RFC 6946 : Processing of IPv6 "atomic" fragments

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 mai 2013. Dernière mise à jour le 20 juin 2013

Date de publication du RFC : Mai 2013

https://www.bortzmeyer.org/6946.html

Une particularité amusante de la gestion de la fragmentation dans IPv6 est qu'un paquet peut très bien avoir un en-tête de fragmentation sans être fragmenté du tout. Cela se nomme un « fragment atomique » (le terme est très malheureux car il n'a **rien** à voir avec les « datagrammes atomiques » du RFC 6864 [1]). Ils n'ont aucun intérêt pratique mais sont engendrés par certains systèmes lorsqu'ils reçoivent un message ICMP "*Packet too big*" indiquant une MTU inférieure à 1280 octets (la taille miminale pour une MTU IPv6). Le problème est que ces fragments atomiques sont ensuite traités par bien des systèmes comme des vrais fragments, permettant ainsi certaines attaques subtiles. Ce nouveau RFC demande donc que les fragments atomiques soient gérés à part, compte tenu de leurs particularités. (Depuis, le RFC 8021 a carrément rendu ces fragments atomiques obsolètes.)

Si vous voulez réviser les règles de la fragmentation dans IPv6, c'est dans le RFC 2460 (depuis remplacé par le RFC 8200, qui a changé ces règles). Contrairement à IPv4, seules les machines de départ (pas les routeurs intermédiaires) peuvent fragmenter. Si une machine de départ doit envoyer un paquet qui est trop grand pour la MTU, elle le fragmente en plusieurs morceaux, chaque morceau indiquant son décalage par rapport au début du paquet original. Même si ce n'est pas explicite dans ce RFC, le recouvrement des fragments était autorisé, ce qui complique sérieusement le réassemblage et pouvait permettre d'échapper à certains IDS. Il n'est donc pas étonnant que le RFC 5722 ait finalement interdit ces fragments recouvrants. Bon, et les fragments atomiques, ils sont fabriqués pourquoi ? La section 5 du RFC 2460 ne laissait pas le choix aux mises en œuvre d'IPv6 : normalement, elles devaient fabriquer un fragment atomique (un paquet qui a l'en-tête de fragmentation, sans pour autant être fragmenté) si la taille indiquée par le paquet ICMP est inférieure aux 1280 octets minimaux d'IPv6 (c'est pour aider certains mécanismes de traduction IPv4¡-¿IPv6). Dans un fragment atomique, le décalage ("*fragment offset*") sera de zéro (puisque ce fragment est le premier) et le bit M sera à Zéro (puisque ce fragment est le dernier du datagramme). Comme ces paquets ICMP "*Packet too big*" ne sont pas authentifiés, même quand c'est possible (section 5.3 du RFC 4443 et RFC 5927), un attaquant peut donc faire fabriquer des

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6864.txt

fragments atomiques assez facilement. Notez que le RFC 8200 a changé cette règle et donc interdit les fragments atomiques.

Ensuite, comme bien des systèmes traitent ces fragments atomiques en même temps que les fragments normaux, ils peuvent, par exemple, être fusionnés avec d'autres fragments prétendant venir de la même source.

Les problèmes de sécurité liés aux fragments sont bien connus : voir notamment le RFC 5722 mais aussi le RFC 6274 qui décrit ces attaques pour IPv4 (beaucoup sont communes aux deux versions). Ce problème est encore aggravé par le fait que certains systèmes génèrent des identificateurs de fragment trop prévisibles, permettant à l'attaquant de savoir où il devra frapper.

La section 4 décrit ensuite les nouvelles règles, visant à résoudre ce problème. En deux mots, en recevant un fragment atomique (que l'on reconnait à la combinaison décalage=0 et bit M =0), on doit le traiter différemment des autres fragments et il ne peut être réassemblé qu'à partir du fragment unique.

L'annexe A contient un tableau listant un certain nombre de systèmes d'exploitation actuels en indiquant s'ils génèrent des fragments atomiques, et s'ils mettent en œuvre ce RFC (les traitent différemment des autres fragments). La plupart des systèmes génèrent ces fragments atomiques lorsqu'ils reçoivent le faux paquet ICMP "*Too big*" (une exception est NetBSD). Et plusieurs d'entre eux (Linux récent, NetBSD, OpenBSD) suivent déjà la recommandation de ce RFC et ne sont donc normalement pas vulnérables aux attaques décrites ici.

Si vous voulez tester vous-même, l'outil frag6 dans la boîte à outils SI6 <http://www.si6networks.com/tools/ipv6toolkit/> permet de le faire facilement. D'abord, on envoie un fragment incomplet qui est le premier du datagramme :

```
# frag6 -v -i em0 --frag-type first --frag-id 1234 -d $MACHINE_UNDER_TEST
```

Dans une autre fenêtre, et avant l'expiration du délai de garde pour le réassemblage (60 secondes par défaut), on envoie un fragment atomique de même identificateur (1234 ici) :

```
# frag6 -v -i em0 --frag-type atomic --frag-id 1234 -d $MACHINE_UNDER_TEST
```

Si les fragments atomiques sont réellement traités à part (ce que notre RFC exige), ce qui est le cas sur un FreeBSD ¿= 9 ou un Linux ¿=¿ 3), le fragment atomique est « réassemblé » avec lui-même et on obtient une réponse :

```
ICMPv6 echo Reply from $MACHINE_UNDER_TEST (RTT: 5 seconds)
```

Si on n'a pas cette réponse, c'est que le système en face traite incorrectement les fragments atomiques avec les autres. Dans les deux cas, système correct ou non, le premier fragment ne sera jamais réassemblé et, au bout des 60 secondes, on aura :

```
Response from $MACHINE_UNDER_TEST: ICMPv6 Time Exceeded error message (Reassembly timeout: 60 seconds)
```

Et rappelez-vous que, normalement, les mises en œuvre d'IPv6 ne doivent plus générer de fragments atomiques depuis le RFC 8021.