RFC 6481 : A Profile for Resource Certificate Repository Structure

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 4 février 2012

Date de publication du RFC : Février 2012

https://www.bortzmeyer.org/6481.html

La RPKI, cette infrastructure de certificats utilisée pour sécuriser le routage sur l'Internet <https://www.bortzmeyer.org/securite-routage-bgp-rpki-roa.html>, repose sur un certain nombre d'objets (certificats mais aussi révocations et surtout les objets signés qui expriment les autorisations, les ROA). Ce RFC décrit comment ils sont stockés dans les **points de publication** à partir desquels ils sont distribués au monde entier. Cela permet de consolider facilement les données obtenues auprès de plusieurs points de publication. (Il n'est pas obligatoire qu'une copie de la RPKI soit complète.)

Rappelez-vous en effet que les routeurs n'accèdent pas aux informations de la RPKI lorsqu'ils en ont besoin (ce qui ferait dépendre le routeur de serveurs extérieurs) mais à l'avance en chargeant dans un validateur la totalité des informations, que le validateur va ensuite vérifier avant de passer au routeur. Comment se présentent ces informations ? Chacun fait comme il veut mais il existe une forme recommandée, celle de notre RFC 6481 [1].

Dans ce format recommandé, chaque objet va être un fichier. Ces fichiers vont être placés dans une arborescence de répertoires, référencée par un URI (typiquement de type `rsync:`).

Dans les exemples ci-dessous, j'utiliserai une copie locale de la RPKI obtenue au RIPE-NCC, avec les outils logiciels de ce même RIPE-NCC (voir mon article sur ces outils <https://www.bortzmeyer.org/rpki-tests.html>).

À chaque point de publication (typiquement, chaque AC, c'est-à-dire en pratique chaque opérateur réseau), on trouve un **manifeste** qui liste les objets publiés. Le fichier a l'extension `.mft`, le type MIME `application/rpki-manifest` (cf. section 7.1 du RFC) et son nom est un condensat cryptographique de la clé publique de l'AC (en suivant par exemple la section 2.1 du RFC 4387). Voici le contenu d'un tel manifeste, à un petit point de publication (seulement quatre ROA) :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6481.txt`

```
% certification-validator -print -file Sxe2vn2-fqZLB5Q578V0Vzf_RfQ.mft
Serial: 39571085
Subject: CN=779caa1de4e5938149c831b097a86ee561bf12f3
Not valid before: 2011-10-31T03:42:07.000Z
Not valid after:  2011-11-01T03:42:07.000Z
...
Filenames and hashes:
    1c_GrZeKKzG5b9BI9Iu4fEEG-QU.roa 20aade4dd3e303103b7facb0d462d89e0cc74753c99bf35a8ff28d16e48f7e6a
    1sOqx9K6uCNTcCTMIe7f5P1bZX0.roa 7e9c4c0be54d180b05743abbd6163f0a812c43d4db59fe0550eb2446ceedbb7e
    4nKGhPF4_7JKjwwBrTCXgE_4YYg.roa f1f55d72eb6e66c36b6312c4c6484a83156d32b7e01067a6a3449327095e409f
    Sxe2vn2-fqZLB5Q578V0Vzf_RfQ.crl 39ec393db6b9a8ec6289304a5c1982ab17ef3bfc5feaffaf4a30174af9054966
    TZw7cXuBc5b9BOlBHqCtAEgTZ-w.roa bd7aa3efae357a8591a7b5d708ccb509ad30f791063ea6fe9aa53730d95cb680
```

On trouve également des révocations, dans des fichiers ayant l'extension `.crl` (et le type MIME `application/pkix-crl` qui vient du RFC 2585) :

```
% certification-validator -print -file Sxe2vn2-fqZLB5Q578V0Vzf_RfQ.crl
CRL type: X.509
CRL version: 2
Issuer: CN=Sxe2vn2-fqZLB5Q578V0Vzf_RfQ
Authority key identifier: Sxe2vn2-fqZLB5Q578V0Vzf_RfQ=
Number: 790
This update time: 2011-10-31T03:42:07.000Z
Next update time: 2011-11-01T03:42:07.000Z
Revoked certificates serial numbers and revocation time:
    32399600 2011-09-10T05:40:48.000Z
```

On trouve bien sûr les certificats, encodés au format DER et stockés dans des fichiers ayant l'extension `.cer` et le type MIME `application/pkix-cert` :

```
% certification-validator -print -file Fx2dYP3STZSiTUjLGrM8oDkK4Ys.cer
Serial: 18282008
Subject: CN=Fx2dYP3STZSiTUjLGrM8oDkK4Ys
Not valid before: 2011-01-12T02:39:31.000Z
Not valid after:  2012-07-01T00:00:00.000Z
Resources: 46.254.48.0/21
```

