Programmeur·ses, méfiez-vous des données contenues dans le DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 mars 2022

https://www.bortzmeyer.org/injection-from-dns.html

Vous êtes programmeuse ou programmeur ? Vous écrivez des programmes qui reçoivent et traitent des données en provenance du grand méchant Internet ? Alors, vous savez certainement qu'il ne faut pas se fier à ces données, qu'elles peuvent être malveillantes, conçues pour créer des problèmes ? Vous le savez, non ? Vous ne passez pas ces données au shell sans précautions ? Ah, zut, d'après cet article <https://www.usenix.org/conference/usenixsecurity21/presentation/jeitner>, tout le monde ne le sait pas.

L'article met en avant le rôle du DNS. Mais, en fait, contrairement à ce que prétendent les auteur·es, le problème n'a pas forcément de rapport avec le DNS. Les failles de sécurité découvertes par les auteur·es de cet intéressant article sont dans deux catégories :

— Certaines viennent simplement d'une utilisation imprudente des données par une application. Que ces données aient été reçues via le DNS est un détail.
— D'autres sont davantage liées à l'utilisation du DNS.

Commençons par les premières. La plus spectaculaire affecte un logiciel très utilisé pour Eduroam, radsecproxy <https://radsecproxy.github.io/>. Pour faire des requêtes DNS, il passe, dans certains cas, des données obtenues via le DNS à un script shell sans aucune précaution particulière <https://github.com/radsecproxy/radsecproxy/security/advisories/GHSA-56gw-9rj9-55rc>. Si ces données contiennent une injection shell (`example.com ; rm -rf /...`, regardez `very-nasty.shaftinc.fr`), le code est exécuté. C'est évidemment très grave et heureusement que les auteur·es de l'article ont découvert cette faille. Mais, répétons-le, contrairement à la présentation sensationnaliste de l'article, ce n'est pas un problème DNS. Un programme qui reçoit des données de l'extérieur (que ce soit par le DNS, par HTTP, par LDAP ou d'autres) et les passe à un script shell sans précautions est mal écrit, point.

L'article affirme que la reponsabilité du DNS est engagée car le DNS passe des données sans les vérifier (sa « transparence »). C'est volontaire (RFC 3597 [1]) car c'est la seule solution pour pouvoir étendre le protocole et ajouter de nouveaux types de données. Si on écoutait les chercheurs en sécurité,

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3597.txt

on ne pourrait jamais déployer des nouveautés. Les auteur·es vont jusqu'à reprocher au DNS de ne pas tripoter les données ! (Un résumé de l'article est sur le blog de l'APNIC <https://blog.apnic.net/2022/02/22/resurrection-of-injection-attacks/>.)

Mais il y a une deuxième catégorie de failles dans l'article, cette fois liées au fonctionnement du DNS. Certains programmes analysent de manière incorrecte les données reçues. Par exemple, les noms de domaine ont une représentation texte (les classiques jolinom.example.com) et une représentation binaire sur le câble qui est très différente <https://www.bortzmeyer.org/representation-texte.html> (elle est de type {longueur, données} donc par exemple {7, jolinom} {7, example} {3, com} {0}). Passer de l'une à l'autre peut créer des surprises. Un nom de domaine peut inclure un point au milieu d'un composant (regardez le nom à droite du CNAME en dot-in-label.bortzmeyer.org), et un nom peut inclure un caractère nul (oui, un octet valant zéro, regardez le nom à droite du CNAME en nul-in-label.bortzmeyer.org) qui sera interprété par certaines bibliothèques comme une fin de chaine. En jouant avec ces subtilités, le gérant d'une zone peut envoyer des données qui seront mal interprétées, et mémorisées à tort par certains résolveurs <https://www.bortzmeyer.org/resolveur-dns.html>.

Dans ce cas et dans ce cas seulement, cela pourrait être intéressant, comme le note l'article, que les bibliothèques valident un peu plus les données. Par exemple, la partie droite d'un enregistrement MX comporte le nom d'un serveur de courrier. Ce nom est un nom de machine et les noms de machines sont soumis à une syntaxe bien plus restrictive que les noms de domaine <https://www.bortzmeyer.org/host-vs-domain.html>. Un sous-programme comme getaddrinfo pourrait donc refuser les noms illégaux. Les auteur·es montrent que la GNU libc le fait bien, mais hélas pas la uClibc.