RFC 7707 : Network Reconnaissance in IPv6 Networks

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 mars 2016

Date de publication du RFC : Mars 2016

https://www.bortzmeyer.org/7707.html

Soit un pauvre petit réseau innocent, et un méchant attaquant qui va essayer de faire subir les derniers outrages au réseau en question. L'attaquant commence souvent par une phase de **reconnaissance**, où il cherche à améliorer sa connaissance du réseau visé. Un outil souvent utilisé dans cette phase est le **balayage** ("*scanning*") systématique du réseau, à la recherche d'adresses IP de machines à attaquer. En IPv6, contrairement à IPv4, la tâche semble colossale, vu le nombre d'adresses IP possibles. Mais, comme l'avait déjà noté le RFC 5157 [1], cette tâche n'est pas impossible. Ce nouveau RFC fait le point sur le balayage IPv6 et détaille les techniques utilisables.

Comme exemple de la différence de situation entre IPv4 et IPv6, prenons un réseau local typique. La **densité** de machines (nombre de machines réellement présentes par rapport au nombre d'adresses possibles) est bien inférieure en IPv6. Sur un /26 IPv4, on a 62 adresses théoriquement possibles et la plupart correspondront sans doute effectivement à une machine. Balayer ce /26 est à la fois très rapide et très avantageux. En IPv6, un simple /64 permet d'avoir plus de $10\hat{\{\}}19$ adresses (c'est bien la principale motivation pour déployer IPv6 : avoir davantage d'adresses). Même si on a bien plus de machines, la densité sera infime. Si les adresses étaient attribuées au hasard et que le balayage se fasse également au hasard, les chances de tomber sur une machine réelle sont quasi-nulles. Heureusement pour l'attaquant, les adresses réelles ne sont pas attribuées au hasard, et le balayage n'est pas forcément fait au hasard. Ce nouveau RFC, deux fois plus long que l'ancien (le RFC 5157) fait le point sur ce que les défenseurs et les attaquants doivent savoir sur le balayage en IPv6.

Avant de tester, installons l'outil de balayage scan6, distribué avec le SI6 toolkit <http://www.si6networks.com/tools/ipv6toolkit/> (et déjà utilisé dans mon article sur les attaques en IPv6 <https://www.bortzmeyer.org/hacking-ipv6.html>) :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5157.txt

```
% wget http://www.si6networks.com/tools/ipv6toolkit/ipv6toolkit-v2.0.tar.gz
% tar xzvf ipv6toolkit-v2.0.tar.gz
% cd ipv6toolkit-v2.0
% make
```

C'est parti, en commençant par les techniques les plus classiques (section 3 de notre RFC). D'abord, lorsque le réseau visé configure les adresses IP avec le SLAAC du RFC 4862. Le SLAAC fonctionne en concaténant un préfixe appris du routeur avec un suffixe, l'IID ("*Interface IDentifier*"). Comment la machine choisit-elle son IID ? Il existe plusieurs techniques, qui ont des conséquences sur la facilité de balayage. Par exemple, une technique courante (mais de moins en moins employée) est de fabriquer l'IID à partir de l'adresse MAC de la machine, en mettant juste le bit 6 à 1 et en ajoutant au milieu 0xfffe. Ainsi, mon adresse MAC `38:59:f9:7d:b6:47` me donne actuellement une adresse IPv6 de `2003:88:6c71:c7f8:3a59:f9ff:fe7d:b647` (préfixe `2003:88:6c71:c7f8` chez Deutsche Telekom et IID `3a59:f9ff:fe7d:b647`). A priori, il y a 64 bits possibles pour l'IID. Mais ce n'est pas tout à fait vrai. D'abord, 16 sont fixes (le 0xfffe). Ensuite, les 24 premiers bits identifient le vendeur de la carte via son OUI (sauf si, comme dans l'exemple ci-dessus, on a changé son adresse MAC...) Tous les OUI possibles n'étant pas encore affectés, cela réduit le champ de recherche. Et ceux qui sont affectés le sont souvent pour du matériel qui n'est plus fabriqué, réduisant encore ce champ. Enfin, un attaquant malin observera qu'une organisation donnée a souvent un parc matériel homogène (par exemple uniquement des ordinateurs Dell). Cela permet de réduire encore l'espace de recherche (option `--tgt-vendor` de scan6). En outre, si les machines ont été achetées en même temps, il est tout à fait possible qu'elles fassent partie du même lot et que leurs adresses MAC soient consécutives. Une fois qu'on en a trouvé une, on peut supposer que les autres sont proches.