Notez qu'il s'agit d'un format standard donc OpenSSL peut l'afficher aussi (pour voir les ressources gérées, ici `46.254.48.0/21`, il faut un OpenSSL compilé avec l'option RFC 3779) :

```
% openssl x509 -inform DER -text -in Fx2dYP3STZSiTUjLGrM8oDkK4Ys.cer
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 18282008 (0x116f618)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=u75at9r0D4JbJ3_SFkZXD7C5dmg
        Validity
            Not Before: Jan 12 02:39:31 2011 GMT
            Not After : Jul  1 00:00:00 2012 GMT
        Subject: CN=Fx2dYP3STZSiTUjLGrM8oDkK4Ys
...
            sbgp-ipAddrBlock: critical
                IPv4:
                  46.254.48.0/21
```

Et, bien sûr, le dépôt contient des autorisations d'annonces de route, les **ROA** ("*Route Origination Authorization*"), normalisés dans le RFC 6482. Elles sont contenues dans des fichiers d'extension `.roa` et ont le type MIME `application/rpki-roa` :

```
% certification-validator -print -file FiUGfSV6BgZqFlJnQ7SSq_700w8.roa
Content type: 1.2.840.113549.1.9.16.1.24
Signing time: 2011-04-11T15:07:23.000Z
ASN: AS56595
Prefixes:
    46.226.0.0/21
    2a00:a600::/32
```

Le ROA ci-dessus autorise l'AS 56595 à annoncer des routes pour `46.226.0.0/21` et `2a00:a600::/32`. À noter que la suite logicielle rcynic <`http://rpki.net/`> permet également l'affichage de ROA :

```
% print_roa FiUGfSV6BgZqFlJnQ7SSq_700w8.roa
Certificates:   1
CRLs:           0
SignerId[0]:    16:25:06:7d:25:7a:06:06:6a:16:52:67:43:b4:92:ab:fe:f4:d3:0f [Matches certificate 0] [signingTime
eContentType:   1.2.840.113549.1.9.16.1.24
version:        0 [Defaulted]
asID:           56595
 addressFamily: 1
     IPaddress: 46.226.0.0/21
 addressFamily: 2
     IPaddress: 2a00:a600::/32
```

La section 3 décrit ensuite les caractéristiques souhaitées du serveur qui va publier toutes ces informations. Il doit notamment être très fiable : même si le serveur n'est évidemment pas accédé de manière synchrone, à chaque mise à jour BGP, il doit néanmoins être accessible la grande majorité du temps, puisque toute panne empêchera les mises à jour de la base et aura potentiellement des conséquencees opérationnelles sur le routage. C'est une tâche nouvelle, par exemple pour les RIR, qui n'avaient pas de rôle opérationnel avant.

Ce serveur doit utiliser rsync (cf. RFC 5781) et peut aussi utiliser d'autres protocoles en prime. Grâce à rsync, la commande :

```
% rsync -av rsync://rpki.ripe.net/repository /var/RPKI
```

va vous récupérer tous les objets décrits ici, tels qu'ils sont distribués par le RIPE-NCC, et les mettre dans le répertoire `/var/RPKI`. rsync travaillant de manière incrémentale, seules les nouveautés seront chargées, pas la totalité de la base (sauf la première fois).

Cette commande peut donc être placée dans le crontab de la machine qui fera la validation. Toutefois, la section 3 recommande des techniques plus subtiles, à base de copie dans un dépôt temporaire puis de renommage (ce que font automatiquement les outils de rcynic), avant d'éviter une incohérence si le rsync est interrompu et qu'on n'a récupéré qu'une partie de la base.

Cette copie locale de la RPKI dans le validateur (machine qui sera proche des routeurs qui l'interrogeront) permettra de fonctionner même en cas de panne des serveurs de distribution (cf. section 5).

Attention à la sécurité, nous avertit la section 6. Les informations de la RPKI sont publiques, il n'y a donc pas besoin de confidentialité, mais il est essentiel que leur intégrité soit préservée. Le protocole rsync utilisé plus haut ne fournit aucune garantie en ce sens. En fait, la protection des données est complètement assurée par les signatures faites avec les clés contenues dans les certificats. Il ne faut donc **pas** utiliser les informations RPKI récupérées avant d'avoir validé (vérifié les signatures). C'est ce que fait automatiquement rcynic en déplaçant les données validées dans un répertoire nommé `authenticated` :

```
% pwd
/var/rcynic/data

% ls -lt
total 12
lrwxrwxrwx  1 rcynic rcynic   34 Oct 31 15:42 authenticated -> authenticated.2011-10-31T13:36:08Z
lrwxrwxrwx  1 rcynic rcynic   34 Oct 31 15:42 authenticated.old -> authenticated.2011-10-31T11:36:11Z
drwxr-xr-x 15 rcynic rcynic 4096 Oct 31 15:42 authenticated.2011-10-31T13:36:08Z
drwxr-xr-x 15 rcynic rcynic 4096 Oct 31 14:31 authenticated.2011-10-31T11:36:11Z
drwxr-xr-x 17 rcynic rcynic 4096 Oct 31 11:31 unauthenticated
```

Les extensions des différents fichiers stockés figurent dans un registre IANA <https://www.iana.org/assignments/rpki/rpki.xml#name-schemes>.