RFC 9726 : Operational Considerations for Use of DNS in Internet of Things (IoT) Devices

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 avril 2025

Date de publication du RFC : Mars 2025

https://www.bortzmeyer.org/9726.html

Les objets connectés, vous le savez, sont une source d'ennuis et de risques de sécurité sans fin. Non seulement ils ne sont pas forcément utiles (faut-il vraiment connecter une brosse à dents au réseau ?) mais en outre, comme ils incluent un ordinateur complet, ils peuvent faire des choses auxquelles l'acheteur ne s'attend pas. Pour limiter un peu les risques, le RFC 8520[1] normalisait MUD ("*Manufacturer Usage Description*"), un mécanisme pour que le fournisseur du machin connecté documente sous un format analysable les accès au réseau dont l'objet avait besoin ce qui permet, par exemple, de configurer automatiquement un pare-feu, qui va s'assurer que la brosse à dents ne se connecte pas <https://www.ietf.org/blog/mud/> à Pornhub. Dans un fichier MUD, les services avec lesquels l'objet communique sont typiquement indiqués par un nom de domaine. Mais le DNS a quelques subtilités qui font que l'utilisation de ces noms nécessite quelques précautions, décrites dans ce nouveau RFC.

En fait, le RFC 8520 permet d'utiliser des noms ou bien des adresses IP, qui seront ensuite ajoutées dans les ACL du pare-feu. Évidemment, les noms de domaine, c'est mieux. Pas pour la raison qu'on trouve dans tous les articles des médias « les noms de domaine sont plus simples à retenir pour les humains » puisque les fichiers MUD ne sont pas prévus pour être traités par des humains. Mais parce qu'ils sont stables (contrairement aux adresses IP, qui changent quand on change d'hébergeur), qu'ils gèrent à la fois IPv4 et IPv6 et parce qu'ils permettent la répartition de charge, par exemple via un CDN.

Mais le pare-feu (sauf s'il espionne la résolution DNS préalable) ne voit pas les noms de domaine, il voit des paquets IP, avec des adresses IP source et destination. (Ainsi que les ports, si le paquet n'est pas chiffré.) Il va donc falloir résoudre les noms en adresses si on veut configurer le pare-feu à partir du fichier MUD, et c'est là que les ennuis commencent. Les exigences de sécurité sont contradictoires (section

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8520.txt

9 du RFC) : on veut bloquer les accès au réseau mais « en même temps » les permettre. Cette permission est nécessaire pour que l'objet connecté puisse mettre à jour son logiciel, par exemple lorsqu'une faille de sécurité est découverte. Et c'est également nécessaire pour que le vendeur de l'objet puisse recevoir des quantités de données personnelles, qu'il revendra (ou se fera voler suite à un piratage).

Le plus simple (section 3 du RFC) pour traduire les noms de domaine du fichier MUD en adresses IP est évidemment que le contrôleur MUD, la machine qui lit les fichiers MUD et les traduit en instructions pour le pare-feu (RFC 8520, sections 1.3 et 1.6) fasse une résolution DNS normale et utilise le résultat. C'est simple et direct. **Mais** notre section 3 décrit aussi pourquoi ça risque de ne pas donner le résultat attendu. Par exemple, le résolveur DNS <https://www.bortzmeyer.org/resolveur-dns.html> peut renvoyer les résultats (les différentes adresses IP) dans un ordre variable, voire aléatoire, comme décrit dans le RFC 1794. C'est souvent utilisé pour faire une forme grossière de répartition de charge. Si le résolveur renvoie **toutes** les adresses IP possibles, OK, le contrôleur MUD les autorise toutes dans le pare-feu. Mais s'il n'en renvoie qu'une partie, on est mal, celle(s) autorisée(s) ne sera pas forcément celle(s) utilisée(s) par l'objet connecté. Et la résolution peut dépendre du client (le RFC parle de « non-déterminisme » ; en fait, c'est parfaitement déterministe mais ça dépend du client, par exemple de sa géolocalisation). Si le serveur faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autor
html> renvoie une seule adresse qui dépend de la localisation physique supposée de son client, et que le machin connecté et le contrôleur MUD n'utilisent pas le même résolveur DNS <https://www.bortzmeyer.org/resolveur-dns.html>, des ennuis sont à prévoir : l'adresse autorisée sur le pare-feu ne sera pas celle réellement utilisée.

Si contrôleur et truc connecté utilisent le même résolveur, le risque d'incohérence est plus faible. Les réponses seront les mêmes pour les deux (contrôleur et chose connectée). Cela peut être le cas dans un environnement résidentiel où tout le monde passe par le CPE (la "*box*") qui fait à la fois résolveur DNS, routeur NAT et pare-feu. Mais que faire si, par exemple, la brosse à dents ou l'aspirateur connecté utilisent un résolveur DNS public, quelque part dans le mythique nuage ?

La section 4 du RFC liste tout ce qu'il ne faudrait **pas** faire en matière d'utilisation du DNS par les objets connectés. Typiquement, l'objet connecté utilise HTTPS pour se connecter à son maitre (le vendeur de l'objet ; le gogo qui a acheté l'objet connecté croit en être propriétaire mais il dépend du vendeur <https://www.twz.com/air/you-dont-need-a-kill-switch-to-hobble-exported-f-35s>, qui peut arrêter le service à tout moment). Pour le cas d'une mise à jour logicielle, le maitre informe alors si une mise à jour du logiciel est nécessaire, en indiquant l'URL où l'objet va pouvoir récupérer cette mise à jour. Évidemment, il est préférable pour la sécurité que cette mise à jour soit signée (RFC 9019). Notez que ce qui est bon pour la sécurité face au risque de piratage par un tiers n'est pas forcément bon pour le consommateur : la vérification de la signature va empêcher les utilisateurs de mettre leur propre logiciel <https://pirg.org/edfund/resources/john-deere-repair-software/> par exemple parce que le vendeur ne fournit plus de mise à jour.

