RFC 6302 : Logging recommendations for Internet facing servers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 juillet 2011

Date de publication du RFC : Juin 2011

https://www.bortzmeyer.org/6302.html

Ce court RFC prend position clairement en faveur d'un enregistrement, par les serveurs Internet, du numéro de port source du client, en plus de l'adresse IP. Cette recommandation est là pour tenir compte de la pratique de plus en plus répandue du partage d'adresses IP, notamment en raison de la pénurie des adresses IPv4.

En effet, depuis l'épuisement <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html> de ces adresses v4 début 2011, le partage d'adresses IP entre machines, par exemple via un routeur NAT, est de plus en plus fréquent. Or, de nombreux serveurs enregistrent dans un journal les adresses IP de leurs clients. Voici un exemple avec Apache :

```
2001:db8:1:4::1972 - - [29/Jun/2011:12:11:27 +0200] "GET /racine-dns-28-juin-2011.html HTTP/1.1" 200 9392 "-" "M
```

et un avec Postfix :

```
Jun 29 12:04:30 aetius postfix/smtpd[29360]: connect from foo.bar.example[198.51.100.79]
```

(Les adresses ont été changées.)

Noter les adresses IP seules n'a guère de sens puisque ces adresses n'identifient plus une machine unique. Le RFC recommande donc que le port source soit également noté. Le processus de discussion qui a mené à ce RFC avait commencé au moins deux ans avant, à la réunion d'Hiroshima <https://www.bortzmeyer.org/loguer-adresse-et-port.html>.

Évidemment, la meilleure solution au partage d'adresses serait le déploiement d'IPv6. Mais comme il va prendre des années, il faut bien gérer la situation existante. Cela implique des solutions comme le NAT44 (le NAT traditionnel, cf. RFC 3022 [1] et mon article sur les différentes formes de NAT <https://www.bortzmeyer.org/nats.html>), le NAT64 (RFC 6146) ou bien DS-Lite (RFC 6333). Ce partage d'adresses pose des tas de problèmes (documentés dans le RFC 6269) mais celui qui nous intéresse ici est le problème de l'enregistrement de l'adresse du client.

Prenons un NAT44 classique. Le client a une adresse privée (RFC 1918), mettons `192.168.42.1` et utilise le port source 52645. Il passe à travers un routeur NAT et ressort avec l'adresse publique `203.0.113.68` et le port source 61921. Le routeur a mémorisé l'association entre les deux tuples, interne {`192.168.42.1`, 52645} et externe {`203.0.113.68`, 61921}. Le serveur auquel se connecte ce client ne verra que le tuple externe. Si le serveur enregistre uniquement l'adresse `203.0.113.68`, il ne pourra pas distinguer les requêtes faites par la machine `192.168.42.1` de celles faites par les autres machines situées derrière le même routeur NAT. Ce n'est pas forcément très grave pour du NAT fait à la maison (après tout, l'HADOPI coupera tout le monde, papa, maman et petit frère, si la fille aînée a téléchargé illégalement) mais c'est bien plus gênant pour du CGN, technique récente où plusieurs abonnés sans lien entre eux partagent une adresse IP.

Solution ? Que le serveur enregistre également le port source (ici, 61921). C'est désormais une recommandation officielle de l'IETF (voilà pourquoi ce document se nomme également « BCP 162 » pour "*Best Common Practices*"). Cette information, **jointe à celle contenue dans le routeur NAT**, permettra de retrouver le client original.

Notez le point mis en évidence : tout ceci n'a un sens que si le routeur NAT enregistre aussi les correspondances entre tuples {adresse, port} interne et externe. Le RFC prend soin de préciser que ses recommandations ne s'appliquent qu'aux serveurs, pas aux routeurs NAT qui, en pratique, à l'heure actuelle, n'enregistrent pas cette information (cf. section 3 sur ce point).

La recommandation exacte du RFC figure en section 2. Je la reprends ici :

— **Si** (et c'est un gros si) le serveur enregistre l'adresse IP de ses clients, il doit aussi noter le port source,
— Ainsi qu'une heure précise à la seconde près, et en utilisant une horloge qui soit synchronisée, par exemple via NTP (RFC 5905), et de préférence en UTC,
— Et des informations comme le protocole de transport utilisé, au cas où le serveur en accepte plusieurs (les correspondances dans le routeur NAT dépendent en général du protocole de transport).

