RFC 7682 : IRR & Routing Policy Configuration Considerations

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 décembre 2015

Date de publication du RFC : Décembre 2015

https://www.bortzmeyer.org/7682.html

On le sait, il n'y a pas de chef <https://www.bortzmeyer.org/qui-est-le-numero-1.html> de l'Internet. Personne ne reste dans un bureau toute la journée avec pour mission l'envoi de règles qui seront immédiatement appliquées par tous les acteurs de l'Internet. C'est même le cas des fonctions les plus essentielles de l'Internet comme l'échange des tables de routage. Tout l'Internet ne tient que parce que les opérateurs sont d'accord entre eux pour s'échanger l'information qui se retrouvera dans ces tables « pour joindre le préfixe 2001:db8:fe2::/48, passez par moi, je connais un chemin que j'ai appris des AS 64499 puis 429496613 ». Cet accord de principe se double d'un accord technique sur le protocole à utiliser : BGP, normalisé dans le RFC 4271 [1]. Et au-delà, rien, aucun accord : chaque opérateur est libre de sa politique et notament de ce qu'il accepte ou refuse. Un opérateur peut refuser les préfixes plus spécifiques que /32, par exemple. Chacun est maître chez soi. En pratique, bien des opérateurs refusent les préfixes qui ne sont pas dans un IRR. C'est quoi, un IRR ? Qui les gère ? Qui décide de ce qu'on met dans un IRR ?. Ce nouveau RFC explore la question.

Un IRR ("*Internet Routing Registry*") est une base de données stockant des préfixes d'adresses IP et des informations de politique associées comme, par exemple, le ou les AS qui sont autorisés à annoncer ce préfixe en BGP ou bien les communautés BGP (RFC 1997) utilisées. Le RFC 1787, dans sa section 7, décrit l'importance de **partager l'information** entre les opérateurs. Certes, chacun est maître chez lui, mais tout serait plus simple si chacun partageait avec les autres, pour limiter le risque de décisions malencontreuses.

Mais les IRR ont des problèmes et des limites (introduction en section 3). Première catégorie de problèmes, l'exactitude et l'intégrité des données stockées (section 4). Comme tous les registres utilisés sur l'Internet, les IRR sont plein de données fausses ou dépassées. Les personnes qui mettent ces données

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4271.txt

dans les registres n'ont guère de motivations pour mettre des données correctes et encore moins pour les mettre à jour. Les données sont donc souvent erronées.

En outre, il n'existe pas de moyen largement déployé de vérifier ces données dès le départ. Si Pakistan Telecom <https://www.bortzmeyer.org/pakistan-pirate-youtube.html> met dans un IRR qu'ils sont autorisés à annoncer le préfixe de YouTube <https://www.bortzmeyer.org/pakistan-pirate-youtube.html>, comment vérifier qu'ils ont le droit de le faire ? Le langage utilisé pour les IRR, RPSL (normalisé dans le RFC 2622) le permet techniquement mais en l'absence d'un mécanisme de signature cryptographique entre le préfixe et l'AS, cela ne permet pas de s'assurer de l'authenticité du contenu de l'IRR. Le RFC note à juste titre que cette absence de mécanisme de signature vient en partie d'un manque d'intérêt de la communauté des opérateurs : tout le monde se plaint des détournements BGP mais personne n'est prêt à faire les considérables efforts qu'il faudrait pour tout certifier.

Bien sûr, certains IRR ont des mesures de sécurité (par exemple, dans la base RIPE, seul le titulaire d'un préfixe peut ajouter des routes pour ce préfixe, cf. RFC 2725 et c'est une obligation contractuelle, cf. ripe-637 <http://www.ripe.net/ripe/docs/ripe-637>) mais cela ne s'étend pas aux autres IRR, qui ne peuvent pas vérifier que cette sécurité a été respectée. Difficile donc de savoir quelle politique de sécurité a été utilisée par un IRR donné.

Même quand les données sont correctes, en l'absence de mécanismes de vérification fondés sur la cryptographie, on peut parfois se retrouver avec des données fausses alors qu'elles étaient justes au départ.

Il existe de nombreuses propositions pour créer de tels systèmes de vérification et de certification (TASRS <http://techreports.verisignlabs.com/tr-lookup.cgi?trid=1130009&rev=1>, HotNets-X <http://dl.acm.org/citation.cfm?id=2070574> et bien sûr la RPKI <https://www.bortzmeyer.org/securite-routage-bgp-rpki-roa.html>) mais aucune n'a encore un déploiement suffisant.

