deux problèmes : elle

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4033.txt

nécessite l'installation et la configuration correcte d'un logiciel supplémentaire (même si cela ne prend que quelques minutes, l'expérience prouve que c'est beaucoup trop difficile pour M. Toutlemonde) **et**, si tout le monde en faisait autant, les serveurs DNS souffriraient sous la charge accrue. En effet, comme il n'y aurait plus de cache partagé (rôle que jouent aujourd'hui les serveurs résolveurs des FAI, qui gardent en mémoire la réponse aux questions déjà posées), les serveurs des différentes zones DNS recevraient bien plus de requêtes.

Cette question est connue depuis un certain temps (voir par exemple une discussion technique approfondie <https://www.bortzmeyer.org/ou-valider-dnsssec.html> en 2011). Mais la nouveauté est qu'on a désormais un logiciel qui résoud ces deux problèmes. dnssec-trigger <http://www.nlnetlabs.nl/projects/dnssec-trigger/> est un outil génial. Développé aux NLnetLabs <http://www.nlnetlabs.nl>, partiellement financé <http://www.afnic.fr/en/about-afnic/news/general-news/2826/show/afnic-actively-supports-nlnet-labs.html> par l'AFNIC, il va permettre de donner la puissance de DNSSEC à M. Michu, en lui permettant de valider sur sa machine, tout en n'écroulant pas les serveurs de l'AFNIC (ou les autres serveurs faisant autorité) sous la charge, comme cela se produirait si chacun avait un résolveur normal sur sa machine.

Comment fonctionne dnssec-trigger ? C'est un logiciel qui tourne sur plusieurs systèmes d'exploitation, plutôt ceux qui sont michuiens ou toutlemondesques (Ubuntu, Fedora, Microsoft Windows, Mac OS). Il s'intègre au système de gestion du réseau de ces systèmes (les programmeurs apprécieront l'exploit que cela représente, vue la variété de ces systèmes), par exemple NetworkManager, et, lorsque celui-ci lui signale une nouvelle connexion réseau, il **teste** les résolveurs indiqués (le réseau les indique typiquement avec DHCP). Pourquoi les tester ? Parce que, dans la nature, on trouve de tout sur les réseaux. Idéalement, les résolveurs devraient fonctionner. Mais très fréquemment (surtout dans les réseaux pourris fournis par les hôtels, les gares, etc), le résolveur est sérieusement cassé : il bloque DNSSEC, ou bien il bloque les réponses plus grandes que 512 octets <https://www.bortzmeyer.org/dns-size.html> (ce qui revient quasiment à bloquer DNSSEC) ou carrément il bloque toutes les requêtes qui utilisent EDNS0 (RFC 6891). Ces résolveurs ne peuvent pas aider l'utilisateur, ils ne sont qu'un obstacle. dnssec-trigger tente alors de joindre directement les serveurs faisant autorité (ce qui marche si le réseau ne bloque pas le port 53 <https://www.bortzmeyer.org/port53-filtre.html>). Si cela échoue, il essaie plusieurs techniques (expérimentales et pas forcément présentes dans la version actuelle du logiciel) comme de joindre un résolveur public qui marche, ou comme de tunneler les requêtes sur le port 443. Ces techniques de contournement sont bien connues des "*hackers*" mais l'intérêt de dnssec-trigger est de les automatiser pour M. Michu.

Une fois ces tests terminés, dnssec-trigger reconfigure au vol le résolveur Unbound (dnssec-trigger pourrait aussi marcher avec BIND, qui a les mêmes capacités mais Unbound est meilleur sur la plupart des plans ; notez que la version Windows de dnssec-trigger inclus Unbound) de façon à :

— Utiliser les résolveurs officiels du réseau comme "*forwarders*", c'est-à-dire qu'ils recevront les requêtes DNS et pourront garder les réponses dans leur cache, limitant ainsi le trafic réseau. C'est donc la meilleure approche, la plus « développement durable ». La validation sera bien faite sur le poste de travail, sur la machine de M. Toutlemonde, mais le trafic sur les serveurs DNS faisant autorité ne sera pas différent de ce qu'il est aujourd'hui, puisqu'on utilise les mêmes caches. (Le fait que la validation fonctionne, qu'on ait obtenu la réponse DNS directement d'un serveur faisant autorité, ou indirectement via un résolveur/cache, est le résultat d'une très utile propriété de DNSSEC : il protège le message, pas le canal.)
— Si les résolveurs officiels sont vraiment trop nuls, dnssec-trigger va dire à Unbound de fonctionner comme résolveur indépendant. La charge sur tous les serveurs DNS va alors augmenter mais c'est inévitable (les serveurs des gros TLD comme `.fr` sont déjà surdimensionnés, pour faire face au risque de DoS distribuées).
— Si le lien direct vers les serveurs faisant autorité ne passe pas (cas où un pare-feu se trouve sur le trajet et bloque le port 53, celui du DNS), dnssec-trigger va alors dire à Unbound de tenter différents trucs pour passer quand même, comme d'utiliser un résolveur ouvert et accessible sur un autre port (ce dernier service est en cours d'évaluation).