Un cas particulier est celui des techniques de virtualisation. Par exemple, VirtualBox utilise l'OUI `08:00:27` et, avec le 0xfffe du milieu, cela fait que l'espace de recherche réel n'est que de 24 bits, bien moins que les 64 théoriques. VMware est encore pire, pour ses adresses MAC automatiques : l'OUI est `00:05:69`, 16 bits sont tirés de l'adresse IPv4 de la console, 8 bits sont un condensat du nom du fichier de configuration. Il peut donc n'y avoir que 8 bits à chercher ! (Les adresses MAC manuelles de VMware ont 22 bits variables.)

Et les adresses temporaires du RFC 8981, n'avaient-elles pas été conçues justement pour éviter qu'on suive à la trace une machine ? L'IID est cette fois aléatoire, et change souvent (par exemple une fois par jour). Ces adresses temporaires sont une très bonne chose, mais elles ont quelques limites. D'abord, elles viennent **en plus** des adresses classiques, auxquelles la machine répond toujours. Les adresses temporaires étant utilisées pour les connexions sortantes, le risque de fuite de l'adresse globale permanente est plus faible mais elles n'empêchent pas le balayage, avec les techniques données plus haut.

C'est en partie pour combler les faiblesses des adresses temporaires que les adresses du RFC 7217 ont été développées. L'IID est un condensat de certaines caractéristiques de la machine et du réseau. Ces adresses sont stables tant que la machine ne change pas de réseau (ce qui permet de n'utiliser que ces adresses et d'oublier les adresses globales classiques). Du point de vue de la vie privée, elles représentent la meilleure solution, et devraient être systématiquement utilisées. Pour l'instant, comme l'indique « "*IPv6 Address Analysis - Privacy In, Transition Out*" <`http://www.internetsociety.org/blog/2013/05/ipv6-address-analysis-privacy-transition-out`> », la grande majorité des clients IPv6 utilisent les adresses temporaires du RFC 8981 (et 7 % utilisent encore les adresses MAC).

Et DHCP ? La politique d'allocation des adresses dépend du serveur DHCP. Parfois, elle mène à des adresses prévisibles, mais pas toujours. Par exemple, si on configure un serveur DHCP pour allouer des adresses dans la plage `2001:db8:a35:b24::/64`, et que le serveur DHCP décide qu'une allocation séquentielle (`2001:db8:a35:b24::1` puis `2001:db8:a35:b24::2`, puis `2001:db8:a35:b24::3`...)

est la meilleure solution, alors les adresses seront prévisibles. C'est pour cela que le RFC 5157 conseillait aux serveurs DHCP d'allouer au hasard à partir de la plage d'adresses configurée et ce conseil est répété ici.

