RFC 7113 : Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 février 2014

Date de publication du RFC : Février 2014

https://www.bortzmeyer.org/7113.html

La sécurité, ce n'est pas facile et il faut faire attention à beaucoup de choses. Ce RFC est entièrement consacré à un oubli de pas mal de programmeurs lorsqu'ils mettent en œuvre une technique de sécurité pour IPv6, le "*RA guard*" : il est en effet courant d'oublier qu'un paquet IPv6 peut avoir plusieurs entêtes, chaînés, et que chercher un motif dans le paquet, à un nombre d'octets fixes depuis le début du paquet, est une **mauvaise** stratégie. (Le RFC décrit également un second problème, lié à la fragmentation.)

Le "*RA guard*" est une protection contre les RAcailles, ces annonces d'un routeur IPv6 (RA = "*Router Advertisement*") qui ne sont pas émises par un routeur légitime. Les RAcailles sont un phénomène courant, en général provoqué par une erreur de configuration mais elles peuvent aussi être utilisées pour une attaque. Dans ce cas, le méchant s'annonce lui-même comme le routeur du réseau, reçoit donc le trafic et peut alors le jeter, l'espionner, le détourner, ou autres méchancetés. (Le RFC 6104 [1] pose le problème.)

Il n'est pas facile de lutter contre ce phénomène, tout en gardant la simplicité de l'auto-configuration sur le réseau local (si on renonce à la simplicité, il existe une solution, le RFC 3971). Le problème est d'ailleurs le même avec DHCP depuis longtemps (le méchant peut générer de fausses réponses DHCP). Comme le problème est ancien, il y a longtemps que des solutions sont déployées en IPv4. Typiquement, le commutateur examine les paquets (oui, de la DPI), regarde si c'est une réponse DHCP et, si c'est le cas et que le port du commutateur n'est pas censé abriter un serveur DHCP, on jette le paquet. (Il existe plusieurs méthodes pour déterminer si un serveur DHCP légitime est sur ce port, de la confiugration manuelle, jusqu'à l'écoute préalable, pour apprendre automatiquement.)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6104.txt

Cette méthode peut s'étendre facilement aux annonces de routeur d'IPv6, les RA : le RFC 6105 la décrit en détail, sous le nom de "*RA guard*". Mais les premières mises en œuvre ont souvent souffert d'un oubli de la part des programmeurs.

En effet, pour examiner la charge utile du paquet, il faut sauter l'en-tête IP. En IPv4, il est de taille variable et il faut donc lire le champ Longueur, sauter le bon nombre d'octets et on regarde alors la charge utile. En IPv6, l'en-tête étant de taille fixe, cela semble plus facile et c'est ce qui a trompé pas mal d'implémenteurs. Car plusieurs ont fait la même erreur : oublier les en-têtes d'extensions. IPv6 permet en effet d'ajouter plusieurs en-têtes entre l'en-tête principal et la charge utile. Pour contourner le "*RA Guard*", tout ce qu'avait à faire le méchant était d'ajouter un de ces en-têtes, décalant ainsi la charge utile et empêchant sa reconnaissance.

Le problème est bien connu, j'avais même fait un article là-dessus il y a quatre ans <https://www.bortzmeyer.org/analyse-pcap-ipv6.html> mais il y avait de nombreuses autres analyses avant. Mais les programmeurs d'engins réseaux comme les commutateurs ne prennent pas toujours le temps de se renseigner : le marketing dit « nos clients réclament du "*RA Guard*", il faut qu'on puisse l'écrire sur la feuille de spécifications, et ça doit être livré avant-hier », le programmeur programme et cela semble marcher avec des paquets normaux, personne ne fait de tests avec des paquets inhabituels, précisement ceux qu'utilisera l'attaquant.

Ce RFC 7113 rappelle donc qu'un commutateur qui met en œuvre "*RA Guard*" **doit** analyser tous les en-têtes, pas seulement sauter le premier. Sinon, cette protection ne vaut pas grand'chose. À noter que le RFC se focalise sur le cas des "*RA Guard*" mais, en fait, des tas de systèmes de sécurité IPv6, des IDS aux limiteurs de trafic ont exactement la même vulnérabilité : en ajoutant un simple en-tête Destination <https://www.bortzmeyer.org/destination-options-ipv6.html>, vous passez outre la plupart des contrôles. (Des logiciels très utiles comme NDPmon <http://ndpmon.sourceforge.net>, ramond <http://ramond.sourceforge.net/> ou rafixd <http://www.kame.net/dev/cvsweb2.cgi/kame/kame/kame/rafixd/> sont apparemment vulnérables.)

