RFC 6108 : Comcast's Web Notification System Design

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/6108.html

Soit un FAI avec beaucoup de clients résidentiels. La plupart utilisent une machine sous Windows. Beaucoup de ces machines sont infectées par un des innombrables virus ou vers qui existent pour cette plate-forme. Devenues des zombies, ces machines participent désormais à diverses opérations illégales comme les dDoS ou l'envoi de spam. Si le FAI détecte, par le comportement de ces machines, ou par des rapports qui lui sont envoyés, qu'un de ses clients a été zombifié, comment le prévenir ? Vues les marges financières existantes dans cette industrie, pas question d'envoyer un technicien compétent chez le client pour lui expliquer. Il n'est même pas possible de l'appeler et de passer une heure à répondre à ses questions idiotes, cela mangerait tout le profit retiré de l'abonnement de ce client. Ce RFC décrit le système qu'utilisait Comcast pour prévenir ses abonnés.

Il existe bien sûr d'autres méthodes, comme de couper purement et simplement la connexion Internet dudit client. Cette méthode est régulièrement réclamée par certains éradicateurs. Outre qu'elle n'est pas dans l'intérêt financier du FAI (le client arrêterait de payer s'il était coupé), elle n'est pas envisageable dans un État de droit (en France, le Conseil Constitutionnel avait cassé une version de la loi Hadopi car elle prévoyait une coupure de l'accès Internet sans jugement). Même dans un pays de cow-boys, cette mesure est largement jugée comme trop radicale (voir la description de la section 12, plus loin). En outre, l'utilisateur risque de ne pas la comprendre, de croire à une panne et d'embêter le support utilisateur. Comcast a donc choisi une autre approche, celle de la notification à l'utilisateur, dont on espère qu'il réagira et prendra des mesures pour éliminer le "*malware*" de sa machine.

Concevoir un tel système de notification n'est pas évident. Il n'existe (heureusement) pas de moyen technique standard permettant à un tiers, le FAI, de faire soudain apparaître des messages sur l'écran de l'utilisateur (un tel mécanisme serait vite détourné par des plaisantins). Envoyer un courrier est simple et automatisable mais rien ne garantit que l'utilisateur les lit (la plupart ne le liraient pas ou bien, peut-être à raison, considéreraient-ils qu'il s'agit de hameçonnage).

L'approche de Comcast est donc de profiter du fait que la plupart des utilisateurs passent beaucoup de temps à regarder des pages Web. Il « suffit » de détourner les pages HTML, d'y insérer le message

et bingo ! Le message apparaîtra à l'écran. Voici une copie d'écran de <http://www.example.com/>, avec l'exemple de modification que donne le RFC en section 8. Cette modification consiste simplement en l'ajout d'un peu de JavaScript :

(On note que la possibilité d'accuser réception, mentionnée plus loin dans le RFC, est absente de cet exemple. Notez aussi le commentaire dans le code CSS qui insiste sur l'importance de ne pas hériter des styles qui seraient déjà présents dans la page.)

Bref, ce mécanisme est une attaque de l'homme du milieu, réalisée pour la bonne cause. Les auteurs insistent beaucoup sur le fait que leur solution repose sur des normes ouvertes et n'utilise pas de DPI mais cela me semble tout à fait secondaire. Le point important est que le FAI joue un rôle plus actif dans la chasse aux zombies et, pour cela, s'insère dans les communications de son client. Ce n'est pas forcément une mauvaise idée mais c'est une modification sérieuse du modèle traditionnel (où le client était seul maître à bord, même si, la plupart du temps, il ne se rendait pas du tout compte de la responsabilité qu'il avait.) Pour l'instant, il n'existe pas de travail organisé à l'IETF sur ce problème des machines zombies et chacun en est donc conduit à expérimenter seul (cf. section 13 du RFC).

Les détails pratiques suivent. La solution décrite très en profondeur est spécifique à DOCSIS mais pourrait être adaptée assez facilement à d'autres types d'accès. Les techniques utilisées sont le protocole ICAP (RFC 3507 [1]) et HTTP, les logiciels étant Squid et Tomcat.

Voyons d'abord le cahier des charges exact. Il figure en section 3. Je ne vais pas répéter tous ses points ici, seulement les plus importants. D'abord, les principes généraux :

— N'est utilisé que pour les notifications de problèmes sérieux (comme le recrutement dans un "*botnet*"), pas pour des publicités.
— Tourne sur le port 80, celui de HTTP, car à peu près tous les utilisateurs s'en servent. Ne doit pas gêner les autres protocoles qui utiliseraient ce port. D'une manière générale, ne doit pas perturber ou casser des applications.
— N'utilise pas de "*ReSeTs*" TCP comme le faisait Comcast <http://www.eff.org/deeplinks/2007/10/eff-tests-agree-ap-comcast-forging-packets-to-interfere> pour démolir les connexions BitTorrent de ses utilisateurs (sauf erreur, c'est la première fois que Comcast reconnait officiellement qu'il utilisait cette méthode (RFC 3360), après l'avoir nié pendant longtemps).
— Permet à l'utilisateur de dire « Oui, je sais » et de faire ainsi cesser les notifications.
— Respecte les pages Web existantes, n'en supprime pas du contenu et n'en insère pas (à vous de voir si l'exemple Javascript montré plus haut respecte ce principe). Le RFC note justement que toute modification des pages Web existantes susciterait une levée de boucliers.

Ensuite, ceux liés au relais HTTP :

— Logiciel libre, avec un client ICAP (Squid a été choisi, entre autres pour cette gestion d'ICAP <http://wiki.squid-cache.org/Features/ICAP>).
— Possibilité d'ACL car les notifications ne doivent être envoyés qu'à certains utilisateurs, et que certaines pages Web ne doivent pas être affectées du tout.

