RFC 6104 : Rogue IPv6 Router Advertisement Problem Statement

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 février 2011

Date de publication du RFC : Février 2011

https://www.bortzmeyer.org/6104.html

Il y a une tension permanente en sécurité des réseaux, entre la sécurité et la facilité d'usage (quelqu'un a une meilleure traduction pour "*convenience*" ? « ergonomie » ?). Ainsi, le protocole IPv6 permet l'**autoconfiguration sans état** des machines, par le biais d'annonces, les **RA** ("*Router Advertisement*") diffusées par les routeurs. C'est très pratique et IPv4 n'a rien de tel. Mais qui dit que la machine qui diffuse est un vrai routeur légitime ? Rien. Les « RAcailles » (mauvaise traduction de "*rogue RA*"), ces annonces envoyées à tort, sont très fréquents dans la réalité. Avant que des solutions soient proposées et discutées, ce court RFC spécifie précisément le problème.

Le mécanisme des **RA** ("*Router Advertisement*") est décrit dans le RFC 4861 [1]. Le principe est que le ou les routeurs officiels du réseau local diffusent des informations telles que le préfixe à utiliser sur ce lien (ce qui permet l'autoconfiguration du RFC 4862) ou l'adresse IP du routeur à utiliser par défaut. Les « RAcailles » sont des annonces RA envoyées par un engin qui n'est **pas** routeur officiel. Cela peut être dû à une attaque délibérée, ou tout simplement à une erreur de configuration, par exemple une machine configurée pour monter un "*hotspot WiFi*" avec réseau ad hoc et qui diffuse donc des RA, se croyant le routeur officiel. Ce phénomène est fréquent sur les réseaux sans-fil partagés (on le voit à chaque réunion IETF, dont les participants sont pourtant censés être des experts, et il faut des annonces répétées au micro pour que les machines émettant des RAcailles se taisent ; un exemple à la réunion d'Hiroshima est décrit dans ce message <http://www.ietf.org/mail-archive/web/76attendees/current/msg00411.html>). Mais on le voit aussi dans des réseaux filaires, par exemple les résidences d'étudiants.

Sur un réseau sans-fil partagé, il n'y a guère de moyen de repérer le coupable, dont on ne connait que l'adresse MAC. Encore aujourd'hui, la solution la plus courante (décrite en section 3.7) est de filtrer les RA, sur la base de l'adresse attendue (liste blanche), ou sur celle des adresses repérées comme erronées (liste noire). Mais cela diminue évidemment sérieusement l'intérêt de l'autoconfiguration (toujours cette tension entre sécurité et facilité d'usage). Sur Linux, cela serait quelque chose comme :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4861.txt

```
% ip6tables -A INPUT -m mac --mac-source 00:1b:77:bc:a4:e6 -j DROP
```

(Exercice pour le lecteur : cette commande bloque absolument tout le trafic en provenance de `00:1b:77:bc:a4:e6`. Comment ne bloquer que les RA ? Benjamin Bachelart `<http://twitter.com/BashyB>` suggère `ip6tables -A INPUT -m mac --mac-source 00:1b:77:bc:a4:e6 -p icmpv6 --icmpv6-type router-advertisement -j DROP` qui semble correct mais je ne l'ai pas testé.)

Quel problème posent ces RAcailles ? La section 1 les résume : si les machines présentes les croient, et utilisent les informations qu'ils contiennent, elles peuvent se trouver partiellement ou totalement coupées du réseau. Par exemple, si elles changent l'adresse IP du routeur par défaut, pour celle indiquée dans le RAcaille, tous les paquets qui ne sont pas envoyés au réseau local sont transmis à un routeur voyou, qui peut les transmettre... ou pas.

Un point important de ce RFC est qu'il se focalise surtout sur les RAcailles accidentels, ceux provoqués par une erreur de configuration, qui sont de loin les plus fréquents aujourd'hui (les méchants sont plus habiles mais les maladroits sont beaucoup plus nombreux...). En outre, si les RAcailles sont émis par un attaquant conscient de ce qu'il fait, protéger contre eux serait très insuffisant : s'il peut émettre des faux RA, il peut aussi bien interférer avec le protocole ND.