Vous avez bien noté que le RFC ne prend pas position sur la question de savoir si un serveur **devrait** enregistrer l'adresse IP de ses clients (note au passage : il existe des serveurs qui n'enregistrent pas cette adresse, justement pour éviter de « fliquer » leurs propres clients ; cela leur évite également la responsabilité d'un fichier sensible, puisque contenant des données à peu près nominatives). Il dit juste que, **si** le serveur le fait, il devrait également noter le port source et l'heure. Enregistrer l'adresse IP seule ne sert pas à grand'chose (sauf, je suppose, si le but est justement de ne pas garder de données trop nominatives. Au passage, un peu de publicité pour le service No Log <http://www.no-log.org/>.)

De la même façon, le RFC évite prudemment les questions de rétention des données (lesquelles, pour combien de temps). Chaque administrateur système doit consulter la loi locale (en France, la LCEN, loi liberticide qui contient des obligations très rigoureuses à ce sujet).

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc3022.txt`

Enregistrer l'heure précise est nécessaire car le serveur ne sait évidemment pas à quel rythme sont réutilisés les numéros de port par le routeur NAT.

Voyons maintenant la mise en œuvre concrète de ces recommandations, avec les deux logiciels cités plus haut. Postfix a cette possibilité depuis assez longtemps (malgré les moqueries assez bêtes de son auteur <http://www.irbs.net/internet/postfix/0401/0392.html> au début) :

```
smtpd_client_port_logging = yes
```

Cela donne :

```
Jun 29 12:04:39 aetius postfix/smtpd[30055]: connect from foo.bar.example[198.51.100.79]:48525
```

Par contre, son concurrent sendmail ne semble pas avoir cette capacité (il faudrait peut-être modifier le source, en `sendmail/srvrsmtp.c` ou bien utiliser un milter `smfi_connect`, peut-être avec la macro `${client_port}`.) Et je n'ai pas d'informations sur d'autres serveurs comme exim.

Pour Apache, cela se fait en ajoutant un `%{remote}p` dans le `LogFormat` :

```
LogFormat "%h %{remote}p %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %v" combined
```

ce qui donnerait :

```
2001:db8:1:4::1972 42425 - - [29/Jun/2011:12:11:27 +0200] "GET /racine-dns-28-juin-2011.html HTTP/1.1" 200 9392
```

(Le 42425 en deuxième champ.) Attention, si on analyse ses fichiers avec un programme de statistique comme analog, il faut changer également sa configuration, pour qu'il ne soit pas surpris par le champ supplémentaire.

J'ai montré un exemple avec une adresse IPv6. C'est sans doute inutile pour ce protocole (pas de traduction d'adresse/port en IPv6) mais c'est plus simple de mettre la même configuration pour les deux. Le paragraphe suivant présente un exemple où IPv4 et IPv6 sont traités différemment.

Si on veut faire plus joli, et écrire les adresses dans le style habituel des tuples {adresse, port} (adresse, deux-points, port dans le cas d'IPv4 et avec des crochets dans le cas d'IPv6, cf. RFC 3986), c'est plus compliqué. La solution que j'utilise (merci à @bitonio <http://twitter.com/bitonio> et @siddartha <http://twitter.com/siddartha>) passe par la définition d'une variable <http://httpd.apache.org/docs/2.2/mod/mod_setenvif.html#setenvif> puis son utilisation dans la définition du format <http://httpd.apache.org/docs/2.0/mod/mod_log_config.html#customlog>. (Une autre solution, due à @Grunt_ <http://twitter.com/Grunt_>, aurait été d'avoir un "*virtual host*" pour IPv4 et un pour IPv6.) Cela se configure ainsi :

```
LogFormat "[%h]:%{remote}p %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %v" combinedv6
LogFormat "%h:%{remote}p %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %v" combinedv4

SetEnvIf Remote_Addr : remote-v6
CustomLog /var/log/apache2/access.log combinedv6 env=remote-v6
CustomLog /var/log/apache2/access.log combinedv4 env=!remote-v6
```

Pour reconnaître une adresse IPv6, je cherche simplement la présence d'un deux-points. Si on veut faire mieux, il existe de jolies expressions rationnelles <http://www.d-sites.com/2008/10/09/regex-ipv4-et-ipv6/> pour tester les familles d'adresses. Quoi qu'il en soit, le résultat est :

```
[2001:db8:1:4::1972]:42425 - - [29/Jun/2011:12:11:27 +0200] "GET /racine-dns-28-juin-2011.html HTTP/1.1" 200

192.0.2.202:53124 - - [01/Jul/2011:08:29:15 +0200] "GET /6269.html HTTP/1.1" 200 37392 "-" "Netvibes (http:/
```

Là aussi, il faut penser aux programmes d'analyse, dont la plupart ne sont pas capables de traiter des cas où le format varie d'une ligne à l'autre. La première solution, moins jolie, était peut-être plus raisonnable.