RFC 7901 : CHAIN Query Requests in DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 juin 2016

Date de publication du RFC : Juin 2016

https://www.bortzmeyer.org/7901.html

Lorsqu'un client DNS parle à un résolveur, il pose une question et obtient une réponse. Avant DNSSEC, ce mode de fonctionnement simple était souvent satisfaisant. Mais, avec DNSSEC, il est beaucoup plus fréquent de devoir faire plusieurs requêtes pour obtenir toute l'information nécessaire pour valider les réponses. (Il faut les clés de toutes les zones situées entre la racine et la zone visée.) Cela coûtait cher en latence <https://www.bortzmeyer.org/latence.html>. Cette extension EDNS expérimentale permet au client DNS de demander au résolveur de chercher et de renvoyer toutes les réponses d'un coup.

Cette extension est particulièrement utile pour le cas de machines terminales <https://www.bortzmeyer.org/terminal-host.html> hébergeant leur propre résolveur validant <https://www.bortzmeyer.org/son-propre-resolveur-dns.html> (ce qui est la meilleure configuration <https://www.bortzmeyer.org/ou-valider-dnsssec.html>, question confiance et sécurité). Ce n'est donc pas un hasard si l'auteur du RFC travaille chez Red Hat, système qui est livré par défaut avec une telle configuration. Mais, lorsqu'un tel résolveur validant veut vérifier les informations obtenues sur `foo.bar.example`, il devra (en supposant qu'il y a une zone par composant du nom de domaine) obtenir la délégation sécurisée de `example` (`dig @la-racine DS example`), la clé de `example` (`dig @ns-example DNSKEY example`), la délégation sécurisée de `bar.example`, etc (la clé de la racine, elle, est en dur dans le résolveur, il faut bien partir de quelque part). À faire séquentiellement, cela serait beaucoup de requêtes, donc du temps passé à attendre les réponses. Sur des liens à latence élevée (ce qui arrive souvent aux machines terminales), cela peut être pénible, même si le cache DNS aidera, pour les requêtes suivantes.

L'idée de cette extension (sections 1 et 3 du RFC) est donc que le résolveur validant local ait un "*forwarder*" (attention, le RFC utilise un vocabulaire erroné, en donnant à "*forwarder*" un sens différent de celui qu'il a dans les RFC 2308[1] et RFC 8499 ; j'utilise, moi, la terminologie standard). Le résolveur

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2308.txt

validant local va demander à ce "*forwarder*", grâce à la nouvelle extension EDNS `CHAIN`, d'envoyer tout d'un coup (tous les DS et DNSKEY nécessaires). Bien sûr, le "*forwarder*", lui, devra faire toutes les requêtes mais, a priori, il a un plus gros cache et sera mieux connecté.

Cette nouvelle extension est donc conçue pour des résolveurs, et est ignorée par les serveurs faisant autorité. Notez que le résolveur validant local peut être un démon autonome (Unbound tournant sur mon portable Unix) ou bien une partie d'une application qui embarquerait ses propres fonctions de résolution de noms.

Le format de l'extension est décrit en section 4 du RFC. C'est une option EDNS (RFC 6891), encodée, comme les autres options EDNS, en TLV. Le type (le code) est 13 <https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-11>. La valeur est composée d'un seul champ, l'ancêtre le plus proche ("*Closest Trust Point*") dont on connaisse les informations nécessaires à la validation. Le résolveur validant local a en effet certaines informations (dans sa configuration ou dans son cache) qu'il n'est pas nécessaire de lui envoyer. Dans l'exemple plus haut, si le résolveur validant local connait déjà la clé DNSSEC de `example`, il mettra dans le champ "*Closest Trust Point*" ce nom de domaine, indiquant au "*forwarder*" qu'il peut se contenter des informations situées plus bas, dans l'arbre du DNS. Ce nom est encodé dans le format habituel du DNS (qui n'est **pas** le format texte avec les points <https://www.bortzmeyer.org/representation-texte.html>).

La section 5 du RFC décrit comment utiliser cette extension au DNS. Si on veut tester les capacités du résolveur qu'on interroge, on peut utiliser une option `CHAIN` vide (longueur nulle). Si le serveur à qui on a envoyé cette option répond avec la même option nulle, c'est bon. Attention, les serveurs récursifs qui mettent en œuvre `CHAIN` n'accepteront des requêtes « réelles » (longueur non nulle) qu'au-dessus d'un transport où l'adresse IP source est vérifiée. Le but est d'éviter les attaques par réflexion avec amplification <https://www.bortzmeyer.org/attaques-reflexion.html> (voir aussi la section 7.2). Pour vérifier l'adresse IP source (ce qui ne se fait normalement pas en UDP), il y a plusieurs solutions, notamment TCP (RFC 7766) et les gâteaux (RFC 7873).

