Limiter le trafic d'un serveur DNS (notamment d'un récursif ouvert)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 mars 2012

https://www.bortzmeyer.org/rate-limiting-dns-open-resolver.html

---

Il peut y avoir plusieurs raisons de limiter le trafic entrant dans un serveur DNS. Pour protéger le serveur lui-même ? Non, pas tellement, car, sur un serveur faisant autorité, cela consomme en général moins de ressources sur la machine pour répondre, que pour décider si on accepte de répondre. Sur un résolveur (un serveur récursif), la limitation du trafic peut déjà être plus utile. Mais elle est surtout indispensable si on gère un résolveur ouvert, accessible à tout l'Internet. Dans ces conditions, une forme de limitation de trafic est indispensable, car, sinon, on ne risque pas que ses propres ressources, mais celles des autres ; un serveur récursif ouvert est en effet une cible tentante pour des attaques par réflexion et amplification.

Le problème est connu depuis des années (mon premier article date de 2006 <https://www.bortzmeyer.org/fermer-les-recursifs-ouverts.html>) et le RFC 5358[1] dit clairement que ces serveurs récursifs ouverts sont **fortement déconseillés**. Toutefois, si on tient absolument à avoir un serveur DNS récursif ouvert, comment peut-on limiter le risque de devenir le relais de l'attaque, le complice involontaire d'une attaque par déni de service ?

La solution réside dans la limitation de trafic. Elle peut se faire sur un boîtier posé en avant du serveur (il en existe plusieurs modèles, tous très chers et au logiciel non libre). Ces boîtiers ont souvent de sérieuses restrictions quant au trafic accepté (si vous en évaluez un, regardez s'il laisse passer IPv6, EDNS, DNSSEC, etc). Et puis, pour un serveur connecté à l'Internet, il est toujours plus simple et plus cohérent qu'il assure sa propre protection.

Si le serveur tourne sur Linux, on a tout ce qu'il faut dans Netfilter. On trouve en ligne une quantité formidable de documentations sur Netfilter et sa commande `iptables`. Mais presque toutes ne parlent que des connexions TCP et utilisent donc un mécanisme qui garde un état, ce qui est coûteux en ressources (j'ai moi aussi fait un article sur le filtrage pour un service TCP <https://www.bortzmeyer.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5358.txt

org/rate-limiting-dos.html>). Le trafic DNS du résolveur étant en général strictement « un paquet pour la requête, un paquet pour la réponse », des solutions Netfilter comme les modules « state » ou « connlimit » n'ont sans doute guère d'intérêt (tous les paquets entrants seront dans l'état « nouveau flot, jamais encore vu »).

La meilleure solution me semble donc être avec le module « hashlimit », qui ne garde pas d'état par « session » (il faut évidemment un peu d'état pour chaque préfixe IP). Par exemple :

```
# iptables -A INPUT -p udp --dport 53 -m hashlimit \
   --hashlimit-name DNS --hashlimit-above 20/second --hashlimit-mode srcip \
   --hashlimit-burst 100 --hashlimit-srcmask 28 -j DROP
```

Cela permet 20 requêtes par seconde à chaque préfixe /28, avec un pic à 100 requêtes si nécessaire (le trafic sur l'Internet est très variable, avec des pics importants). Si iptables vous répond `Unknown arg [Caractère Unicode non montré [2] ]--hashlimit-above'`, c'est que vous avez une version trop ancienne pour faire de la limitation de trafic sérieuse (les options que j'utilise sont apparues avec la 1.4.)

Notez bien que cette règle s'applique à toutes les requêtes DNS, quels que soient le nom demandé et le type de données demandé. Il existe des solutions pour se limiter à certaines requêtes mais, avec le protocole DNS, ce n'est pas facile <https://www.bortzmeyer.org/dns-netfilter-u32.html>.

julienth37 a adapté à IPv6 (masque de préfixe plus long) :

```
# ip6tables -A INPUT -p udp -m udp --dport 53 -m hashlimit \
   --hashlimit-above 10/sec --hashlimit-burst 20 --hashlimit-mode srcip --hashlimit-name DNS \
   --hashlimit-srcmask 64 -j DROP
```

Testons le un peu avec queryperf <https://www.bortzmeyer.org/performances-serveur-dns.html>. Avec un cache rempli et la limitation de trafic :

```
Statistics:

  Queries sent:         10000 queries
  Queries completed:    7039 queries
  Queries lost:         2961 queries

  Percentage completed:  70.39%
  Percentage lost:       29.61%
```

On peut aussi demander à Netfilter ce qu'il a vu :

```
# iptables -v -n -L INPUT
Chain INPUT (policy ACCEPT 21341 packets, 3194K bytes)
 pkts bytes target     prot opt in     out     source               destination
 2944  190K DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0        udp dpt:53 limit: above
```

2. Car trop difficile à faire afficher par LaTeX

Et le nombre de paquets jetés correspond en effet à celui des requêtes pour lesquelles queryperf n'a pas eu de réponse. On a donc limité l'attaquant (queryperf est, par défaut, nettement moins agressif qu'un vrai attaquant, qui enverrait les paquets à un rythme plus soutenu et en perdrait donc davantage).

Notez que la même machine, sans règle de limitation du trafic, fait :

```
Statistics:

  Queries sent:         10000 queries
  Queries completed:    9981 queries
  Queries lost:         19 queries

  Percentage completed:  99.81%
  Percentage lost:        0.19%
```

Bien sûr, un vrai test serait plus complexe : il faudrait un grand nombre de machines clientes, pour tenter d'épuiser les ressources du résolveur DNS utilisé comme relais. Mais c'est un premier pas vers la sécurisation d'un serveur récursif ouvert.

Un exemple d'un résolveur ouvert, pour de bonnes raisons est celui de l'OARC <https://www.dns-oarc.net/oarc/services/odvr>, pour tester DNSSEC. Un autre est celui de Telecomix <http://dns.telecomix.org/>, pour fournir un service de résolution qui ne censure pas. Bien sûr, la majorité des résolveurs ouverts le sont par négligence et ignorance et les non-administrateurs de ces machines ne la protégeront sans doute pas par un limiteur de trafic mais, pour les rares résolveurs ouverts sérieux, c'est une bonne idée.

---

https://www.bortzmeyer.org/rate-limiting-dns-open-resolver.html