Comme indiqué plus haut, une partie du problème est le manque de motivation : le titulaire d'une ressource Internet (un préfixe IP, par exemple) peut avoir un intérêt direct à ajouter une information dans un IRR (par exemple parce que son transitaire est strict, et lui impose cet ajout, avant d'accepter l'annonce BGP du titulaire) mais il a rarement de raison objective de le maintenir à jour ou, surtout, de le détruire quand il n'est plus utilisé. Certains acteurs bâtissent leurs filtres d'annonces BGP à partir des IRR, ce qui fait qu'une information manquante se paie d'un manque de visibilité BGP. Mais il n'y a aucune pénalité à avoir de l'information en trop. Si un client, titulaire d'un préfixe IP, passe d'un transitaire strict, qui lui imposait une entrée dans un IRR (par exemple un objet `route` dans la base RIPE-NCC) vers un transitaire laxiste qui n'impose rien, le client n'a aucune motivation (à part la pureté de l'IRR, qui n'intéresse qu'une poignée de barbus fanatiques utilisant OpenBSD) à changer l'ancienne entrée. Il la laissera probablement telle quelle. Le nouveau transitaire, qui ignore l'IRR, n'en tiendra pas compte.

Une des rares motivations concrètes qui pourrait mener à un nettoyage des IRR est le manque de mémoire dans les routeurs : les filtres pourraient devenir trop gros, si personne ne nettoie jamais. Heureusement pour les utilisateurs des routeurs, mais malheureusement pour la pureté de l'IRR, les routeurs PE modernes ont en général assez de mémoire dans leur module de routage ("*control plane*") pour que cela ne soit plus un problème.

Autre problème avec les IRR, le fait que leur modèle de sécurité soit en général basé sur « le titulaire peut tout, les autres ne peuvent rien ». Si une information est dépassée, un tiers qui s'en aperçoit ne peut

rien faire, seul le titulaire est autorisé à la modifier. Pas moyen de nettoyer pour compenser la négligence d'un titulaire. C'est pour cela que le RFC 2725 avait prévu le mécanisme `auth-override`, qui semble n'avoir jamais été mis en œuvre.

Pour accéder aux données des IRR, on utilise souvent whois. Un exemple (une liste des IRR <`http://www.irr.net/docs/list.html`> existe en ligne) :

```
%  whois -h whois.radb.net 192.134.5.10
route:          192.134.5.0/24
descr:          NIC-FR-SITE-TH3
origin:         AS2485
mnt-by:         FR-NIC-MNT
mnt-lower:      FR-NIC-MNT
mnt-routes:     FR-NIC-MNT
changed:        jean-philippe.pick@nic.fr 20110214
remarks:        Peering:   peering@nic.fr
remarks:        NOC:       noc@nic.fr
remarks:        Info:      http://www.nic.fr/
source:         RIPE
remarks:        ****************************
remarks:        * THIS OBJECT IS MODIFIED
remarks:        * Please note that all data that is generally regarded as personal
remarks:        * data has been removed from this object.
remarks:        * To view the original object, please query the RIPE Database at:
remarks:        * http://www.ripe.net/whois
remarks:        ****************************
```

On y voit que seul l'AS 2485 est autorisé à être l'origine d'une annonce pour le préfixe `192.134.5.0/24`.

Un plaisir des IRR existants est qu'on n'a pas de moyen simple, étant donné une ressource Internet (par exemple un préfixe IP), de savoir quel IRR utiliser. Les clients logiciels existants doivent utiliser diverses heuristiques. Par exemple GNU whois <`https://github.com/rfc1036/whois`> utilise un fichier de configuration local - `/etc/whois.conf`, des règles incluses dans le client - fichier `ip_del_list` dans le source, et compte sur le serveur whois de l'ARIN pour le rediriger, pour les ressources inconnues (cette fonction de redirection étant une extension absente du protocole officiel whois, normalisé dans le RFC 3912).

À noter que whois ne fournit aucune confidentialité : le protocole est du simple TCP en clair sur le port 43. (Le protocole RDAP <`https://www.bortzmeyer.org/weirds-rdap.html`>, lui, fournit un mécanisme de chiffrement via TLS.)

La section 5 de notre RFC se penche sur le fonctionnement interne des IRR et notamment sur leur synchronisation. Il est fréquent en effet que les IRR copient les données gérées par d'autres IRR. Cela se fait en général avec le protocole NRTM (protocole ressemblant à whois, et qui est peu ou pas documenté). Ce protocole n'a aucun mécanisme de sécurité (en pratique, la seule protection est la limitation d'accès à certaines adresses IP) et est peu robuste (des erreurs de synchronisation ont déjà eu lieu). Il fonctionne en mode "*pull*" et il n'y a pas de mécanisme de notification pour indiquer la présence de données récentes. Une autre façon de synchroniser des IRR est de télécharger un fichier plat, par exemple en FTP, qu'on applique ensuite à la copie locale (ce qui facilite l'application de modifications locales). Ce téléchargement et cette synchronisation sont typiquement faits toutes les 24 heures (une période qui date de l'époque où les réseaux étaient plus lents qu'aujourd'hui, et qui n'a pas toujours été réévaluée depuis), ce qui fait que les données ne sont pas forcément de la première fraicheur. On notera que le logiciel irrd <`http://www.irrd.net/`> synchronise toutes les dix minutes, par défaut.