Donc, première question (section 2), quel phénomène peut mener à l'envoi de RAcailles par une machine ? Il y en a trois :

— La traditionnelle erreur commise par l'administrateur réseaux (qui tape les mauvaises commandes sur le routeur, ou bien sélectionne la mauvaise option dans un cliquodrome),
— L'erreur commise par un utilisateur qui n'est pas l'administrateur réseaux (comme dans l'exemple du réseau ad hoc cité plus haut, par exemple suite à l'activation de ICS, qui transformera également la machine en un serveur DHCP IPv4, ou bien dans le cas d'un portable configuré pour un réseau isolé et connecté ensuite à un autre réseau),
— L'attaque délibérée par le méchant, prélude à une DoS ou bien à une attaque de l'intermédiaire. Notons bien que ce genre d'attaques est rarement signalé (alors que les récits de RAcailles accidentels abondent). Le RFC relègue donc ce problème pour de futures études et se concentre sur le problème le plus fréquent.

Et quelles sont les solutions possibles ? La plupart seront détaillées dans d'autres RFC comme le RFC 6105 mais la section 3 du nôtre donne quelques pistes. La plus évidente est de laisser tomber l'autoconfiguration et de revenir aux solutions à configuration manuelle. Sûr mais pas forcément enthousiasmant pour l'administrateur du réseau. D'autant plus que cela n'éliminera pas les erreurs... Pour être cohérente, une telle solution implique, outre la configuration manuelle de l'adresse IP et du routeur, d'ignorer les éventuels RA (avec Linux, il faut mettre la variable sysctl `net.ipv6.conf.all.accept_ra` à 0).

Autre solution, profiter de ce que les RA soient diffusés à tous pour les observer et détecter ainsi rapidement les RAcailles. C'est ce que fait NDPMon, par exemple. Fait dans le commutateur, comme le propose le RFC 6105, cela pourrait permettre le blocage des RAcailles en temps réel. Le filtrage par les commutateurs peut aussi être configuré manuellement, par le biais d'ACL. Après tout, l'administrateur réseaux sait sur quel(s) port(s) du commutateur se trouvent les routeurs légitimes et il peut donc bloquer les RA sur les autres (quelqu'un a des exemples concrets avec des commutateurs réels ? Je n'ai jamais essayé.) Notez que cela marche seulement si toutes les communications passent par le commutateur, ce qui est typiquement le cas aujourd'hui des réseaux filaires. Avec un réseau sans-fil ad hoc, il faudra trouver autre chose.

De plus haute technologie, il y a aussi l'inévitable solution utilisant la cryptographie, SEND (RFC 3971), où les annonces des routeurs légitimes sont signées. SEND est très peu déployé en pratique, sans doute en raison de sa complexité, et du fait qu'il faille configurer chaque machine avec le certificat de la

CA. Du point de vue sécurité, SEND est la solution idéale (cf. section 6), mais il est probablement peu adapté aux environnements assez ouverts, comme un campus ou bien un café avec WiFi, où chacun administre sa machine. Le principal avantage de SEND est qu'il fonctionne aussi en cas d'attaque délibérée et intelligente et il convient donc bien aux environnements qui exigent un haut niveau de sécurité.

D'autres solutions techniques ? Mais oui, l'imagination des experts étant inépuisable. Par exemple, puisque le RFC 4191 ajoute une option Préférence aux RA, pourquoi ne pas mettre cette option à la valeur maximale pour les routeurs légitimes, en espérant que les accidentels la laissent à une valeur plus faible ? Mais le RFC n'indique pas si cette option est souvent reconnue par les clients. Autre idée, se servir des protections de la couche 2 (par exemple 802.1X) pour limiter l'accès au réseau. Le problème est qu'authentification n'est pas autorisation. Une machine peut avoir légitimement un accès, sans être pour autant autorisée à envoyer des RA. Il reste le filtrage au niveau de la machine, déjà mentionné. Son principal inconvénient est qu'il faut manuellement maintenir à jour les ACL, ce qui supprime une bonne partie de l'intérêt de l'autoconfiguration.