Cette première attaque est exposée en section 2.1. On peut noter qu'il n'existe actuellement aucun usage légitime des en-têtes d'extension dans une annonce RA. Mais l'attaque marche car les mises en œuvre de "*RA guard*" n'en tiennent pas compte.

Une deuxième attaque, fondée sur la fragmentation, est décrite en section 2.2. L'idée est de fragmenter l'annonce RA en deux paquets IP, les bits qui permettent d'identifier le paquet comme une annonce RA étant dans le deuxième fragment. Si le commutateur ne fait pas de réassemblage des paquets, il ne pourra pas détecter la RAcaille et donc pas le bloquer. **Toutes** les implémentations testées par l'auteur du RFC sont vulnérables.

Bon, c'est bien joli de décrire les attaques mais comment réparer ? La section 3 décrit les obligations d'un système de "*RA GUard*" sérieux :

— Si le paquet a une adresse source qui n'est pas locale au lien (les adresses locales au lien sont dans le préfixe `fe80::/10`), le laisser passer. Un RA doit toujours avoir une adresse source locale au lien (puisqu'il est forcément générer par un routeur connecté au lien) donc inutile d'examiner ce paquet, si c'est une RAcaille, il sera refusé de toute façon (RFC 4861, section 6.1.2).
— Si le nombre de sauts maximum (champ "*Hop Limit*") n'est pas à 255, laisser passer le paquet. Les RA utilisent ce champ comme protection contre une injection de RA par une machine extérieure au réseau local. S'il est à moins de 255, ce qui indique qu'il a traversé au moins un routeur, il sera rejeté par les machines terminales donc, comme dans le cas précédent, pas la peine de se fatiguer à l'examiner.
— Si le paquet est un fragment et n'est pas le premier fragment d'une série, le laisser passer (en raison de la règle suivante).

— Le système "*RA Guard*" **doit** analyser tous les en-têtes, pas juste le premier. Cette tâche est complexe <https://www.bortzmeyer.org/analyse-pcap-ipv6.html> mais elle a été rendue plus facile par le RFC 6564. En outre, le RFC 7112 impose que la totalité des en-têtes soient dans le premier fragment, ce qui fait qu'il n'y a pas besoin de réassembler le paquet d'origine pour effectuer cette analyse (on a désormais le droit de jeter les paquets où le premier fragment ne contient pas toute la chaîne d'en-têtes). Par contre, le RFC interdit à "*RA Guard*" de se limiter aux N premiers octets : il faut analyser tout le paquet (les en-têtes d'extension peuvent être arbitrairement grands).
— Ensuite, on peut appliquer l'algorithme "*RA Guard*" classique : si le paquet est un RA et que ce n'est pas autorisé sur ce port, le jeter. Autrement, le laisser passer.

À noter que les paquets ESP (RFC 4303) seront toujours acceptés (puisque le RA est après l'en-tête ESP, qui est considéré comme l'en-tête final d'IP). C'est logique : c'est IPsec qui, dans ce cas, assure la sécurité des paquets, "*RA Guard*" s'efface alors devant IPsec.

Un dernier détail lié à la fragmentation : IPv6 permettait à l'origine des fragments non-disjoints. Ces fragments arriveraient à passer les règles ci-dessus. Mais ils ont été interdits par le RFC 5722 et l'étude « "*IPv6 NIDS evasion and improvements in IPv6 fragmentation/reassembly*" <http://blog.si6networks.com/2012/02/ipv6-nids-evasion-and-improvements-in.html> » montrait, qu'en 2012, la plupart des mises en œuvre d'IPv6 faisaient respecter cette interdiction. Plus radical, le RFC 6980 interdit complètement la fragmentation pour les RA mais il ne sera pas déployé immédiatement dans toutes les machines terminales, donc "*RA Guard*" ne peut pas encore compter dessus.

Un document équivalent pour DHCP sera publié comme RFC, pour l'instant, c'est l'"*Internet-Draft*" draft-ietf-opsec-dhcpv6-shield, qui utilise les mêmes techniques que notre RFC 7113.

Si vous voulez tenter les attaques décrites ici, vous avez la boîte à outils de SI6 <http://www.si6networks.com/tools/ipv6toolkit> ou bien l'ensemble logiciel THC <http://www.thc.org/thc-ipv6/>.