Où doit se faire la validation DNSSEC ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 avril 2011

https://www.bortzmeyer.org/ou-valider-dnsssec.html

Une question fréquemment posée par les débutants en DNSSEC est : qui doit faire la validation, c'est-à-dire vérifier les signatures cryptographiques ? Le résolveur habituel (typiquement celui du FAI) ? Ou bien un logiciel (par exemple le navigateur Web) situé sur le poste de travail de l'utilisateur ? La question est revenue sur le tapis lors de la conférence SATIN <https://www.bortzmeyer.org/satin.html> hier.

Le problème avait été soulevé par Wesley Hardaker lors d'un bref, mais excellent, exposé sur les tests des résolveurs DNS, pour déterminer s'ils pouvaient servir de relais pour DNSSEC. Mais un peu de contexte d'abord : DNSSEC permet au titulaire d'une zone DNS de signer celle-ci cryptographiquement. Pour que cela serve à quelque chose, une entité, le **validateur**, doit vérifier ces signatures (et leur lien avec une clé connue). Plusieurs logiciels savent faire cela (en logiciel libre, les résolveurs BIND, Unbound, les bibliothèques libval <https://www.dnssec-tools.org/wiki/index.php/Libval_and_libsres> ou ldns <http://www.nlnetlabs.nl/projects/ldns/>, etc).

Quel est le meilleur endroit pour faire la validation. A priori, plusieurs endroits sont possibles :

— Dans le résolveur DNS existant, géré par le FAI ou par le service informatique du réseau local. Avantages : ce logiciel existe déjà, il est sans doute déjà capable de valider DNSSEC, il suffit d'activer cette fonction (dans BIND : `dnssec-validation yes;`), il n'y a ainsi rien à faire sur les machines de l'utilisateur. Inconvénients : si l'attaque se produit entre le résolveur et la machine de l'utilisateur, DNSSEC n'aura servi à rien. Or, ce « dernier kilomètre » est loin d'être fiable. D'autre part, le FAI est souvent le premier ennemi (résolveurs DNS menteurs <https://www.bortzmeyer.org/dns-menteur.html>, censure genre LOPPSI).
— Dans un résolveur installé sur la machine de l'utilisateur ? Avantage : cela se produit sur une machine que l'utilisateur contrôle complètement, nul ne peut plus lui mentir. Inconvénient : aujourd'hui, il reste encore un peu de code à écrire pour que cela soit complètement automatique. Si installer Unbound, par exemple sur une Ubuntu, est trivial, encore faut-il configurer la machine pour utiliser ce résolveur local (fichier `/etc/resolv.conf` sur Unix). Un autre inconvénient potentiel, la charge sur les serveurs DNS, est traité plus loin.

— Dans une application, comme Firefox ou OpenSSH ? Avantages : l'application sait ainsi tout ce qui se passe et peut présenter à l'utilisateur un message approprié, voire lui permettre de passer outre une erreur DNSSEC. (Avec les deux premières solutions, la seule possibilité du résolveur, en cas d'échec de la validation, était de renvoyer un code d'erreur très général, genre `SERVFAIL`, "*Server Failure*".) L'application peut aussi adapter son niveau de sécurité (on peut penser que la récupération d'un enregistrement `SSHFP` (RFC 4255 [1]) doit être davantage sécurisée que celle d'un enregistrement `MX`). Inconvénient : il faut modifier toutes les applications. S'il existe des bibliothèques de validation (citées plus haut), ce n'est quand même pas une tâche triviale, d'autant plus qu'il n'existe pas d'API standard (chaque bibliothèque a son interface).

Wesley Hardaker avait fait également un exposé plus long (« "*Enabling DNSSEC in Applications*" <`http://conferences.npl.co.uk/satin/papers/satin2011-Hardaker.pdf`> ») sur la troisième solution, avec une jolie démonstration où, pour montrer que la validation DNSSEC était peu coûteuse en ressources, il utilisait un Firefox et un OpenSSH « DNSSEC » sur son téléphone portable (utilisant Maemo). Mais son exposé rapide portait sur la deuxième solution, un résolveur sur la machine de l'utilisateur. Demain, tout "*smartphone*" aura-t-il un Unbound qui tourne ?

