Un exemple du forçage des TTL DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 mars 2013

https://www.bortzmeyer.org/forcer-ttl.html

Le DNS ne marche que parce que les données sont conservées dans les caches par les résolveurs. Autrement, si chaque requête remontait aux serveurs faisant autorité, ceux-ci souffriraient beaucoup. Combien de temps le résolveur peut-il garder l'information dans les caches ? C'est le serveur faisant autorité qui choisit, via le champ TTL de la réponse. Certains abusent en mettant des TTL très bas, poussant d'autres à ne pas les respecter.

Voici un exemple d'une requête DNS avec dig, montrant le TTL :

```
% dig @ns1.fdn.org MX fdn.fr
...
;; ANSWER SECTION:
fdn.fr. 86400 IN MX 20 mx2.fdn.fr.
fdn.fr. 86400 IN MX 10 mail.fdn.fr.
...
```

Le TTL est ici de 86 400 (secondes, soit une journée). C'est ce que sert `ns1.fdn.org`, qui est un des serveurs faisant autorité pour `fdn.fr`. Sur le résolveur, le TTL sera égal ou inférieur puisqu'il vaut la valeur originelle, au début, puis diminue avec le temps. Sur le résolveur de ma machine :

```
% dig MX fdn.fr
...
;; ANSWER SECTION:
fdn.fr. 4250 IN MX 20 mx2.fdn.fr.
fdn.fr. 4250 IN MX 10 mail.fdn.fr.
```

Ici, l'information est dans le cache depuis 22 heures (86 400 secondes - 4 250).

Jusqu'ici, tout va bien. Le problème est que certains gérants DNS mettent des valeurs ridiculement basses, en général parce qu'ils n'ont pas compris à quoi servaient les caches, et qu'ils fantasment sur une information distribuée en temps réel. On voit ainsi des TTL de moins d'une minute, par exemple pour `dyn.com` :

```
% dig @ns1.p01.dynect.net. A dyn.com
...
;; ANSWER SECTION:
dyn.com. 30 IN A 134.0.76.51
```

C'est très bas. C'est très égoïste puisque cela va augmenter le trafic réseau.

On voit aussi des administrateurs de résolveurs DNS qui répondent en violant le protocole DNS. Ce dernier est normalisé dans le RFC 1034 [1] (le TTL est décrit dans sa section 3.6). Un résolveur peut garder l'information en cache moins longtemps que le TTL (par exemple s'il redémarre, le résolveur typique perd tout). Il existe aussi des options des logiciels serveurs pour imposer un TTL maximal, pour économiser la mémoire (`max-cache-ttl` dans BIND, `cache-max-ttl` dans Unbound). Mais BIND n'a pas d'option pour mettre une valeur minimale, et pour cause, ce serait une violation du protocole : le TTL est fixé par le serveur faisant autorité, le résolveur n'a normalement pas le droit de le prolonger.

Certains, pourtant, ont modifié leur logiciel pour le faire. Voici le résolveur d'une Freebox :

```
% dig A dyn.com
...
;; ANSWER SECTION:
dyn.com. 823 IN A 134.0.76.51
```

823 est supérieur à 30. Le TTL a donc été remonté à une valeur non décidée par le serveur faisant autorité et va donc rester dans le cache plus longtemps que souhaité par le gérant de `dyn.com`. (Le résolveur de la Freebox étant un dnsmasq, qui n'a pas non plus d'option pour cela, il s'agit sans doute d'une version localement modifiée. Notez qu'Unbound, lui, a une option pour forcer un TTL en ne respectant pas les valeurs indiquées, `cache-min-ttl`, merci à Laurent Frigault pour ce rappel.)

Il faut aussi savoir que les signatures DNSSEC protègent également le TTL (RFC 4034, section 3.1.4) et que ce forçage de TTL, comme toute modification non autorisée, peut potentiellement invalider les signatures (merci à Phil Regnauld pour avoir noté ce point).

À noter un article des gens de dyn.com <https://help.dyn.com/everything-you-ever-wanted-to-kno> expliquant leur choix. Merci à m0_o <https://twitter.com/m0_o> pour l'idée et les données.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1034.txt`