RFC 7115 : RPKI-Based Origin Validation Operation

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 janvier 2014

Date de publication du RFC : Janvier 2014

https://www.bortzmeyer.org/7115.html

Le mécanisme de sécurisation des annonces de route pour BGP, connu sous le nom de RPKI+ROA <https://www.bortzmeyer.org/securite-routage-bgp-rpki-roa.html>, est normalisé depuis un an et il existe désormais un peu d'expérience opérationnelle. Que nous enseigne-t-elle ? Peut-on commencer à définir des « bonnes pratiques » de déploiement et de fonctionnement de la RPKI ?

Outre la publication des normes sur RPKI+ROA, comme le RFC 6480 [1], il existe plusieurs logiciels <https://www.bortzmeyer.org/rpki-tests.html> permettant de mettre en œuvre ces normes. On trouve encore peu de comptes-rendus d'expérience concrète avec la RPKI. En français, je vous recommande fortement l'excellent mémoire de master de Guillaume Lucas <http://www.guiguishow.info/wp-content/uploads/2013/09/RPKI-ROA/Rapport/Rapport-securiser-routage-internet.pdf>. Il faut dire qu'on en est encore aux débuts (ce qui explique la faible taille de ce nouveau RFC). Par exemple, il n'existe toujours pas de point de départ ("*trust anchor*") unique pour la validation, chaque RIR ayant son propre certificat, chacun étant une racine pour la validation. Le RFC estime qu'une racine unique <http://www.iab.org/documents/correspondence-reports-documents/docs2010/iab-statement-on-the-rpki/> finira par apparaître, ce dont je doute fort, en raison des nombreux conflits de gouvernance à ce sujet.

Notre nouveau RFC, ce RFC 7115, a une vision très optimiste : il affirme même que, déployée intelligemment et prudemment, RPKI+ROA entraînera peu de perturbations du système de routage de l'Internet. L'expérience récente de DNSSEC semble pourtant indiquer le contraire, à savoir qu'il n'est pas trivial de déployer de la sécurité cryptographique et qu'on casse parfois quelque chose en le faisant. À moins que les gens de BGP soient bien plus compétents que ceux du DNS ?

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6480.txt

Bon, assez de généralités, place aux recommandations concrètes. N'oubliez pas d'abord de lire les RFC RPKI+ROA <https://www.bortzmeyer.org/securite-routage-bgp-rpki-roa.html>. Puis, attaquons la section 3, le cœur de ce nouveau RFC. La RPKI est la base de données répartie (RFC 6481) stockant les autorisations de routage (les ROA, RFC 6482), les certificats (RFC 6484), les révocations, les enregistrements Ghostbuster (RFC 6493), etc. Elle a été conçue pour être distribuée de manière hiérarchique, des dépôts mondiaux nourrissant des dépôts plus locaux qui à leur tout nourriraient des dépôts très locaux. En pratique, cela ne semble pas encore très fréquent : l'architecture la plus courante semble être encore à deux étages, le dépôt du RIR et celui de l'opérateur. Notez qu'on peut parfaitement utiliser des dépôts et des caches validateurs situés dans un autre réseau que le sien, par exemple chez son transitaire : si on lui fait confiance pour router les paquets, on peut bien lui faire confiance pour gérer la RPKI, et s'appuyer sur son travail.

Le **cache validateur**, dans le réseau de l'opérateur, collecte toutes les données de la RPKI (en utilisant rsync) et les valide en utilisant un logiciel comme rcynic ou le validateur du RIPE-NCC <https://www.bortzmeyer.org/rpki-tests.html>. Les routeurs qui font confiance à ce cache iront alors chercher ces informations sur la validité des routes (typiquement en utilisant le protocole RTR du RFC 6810), sans avoir à faire de crypto eux-mêmes. Cela impose que les routeurs soient configurés uniquement avec des adresses de caches validateurs de confiance. Et que la connexion entre le routeur et le cache soit bien sécurisée, par exemple par SSH, particulièrement si le cache validateur est dans un AS différent. Pour des raisons de résilience, le cache doit être joignable sans avoir besoin de ressources extérieures, comme le DNS ou comme la RPKI elle-même (si le routeur devait valider la route vers le cache, on aurait un beau problème de "*bootstrap*"). Il vaut sans doute mieux que les routes entre le routeur et le cache validateur ne dépendent pas du tout de BGP. Et, toujours pour la résilience, un routeur devrait utiliser plusieurs caches (comme une machine a plusieurs résolveurs DNS).

