RFC 6092 : Recommended Simple Security Capabilities in Customer Premises Equipment for Providing Residential IPv6 Internet Service

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 janvier 2011

Date de publication du RFC : Janvier 2011

https://www.bortzmeyer.org/6092.html

Ah, la sécurité informatique... Sujet de polémiques sans fin, à l'IETF comme ailleurs. Faut-il des pare-feux ? Faut-il protéger le PC de M. Michu par défaut ? Vaut-il mieux la liberté ou bien la sécurité (réponse en un paragraphe, de préférence) ? Il est très difficile de réunir un consensus sur ce thème et, après avoir essayé, le groupe de travail v6ops <http://tools.ietf.org/wg/v6ops> de l'IETF a décidé de produire uniquement des recommandations sur la configuration d'un pare-feu pour les CPE (les « "*boxes*" ») IPv6, sans s'engager sur la question de savoir s'il fallait un pare-feu ou pas...

Le CPE ("*Customer Premises Equipment*") est le petit boîtier qui connecte le réseau du client résidentiel (M. Michu) au FAI. Parfois fourni par le FAI, parfois acheté en supermarché, il est typiquement laissé tel quel, sans changement de la configuration, et sa configuration par défaut a donc une grande importance. Aujourd'hui, où le principal protocole de l'Internet est IPv4, le CPE a en général une configuration par défaut qui inclut l'allocation d'adresses IP privées (RFC 1918 [1]) aux machines du réseau local, et une connexion au FAI, par laquelle arrivera une adresse IP publique. Certaines personnes croient que le fait que les machines « internes » ont une adresse IP privée leur assure une certaine sécurité. C'est largement faux mais, ce qui est sûr, c'est que le mécanisme de traduction d'adresses entre l'intérieur et l'extérieur empêche l'ouverture de connexion à la demande des machines extérieures. Si une imprimante a un serveur HTTP pour sa configuration, le monde extérieur ne peut pas s'y connecter par des moyens normaux (il existe plusieurs moyens « anormaux » comme le changement DNS <https://www.bortzmeyer.org/dns-rebinding-pinning.html>, et c'est pour cela que le bénéfice en terme de sécurité est douteux). Parfois, on est content de ce barrage (qui empêche les méchants barbares de se connecter à notre innocente imprimante, qui n'est probablement pas configurée de manière très sûre, avec son mot de passe d'usine jamais changé), parfois on le regrette (quand le transfert de fichiers

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1918.txt

en pair-à-pair échoue ou bien quand un coup de téléphone SIP s'établit mais que la voix ne passe pas). Ce filtrage de fait peut en effet frapper aussi bien les applications légitimes que les autres.

Et en IPv6 ? Le NAT s'est imposé pour IPv4 car, de toute façon, vu le manque d'adresses IPv4 <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html>, il n'y avait pas le choix. Mais en IPv6, on revient à l'abondance. Chaque machine du réseau local a une adresse IP unique au niveau mondial et redevient donc contactable. La sécurité largement illusoire <https://www.bortzmeyer.org/nat-et-securite.html> que fournissait le NAT disparait. Que faut-il mettre à la place ? Un pare-feu ? Et, si oui, configuré comment à sa sortie d'usine ? (Rappelez-vous que M. Michu ne changera probablement jamais la configuration. La règle d'ergonomie que rappelle notre RFC est que le réglage de la sécurité dans le CPE doit être facile à faire et encore plus facile à ignorer.)

Notre RFC 6092 ne répond pas à toutes les questions sur la sécurité d'IPv6. Il vise le cas où on veut fournir une sécurité **simple** dans un environnement simple, celui de l'accès Internet résidentiel. Les cas plus complexes, par exemple celui d'un réseau d'une entreprise, avec des ingénieurs compétents, sont délibérement laissés de côté (section 1 et RFC 4864). Ici, ce RFC propose un compromis jugé raisonnable dans ce contexte, entre la sécurité et le respect du principe de connexion de bout en bout, qui est à la base du succès d'Internet.

Chaque virgule de ce RFC a été négociée avec acharnement. Ainsi, le statut de ce document est « pour information » mais il utilise le vocabulaire normatif du RFC 2119, tout en précisant en section 1.2 que c'est uniquement pour la précision du discours.