Et les adresses attribuées manuellement, par l'administrateur système ? C'est utile pour les routeurs (qui ne peuvent pas utiliser SLAAC) et les serveurs (qui ne cherchent pas à se cacher, et pour qui il est très souhaitable d'avoir une adresse IP stable). En théorie, l'administrateur système peut choisir l'IID librement parmi 2^64 valeurs. En pratique, on observe que ces IID ne sont pas choisis au hasard mais que les ingénieurs suivent un de ces quatre schémas, utilisant :

— Un nombre de faible valeur (schéma dit "*low byte*") : `PREFIX::1`, `PREFIX::2`, etc. L'option `--tgt-low-byte` de scan6 permet de tester en premier ces valeurs. Dans le cas le plus simple, il suffit de tester les 256 valeurs correspondant aux huits bits finaux (certaines variantes de ce schéma utilisent les deux ou trois derniers octets, ce qui rend le balayage un peu plus long). Ce schéma est de loin le plus fréquent pour les serveurs et les routeurs (voir « "*IPv6 Network Reconnaissance : Theory & Practice*" <http://www.si6networks.com/presentations/lacnic19/lacsec2013-fgont-ipv6-network-reconnaissance.pdf> », cité en section 3.1.5).
— Une adresse IPv4, en profitant de ce que la représentation texte d'une adresse IPv6 peut se terminer par une adresse IPv4, comme dans `2001:db8:88:12a::192.0.2.21`. Balayer ces adresses revient à balayer l'espace IPv4 correspondant (option `--tgt-ipv4`, de scan6).
— Un numéro de port, le serveur DNS sera `PREFIX::53`, le serveur IMAP `PREFIX::993`, etc. Le cas le plus simple (numéro de port dans le dernier octet) est très rapide à balayer (option `--tgt-port` de scan6, il n'y a que quelques dizaines de ports populaires), certaines variantes sont un peu plus lentes.
— Un terme amusant en hexspeak comme `::dead:beef` ou `::a11`. Balayer ces termes se fait à partir d'un dictionnaire.

Bon, armés de ces connaissances, voyons maintenant concrètement comment balayer un réseau. D'abord, un réseau distant (section 3.2). Un balayage par la force brute est impossible en IPv6 : un seul /64 peut avoir dans les 10^20 machines, ce qui, même à un million de paquets par seconde, prendrait trois millions d'années à examiner. Un balayeur IPv6 doit donc être plus astucieux que la force brute et exploiter les propriétés des adresses IPv6, qui font que l'espace réel à explorer est plus réduit que l'espace théorique. Ainsi, en balayant dans l'ordre, de `PRÉFIXE::` puis `PRÉFIXE::1`, `PRÉFIXE::2`, etc, on tire profit des adressses de faible valeur. De même, on peut deviner le plan d'adresssage et utiliser les adresses IPv4 du réseau, les numéros des bâtiments, etc.

À noter que le balayage peut aussi être utilisé pour faire une attaque par déni de service. Certaines mises en œuvres d'IPv6 gèrent mal leur cache NDP et un grand nombre de requêtes pour des machines inexistantes peut planter certains routeurs IPv6 (RFC 6583).

Revenons au balayage fait dans le but de découvrir des adresses actives. Si on balaye le réseau local, celui où on se trouve, il y a encore d'autres possibilités, notamment les adresses "*multicast*". Par exemple, en envoyant un seul paquet à l'adresse "*multicast*" `ff02::1`, on récupère plein de machines :

```
% ping6 -I eth0 ff02::1
PING ff02::1(ff02::1) from fe80::e349:a3a5:ad58:a21 eth0: 56 data bytes
64 bytes from fe80::e349:a3a5:ad58:a21: icmp_seq=1 ttl=64 time=0.250 ms
64 bytes from fe80::21e:8cff:fe76:29b6: icmp_seq=1 ttl=64 time=1.17 ms (DUP!)
64 bytes from fe80::f6ca:e5ff:fe4d:1f41: icmp_seq=1 ttl=64 time=4.96 ms (DUP!)
64 bytes from fe80::ce0d:dad3:6bc2:6da4: icmp_seq=1 ttl=64 time=4.96 ms (DUP!)
...
```

