RFC 8165 : Design considerations for Metadata Insertion

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 mai 2017

Date de publication du RFC : Mai 2017

http://www.bortzmeyer.org/8165.html

Ce court RFC déconseille l'insertion de métadonnées dans les paquets IP, si cette insertion est faite en route, par des intermédiaires. Pourquoi ? (Essentiellement pour des raisons de vie privée.)

Le problème de la surveillance de masse que pratiquent la plupart des États (en tout cas ceux qui en ont les moyens financiers) est maintenant bien documenté (par exemple dans les RFC 7258 [1] et RFC 7624). Une solution fréquente pour limiter cette surveillance, la rendre plus coûteuse et moins efficace est de chiffrer ses communications. Dans l'éternelle lutte de l'épée et de la cuirasse, les surveillants réagissent au chiffrement en utilisant davantage les métadonnées, en général non protégées par le chiffrement. Qui met des métadonnées dans les paquets, affaiblissant ainsi l'effet du chiffrement ?

Certaines métadonnées sont absolument indispensables au fonctionnement de l'Internet. Par exemple, l'adresse IP de destination dans un paquet doit être en clair car tous les routeurs situés sur le trajet doivent la voir, pour prendre leurs décisions. Certaines métadonnées sont inutiles au fonctionement de l'Internet, mais difficiles à dissimuler, la taille des paquets, par exemple. (C'est également un exemple d'une métadonnée implicite : contrairement à l'adresse IP, elle n'apparait pas explicitement dans le paquet.) Normalement, pour gêner la surveillance, il faut envoyer le moins de métadonnées possible.

L'Internet est souvent décrit comme reposant sur une liaison de bout en bout, où seules les deux machines situées aux extrémités de la communication ont accès à tout le contenu de la communication. Mais, en pratique, il existe souvent des équipements intermédiaires qui ont accès à des informations poour faire leur travail. Si ces "*middleboxes*" ont la mauvaise idée de mettre ces informations dans les métadonnées d'un paquet, elles affaiblissent la confidentialité des échanges. Imaginons par exemple (ce n'est pas forcément fait aujourd'hui : le RFC met en garde contre une mauvaise idée, pas toujours contre des pratiques existantes, voir à ce sujet l'examen par la direction Sécurité <https://datatracker.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7258.txt

ietf.org/doc/review-hardie-privsec-metadata-insertion-05-secdir-lc-nir-2017-02-07/ >), imaginons par exemple un VPN qui déciderait d'indiquer l'adresse IP originale dans la communication...Notre RFC mentionne deux exemples qui sont décrits dans des RFC : le RFC 7239 qui décrit l'en-tête HTTP `Forwarded:` qu'un relais HTTP peut mettre pour indiquer l'adresse IP d'origine du client, et bien sûr le RFC 7871, où un résolveur DNS transmet aux serveurs faisant autorité l'adresse IP du client original.

La section 4 du RFC est la recommandation concrète : les métadonnées ne doivent pas être mises par les intermédiaires. Si ces informations peuvent être utiles aux destinataires, c'est uniquement au client d'origine de les mettre. Autrement, on trahit l'intimité du client.

Le RFC 7871, par exemple, aurait dû spécifier un mécanisme où l'adresse IP est mise par le client DNS de départ, celui qui tourne sur la machine de l'utilisateur. Cela permettrait un meilleur contrôle de sa vie privée par l'utilisateur.

Et si cette machine ne connait pas sa propre adresse IP publique, par exemple parce qu'elle est coincée derrière un NAT ? Dans ce cas, notre RFC 8165 dit qu'il faut utiliser une technique comme STUN (RFC 5389) pour l'apprendre.

Bon, la section 4, c'était très joli, c'était les bons conseils. Mais la cruelle réalité se met parfois sur leur chemin. La section 5 de notre RFC est le « "*reality check*" », les problèmes concrets qui peuvent empêcher de réaliser les beaux objectifs précédents.

D'abord, il y a le désir d'aller vite. Prenons l'exemple du relais HTTP qui ajoute un en-tête `Forwarded:` (RFC 7239), ce qui permet des choses positives (adapter le contenu de la page Web servie au client) et négatives (fliquer les clients). Certes, le client HTTP d'origine aurait pu le faire lui-même, mais, s'il est derrière un routeur NAT, il faut utiliser STUN. Même si tous les clients HTTP décidaient de la faire, cela ne serait pas instantané, et la longue traine du déploiement des navigateurs Web ferait qu'un certain nombre de clients n'aurait pas cette fonction. Alors que les relais sont moins nombreux et plus susceptibles d'être rapidement mis à jour.

En parlant d'adaptation du contenu au client, il faut noter que c'est une des principales motivations à l'ajout de tas de métadonnées. Or, comme dans l'exemple ci-dessus, si on demande au client de mettre les métadonnées lui-même, beaucoup ne le feront pas. De mon point de vue, ils ont bien raison, et le RFC note qu'une des motivations pour la consigne « ne pas ajouter de métadonnées en route » est justement de rendre le contrôle à l'utilisateur final : il pourra choisir entre envoyer des métadonnées lui permettant d'avoir un contenu bien adapté, et ne pas en envoyer pour préserver sa vie privée. Mais ce choix peut rentrer en conflit avec ds gens puissants, qui exigent, par exemple dans la loi, que le réseau trahisse ses utilisateurs, en ajoutant des informations qu'eux-mêmes ne voulaient pas mettre.

Enfin, il y a l'éternel problème de la latence <http://www.bortzmeyer.org/latence.html>. L'utilisation de STUN va certainement ralentir le client.

Un dernier point (section 7 du RFC) : si on passe par Internet pour contacter des services d'urgence (pompiers, par exemple, ou autre PSAP), ils ont évidemment besoin du maximum d'informations, et, dans ce cas, c'est peut-être une exception légitime à la règle de ce RFC.