La section 2 du RFC décrit le modèle de réseau considéré. Le CPE est un engin simple (on ne peut donc pas lui demander des services complexes), routeur par défaut du réseau local (RFC 8504), et n'ayant typiquement qu'une seule interface avec l'extérieur, celle qui mène au FAI, plus un petit nombre d'interfaces vers l'intérieur (par exemple Ethernet et WiFi). Pour ce qui concerne la sécurité, leur vision du monde est également très simple : un univers rassurant et civilisé du côté du réseau local, un monde hostile et dangereux de l'autre côté. Le RFC 4864 résume les services de sécurité attendus : filtrer les paquets clairement anormaux (comme les paquets venant de l'extérieur mais prétendant avoir une adresse IP source interne, ou comme les paquets martiens du RFC 4949), ne pas permettre de paquets entrants sans qu'une « connexion » ait d'abord été initiée de l'interieur, permettre des exceptions manuellement configurées (sur la base des adresses IP et ports source). En IPv4, un routeur NAT fournissait plusieurs de ces services sans même y penser, comme effet de bord de la traduction d'adresses (qui, en IPv4, impose de garder un état, donc de se souvenir des connexions ouvertes). En IPv6, routage et filtrage sont nettement séparés. Le RFC 4864 proposait donc de répondre au besoin des utilisateurs (qui s'attendent au même niveau de protection en IPv4 et IPv6) en ayant par défaut un pare-feu à état, qui bloque les connexions entrantes.

Il faut noter que certaines applications voudraient bien autoriser les connexions entrantes, au moins depuis certaines adresses. Il n'existe pas actuellement de mécanisme normalisé pour qu'une application puisse dire au CPE « je veux bien recevoir des connexions sur mon port 443, s'il te plait, laisse-les passer ». Plusieurs projets ont été discutés à l'IETF mais sans résultat consensuel pour l'instant.

Autre point à garder en tête, le CPE est un routeur et doit donc se comporter comme tel (RFC 8504, sur les obligations des routeurs). Il doit également (section 2.1) bloquer certains paquets dangereux, comme les usurpateurs ou les martiens cités plus haut, ainsi que ceux ayant l'en-tête de routage 0 (RFC 5095). En revanche, le CPE ne doit pas empêcher le développement de nouveaux services et applications sur l'Internet. Il ne faut donc pas qu'il bloque aveuglément les paquets IPv6 ayant des en-têtes de sécurité ou de routage.

Au niveau 3, le CPE va devoir bloquer par défaut les tunnels, sauf ceux d'IPsec (car il serait paradoxal qu'un pare-feu, censé améliorer la sécurité, bloque par défaut un protocole qui permet de sécuriser le trafic). Pour la même raison, HIP doit être autorisé.

Et au niveau 4 (section 2.3) ? Le principe posé est celui d'une fermeture par défaut : pas de paquets TCP ou UDP entrants s'ils n'ont pas été sollicités de l'intérieur, par une application se connectant à l'extérieur. C'est cette simple règle, qui prend juste une phrase du RFC, qui a suscité le plus de tempêtes de discussions autour de ce document, puisqu'elle revient à renoncer, par écrit, au principe de la connectivité de bout en bout.

La section 3 du RFC transforme ensuite ces principes généraux en recommandations concrètes. Si on est pressé, on peut lire à la place la section 4, qui est un résumé de ces recommandations. Les recommandations sont identifiées par un numéro unique et je vais reprendre ici la syntaxe du RFC, qui les désigne par REC-N où N est le numéro. Dans ses recommandations, cette section s'appuie sur la classification des paquets et des flots (ensemble de paquets liés à la même « connexion ») :

— Un paquet va vers l'extérieur s'il est issu d'une machine interne, à destination d'une machine externe et vers l'intérieur si c'est le contraire.
— Pour les flots, c'est plus complexe. Si une machine interne fait une connexion TCP vers le port 80 d'une machine externe, la majorité des octets sera de l'extérieur (le serveur HTTP) vers l'intérieur alors qu'on considérerait plutôt cette connexion comme allant vers l'extérieur. La définition d'un flot allant vers l'extérieur considère donc que c'est le **premier** paquet qui détermine la direction du flot. Notons que c'est très simpliste et que cette définition ne couvre pas des cas comme SIP/RTP où l'appelant peut demander à recevoir des paquets sur une toute autre « connexion ».

