RFC 6039 : Issues with existing Cryptographic Protection Methods for Routing Protocols

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 octobre 2010

Date de publication du RFC : Octobre 2010

https://www.bortzmeyer.org/6039.html

Le groupe de travail IETF opsec <http://tools.ietf.org/wg/opsec> travaille à améliorer la sécurité des protocole de routage. Ce RFC du groupe (l'un des premiers, après les très intéressants RFC 4778 [1] et RFC 5635) fait le point sur l'utilisation actuelle de la cryptographie dans ces protocoles et ses limites, les problèmes qu'elle soulève, notamment de gestion des clés.

Plusieurs protocoles de routage ont des mécanismes standards pour protéger par la cryptographie les informations transmises. OSPF v2 (RFC 2328), OSPF v3 (RFC 5340), IS-IS (RFC 1195), BGP (RFC 4271), ou des protocoles qui, sans être à proprement parler « de routage », en sont néanmoins proches comme BFD (RFC 5880), sont tous capables de prendre un condensat cryptographique de l'information transmise, en utilisant une fonction de hachage comme MD5 (RFC 1321) et une clé secrète. Le routeur récepteur, s'il a la même clé secrète, peut refaire le calcul du condensat et, s'il trouve la même valeur, être rassuré que le message vient bien d'un routeur autorisé, et qu'il n'a pas été modifié en route. (On notera que l'utilisation de protocoles à clés publiques est infiniment plus rare, au point d'être à peine mentionné dans le RFC.)

Notons que ces protocoles n'assurent pas de service de confidentialité : un méchant qui espionne la communication entre deux routeurs peut toujours apprendre les routes échangées.

Comme le note la section 1 du RFC, qui étudie les problèmes génériques, communs à tous les protocoles de routage, cette approche a des limites : l'utilisation de clés secrètes partagées entraine une gestion manuelle des clés, plutôt pénible. Remplacer une clé compromise est ainsi une opération longue

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4778.txt

et risquée (il est facile d'oublier un routeur, ce qui cassera les communication entre les routeurs ayant la nouvelle clé et les autres). D'autre part, plusieurs cas existent où les attaques par rejeu sont possibles.

Comme le note la section 1.1, la fonction de hachage la plus couramment utilisée est MD5. Celle-ci n'est pas sans reproche, cryptographiquement parlant. Des attaques par collision (trouver deux messages qui ont le même condensat cryptographique) ont déjà été publiées par exemple par Xiaoyun Wang, Xuejia Lai, Dengguo Feng & Hongbo Yu, dans « "*Collisions for hash functions MD4, MD5, HAVAL-128, and RIPEMD*" <http://web.mit.edu/fustflum/documents/crypto.pdf> », Crypto 2004 Rump Session. En revanche, des attaques dites « pré-image » (où on produit un message qui a le même condensat qu'un message donné, ce qui est beaucoup plus difficile qu'une attaque par collision) ne semblent pas avoir été publiées. Pour le cas particulier des protocoles de routage, les collisions ne sont pas trop à craindre car beaucoup de champs dans un message ne peuvent pas prendre des valeurs arbitraires (par exemple, dans OSPF le champ « "*Link State Advertisement function code*" - section A.4.2.1 du RFC 5340 - peut valoir seulement de 1 à 14 <https://www.iana.org/assignments/ospfv3-parameters/ospfv3-parameters.xhtml#ospfv3-parameters-2>). Pour les attaques pré-image, elle seraient sans doute plus graves, mais la taille relativement limitée des messages de routage limite sans doute les risques d'exploitation effective.

SHA-1 a aussi ses faiblesses (section 1.2) mais il faut bien voir qu'elles ne sont pas réellement utilisables pour une attaque, dans les conditions spécifiques des protocoles de routage (cf. RFC 4270 pour un bon arrière-plan sur la question.)

Les sections suivantes détaillent ensuite les problèmes pour le cas spécifique de chaque protocole. À chaque fois, sont indiqués les problèmes d'ordre pratique (gestion des clés) et ceux d'ordre technique (faiblesses dans le protocole). OSPF v2 fait l'objet de la section 2. Les routeurs OSPF d'un même réseau peuvent s'authentifier en condensant les paquets et une clé secrète avec MD5 (RFC 2328, annexe D.3). Cette clé se configure avec :

```
interface eth0
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 MACLESECRETE
```

sur Quagga et avec :

```
authentication-type md5;
interface fe-0/0/1.0 {
     authentication {
         md5 1 key "$9$dEbgoZUjqP5GUApO1hcgoaJHq"; MACLESECRETE
     }
}
```

sur JunOS. Il existe aussi un mécanisme expérimental de signature des messages par des clés asymétriques (RFC 2154) mais qui n'a apparemment jamais été déployé.

Le condensat cryptographique s'applique aux messages échangés entre routeurs, pas à l'information de routage elle-même. Autrement dit, MD5 dans OSPF sécurise le canal mais pas le message (le LSA). OSPF ne fournit aucun mécanisme de négociation de paramètres comme l'algorithme de hachage utilisé. Si on veut utiliser SHA (RFC 5709), tous les routeurs d'un même lien doivent être manuellement configurés de manière identique. D'une manière générale, tout changement dans les clés doit faire l'objet d'une prudente intervention planifiée et manuellement exécutée.