La validation dans la RPKI est arborescente : on part d'un certificat d'une autorité de confiance, le TAL ("*Trust Anchor Locator*", cf. RFC 7730 et, oui, je simplifie parce que le TAL est en fait un pointeur vers le certificat), et on valide récursivement les certificats des titulaires de ressources (adresses IP, par exemple) à partir de ces TAL. Si le gérant d'un cache validateur met n'importe quoi comme TAL (des certificats mal ou pas gérés), toute la confiance s'écroule. Il faut donc bien choisir ses TAL ! Voici un contre-exemple, une installation de rcynic où ont été installées toutes les TAL possibles, test et production mêlées :

```
% ls etc/trust-anchors
afrinic.tal  lacnic.tal     testbed-apnicrpki.tal
apnic.tal  ripe-ncc-root.tal  testbed-arin.tal
bbn-testbed.tal  testbed-apnic.tal  testbed-ripe.tal
% cat etc/trust-anchors/ripe-ncc-root.tal
rsync://rpki.ripe.net/ta/ripe-ncc-ta.cer
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0URYSGqUz2m
yBsOzeW1jQ6NsxNvlLMyhWknvnl8NiBCs/T/S2XuNKQNZ+wBZxIgPPV
...
```

Le RFC insiste ensuite sur un des points les plus faibles de la RPKI : les délais de mise à jour ne sont spécifiés nulle part (voir aussi la section 6 du RFC, qui revient sur ce problème). Un cache validateur pourrait, par exemple, ne charger les données avec rsync que deux fois par semaine et travailler ainsi avec des données relativement anciennes, sans violer aucune norme technique. (La section 6 du RFC prétend que c'est la même chose avec le DNS, qui n'est que « synchrone à terme ». Mais c'est tout à fait faux : dans le DNS, les délais sont parfaitement connus et standards, le principal étant le TTL.) À l'intérieur de son réseau, un opérateur qui a plusieurs caches peut décider de les synchroniser très fréquemment : ce sont ses machines, il décide. Mais, à l'extérieur, même si ce RFC n'en parle pas, il n'est pas du tout sûr que les serveurs rsync des RIR résistent en cas de synchronisation trop agressive.

Après ces questions d'architecture de la validation, la section 3 de notre RFC continue avec la création des ROA ("*Route Origin Authorizations*", les objets signés qui authentifient la prétention d'un AS à être à

l'origine de l'annonce BGP d'un préfixe). D'abord, bien se rappeler qu'IP est arborescent : un ROA pour `192.0.2.0/24` peut concerner aussi `192.0.2.128/26`. Avant d'émettre un ROA couvrant un gros préfixe, il faut donc être sûr que tous les préfixes inférieurs qui auront besoin d'être routés ont déjà des ROA corrects. Sinon, toutes leurs annonces deviendront subitement invalides.

Les ROA disposent d'un attribut `maxLength` qui permet d'être un peu laxiste en cas d'annonce d'un sous-préfixe. Si le ROA concerne le préfixe `192.0.2.0/24-26` (avec 24 = longueur du préfixe, et 26 = longueur maximale des annonces), une annonce `192.0.2.128/26` sera acceptée (alors que `192.0.2.128/27` serait refusée). Mais c'est dangereux car une annonce plus spécifique (préfixe plus long) créera une entrée prioritaire dans la table de routage (IP donnant la priorité au préfixe le plus long), facilitant le détournement que RPKI+ROA avait pour mission d'empêcher. Le RFC conseille donc, d'abord que les logiciels de gestion de la RPKI mettent, par défaut, une longueur maximale égale à la longueur du préfixe (aucun sous-préfixe autorisé) et ensuite que `maxLength` soit utilisé avec précaution. Si on a juste trois sous-préfixes, il vaut sans doute mieux générer trois ROA pour eux, plutôt que d'affaiblir la sécurité en ayant une longueur maximale supérieure à la longueur du préfixe. (Ou un seul ROA contenant trois préfixes s'ils ont tous le même AS d'origine.)

On l'oublie souvent, mais un préfixe peut être « multi-origine », c'est-à-dire être annoncé par plusieurs AS. C'est même recommandé dans certains cas, par le RFC 6382. Dans ce cas, il faut faire autant de ROA qu'il y a d'AS d'origine possibles (un ROA peut contenir plusieurs préfixes mais pas plusieurs AS).

Enfin (il y a aussi plein d'autres recommandations que je n'ai pas reprises ici), le RFC recommande d'utiliser des outils automatiques pour superviser ses objets RPKI et, par exemple, avertir lors de l'expiration proche d'un certificat (comme on devrait le faire pour tous les certificats <`https://www.bortzmeyer.org/tester-expiration-certifs.html`>).