---

https://www.bortzmeyer.org/dnssec-trigger.html

Avec ce logiciel, on aura donc enfin le beurre et l'argent du beurre. On pourra donc valider en local, sur sa machine, sans pour autant massacrer les serveurs DNS sous les requêtes. Mais je parle au futur car dnssec-trigger est encore en béta-test. Il faut des volontaires <http://www.nlnetlabs.nl/projects/dnssec-trigger/> pour le tester sur plein de réseaux différents (surtout les pénibles, aéroports, hôtels, etc). Ensuite, il reste à l'intégrer dans les systèmes d'exploitation, pour que M. Toutlemonde n'ait rien à installer (les discussions ont déjà commencé <http://mail.gnome.org/archives/networkmanager-list/2011-September/msg00153.html> avec Fedora).

Après cet appel au peuple, un peu de technique. dnssec-trigger a quatre parties, un démon qui doit tourner en permanence, `dnssec-triggerd`, un composant enregistré auprès du gestionnaire de réseau (NetworkManager sur Ubuntu, le script d'installation le détecte en général seul et se débrouille) pour être prévenu des changements de connectivité, un logiciel de contrôle, `dnssec-trigger-control` et bien sûr le résolveur validant, actuellement Unbound (toute la cuisine de communication sécurisée entre dnssec-trigger et Unbound, incluant des certificats X.509 pour l'authentification, est faite automatiquement à l'installation de dnssec-trigger). On peut regarder l'état du système de résolution :

```
% dnssec-trigger-control status
at 2011-10-15 16:27:42
cache 212.27.40.240: OK
cache 212.27.40.241: OK
state: cache secure
```

Ici, on voit que dnssec-trigger a été informé (par NetworkManager) que le réseau local a deux résolveurs, qu'ils ont été testés et trouvés corrects, et que dnssec-trigger est donc heureux ("*secure*") et utilise un cache. C'était sur un réseau local connecté à Free. On a eu de la chance ici car les deux résolveurs de Free sont composés de plusieurs machines ayant des configurations différentes (!) et il est fréquent qu'une des adresses ne gère pas DNSSEC. Ici, ce n'est heureusement pas le cas, dnssec-trigger a donc indiqué à Unbound d'utiliser ces deux résolveurs, ce qu'on peut vérifier, en demandant à Unbound :

```
# unbound-control forward
212.27.40.241 212.27.40.240
```

Et tcpdump nous le confirme :

```
17:06:26.848768 IP 192.168.2.26.37031 > 212.27.40.241.53: 31541+% [1au] A? ubuntu-archive.mirrors.proxad.net. (6
17:06:26.871841 IP 212.27.40.241.53 > 192.168.2.26.37031: 31541 1/2/3 A 88.191.250.131 (146)
```

Les requêtes DNS (ici une demande de l'adresse IPv4 de `ubuntu-archive.mirrors.proxad.net`) sont bien envoyées au "*forwarder*", `212.27.40.241` (qui pourra répondre à partir de son cache), et pas directement aux serveurs faisant autorité.

Si l'un des résolveurs indiqués ne gère pas DNSSEC (ici, `192.134.4.163` est un BIND avec la configuration `dnssec-enable no;`), dnssec-trigger utilise les autres :

```
%  dnssec-trigger-control status
at 2011-09-20 10:01:09
cache 192.134.4.163: error no RRSIGs in reply
cache 192.134.4.162: OK
state: cache secure
```

On est quand même "*secure*", seul `192.134.4.162` sera utilisé comme "*forwarder*".

dnssec-trigger sait gérer les annonces RA ("*Router Advertisment*") du RFC 8106 et peut donc utiliser des résolveurs IPv6. Ici, toujours chez Free :

```
% dnssec-trigger-control status
at 2011-10-15 17:04:32
cache 2a01:e00::1: OK
cache 2a01:e00::2: error no EDNS
cache 212.27.40.240: OK
cache 212.27.40.241: OK
state: cache secure
```

Un des deux résolveurs IPv6 ne gère pas EDNS0 (je l'ai dit, c'est un problème courant chez Free) donc dnssec-trigger n'utilise que les résolveurs qui marchent :

```
# unbound-control forward
212.27.40.241 212.27.40.240 2a01:e00::1
```

tcpdump nous confirme que la résolution se passe bien en IPv6 (ici, demande de la clé de `.org`, nécessaire pour valider le nom `www.bortzmeyer.org` qui avait été demandé) :

```
17:07:13.651758 IP6 2a01:e35:8bd9:8bb0:1a03:73ff:fe66:e568.31040 > 2a01:e00::1.53: 54769+% [1au] DNSKEY? org
17:07:13.771831 IP6 2a01:e00::1.53 > 2a01:e35:8bd9:8bb0:1a03:73ff:fe66:e568.31040: 54769 6/0/1 DNSKEY, DNSKE
```

Jusqu'à présent, nous n'avons vu que des résolveurs assez sympa. Mais, lorsqu'on se promène de "*hotspot*" en "*hotspot*", on tombe sur bien pire. Ici, aucun des résolveurs officiels ne gère DNSSEC :

```
% dnssec-trigger-control status
at 2011-10-07 09:04:06
authority 128.63.2.53: OK
cache 10.150.6.1: error no RRSIGs in reply
cache 10.150.2.1: error no RRSIGs in reply
state: auth secure
```

Mais dnssec-trigger a vu qu'il pouvait parler directement aux serveurs faisant autorité (il indique avoir testé avec `128.63.2.53`, un des serveurs de la racine, le H). On est donc bien "*secure*" mais l'indication "*cache*" a été remplacée par "*auth*". Désormais, le seul cache est celui d'Unbound sur la machine.