Un standard existe pour cette réplication d'IRR, le RFC 2769, mais il ne semble pas avoir jamais eu le moindre succès, peut-être parce qu'il dépend du standard de sécurité RFC 2725, également peu ou pas répandu.

Les filtres effectivement utilisés par les routeurs de l'opérateur sont créés à partir d'un IRR, et cette création implique une autre étape (et les délais associés). Typiquement, l'opérateur utilise un logiciel comme IRRtoolset <http://irrtoolset.isc.org/>, qui va traduire les objets trouvés dans l'IRR en règles pour un type de routeur donné. Ce logiciel ne tourne pas en permanence. Conséquence pratique : un changement dans l'IRR a intérêt à ne pas être urgent ! Il n'y a aucun moyen de forcer les opérateurs à télécharger tout de suite les nouvelles données, et à faire tourner la « moulinette » qui va produire les règles pour les routeurs. Si on est un client payant de l'opérateur, on peut parfois les appeler et obtenir que cette opération soit faite de suite, mais ce n'est pas toujours le cas.

Idéalement, lorsqu'une politique change (par exemple un AS annonce un nouveau préfixe), il « suffit » de changer l'IRR, d'attendre les copies dans les autres IRR, puis l'exportation des données depuis les IRR vers les routeurs. Mais le protocole BGP ne permettait pas forcément de changer les caractéristiques d'une session en vol (section 6 de notre RFC) et exigeait une opération de réinitialisation, pour que le routeur accepte les nouvelles routes. Pendant cette opération, tout ou partie du travail normal du routeur était mis en attente... Résultat, cette opération (`clear ip bgp neighbor...` sur les Cisco...) ne pouvait se faire que pendant les périodes de maintenance. C'est un bon exemple d'un cas où les « bonnes pratiques » (n'accepter que les préfixes décrits dans les IRR) sont irréalisables dans un environnement de production (cf. section 8). Le yakafokon ne marche pas bien dans le monde des réseaux.

Heureusement, ce problème spécifique appartient largement au passé : les extensions à BGP des RFC 2918 et RFC 7313 ont largement supprimé ces obligations.

Il y a aussi les limites inhérentes à certaines mises en œuvre de BGP (section 7 du RFC). Par exemple, au milieu des années 1990, la plupart des routeurs ne permettaient pas de modifier les listes de préfixes acceptés de manière incrémentale : il fallait effacer la liste, puis installer une nouvelle liste, créant ainsi une fenêtre de vulnérabilité pendant laquelle des routes pouvaient fuir. Là aussi, ce problème a largement disparu avec les progrès des routeurs (voir aussi les sections 1 et 8, qui expliquent bien que certains problèmes historiques d'utilisation des IRR sont désormais du passé).

Rien n'est gratuit en informatique : le stockage des listes de préfixes IP acceptés nécessite de la mémoire et, pendant longtemps, les routeurs disposaient de trop peu de mémoire (une NVRAM bien limitée, et très lente en écriture). Les routeurs modernes ont des mémoires flash ou SSD, voire des disques durs et ont donc une bien meilleure capacité. D'un autre côté, les exigences ont crû et la taille de certaines configurations peut toujours poser des défis aux mémoires des routeurs (voir « "*NTT BGP Configuration Size and Scope*" <http://iepg.org/2014-03-02-ietf89/ietf89_iepg_jmauch.pdf> »).

Dernier problème, l'envoi aux routeurs des changements. Autrefois, il n'y avait aucun standard en ce sens. Chaque routeur avait son CLI et il fallait générer depuis l'IRR les quelques lignes de commandes qui allaient changer la configuration du routeur, lui faisant accepter de nouveaux préfixes, ou bien refuser ceux qui étaient acceptés. Le routeur recevait ensuite en telnet, puis SSH, l'ordre de charger ces quelques lignes en TFTP ou FTP, puis, plus tard, en SCP. On pouvait alors activer la nouvelle configuration.

De nos jours, il existe une norme pour cela, NETCONF (RFC 6241). On génère toujours des données depuis l'IRR, puis on utilise NETCONF pour les charger. Les données issues de la RPKI peuvent, elles, être envoyées en RTR (RFC 6810) mais cela ne concerne pas tout le reste de la configuration du routeur.

Un petit mot de sécurité pour finir (section 9 du RFC). Le but des IRR étant d'influencer le routage à distance, ces IRR sont donc des ressources sensibles : si quelqu'un peut pirater un IRR et y injecter de fausses données, il peut perturber le routage Internet. Si ce problème est trop fréquent, les opérateurs pourraient en arriver à ne plus utiliser les IRR. Une gestion rigoureuse de leur sécurité est donc cruciale.

Voilà, si vous voulez en savoir davantage sur les IRR, il y a bien sûr l'incontournable site <http://www.irr.net/>.