Secure Inter-Domain Routing
Internet-Draft
Intended status: Best Current Practice
Expires: November 14, 2015

D. Mandelberg
BBN Technologies
May 13, 2015

Simplified Local internet nUmber Resource Management with the RPKI
draft-dseomn-sidr-slurm-02

Abstract

The Resource Public Key Infrastructure (RPKI) is a global authorization infrastructure that allows the holder of Internet Number Resources (INRs) to make verifiable statements about those resources. Network operators, e.g., Internet Service Providers (ISPs), can use the RPKI to validate BGP route origination assertions. In the future, ISPs also will be able to use the RPKI to validate the path of a BGP route. Some ISPs locally use BGP with private address space or private AS numbers (see RFC6890). These local BGP routes cannot be verified by the global RPKI, and SHOULD be considered invalid based on the global RPKI (see RFC6491). The mechanisms described below provide ISPs with a way to make local assertions about private (reserved) INRs while using the RPKI's assertions about all other INRs.

Table of Contents

1. Introduction

The Resource Public Key Infrastructure (RPKI) is a global authorization infrastructure that allows the holder of Internet Number Resources (INRs) to make verifiable statements about those resources. For example, the holder of a block of IP(v4 or v6) addresses can issue a Route Origination Authorization (ROA) [RFC6482] to authorize an Autonomous System (AS) to originate routes for that block.

Internet Service Providers (ISPs) can then use the RPKI to validate BGP routes. (Validation of the origin of a route is described in [RFC6483], and validation of the path of a route is described in [I-D.ietf-sidr-bgpsec-overview].) However, some ISPs locally use BGP with private address space ([RFC1918], [RFC4193], [RFC6598]) or private AS numbers ([RFC1930], [RFC6996]). These local BGP routes cannot be verified by the global RPKI, and SHOULD be considered invalid when using the RPKI. For example, [RFC6491] recommends the creation of ROAs that would invalidate routes for reserved and unallocated address space.

This document specifies two new mechanisms to enable ISPs to make local assertions about some INRs while using the RPKI's assertions about all other INRs. These mechanisms support the second and third use cases in [I-D.ietf-sidr-lta-use-cases]. The second use case describes use of [RFC1918] addresses or use of public address space not allocated to the ISP that is using it. The third use case describes a situation in which an ISP publishes a variant of the RPKI hierarchy (for its customers). In this variant some prefixes and/or AS numbers are different from what the RPKI repository system presents to the general ISP population. The result is that routes for consumers of this variant hierarchy will be re-directed (via routing).

Both mechanisms are specified in terms of abstract sets of assertions. For Origin Validation [RFC6483], an assertion is a tuple of {IP prefix, prefix length, maximum length, AS number} as used by rpki-rtr version 0 [RFC6810] and version 1 [I-D.ietf-sidr-rpki-rtr-rfc6810-bis]. For BGPsec [I-D.ietf-sidr-bgpsec-overview], an assertion is a tuple of {AS number, subject key identifier, router public key} as used by rpki-rtr version 1. Output Filtering, described in Section 2, filters out any assertions by the RPKI about locally reserved INRs. Locally Adding Assertions, described in Section 3, adds local assertions about locally reserved INRs. The combination of both mechanisms is described in Section 5.

To ensure local consistency, the effect of SLURM MUST be atomic. That is, the output of the relying party must be either the same as if SLURM were not used, or it must reflect the entire SLURM configuration. For an example of why this is required, consider the case of two local routes for the same prefix but different origin AS numbers. Both routes are configured with Locally Adding Assertions. If neither addition occurs, then both routes could be in the unknown state [RFC6483]. If both additions occur then both routes would be in the valid state. However, if one addition occurs and the other does not, then one could be invalid while the other is valid.

In general, the primary output of an RPKI relying party is the data it sends to routers over the rpki-rtr protocol. The rpki-rtr protocol enables routers to query a relying party for all assertions it knows about (Reset Query) or for an update of only the changes in assertions (Serial Query). The mechanisms specified in this document are to be applied to the result set for a Reset Query, and to both the old and new sets that are compared for a Serial Query. Relying party software MAY modify other forms of output in comparable ways, but that is outside the scope of this document.

This document is intended to supersede [I-D.ietf-sidr-ltamgmt] while focusing only on local management of private INRs. Another draft [I-D.kent-sidr-suspenders] focuses on the other aspects of local management.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Validation Output Filtering

To prevent the global RPKI from affecting routes with locally reserved INRs, a relying party may be locally configured with a list of IP prefixes and/or AS numbers that are used locally, and taken from reserved INR spaces. Any Origin Validation assertions where the IP prefix is equal to or subsumed by a locally reserved IP prefix, are removed from the relying party’s output. Any Origin Validation assertions where the IP prefix contains a locally reserved IP prefix are removed; the relying party software SHOULD issue a warning when this action is taken. (Note that an Origin Validation assertion is not removed due to its AS number matching a locally reserved AS number.) Any BGPsec assertion where the AS number is equal to a locally reserved AS number is removed from the relying party’s output.

3. Locally Adding Assertions

Each relying party is locally configured with a (possibly empty) list of assertions. This list is added to the relying party’s output.

4. Configuring SLURM

Relying party software SHOULD support the following configuration format for Validation Output Filtering and Locally Adding Assertions. The format is defined using the Augmented Backus-Naur Form (ABNF) notation and core rules from [RFC5234] and the rules <IPv4address> and <IPv6address> from Appendix A of [RFC3986]. See Appendix A for an example SLURM file.