Il y a aussi quelques exigences techniques pour le serveur ICAP (un protocole, normalisé dans le RFC 3507, permettant à un serveur ou relais de demander au serveur ICAP des changements à apporter au contenu qu'ils servent) et pour l'arrière-plan du système (la partie qui va noter les comportements bizarres des machines, ainsi que les actions de réparation des utilisateurs).

Armée de ces bons principes, les sections 4, 5, 6 et 7 présentent le système effectivement déployé. Le relais Squid... relaie les requêtes HTTP et, lorsqu'il reçoit une réponse, demande au serveur ICAP si et comment la modifier. Le serveur ICAP utilise GreasySpoon <http://greasyspoon.sourceforge.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3507.txt

net/>. Si la notification est nécessaire, il répond avec un bout de code Javascript à insérer dans les pages. Un serveur Tomcat garde les messages à envoyer aux utilisateurs (il peut y avoir plusieurs types de messages). Un équipement de "*load-balancing*" est utilisé pour séparer le trafic HTTP du reste (même s'il tourne sur le port 80) et pour n'aiguiller vers le relais Squid que les clients listés comme suspects. Le dessin n° 1 montre tous ces composants et leur interaction. Le dessin n° 2, plus concret, illustre les connexions réseau entre ces composants. À noter qu'il ne semble pas y avoir de protection contre un "*malware*" qui supprimerait la notification (après tout, il contrôle la machine de l'utilisateur).

Voici pour le fonctionnement technique. Maintenant, quelques considérations de sécurité, en section 10. D'abord, la rédaction du message de notification doit être très soignée, notamment pour éviter que l'utilisateur ne la considère comme du hameçonnage. Le message ne doit pas mener à une page où l'utilisateur est invité à taper son mot de passe. Il doit fournir un moyen d'obtenir plus de détails, par exemple un numéro de téléphone ou tout autre mécanisme de validation indépendante.

Un tel système de modification des pages Web vues, à des fins de notification, est-il Bon ou Méchant ? La section 11 discute cette question philosophique. Certains peuvent penser que le changement, par un intermédiaire technique, le FAI, est un franchissement de la ligne rouge. Selon ce point de vue, les intermédiaires ne devraient **jamais** tripoter le contenu des communications. Le RFC ne répond pas réellement à ces objections (à part en affirmant que Comcast a de « "*good motivations*" » et par l'argument traditionnel de chantage que, si on refuse ce mécanisme, on aura pire, par exemple à base de DPI) mais note que, au minimum, l'existence de ce système doit être annoncée aux clients (actuellement, la plupart des FAI gardent un silence complet, vis-à-vis de leurs propres clients, sur les éventuelles opérations de « gestion du réseau » qu'ils effectuent).

Et, compte-tenu des objections que peut soulever ce système, fallait-il publier ce RFC ? C'est un débat traditionnel à l'IETF : lorsqu'une idée ne fait pas l'objet d'un consensus, mais est largement déployée sur l'Internet, faut-il la documenter dans un RFC (évidemment sans que celui-ci ait le statut de norme, cf. RFC 1796 et RFC 2026), au risque que des gens croient que l'idée bénéficie d'un soutien de l'IETF, ou bien l'ignorer, au risque que les RFC ne reflètent plus la réalité de l'Internet ? La section 11 répond également à ce débat de fond en considérant qu'il vaut mieux publier, ne serait-ce que pour aider aux futurs débats.

Le problème de ces notifications est d'autant plus complexe que le FAI n'est pas tout seul face aux décisions à prendre : de nombreux organismes, au gouvernement ou ailleurs, font lourdement pression sur les FAI pour que ceux-ci fassent quelque chose contre les machines zombies. Est cité Howard Schmidt, conseiller d'Obama pour la cybersécurité, qui dit « "*As attacks on home-based and unsecured networks become as prevalent as those against large organizations, the need for ISPs to do everything they can to make security easier for their subscribers is critical for the preservation of our nation's information backbone.*" » Le RFC dit clairement qu'un des buts de cette technique de notification est de satisfaire de telles demandes.

Mais, pour atteindre ce but, y a-t-il d'autres méthodes ? La section 12 présente une alternative possible, le jardin fermé. Dans un tel système, les clients détectés comme zombifiés sont limités à l'accès à un petit nombre de sites Web, par exemple ceux de sites liés la sécurité, ou à la mise à jour de leur système d'exploitation. De facto, cela notifie l'utilisateur qu'il y a un problème. On peut aussi rediriger toutes les autres connexions HTTP vers un serveur présentant un message d'alerte. Mais c'est beaucoup plus violent, en coupant l'accès à des services que l'utilisateur peut considérer comme essentiels, comme une conversation téléphonique qui serait alors brutalement interrompue. La responsabilité du FAI serait alors clairement engagée. Le RFC rappelle à juste titre qu'il n'y a pas que le Web ! Et rejette donc l'idée du jardin fermé.

Ah, et mon opinion personnelle ? Je pense que cette méthode est un compromis acceptable. Sur le principe, c'est une attaque de l'intermédiaire et une violation de la neutralité du réseau mais l'ampleur du problème des machines Windows infectées, qui attaquent ensuite les autres, est tel qu'il peut être

justifié de prendre des mesures un peu dures. Après, tout dépendra des détails concrets de réalisation (taux de fausses alertes, par exemple).

Un article avait été fait à l'occasion des premiers tests, « "*Comcast to send infected-PC alerts*" <http://www.zdnet.com/news/comcast-to-send-infected-pc-alerts/350878> ».