Vider le cache d'un résolveur DNS, pour un seul domaine

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 février 2012. Dernière mise à jour le 2 mars 2012

https://www.bortzmeyer.org/vider-cache-resolveur.html

Lorsqu'on administre un résolveur DNS, il arrive que les informations concernant un domaine soient erronées, et qu'il faille les oublier, les supprimer du cache. La plupart des administrateurs système redémarrent le démon. Mauvaise méthode, trop radicale (elle fait perdre l'intégralité du cache). Tous les logiciels résolveurs sérieux permettent au contraire de ne supprimer qu'un seul domaine du cache.

Il peut y avoir plusieurs raisons pour ces mauvaises données dans le cache : empoisonnement délibéré (attaque Kaminsky <https://www.bortzmeyer.org/comment-fonctionne-la-faille-kaminsky.html>), erreur dans le domaine (comme la panne de .SE <https://www.bortzmeyer.org/panne-de-point-se.html>), embrouille DNSSEC (signature expirée...), etc. Normalement, aucune de ces causes n'est permanente. Mais, comme le résolveur DNS a un cache, une fois la mauvaise information acquise, il va la garder pendant une durée indiquée par le TTL alors qu'on voudrait, sachant que le problème a été corrigé, repartir tout de suite du bon pied.

À ce stade, beaucoup d'administrateurs système ont appliqué la méthode Windows : « dans le doute, reboote » et redémarré le serveur DNS voire, pour les plus windowsiens, toute la machine. Le résultat est qu'ils perdent la totalité du cache et qu'ils privent de service DNS les clients pendant quelques secondes ou quelques minutes. Une bien meilleure méthode est de ne vider ("*to flush*", dans la langue de Jean Auel) que le domaine en cause (`foobar.example` dans les exemples qui suivent).

Avec Unbound, cela se fait ainsi :

```
# unbound-control flush_zone foobar.example
ok removed 78 rrsets, 60 messages and 1 key entries
```

Et, avec BIND :

```
# rndc flushname foobar.example
```

Notez au passage que, pour d'évidentes raisons de sécurité, ces commandes ne sont en général accessibles qu'à root (elles nécessitent la lecture d'un fichier de configuration contenant les informations d'authentification, fichier lisible par root seul, autrement vous aurez un message comme `fopen:Permission denied:bss_file.c:356:fopen('/etc/unbound/unbound_control.pem','r')` ou comme `open: /etc/bind/rndc.key: permission denied`).

Les deux commandes n'ont pas tout à fait la même sémantique (merci à David Gavarret pour le rappel). `unbound-control flush_zone` vide récursivement tous les noms situés sous le nom indiqué (la commande `unbound-control flush` est son équivalent en non-récursif). Au contraire, BIND n'a qu'une commande, `flushname`, et elle est non-récursive. On ne peut donc pas l'utiliser pour des cas comme celui de `.se` cité plus haut, où il fallait vider le cache de tous les noms se terminant par `.se`. BIND 9.9 (pas encore sorti aujourd'hui) apportera la commande `rndc flushtree` qui lui donnera les mêmes possibilités qu'à Unbound.

Ces deux commandes (celle pour Unbound et celle pour BIND) nécessitent une certaine configuration pour assurer l'authentification. Elle est typiquement faite automatiquement lors de l'installation mais cela peut dépendre du paquetage utilisé. Pour Unbound, il faut en outre veiller à ce que le fichier de configuration `unbound.conf` contienne `control-enable: yes`. La création des clés X.509 utilisées par Unbound pour l'authentification se fait avec un programme livré avec Unbound :

```
# unbound-control-setup
setup in directory /etc/unbound
generating unbound_server.key
Generating RSA private key, 1536 bit long modulus
.....................++++
...................++++
e is 65537 (0x10001)
generating unbound_control.key
Generating RSA private key, 1536 bit long modulus
.........................++++
.............................................++++
e is 65537 (0x10001)
create unbound_server.pem (self signed certificate)
create unbound_control.pem (signed client certificate)
Signature ok
subject=/CN=unbound-control
Getting CA Private Key
Setup success. Certificates created. Enable in unbound.conf file to use
```

Pour BIND, la configuration par défaut inclus en général ce qu'il faut avec un fichier `/etc/bind/rndc.key` lu par le serveur et par `rndc`. Si ce n'est pas le cas, voir la fin de mon article sur le RFC 6168 [1], qui donne tous les détails.