Une fois qu'on a un tel transport, et que le client DNS a testé que le serveur qu'il interroge gère bien `CHAIN`, on peut y aller. Le client met l'ancêtre le plus proche (dont il a les clés) du nom demandé dans le champ "*Closest Trust Point*". Dans le cas le plus courant (résolveur validant configuré avec une seule clé, celle de la racine), le résolveur « froid », qui vient de démarrer et dont le cache est vide, il commencera par mettre la racine en "*Closest Trust Point*" puis, au fur et à mesure qu'il se « réchauffera » (que son cache se peuplera), il pourra mettre des noms plus proches du nom demandé (et donc plus éloignés de la racine). Par exemple, si le résolveur validant local est configuré avec la clé de la racine, et qu'il a appris par les réponses précédentes la clé de `example`, mais pas celle de `bar.example`, et qu'il veut des informations sur le nom `foo.bar.example`, son option `CHAIN` vaudra {type = 13, longueur = 9, valeur = 0x07 0x65 0x78 0x61 0x6d 0x70 0x6c 0x65 0x00} (la longueur est celle de la partie « valeur » uniquement, le nom `example` est encodé selon la norme DNS). S'il connaissait également la clé de `bar.example`, son option `CHAIN` vaudrait {type = 13, longueur = 13, valeur = 0x03 0x62 0x61 0x72 0x07 0x65 0x78 0x61 0x6d 0x70 0x6c 0x65 0x00}. D'autres exemples figurent en section 9 du RFC.

Faut-il envoyer l'option `CHAIN` à chaque requête ? On peut mais il est recommandé de se souvenir de quels serveurs la gèrent et de n'envoyer qu'à ceux-ci (autrement, non seulement on fait du travail inutile mais on renseigne des serveurs extérieurs sur l'état de son cache). Comme il existe des "*middleboxes*" boguées sur certains trajets, la stratégie de repli du RFC 6891, section 6.2.2 peut être utile.

Et le serveur interrogé, que fait-il ? S'il accepte de répondre à une requête contenant l'extension `CHAIN`, il doit :

— Ajouter à la réponse, dans la section Autorité, les enregistrements DNSSEC nécessaires (DS, DNSKEY et NSEC),

— Mettre une CHAIN dans la réponse, avec la valeur "*Closest Trust Point*" mise au nom le plus bas (le plus éloigné de la racine) pour lequel ces informations sont nécessaires. (C'est surtout utile lorsque le serveur n'envoie pas une chaîne complète, par exemple pour économiser le réseau.)

Évidemment, si la question avait une erreur de syntaxe (taille de la partie Valeur inférieure à la Longueur, par exemple), le serveur répond FORMERR ("*FORmat ERRor*").

La section 7 sur la sécurité étudie quelques programmes que peut poser cette extension au DNS. D'abord, mettre en œuvre cette option fatigue davantage le serveur interrogé, en terme de travail et de capacité du réseau. Un serveur est donc toujours libre d'ignorer les options CHAIN et de s'en tenir au service minimum.

Ensuite, comme vu plus haut, les réponses suivant une question qui utilise CHAIN vont être évidemment plus grosses que les réponses DNS habituelles. Il y a donc un risque d'attaques par réflexion avec amplification <https://www.bortzmeyer.org/attaques-reflexion.html>, si un attaquant usurpe l'adresse IP de sa victime. C'est pour cela que notre RFC impose de ne répondre avec une chaîne complète que si l'adresse IP du client a été vérifiée (par exemple parce qu'il utilise TCP, ou bien les "*cookies*" du RFC 7873).

CHAIN a aussi quelques effets sur la vie privée. Le résolveur validant local va indiquer à son "*forwarder*" (et à tout espion qui surveille le trafic) comment il est configuré et ce qu'il y a dans son cache.

Il ne semble pas qu'il existe de mise en œuvre de cette option CHAIN pour l'instant, même si c'est en projet pour Go <https://github.com/miekg/dns/issues/381>.

Si vous vous intéressez à la conception des protocoles réseaux, notez que cette extension a fait l'objet d'une discussion <https://mailarchive.ietf.org/arch/search/?email_list=dnsop&gbt=1&index=YAOKdXMZe4iMt2HV0CT-cAtjVKQ> pour savoir s'il ne valait pas mieux, pour réduire la latence, envoyer toutes les requêtes possibles en parallèle (cette idée a finalement été rejetée).