Questions bonnes pratiques, d'abord, il faut utiliser le DNS, donc un nom de domaine dans l'URL, pas des adresses IP littérales. Cette approche aurait certes l'avantage de fonctionner même quand la résolution DNS est défaillante, mais elle souffre des problèmes suivants :

— Il faut choisir entre une adresse IPv4 et IPv6, alors que le maitre ne sait pas forcément de quelle connectivité dispose l'objet. Si l'objet se connecte en IPv4 au maitre, celui-ci pourrait croire qu'une adresse IPv4 littérale est acceptable ce qui n'est pas vrai si l'objet est connecté via NAT64 (RFC 6146). Dans cet exemple, le DNS marcherait (RFC 6147) mais pas l'adresse littérale.
— Ensuite, l'adresse IP peut changer fréquemment, par exemple si le vendeur change d'hébergeur. C'est justement le principal avantage du DNS que de permettre une stabilité des identificateurs sur le long terme et non pas, comme on le lit souvent parce que « les noms sont plus lisibles et plus mémorisables que les adresses », argument qui ne s'applique pas aux objets connectés.

— Enfin, si la mise à jour est récupérée via HTTPS, ce qui est évidemment recommandé, le serveur doit obtenir un certificat et peu d'AC acceptent d'émettre des certificats pour des adresses IP (à l'heure de la publication de ce RFC, Let's Encrypt a annoncé qu'ils allaient le permettre mais cela ne semble pas encore fait).

Autre problème, des noms de domaine qui, en raison des systèmes utilisés sur le service HTTP, par exemple un CDN, changent souvent et de manière qui semble non déterministe. Modifier les fichiers MUD ne sera alors pas possible. Le RFC recommande, s'il faut absolument changer les noms souvent, de mettre un alias dans le fichier MUD, modifié à chaque fois que le nom change.

Toujours du côté des CDN, même s'ils ne sont pas les seuls à poser ce genre de problème, les noms trop génériques. Si tous les clients du service d'hébergement sont derrière le même nom de domaine, celui-ci sera difficile à modifier, et les effets d'une compromission pourront être plus étendus que souhaitable. Il vaut mieux des noms spécifiques à chaque client <https://techmonitor.ai/techonology/cloud/aws-s3-path-deprecation>.

Les requêtes faites par le contrôleur MUD peuvent poser des problèmes de confidentialité (section 5 du RFC). Si le résolveur local n'est pas jugé digne de confiance, et que le contrôleur MUD utilise un résolveur distant, par exemple un résolveur public comme ceux de Cloudflare ou Quad9 (de préférence avec chiffrement, via DoH RFC 8484, DoT RFC 7858 ou DoQ RFC 9250), il faut s'assurer que l'objet va utiliser ce même résolveur (il n'y a pas aujourd'hui de mécanisme dans MUD pour faire cela automatiquement).

Les recommandations positives, maintenant (section 6). Le fichier MUD peut être obtenu de diverses façons (par exemple via un code QR comme documenté dans le RFC 9238), c'est son contenu qui compte. Les conseils de la section 6 sont en général du simple bon sens, qui correspondent aux bonnes pratiques habituelles d'utilisation des URL, mais qui ne sont pas toujours appliquées par les objets connectés. Notre RFC recommande :

— D'utiliser le DNS (et pas des adresses IP littérales comme le font trop de vendeurs),
— d'utiliser des noms contrôlés par le vendeur de l'objet (et pas par son hébergeur Web), éventuellement en utilisant des alias (RFC 9499, section 2),
— d'utiliser des CDN dont les serveurs faisant autorité retournent plusieurs réponses, quitte à les réordonnancer pour la répartition de charge, pour augmenter la probabilité que le contrôleur MUD et l'objet aient des adresses en commun,
— de ne pas utiliser des réponses adaptées au client DNS ; un mécanisme comme ECS (RFC 7871) permet de donner des réponses différentes selon l'adresse IP du client final (et pas celle du client que voit le serveur faisant autorité), ce qui va créer des problèmes si le contrôleur MUD et l'objet passent par des connexions Internet différentes,
— d'utiliser le résolveur local (celui indiqué via DHCP ou RA - RFC 8106), en s'appuyant sur les techniques décrites dans les RFC 9462 et RFC 9463 ; l'utilisation de résolveurs DNS publics est découragée, à mon avis pour de mauvaises raisons.

Notez que l'objet peut aussi utiliser des noms mais ne pas se servir du DNS pour les traduire en adresses IP. C'est le cas par exemple s'il utilise mDNS (RFC 6762 et RFC 8882) qui, en dépit de son nom, n'est pas du DNS, ce qui peut également poser des problèmes d'incohérence (l'objet n'obtenant pas la même adresse IP que le contrôleur MUD).

La section 8 revient sur les questions de vie privée (RFC 7626). Elle dit (ce qui est peut-être contestable) que les requêtes DNS d'un four à micro-ondes ne sont pas forcément un enjeu d'intimité mais que celles d'un "*sextoy*" le sont davantage. L'utilisation de DoT ou DoH va supprimer le risque d'une écoute par un tiers mais le résolveur, dans tous les cas, voit la requête et il faut donc choisir un résolveur de confiance, sauf à utiliser des techniques qui sont pour l'instant très rares, comme celle du RFC 9230. Ah, et le RFC discute aussi des risques posés par la divulgation de la version de logiciel actuellement utilisée par l'objet (qui va lui servir à savoir s'il faut une mise à jour) mais le RFC 9019 a déjà traité cela.