Sécuriser le DNS, les deux approches

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 novembre 2006. Dernière mise à jour le 5 juin 2009

https://www.bortzmeyer.org/securiser-le-dns.html

Presque toutes les applications de l'Internet dépendent du DNS. Pourtant, ce protocole n'est absolument pas sécurisé et il est relativement facile d'y injecter de fausses données. Il existe deux approches (pas forcément concurrentes) pour sécuriser le DNS : signer cryptographiquement les données et s'assurer que la réponse vient bien du serveur interrogé.

Les failles du DNS, en matière de sécurité, sont bien décrites dans le RFC 3833 [1].

La seule manière fiable de sécuriser la distribution des données DNS est clairement la signature cryptographique, qui fournit une authentification forte des données. La norme pour cela se nomme DNSSEC (RFC 4033 et suivants). DNSSEC résout complètement le problème de l'injection de fausses données dans le DNS, et ceci indépendamment du serveur qui a répondu à la requête. Cela permet même d'abandonner le traditionnel système des serveurs « faisant autorité » au profit de systèmes de résolution "*peer to peer*" comme CoDoNS <http://www.cs.cornell.edu/People/egs/beehive/codons.php> (CoDoNS impose d'utiliser DNSSEC).

Mais, en pratique, DNSSEC est très peu déployé (certains TLD sont signés mais quasiment aucun serveur récursif ne valide avec DNSSEC). C'est en partie dû au fait que, si le protocole est stable et correct, les problèmes pratiques ont été peu étudiés :

— Comment remonter l'information aux applications (`getaddrinfo` ne le permet pas actuellement) ? Le résolveur DNS doit-il prendre des décisions et prétendre qu'un domaine n'existe pas si la vérification de la signature a échoué ? (Ce que font BIND ou Unbound <https://www.bortzmeyer.org/unbound-dnssec.html> par défaut.)
— Qui a la légitimité politique et technique pour signer la racine ? (D'où le système DLV, dans le RFC 4431.)

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

— Qui va assurer le support utilisateurs avec tous les problèmes que crée en général le déploiement d'un nouveau système de sécurité (clés perdues, clés qui expirent, comme cela s'est produit plusieurs fois sur les bancs de test où ne participaient pourtant que des experts, bogues logicielles, domaines parfaitement légaux qui deviennent « invalides » par suite d'un problème de signature), etc.

D'où l'intérêt de ne pas laisser tomber l'autre approche : vérifier que l'information vient bien du serveur « officiel ». C'est une authentification faible mais, en pratique, elle résoudrait bien des problèmes. En termes techniques (voir par exemple le RFC 3552), il s'agit de protéger le canal, DNSSEC protégeant le message. Bien sûr, protéger le canal ne sert pas à grand'chose si l'un des serveurs intermédiaires ment (c'est le cas des serveurs récursifs de beaucoup de FAI). Mais cette sécurité, complémentaire de celle qu'offre DNSSEC, est typiquement plus simple à déployer.

Il y a très longtemps, la plupart des serveurs de noms acceptaient n'importe quelle réponse, même « martienne » (une réponse martienne est une réponse à une question qu'on n'a pas posée). Il était donc facile de les tromper. Peu à peu, les implémentations du DNS sont devenues plus paranoïaques, en partie sous l'influence de Daniel Bernstein, dont l'épouvantable caractère et l'impossibilité à travailler avec les autres ne doivent pas faire oublier le rôle positif dans la compréhension des problèmes de sécurité du DNS <http://cr.yp.to/djbdns/notes.html> (utilisez avec précaution ce que vous trouvez sur son site Web <http://cr.yp.to/> : il y a beaucoup d'énormités). C'est par exemple Bernstein qui a conceptualisé l'importance de ne pas accepter les données hors-bailliage, c'est à dire servies par un serveur qui n'a pas autorité pour la zone : par exemple, la réponse d'un serveur de `.fr` incluant des adresses IP d'une machine en `.net`. Ces principes ont été mis en œuvre dans beaucoup de logiciels, comme BIND.

Il est donc plus difficile aujourd'hui de tricher ("*spoofing*") mais les résolveurs DNS ne mettent pas encore tous en œuvre toutes les recommandations qui figurent dans le RFC 5452. S'ils le faisaient, en pratique, nous aurions moins besoin de DNSSEC, et moins rapidement.