Pire, voici un "*hotspot*" qui ne laisse même pas parler aux serveurs faisant autorité (ici le serveur racine K) :

```
% dnssec-trigger-control status
at 2011-10-07 09:02:21
authority 193.0.14.129: error no EDNS
cache 192.168.16.1: error no EDNS
state: nodnssec insecure
```

dnssec-trigger affiche alors un avertissement disant qu'il ne sait pas faire (dans cette version, qui n'a pas encore les techniques de contournement) et laisse le choix à l'utilisateur de se connecter de manière non sûre (ce qui a été choisi ici) ou bien de se déconnecter. Notez qu'on est en sécurité dans ce dernier cas, et dnssec-trigger affichera "*secure*"... :

```
% dnssec-trigger-control status
at 2011-10-07 09:06:49
no cache: no DNS servers have been supplied via DHCP
state: disconnected secure
```

Un cas beaucoup plus vicieux est celui d'un "*hotspot*" où les résolveurs ne gèrent pas DNSSEC, mais on a accès aux serveurs faisant autorité, sauf que le portail captif ne marche pas si on n'utilise pas les résolveurs officiels...

```
at 2011-10-07 09:08:46
authority 193.0.14.129: OK
cache 10.150.6.1: error no RRSIGs in reply
cache 10.150.2.1: error no RRSIGs in reply
state: auth secure
```

dnssec-trigger utilise alors l'accès direct aux serveurs faisant autorité, qui marche (vérifié ici avec dig) :

```
; <<>> DiG 9.7.3 <<>> @193.0.14.129 DNSKEY .
...
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
.                       172800  IN      DNSKEY  256 3 8
...
```

Mais dès qu'on essaie d'aller sur le Web, on est redirigé d'autorité vers un portail captif nommé `bsc-lsh3.essec.fr` et on reçoit un message d'erreur disant qu'il n'existe pas. En effet, ce nom n'est pas présent dans le vrai DNS :

```
; <<>> DiG 9.7.3 <<>> A bsc-lsh3.essec.fr
...
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 28170
```

mais il l'est dans les serveurs officiels du "*hotspot*" :

```
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2500
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
bsc-lsh3.essec.fr.      28800   IN      A       194.254.137.123
```

Seule solution, régler le problème à la main. dnssec-trigger fournit un GUI pour le faire, mais on peut aussi utiliser la ligne de commande :

```
... On dit à dnssec-trigger d'utiliser temporairement les résolveurs officiels ...
% dnssec-trigger-control hotspot_signon
... On se connecte au portail captif, acceptant les conditions d'utilisation ...
... On re-teste ...
% dnssec-trigger-control reprobe
... Et cette fois, on a bien du DNSSEC ...
```

Pour la petite histoire, voici quel était le cahier des charges original de dnssec-trigger, ceci était le compte-rendu d'une série de réunions informelles et de discussions en ligne.

"My wishlist for Xmas : an easy (easy as in "works for M. Smith or M. Jones") way to have an Unbound resolver which :"

- *"uses the DHCP-assigned name servers as forwarders after testing them (ldns-test-edns may be useful here), to save traffic on authoritative name servers"*
- *"or talks directly to the authoritative name servers (including the root) if the DHCP-assigned name servers are broken (no EDNS, mangles DNSSEC, etc) and if there is a clean path to the auth. name servers (again, something to test, because of the GF in China, because of broken "transparent" DNS proxies)"*
- *"or tunnels DNS requests to a forwarder hardwired somewhere (or configured by the user) is there is no other solution."*

"All of this automatically. Please ship in three months and for free. "

À propos d'anglais, si vous préférez lire en anglais, ou simplement si vous voulez voir davantage de copies d'écran, Jan-Piet Mens a écrit sur le même sujet en anglais <http://jpmens.net/2011/10/21/automating-unbound-for-dnssec-on-your-workstation/>. Il y a également l'exposé d'Olaf Kolkman au RIPE 63 <http://ripe63.ripe.net/presentations/172-RIPEWG-DNSSEC-tri pdf>. Voir aussi L'article de Dmitry Kohmanyuk <http://dk379.posterous.com/dnssec-results-fetched (mais son titre contient une grosse erreur : le principe de dnssec-trigger est justement de tout faire pour éviter de demander aux serveurs faisant autorité).