La section 3.1 traite ensuite du cas du filtrage le plus simple, le filtrage **sans état**, où chaque paquet est examiné uniquement pour son contenu, indépendamment de tout contexte. Il ne s'agit donc que de filtrage de paquets, sans considérer les flots. Ce filtrage sans état devrait éliminer les usurpateurs (REC-5, REC-6, RFC 2827 et RFC 3704). De même, les adresses IPv6 appartenant à des services obsolètes (RFC 3879) ou non admises sur l'Internet public (RFC 6890) devraient être rejetées (REC-3). Les requêtes DNS de l'extérieur devraient être rejetées (REC-8).

Les sections 3.2 et 3.3 traitent ensuite du cas du filtrage **avec état**, où le CPE filtrant se souvient des paquets précédents et peut donc reconstituer des flots. Le cas facile (section 3.3) est celui où le protocole de couche 4 a une sémantique de connexion. Le pare-feu peut alors savoir exactement quand la connexion est ouverte et quand elle se termine (ce qui permet de libérer les ressources allouées pour le filtrage). Parmi les protocoles dans ce cas, TCP, SCTP mais aussi tous les futurs protocoles de transport à connexion. Un problème typique des pare-feux avec état est en effet qu'ils regardent l'établissement et la fermeture de la connexion et que cela implique qu'ils connaissent le protocole de transport utilisé (beaucoup de pare-feux de bas de gamme ne connaissent pas SCTP et le bloquent donc aveuglément, ce qui rend difficile le déploiement de ce « nouveau » protocole).

Commençons par TCP, le cas le plus simple (section 3.3.1). REC-31 demande que les CPE qui gèrent TCP le gèrent **complètement** et de manière conforme à la machine à états du RFC 793, y compris dans ses aspects moins connus comme l'ouverture simultanée (les deux pairs envoyant un SYN presque en même temps). C'est quand même un comble qu'il faille un RFC pour dire que les autres RFC doivent être respectés et cela donne une idée des problèmes auquel est confronté l'Internet aujourd'hui. Afin de maximiser les chances qu'une application nouvelle fonctionne correctement, REC-33 demande que le filtrage ait un comportement indépendant de la destination, c'est-à-dire que l'adresse de destination ne soit pas prise en compte pour mettre en correspondance un paquet avec un flot. Le comportement dépendant de l'adresse peut être adopté si on souhaite contrôler davantage. Voir le RFC 5382 pour plus de détails sur la gestion de TCP dans un CPE.

Autre demande de notre RFC qui ne correspond **pas** à l'état actuel de la plupart des CPE : REC-34 exige que, par défaut, les paquets TCP SYN entrants bloqués suscitent l'envoi d'un message ICMP

"*Destination unreachable*". Aujourd'hui, il est probable que la grande majorité des CPE avalent silencieusement le paquet TCP, sans prévenir l'envoyeur.

Et les délais de garde ? A priori, pour les protocoles orientés connexion comme TCP, ils ne sont pas nécessaires, puisque la fin d'une connexion sera toujours explicite, avec les paquets FIN (ou RST). Mais, en pratique, le CPE peut rater la fin d'une connexion (par exemple parce qu'un des pairs TCP a été éteint et n'a donc pas pu terminer proprement la connexion). Pour gérer ce cas, tous les équipements intermédiaires qui gèrent les états des connexions TCP ajoutent un délai de garde et, si aucun paquet n'est passé pendant ce délai, ils suppriment l'état, ce qui empêchera les paquets ultérieurs de passer. C'est souvent un problème pour les applications comme SSH, qui peuvent laisser passer des heures sans envoyer de paquets (avec OpenSSH, la directive `ServerAliveInterval` permet d'envoyer des paquets pour maintenir la connexion ouverte). Il existe aussi des mécanismes pour envoyer des paquets TCP sans réel contenu, uniquement chargés de maintenir une connexion ouverte (RFC 1122, qui spécifie une durée par défaut de deux heures, bien trop longue pour la plupart des CPE). REC-35 demande donc un délai de garde minimal par défaut de deux heures et quatre minutes (voir le RFC 5382 pour les raisons de ce choix), bien plus long que celui des CPE d'aujourd'hui <https://www.bortzmeyer.org/home-gateway.html>.