Ça ne marche pas avec les machines Windows, qui ne répondent pas à ce ping. Mais il y a d'autres astuces comme d'envoyer à cette adresse multicast un paquet IPv6 ayant une option inconnue et dont le type commence par 10 (qui signifie « si cette option n'est pas connue, jeter le paquet et envoyer une erreur ICMP », cf. RFC 2460, section 4.2). Windows répondra alors par un message ICMP "*Parameter problem*" <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-codes-5>. Voici le résultat de scan6 vu avec tcpdump, montrant cette technique :

```
19:01:06.204431 IP6 fe80::e349:a3a5:ad58:a21 > ff02::1: DSTOPT ICMP6, echo request, seq 34994, length 64
19:01:06.205469 IP6 fe80::21e:8cff:fe76:29b6 > fe80::e349:a3a5:ad58:a21: ICMP6, parameter problem, option -
19:01:06.214792 IP6 fe80::ba27:ebff:feaa:78b9 > fe80::e349:a3a5:ad58:a21: ICMP6, parameter problem, option -
...
```

Ces techniques ne sont pas purement théoriques. Certes, un outil comme nmap sait faire du balayage en IPv4 mais pas en IPv6 à distance, mais d'autres outils existent. Ces techniques sont mises en œuvre dans l'outil scan6 du "*SI6 toolkit*" <http://www.si6networks.com/tools/ipv6toolkit/>, cité plus haut. Une fois compilé et installé, on peut utiliser cet outil pour découvrir ses voisins (sur le même réseau local) :

```
 % sudo scan6 -L -i eth0
 ...
2a01:f23:a65:6721:21e:8cff:fe76:29b6
2a01:f23:a65:6721::1
2a01:f23:a65:6721:666:6c7c:9bed:b390
2a01:f23:a65:6721:b5a5:dfd6:4e7b:2584
...
```

Ou bien tester un réseau distant :

```
% sudo scan6 -d 2001:7b2:dc0:41::/64
2001:7b2:dc0:41::250
...
```

Le dernier exemple ci-dessus est évidemment complètement irréaliste (balayage d'un /64 en force brute), même si ici on a trouvé tout de suite une machine « "*low-byte*" ». Ne soyez pas trop optimiste : scan6, livré à lui-même, est très lent et vous trouverez rarement quelque chose. Un balayage d'un site où on connait le vendeur des cartes Ethernet (et donc l'OUI) :

```
 %  sudo scan6 --tgt-ieee-oui b8:27:eb -d 2001:db8:8469:a30::/64 -i eth0 -e print-global
```

prend de très nombreuses heures et risque fortement d'être détecté. En pratique, pour utiliser scan6 avec succès, il faut récolter beaucoup d'informations sur le réseau qui vous intéresse, et guider scan6 en lui mettant beaucoup d'options. Un projet intéressant serait de tenter d'automatiser cette phase heuristique.

Une des conséquences de la difficulté de balayer en IPv6 est que la gestion de réseaux devient plus compliquée. L'administrateur réseaux ne peut pas découvrir facilement tout ce qui a été branché et fonctionne sur son réseau. Il doit donc changer ses pratiques. Une des approches, pour connaitre tout son réseau, est l'écoute passive, avec ndpmon, qui permet de se constituer automatiquement une base des machines et de leur adresse MAC.

Et l'administrateur réseaux qui veut défendre son réseau et limiter le balayage, que peut-il faire (section 3.5 du RFC) ? Il existe plusieurs techniques qui aident, en rendant les adresses moins prévisibles :

— Utiliser les adresses stables mais opaques du RFC 7217. (Au passage, je ne connais pas encore de système d'exploitation où ce soit facile ou même possible.)
— Utiliser un IDS pour détecter (ou un IPS pour bloquer) : les balayages IPv6 ne peuvent pas être faits discrètement, ils allument les systèmes d'alarme comme un arbre de Noël.
— Peut-être filtrer certains types de paquets (avec prudence : voir le RFC 4890).
— Configurer manuellement les adresses MAC (par exemple avec macchanger <https://github.com/alobbs/macchanger>). Ainsi, même si les adresses IPv6 son dérivées de l'adresse MAC, elles ne seront plus prévisibles.
— Si on utilise DHCP, configurer le serveur DHCP pour allouer dans un ordre aléatoire, et pas bêtement en partant de 1. (Je n'ai pas regardé les possibilités des serveurs DHCP actuels.)