Et la contre-attaque ? Si une machine envoie des RAcailles sans y être autorisée, on n'a pas de raison d'avoir des scrupules à la faire taire. Une autre solution aux RA non autorisés est donc de surveiller le trafic (comme avec NDPMon) et de générer des faux RA, ayant la même adresse source que les RAcailles, avec une durée de vie de zéro. Cet empoisonnement devrait donc masquer le RAcaille par le RA à durée de vie nulle (et qui ne sera donc pas utilisé par les machines). Cette technique est mise en œuvre dans la suite logicielle KAME et a déjà été déployée lors de réunions IETF. On peut télécharger le source en <http://ramond.sourceforge.net>. À noter qu'un tel outil pourrait aussi parfaitement être utilisé pour lancer une belle attaque par déni de service (cf. section 7).

Enfin, dernière solution proposée par notre RFC, configurer les machines pour n'utiliser que DHCPv6 et renoncer à l'autoconfiguration sans état. DHCP peut allouer les adresses. Une nouvelle option, en cours de normpalisation, lui permettra d'indiquer également le routeur à utiliser. DHCP présente de nombreux avantages pour l'administrateur réseaux, notamment d'un meilleur contrôle de la configuration des machines. Mais on n'a fait que déplacer le problème. Comme le montre l'expérience d'IPv4, les serveurs DHCP non autorisés sont un problème, autant que les RA non autorisés... Le seul avantage serait que le problème est mieux connu avec DHCP (voir aussi la section 5.2).

Les différentes solutions, avec leur domaine d'application, sont résumées dans un tableau en section 4. La section 5 discute ensuite d'autres problèmes liés à celui des RAcailles. Par exemple, un RA n'est pas forcément envoyé par diffusion, il peut être transmis directement à une machine et, dans ce cas, a des chances de ne pas être détecté par d'éventuels outils de surveillance. Les RAcailles accidentels seront sans doute en "*multicast*" mais un attaquant, lui, choisira peut-être donc l'"*unicast*".

La section 5.2 compare les menaces de RA par rapport à celles de DHCP. De même que les RA peuvent être protégés avec SEND, DHCP peut être protégé par des options d'authentification (RFC 3315, section 21). Mais ne nous voilons pas la face : personne n'utilise ces options de sécurité (elles ont même été larguées par le RFC 8415). Elles sont bien trop contraignantes, alors qu'on choisit justement RA ou DHCP pour se simplifier la vie.

L'importance d'une surveillance du réseau, avec un logiciel comme NDPMon, est rappelée dans la section 5.4. Si cette recommandation est appliquée, l'administrateur réseaux saura au moins ce qui se passe. Actuellement, il est probable que la plupart de ces administrateurs ne savent même pas si le problème existe dans leur réseau.

Toujours dans la série « guérir si on n'a pas pu prévenir », la section 5.5 est consacrée aux conséquences. Si les RAcailles existent et que certaines machines en ont tenu compte, que faire ? Le comportement de la

machine touchée est très imprévisible (plusieurs adresses, plusieurs routes par défaut, et, avec SHIM6, la machine peut même croire qu'elle est réellement connectée à plusieurs réseaux). Aujourd'hui, il n'y a pas de moyen simple, une fois les RAcailles supprimés, de ramener ces machines à la normale, on risque d'attendre jusqu'à deux heures (RFC 4862, section 5.5.3) avant que la machine n'oublie le RAcaille.

Quelques documents à lire sur ce sujet et des logiciels à essayer :

— Le site web de NDPMon <http://ndpmon.sourceforge.net/>, excellent outil de détection des annonces v6, notamment des RA.
— Le dépôt de rafixd <http://www.kame.net/dev/cvsweb2.cgi/kame/kame/kame/rafixd/>, un outil d'empoisonnement actif des RAcailles.
— Le site Web <http://ramond.sourceforge.net/> de Ramon, outil de détection utilisé par exemple à la réunion RIPE 63 <http://ripe63.ripe.net>, avec succès.
— Une discussion des différentes solutions <http://ipv6samurais.com/ipv6samurais/demystified/rogue-RA.html> (en anglais).
— Très détaillé, très concret : « "*IPv6 security*" <http://www.katta.name/rd/Amit/IPV6/1587055945_20081205.pdf> », chez Cisco press, couvre largement les RAcailles..
— En français, le problème est décrit en détail dans « La sécurité dans une transition vers IPv6 <http://www.urec.cnrs.fr/IMG/pdf/secu.articles.Archi.Securite.IPv6.pdf> » de Gaël Beauquin.