Comme TCP a besoin d'ICMP (notamment pour découvrir la MTU du chemin), REC-36 rappelle que, si un équipement transmet du TCP, il doit aussi transmettre les messages ICMP essentiels, "*Destination unreachable*" et "*Packet too big*". A contrario (REC-37), le passage d'un paquet ICMP ne doit pas entraîner le CPE à supprimer les états des connexions TCP (décider, lors de la réception d'un paquet ICMP, si on coupe la connexion TCP associée est un processus complexe, voir le RFC 5927).

TCP est très répandu et relativement bien connu. On peut espérer (même si c'est illusoire) que tous les CPE le gèrent bien. Mais qu'en est-il de SCTP, son concurrent bien moins connu (section 3.3.2) ? D'abord, il faut noter que SCTP a des particularités, notamment le fait qu'une même connexion puisse utiliser plusieurs adresses IP. Cela rend le modèle simple de notre RFC 6092 inapplicable puisqu'il faudrait que les différents routeurs qui voient passer les paquets d'une même connexion se coordonnent (ce qui ne serait pas « simple »). Toutefois, les applications SCTP qui n'exploitent pas toutes les fonctions de ce protocole peuvent quand même s'en tirer. Si tous les paquets passent au même endroit (un seul routeur) et que cet endroit suit le RFC 4960, comme indiqué par REC-38, alors tout peut bien se passer. En pratique, SCTP souffre bien plus du fait que beaucoup de CPE bloquent tous les protocoles inconnus, ce qui signifie en général tous ceux ayant été créés il y a moins de vingt ans (les "*boxes*" sont très conservatrices <https://www.bortzmeyer.org/home-gateway.html>).

La section 3.2, elle, s'occupe du cas moins facile où, contrairement à ce qui se passe avec TCP ou SCTP, il n'y a pas de connexion explicite. Le CPE ne peut pas savoir quand une « session » se termine. La seule solution générale est d'attendre un certain temps après le dernier paquet vu. Autrement, il faut que le CPE connaisse le protocole applicatif utilisé (et sache ainsi, par exemple, qu'une « session » DNS, c'est typiquement seulement deux datagrammes UDP, un de requête et un de réponse). La section 3.2.2 avertit que cette méthode ne doit pas mener à une interdiction de fait des applications nouvelles, donc inconnues du CPE.

Pour ICMPv6, les recommandations concernant son filtrage figurent dans le RFC 4890 (pour le résumer : il ne faut **surtout pas** filtrer aveuglément la totalité d'ICMP). La règle ajoutée par notre RFC 6092 est de vérifier, avant de faire suivre un paquet ICMP, que les données de la couche transport qu'il contient correspondent à une session en cours (section 3.2.1).

Pour les autres protocoles, afin de maximiser les chances qu'une application nouvelle ou un protocole de couche 4 nouveau fonctionnent correctement, la section 3.2.2 recommande fortement que le filtrage ait un comportement indépendant de la destination (REC-11, "*endpoint independent filter*"), c'est-à-dire que

l'adresse de destination ne soit pas prise en compte pour mettre en correspondance un paquet avec un flot. Le comportement dépendant de l'adresse peut être adopté si on souhaite une sécurité plus fasciste, peut-être via une option de configuration du pare-feu.

Le cas spécifique d'UDP (rappelons qu'il existe d'autres protocoles de transport que TCP et UDP) est traité en section 3.2.3. Le RFC 4787 donnait déjà des règles pour le comportement attendu d'un routeur NAT avec UDP. Ces règles s'appliquent toujours dans le cas de notre CPE IPv6. Par exemple, REC-14 demande un délai de garde minimum, avant de détruire l'état associé à un flot, de deux minutes (moins si les ports sont dans la zone bien connue, cf. REC-15). Beaucoup d'applications avaient en effet souffert de CPE qui détruisaient l'état trop tôt, empêchant les paquets ultérieurs de passer.

Et IPsec ? Il est en section 3.2.4. Comme son utilisation permet d'améliorer sérieusement la sécurité de l'Internet, il serait paradoxal qu'un équipement assurant des fonctions de sécurité le bloque. REC-21 et REC-22 demandent donc qu'IPsec passe par défaut. Même chose pour HIP avec REC-26. Notons que de telles recommandations laissent de côté le cas de **futurs** protocoles de sécurité, dont le déploiement risque d'être bloqué par un CPE qui interdit trop de choses par défaut.