L'authentification cryptographique dans OSPF a aussi des problèmes techniques (section 2.2) comme le fait que la même clé soit utilisé pour tous les routeurs d'un même réseau Ethernet, obligeant à largement partager le secret.

Pour OSPF v3 (RFC 5340 et section 3), encore peu déployé, la situation est toute différente. Il n'a pas de mécanisme d'authentification intrinsèque et ses utilisateurs sont censés s'appuyer sur IPsec (par exemple RFC 4303, sans chiffrement, comme spécifié dans le RFC 2410, pour avoir juste la signature). La norme de sécurité d'OSPF v3 (RFC 4552) précise que les clés IPsec doivent être gérées manuellement, pas avec un protocole comme IKE parce que le modèle de fonctionnement d'OSPF (un-vers-plusieurs) est très particulier.

Même si le RFC n'en dit pas un mot, le principal problème d'OSPF v3, aussi bien technique que pratique, est qu'IPsec est très peu déployé et que peu d'administrateurs réseaux ont envie de s'y mettre pour assurer une fonction qui, en OSPF v2, ne prenait que deux lignes de configuration.

Et le concurrent d'OSPF, IS-IS (RFC 1195 et section 4 de notre RFC) ? Lui aussi peut utiliser MD5 pour l'authentification (voir RFC 5304). Mais, contrairement à OSPF, la signature porte sur l'annonce d'état d'un lien, pas sur un message échangé entre deux routeurs. Cela veut dire qu'on peut authentifier un routeur distant, mais cela oblige (section 4.1) tous les routeurs à avoir la même clé... Comme OSPF, pas moyen de négocier la clé, ou l'algorithme de condensation. Si on veut remplacer MD5 par SHA-256, un gros travail de gestion est à prévoir.

Faut-il également parler de l'antédiluvien RIP ? La section 6 lui consacre quelques paragraphes. La version historique, dans le RFC 1058, avait **zéro** service de sécurité. La version 2, actuellement normalisée dans le RFC 2453, et dispose désormais d'un mécanisme d'authentification analogue à ceux des autres protocoles (cf. RFC 4822).

Bien que n'étant pas un protocole de routage, BFD (RFC 5880) est tellement souvent utilisé dans les routeurs qu'il mérite sa section de ce RFC, la 7. BFD sert à détecter la panne d'une liaison. Il dispose d'un mécanisme d'authentification optionnel (sections 4.2 et 6.7 du RFC 5880), à clés partagées. Il hérite donc des problèmes de gestion de clés de ce système.

Et BGP, vedette des protocoles utilisés sur l'Internet, puisque c'est lui qui assure le routage entre opérateurs ? La section 5 détaille ce protocole. BGP est forcément transporté sur TCP. Il peut donc utiliser les options d'authentification de TCP comme celle du RFC 2385 (très répandue en pratique dans le monde BGP) ou bien celle du RFC 5925, plus récente, bien meilleure et plus rare en pratique. Pour cette raison, notre RFC 6039 ne couvre que l'authentification MD5 du RFC 2385. Parmi les problèmes de gestion, il y a le fait que BGP est typiquement utilisé de part et d'autre d'une frontière administrative. Cela complique sérieusement les opérations de gestion de clés. Par exemple, quand avez-vous pour la dernière fois changé un mot de passe MD5 sur une session BGP avec un pair ? (Réponse : vous ne l'avez probablement jamais fait.) Sur un gros point d'échange, un routeur BGP peut se connecter à des dizaines, voire des centaines de pairs, et la gestion des « mots de passe MD5 » pour tous ces pairs est un défi. En général, on ne complique pas ce défi en essayant de suivre des bonnes pratiques, comme le changement des mots de passe...

Et les problèmes techniques de BGP ? Comme il utilise TCP, il profite de ses mécanismes de sécurité (comme les numéros de séquence qui protègent des rejeux) et il hérite de ses faiblesses (voir l'excellent exposé de Convery & Franz à Black Hat, « "*BGP Vulnerability Testing : Separating Fact from FUD*" <http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-convery-franz-v2.pdf> », un modèle pour toute analyse de la sécurité). À plusieurs reprises, les fabricants de routeurs BGP ont dû mettre à jour leur logiciel pour traiter ces vulnérabilités.

À noter que le RFC ne mentionne pas du tout ce qui est probablement la plus grosse faiblesse de sécurité de BGP : le fait qu'on puisse authentifier le voisin, mais pas l'information qu'il envoie (ou relaie). Savoir que c'est bien mon pair `2001:db8:face:b00c::1` avec qui je parle, c'est très bien mais, lorsqu'il annonce une route vers `2001:db8:137::/48`, comment savoir s'il a le droit de le faire ou bien si Pakistan Telecom a fait mumuse <`https://www.bortzmeyer.org/pakistan-pirate-youtube.html`> ? Il n'existe pour l'instant aucun protocole standard pour cela.