Au fait, à quel endroit du réseau doit-on installer les routeurs qui utiliseront la validation ? La section 4 se penche sur cette question. A priori, seuls les routeurs de bord, ceux situés au contact d'un autre AS, seront dans ce cas. Les autres routeurs font confiance aux routes distribuées en interne. On peut même déployer cette utilisation de la validation de manière incrémentale, un routeur après l'autre. Mais attention à la répartition du trafic : les premiers routeurs qui utiliseront la validation pourront refuser certaines routes, que les autres accepteront, menant à des déplacements de trafic d'un lien vers un autre.

J'ai dit que les routeurs qui utilisent la validation pourraient refuser certaines routes. Justement, que doit faire un routeur qui se connecte à un cache validateur lorsque ledit cache a marqué une route comme invalide (cf. RFC 6811 pour les différents résultats de validation) ? La section 5 du RFC décrit les politiques possibles. Le principe cardinal est « chaque opérateur décide ». Ce n'est pas le RFC qui décide de ce que doit faire le routeur.

L'approche radicale serait évidemment que le routeur rejette les routes invalides. Mais, si on se fie au déploiement d'autres techniques de sécurité comme DNSSEC, au début, il y aura plein d'erreurs par les opérateurs et bien des routes seront marquées à tort comme invalides. Rejeter ces routes dès le début serait sans doute une erreur. Il vaudrait sans doute mieux commencer par une approche moins violente, se contentant d'attribuer une priorité plus basse aux routes invalides.

Le problème est que cela permet des attaques avec des préfixes plus spécifiques, comme dans l'exemple plus haut. Soit un ROA `192.0.2.0/24-24` (le 24-24 indiquant que la longueur du préfixe et la longueur maximale admise sont la même, 24 bits). Un méchant annonce `192.0.2.128/26`. Cette annonce sera marquée comme invalide par le système RPKI+ROA. Mais si elle est quand même acceptée, même avec la priorité la plus basse, elle s'installera dans les tables de routage et, étant plus spécifique que

192.0.2.0/24, elle gagnera, et les paquets IP seront envoyés à l'endroit désigné par l'annonce du méchant. Bref, pas de solution idéale à ce dilemme, le RFC suggère néanmoins de ne pas accepter les routes invalides sans une bonne raison.

Et si le résultat de la validation est qu'aucun ROA couvrant cette annonce n'a été trouvé (le cas le plus fréquent aujourd'hui, où seule une minorité de préfixes sont signés) ? Le RFC recommande évidemment d'accepter ces routes. Ce n'est pas demain la veille qu'on pourra dire « quasiment tout le monde a des ROA, arrêtons de router les autres ».

Enfin, la section 6 couvre quelques recommandations diverses : avoir des mécanismes spéciaux pour les préfixes « importants », par exemple ceux des serveurs racine du DNS, afin d'être prévenu si la RPKI veut tout à coup les invalider, gérer les AS sur quatre octets du RFC 6793, et avoir des horloges à l'heure, puisque certificats et ROA peuvent expirer.

Le but de RPKI+ROA est d'améliorer la sécurité du routage Internet, notamment face à des détournements comme celui de YouTube par Pakistan Telecom <https://www.bortzmeyer.org/pakistan-pirate-youtube.html>. Cet objectif est-il atteint ? La section 7 de notre RFC appelle à la prudence : les ROA, comme leur nom l'indique, ne protègent que l'origine d'une annonce (l'AS le plus à droite dans un chemin d'AS). Dans quasiment tous les détournements accidentels (comme celui de Pakistan Telecom cité plus tôt), l'origine est changée et RPKI+ROA le détecterait donc. Mais, si le détournement n'est pas un accident mais une attaque délibérée, on peut penser que le méchant va soigner ses annonces BGP. Il mettra probablement la vraie origine et ne bricolera que la suite du chemin. Pour l'instant, la RPKI n'a pas de mécanisme pour faire face à cela. De même, toute manipulation du chemin faite après l'émission par l'AS d'origine ne sera pas détectée, tant que l'origine est respectée.

Ah, et une autre limite de ce système : BGP ne fait que transmettre des annonces de route, que les routeurs ne respectent pas forcément (« les données ne suivent pas forcément le contrôle »). Donc, même si BGP est parfaitement sécurisé, des routeurs défaillants ou piratés peuvent encore faire échouer la sécurité.

Une bonne lecture **concrète** en français sur le système RPKI+ROA ? « Sécuriser le routage sur Internet <http://www.guiguishow.info/2013/09/18/securiser-le-routage-sur-internet/> » de Guillaume Lucas. Sinon, il existe une liste de diffusion des opérateurs RPKI <https://lists.rpki.net/mailman/listinfo/rpki> mais avec très peu de trafic.

Merci à Guillaume Lucas pour sa relecture très attentive, mais cela ne veut pas dire qu'il est d'accord avec tout.