Enfin, les applications de gestion font l'objet de la section 3.5. La plupart des CPE ne verront jamais leur configuration par défaut modifiée et, si c'est le cas, celui ou celle qui change la configuration ne sera en général pas un expert. Le RFC ne donne pas plus de détails, à part l'insistance sur le fait que ces fonctions de configuration ne doivent pas être accessibles depuis l'Internet, seulement depuis le réseau local.

On l'a vu, le comportement par défaut d'un CPE qui suit ce RFC est d'interdire les connexions entrantes. Or, certaines applications souhaiteraient pouvoir en recevoir (section 3.4), soit parce qu'elle jouent un rôle de serveur, soit parce que le fonctionnement du protocole sous-jacent implique que l'application se mette en écoute (c'est le cas de SIP). Il n'existe pas de mécanisme normalisé pour l'application de dire au pare-feu « laisse entrer les paquets RTP en provenance de `2001:db8:1010::47:1` », même s'il existe des solutions non-standard comme UPnP. Avec les CPE traditionnels, gérant de l'IPv4 NATé, les applications se servent typiquement de STUN pour « ouvrir (indirectement) une brèche » dans le pare-feu. Et en IPv6 ? Même s'il n'y a pas de NAT, les recommandations de notre RFC 6092 ont le même résultat d'empêcher les connexions entrantes. L'idéal serait un mécanisme standard équivalent à UPnP ou à des propositions comme l'"*Application Listener Discovery Protocol*" <`http://tools.ietf.org/id/draft-woodyatt-ald`>. Mais il n'existe pas et l'état actuel des discussions à l'IETF ne permet pas de penser qu'il apparaitra bientôt, bien que des propositions existent comme une extension des RFC 5189 ou RFC 4080.

En attendant, les recommendations pour les CPE sont d'offrir un mécanisme pour permettre aux applications de solliciter l'ouverture du pare-feu (REC-48) et de permettre, via une action de l'administrateur, de couper le pare-feu ou de changer sa politique pour laisser entrer les connexions (REC-49). Bien sûr, ce dernier mode impose que les machines sur le réseau local soient proprement sécurisées (le RFC utilise une formulation plus restrictive, en demandant que les machines aient leur propre pare-feu, alors que ce n'est qu'une solution de sécurisation parmi d'autres) mais il est nécessaire que cette option soit présente, car c'est la seule qui permet l'accès à toutes les potentialités de l'Internet. Sans elle, le CPE n'est pas un point d'accès à l'Internet mais un Minitel amélioré.

La section 7, consacrée spécifiquement à la sécurité, prend de la hauteur et rappelle quelques bons principes de sécurité. Elle note ainsi que le RFC ne prétend pas que la politique par défaut proposée (qu'on peut résumer par « zéro connexion entrante par défaut ») soit effective, simplement qu'elle assure le même niveau de sécurité ou d'insécurité que celui fourni par un routeur NAT d'aujourd'hui. Comme le rappelle le RFC 2993, cette politique peut même diminuer la sécurité (par exemple par

l'« illusion du "*limes*" », ce qui fait qu'on se fie aveuglément à la protection qu'elle offre et qu'on oublie de sécuriser l'intérieur). Les bonnes pratiques de sécurité, et des protections comme celles recommandées dans le RFC 4864 sont donc toujours d'actualité. Après tout, les attaques ne viennent pas que de l'extérieur. Même quand l'attaquant humain n'a pas accès au réseau local, il peut trouver un moyen (via du "*malware*", par exemple) de l'attaquer de l'intérieur.

Cette section sur la sécurité note que la grande majorité des ordinateurs dont le système d'exploitation gère IPv6 ont également un mécanisme de filtrage compatible IPv6. Ce n'est toutefois pas forcément le cas pour les autres machines, comme les imprimantes.

Petite exception au principe comme quoi la sécurité d'un réseau situé derrière un CPE IPv6 conforme à ce RFC est à peu près la même que celle d'un réseau NATé : IPsec n'est pas, par défaut, autorisé par le CPE typique actuel. De toute façon, le dernier paragraphe du RFC, écrit dans le pur style juridique états-unien, note bien que l'IETF n'est responsable de rien et ne garantit pas que les règles énoncées dans ce RFC améliorent la sécurité... C'est une innovation dans un RFC : verra-t-on le RFC sur TCP dire « L'IETF ne promet pas que vous échapperez à la congestion si vous suivez cet algorithme » et celui sur SMTP affirmera-t-il « Attention, utiliser le courrier vous expose au spam » ?