Ces solutions ne sont pas parfaites. Consolez-vous avec l'idée que la sécurité de vos machines ne doit de toute façon pas dépendre uniquement de la résistance de votre réseau au balayage. Tôt ou tard, l'attaquant découvrira vos adresses IP, et les machines doivent donc être préparées à tout.

Le balayeur n'utilisera pas que des techniques de couche 3 comme décrit dans la précédente section, la section 3 du RFC. Il peut aussi se servir, par exemple, du DNS (section 4). Par exemple, s'il est possible de récupérer la zone DNS, le balayeur obtient beaucoup d'adresses IPv6. Mais il peut aussi tester des noms communs dans le DNS et obtenir ainsi des adresses (et, si votre algorithme d'allocation est prévisible, un petit nombre d'adresses lui permettra d'en déduire d'autres). Un tel balayage est facile à automatiser.

Mieux (du point de vue de l'attaquant), on peut énumérer les enregistrements PTR dans votre sous-arbre de ip6.arpa. (C'est automatisable, avec l'outil dnsrevenum6, qui fait partie de la boîte à outils THC <https://www.thc.org/thc-ipv6/>.)

Outre le DNS classique, mDNS (RFC 6762) peut également aider (section 5) : des requêtes envoyées à la cantonade peuvent découvrir des machines qui se signaleraient imprudemment.

L'attaquant en mission de reconnaissance peut aussi utiliser des archives publiques (section 6). Par exemple, une liste de diffusion dont les messages seraient stockées intégralement, y compris les en-têtes Received:, qui contiennent souvent des adresses IP. Difficile d'être discret sur l'Internet !

Certaines applications peuvent également aider l'attaquant à récolter des adresses IP (section 7). Par exemple, BitTorrent ne fait aucun effort pour dissimuler l'adresse du pair avec qui on échange des fichiers. Même chose avec NTP, si les machines de votre réseau local se synchronisent directement à l'extérieur, ce qui les rend visibles à Shodan <https://netpatterns.blogspot.fr/2016/01/the-rising-sophistication-of-network.html>.)

On trouve également des adresses IP dans les caches NDP (section 8) - et ces caches peuvent parfois être examinés à distance avec SNMP (section 13), dans les journaux (gérer un serveur Web populaire permet d'en récolter beaucoup, cf. section 9), via les protocoles de routage (section 10), via traceroute (section 12)... Au passage, il est recommandé de lire l'excellent article de Bellovin, S., Cheswick, B., et A. Keromytis, « "*Worm propagation strategies in an IPv6 Internet*" <https://www.cs.columbia.edu/~smb/papers/v6worms.pdf> » dans "*login :*" en 2006.

L'annexe A du RFC décrit une mise en œuvre des principales idées expliquées dans ce document. Si vous voulez tous les détails techniques sur le balayage en IPv6, c'est le texte à lire. Il faut par exemple être prudent : un balayage exhaustif en IPv6 peut nécessiter un nombre colossal de paquets et, si on les envoie trop vite, l'augmentation du trafic vous fera repérer (variante : vous surchargerez le réseau, des paquets seront perdsu et vous raterez ainsi des réponses). Cette annexe couvre également les méthodes de dissimulation, pour éviter d'être détecté (avec la plupart des IDS IPv6, il suffit d'ajouter des en-têtes d'extension au paquet). Le tout est évidemment très inspiré de l'expérience de l'un des auteurs du RFC avec l'outil scan6, déjà cité.

Je ne fais pas ici un résumé des différences avec son prédécesseur, le RFC 5157, car il y en a beaucoup trop (notre nouveau RFC est bien plus détaillé).