Vous pouvez tester que la configuration fonctionne avec une commande inoffensive comme :

```
# unbound-control status
```

ou :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6168.txt`

```
# rndc status
```

Par contre, je n'ai pas d'expérience pratique de cette technique avec le troisième grand logiciel libre sur ce créneau, PowerDNS recursor. Mais Jean-Eudes Onfray s'en est chargé et voici ses conclusions :

```
# /usr/bin/rec_control wipe-cache foobar.example
```

Pour les versions antérieures à 3.1, il faut ajouter un point à la fin du domaine :

```
# /usr/bin/rec_control wipe-cache foobar.example.
```

Comme la `rndc flushname` de BIND, cette commande est non-récursive.

Quant au serveur DNS de Windows, il est très difficile de trouver de l'information fiable et je n'ai pas testé. Mais Clément Fender l'a fait et voici ses conclusions :

J'ai rapidement maquetté (deux serveurs DNS : un hébergeur et un résolveur) et longuement pris le temps de tester sur un environnement composé de deux serveurs fonctionnant sous Windows 2008. J'ai établi les conclusions suivantes :

— les commandes que vous proposiez (`DnsCmd 127.0.0.1 /NodeDelete foobar.example foobar.example`), et celles énoncées sur Technet <`http://technet.microsoft.com/en-us/library/cc772069%28v=ws.10%29.aspx`>, n'ont pas fonctionné.
— afin d'obtenir le résultat escompté (suppression du cache des informations d'une zone) je suis (après moult essais) parvenu à faire fonctionner la commande décrite plus loin.

```
C:\Users\Administrateur> dnscmd 127.0.0.1 /nodedelete . dns1.zone. /tree
Êtes-vous sûr de vouloir supprimer le noud ? (o/n)y
Le serveur DNS 127.0.0.1 a supprimé le noud au niveau de dns1.zone. :
    Statut = 0 (0x00000000)
La commande s'est terminée correctement.
```

Cette commande a aussi supprimé les informations relatives à des zones filles de `dns1.zone`. Il faut indiquer le commutateur `/tree` car sans son utilisation, la commande abouti mais ne supprime que les enregistrement directement liés à la zone (SOA et NS). Les enregistrements A présents directement dans cette zone ne sont pas supprimés du cache.

Pour la commande utilisant le commutateur `/zonedelete`, elle fonctionne mais ne répond pas au besoin exprimé : elle n'agit pas sur les information stockées en cache sur le serveur. Dans mon cas de maquettage, elle a abouti à la suppression pure et simple de la configuration du redirecteur conditionnel ... sans toucher aux informations stockées en cache.

```
C:\Users\Administrateur> dnscmd 127.0.0.1 /zonedelete dns1.zone.
Êtes-vous sûr de vouloir supprimer la zone ? (o/n)y
Le serveur DNS 127.0.0.1 a supprimé la zone dns1.zone. :
    Statut = 0 (0x00000000)
La commande s'est terminée correctement.
```

Afin de supprimer du cache un enregistrement d'adresse (A) présent dans la zone :

```
C:\Users\Administrateur> dnscmd 127.0.0.1 /nodedelete . arecord.dns1.zone.
Êtes-vous sûr de vouloir supprimer le noud ? (o/n)y
Le serveur DNS 127.0.0.1 a supprimé le noud au niveau de arecord.dns1.zone. :
    Statut = 0 (0x00000000)
La commande s'est terminée correctement.
```

Attention cependant sur les versions françaises du système, si l'on n'utilise pas le commutateur `/f` pour forcer, à la demande de confirmation (invitant à saisir o/n) il faut répondre y(es) car o ne fonctionne pas. (Fin du compte-rendu de Clément Fender.)

Et, enfin, certains résolveurs DNS publics offrent une interface Web pour vider le cache pour un domaine donné. C'est par exemple le cas de Google Public DNS <https://developers.google.com/speed/public-dns/cache>.

Petit avertissement : dans les entretiens d'embauche pour recruter un administrateur système, je pose cette question :-)