Appliquée telle quelle, cette solution coûterait cher en trafic DNS supplémentaire puisque ce résolveur local n'aurait plus accès au cache partagé du résolveur commun. Les serveurs faisant autorité (par exemple ceux des TLD) risqueraient de souffrir. Une solution est possible : utiliser le résolveur commun, celui du FAI (ou du réseau local de l'organisation) comme "*forwarder*", par exemple, pour Unbound :

```
forward-zone:
     name:   "." # La racine, donc on fait tout suivre
     forward-addr: 198.51.100.53
     forward-addr: 203.0.113.129
```

Ainsi, Unbound fera la validation, mais les requêtes passeront par les deux serveurs `198.51.100.53` et `203.0.113.129`, qui pourront garder les résultats dans leur cache, économisant les ressources du réseau. Comme DNSSEC est une solution de bout en bout, qui sécurise le message et pas le canal par lequel le message est passé, cela fonctionnera et cela sera sûr.

Cela pose deux problèmes, un petit et un gros. Le petit est, on l'a vu, que sur une machine qui se déplace (un portable qui va au café ou à l'hôtel), il faut réécrire le fichier `unbound.conf` (ou `named.conf` pour BIND) à chaque fois qu'un serveur DHCP transmet de nouvelles adresses d'un serveur de noms résolveur. Pas un travail énorme mais il n'est pas fait aujourd'hui donc cette solution n'est pas encore accessible à l'utilisateur de base.

Le gros problème, et le cœur de l'exposé d'Hardaker, est que beaucoup de résolveurs (surtout dans les hôtels, trains et cafés, où personne ne sait exactement ce qui a été installé, mais également chez des FAI) ne sont pas des résolveurs corrects et massacrent plus ou moins les paquets DNS (à moins que ce massacre ne soit commis par la "*box*" insérée sur le chemin). Parmi les erreurs relevées par Hardaker :

— Résolveurs qui ne respectent pas le bit DO ("*DNSSEC OK*", cf. RFC 3225) et transmettent la demande sans ledit bit, ce qui produit des résponses sans signature,
— Résolveurs qui ne peuvent pas recevoir les réponses de grande taille (DNSSEC est bavard), problème en général lié à la fragmentation <`https://www.bortzmeyer.org/dns-size.html`>,
— Résolveurs qui filtrent les réponses NSEC3 (cf. RFC 5155), peut-être parce que le nom en partie gauche n'est pas le nom demandé mais un condensat cryptographique,
— Et bien d'autres encore.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4255.txt`

Résultat, on ne peut souvent pas utiliser le résolveur officiel. Il faut alors que le résolveur de la machine se résigne à parler directement à la racine et aux serveurs faisant autorité (sauf si le port 53 est filtré <https://www.bortzmeyer.org/port53-filtre.html>). C'est ce que teste un outil développé par l'auteur <http://www.my-maemo.com/software/applications.php?name=DNSSEC-Check&fldAuto=2183&faq=37>, qui fait tous ces tests et détermine si le résolveur officiel peut être utilisé comme "*forwarder*".

Un bémol toutefois : cette idée très « développement durable » (on utilise le résolveur officiel si on peut, pour économiser des ressources), n'est pas forcément dans l'intérêt de l'utilisateur. Un autre exposé à cette même conférence SATIN, par Nicholas Weaver, « "*Implications of Netalyzr's DNS Measurements*" <http://conferences.npl.co.uk/satin/papers/satin2011-Weaver.pdf> » montrait que, dans beaucoup de cas, le résolveur officiel est **plus lent**, pour une requête déjà en cache, que de demander au serveur faisant autorité ! Cela s'explique par le peu d'attention que beaucoup de gérants de réseau portent à leur résolveur.

À noter qu'un tel outil est également en cours de développement <https://www.bortzmeyer.org/dnssec-trigger.html> pour Unbound, pour permettre de configurer simplement son Unbound en résolveur local validant. Il pourra être utilisé par exemple via le très récent outil de contrôle d'Unbound pour faire un `unbound-control forward $(ldns-test-edns $IP_ADDRS)` qui activera le "*forwarding*" si et seulement si `ldsn-test-edns` ne répond pas `off`. D'autre part, ce sujet avait déjà fait l'objet d'une très bonne discussion sur la liste dns-operations, « "*DNSSEC validating clients that use upstream caching resolvers ?*" <https://lists.dns-oarc.net/pipermail/dns-operations/2011-February/006751.html> ».

---

https://www.bortzmeyer.org/ou-valider-dnsssec.html