A SLURM configuration file, <SLURMFile>, consists of a head and a body. The head identifies the file as a SLURM configuration file, specifies the version of SLURM for which the file was written, and optionally contains other information described below. The body contains the configuration for Validation Output Filtering and Locally Adding Assertions.

```
   SLURMFile = head body

   head = firstLine *(commentLine / headLine)

   body = *(commentLine / bodyLine)

   firstLine = %x53.4c.55.52.4d SP "1.0" EOL ; "SLURM 1.0"

   commentLine = *WSP [comment] EOL

   headLine = *WSP headCommand [ 1*WSP [comment] ] EOL

   bodyLine = *WSP bodyCommand [ 1*WSP [comment] ] EOL

   comment = "#" *(VCHAR / WSP)

   EOL = CRLF / LF
```

The head may specify a target. If present, the target string identifies the environment in which the SLURM file is intended to be used. The meaning of the target string, if any, is determined by the user. If a target is present, a relying party SHOULD verify that that the target is an acceptable value, and reject the SLURM file if the target is not acceptable. For example, the relying party could be configured to accept SLURM files only if they do not specify a target, have a target value of "hostname=rpki.example.com", or have a target value of "as=65536". If more than one target line is present, all targets must be acceptable to the RP.

```
   headCommand = target

   target =
      %x74.61.72.67.65.74 1*WSP ; "target"
      1*VCHAR
```

The body contains zero or more configuration lines for Validation Output Filtering and Locally Adding Assertions. Each <del> command specifies an INR to use for Validation Output Filtering. Each <add> command specifies an assertion to use for Locally Adding Assertions.

```
   bodyCommand = add / del

   add =
      %x61.64.64 1*WSP ; "add"
      addItem

   del =
      %x64.65.6c 1*WSP ; "del"
```

```
      delItem

   addItem = addItemPrefixAS / addItemASKey

   ; Add a mapping from a prefix and max length to an AS number.
   addItemPrefixAS =
      %x6f.72.69.67.69.6e.61.74.69.6f.6e 1*WSP ; "origination"
      IPprefixMaxLen 1*WSP
      ASnum

   ; Add a mapping from an AS number to a router public key.
   addItemASKey =
      %x62.67.70.73.65.63 1*WSP ; "bgpsec"
      ASnum 1*WSP
      RouterSKI 1*WSP
      RouterPubKey

   delItem = delItemPrefix / delItemAS

   ; Filter prefix-AS mappings, using the given prefix
   delItemPrefix =
      %x6f.72.69.67.69.6e.61.74.69.6f.6e 1*WSP ; "origination"
      IPprefix

   ; Filter AS-key mappings for the given AS
   delItemAS =
      %x62.67.70.73.65.63 1*WSP ; "bgpsec"
      ASnum

   IPprefix = IPv4prefix / IPv6prefix

   IPprefixMaxLen = IPv4prefixMaxLen / IPv6prefixMaxLen

   IPv4prefix = IPv4address "/" 1*2DIGIT
   IPv6prefix = IPv6address "/" 1*3DIGIT

   ; In the following two rules, if the maximum length component is
   ; missing, it is treated as equal to the prefix length.
   IPv4prefixMaxLen = IPv4prefix ["-" 1*2DIGIT]
   IPv6prefixMaxLen = IPv6prefix ["-" 1*3DIGIT]

   ASnum = 1*DIGIT

   ; This is the Base64 [RFC4648] encoding of a router certificate’s
   ; Subject Key Identifer, as described in
   ; [I-D.ietf-sidr-bgpsec-pki-profiles] and [RFC6487]. This is the
   ; value of the ASN.1 OCTET STRING without the ASN.1 tag or length
   ; fields.
```

```
RouterSKI = Base64

; This is the Base64 [RFC4648] encoding of a router public key's
; subjectPublicKeyInfo value, as described in
; [I-D.ietf-sidr-bgpsec-algs]. This is the full ASN.1 DER encoding
; of the subjectPublicKeyInfo, including the ASN.1 tag and length
; values of the subjectPublicKeyInfo SEQUENCE.
RouterPubKey = Base64

Base64 = 1*(ALPHA / DIGIT / "+" / "/") 0*2"="
```

An implementation MAY support the concurrent use of multiple SLURM files. In this case, the resulting inputs to Validation Output Filtering and Locally Adding Assertions are the respective unions of the inputs from each file. The typical use case for multiple files is when the files have distinct scopes. For example, an organization may belong to two separate networks that use different private-use IP prefixes and AS numbers. To detect conflict between multiple SLURM files, a relying party SHOULD issue a warning in the following cases:

1. There may be conflicting changes to Origin Validation assertions if there exists an IP address X and distinct SLURM files Y,Z such that X is contained by any prefix in any <addItemPrefixAS> or <delItemPrefix> in file Y and X is contained by any prefix in any <addItemPrefixAS> or <delItemPrefix> in file Z.

2. There may be conflicting changes to BGPsec assertions if there exists an AS number X and distinct SLURM files Y,Z such that X is used in any <addItemASKey> or <delItemAS> in file Y and X is used in any <addItemASKey> or <delItemAS> in file Z.

5. Combining Mechanisms

In the typical use case, a relying party uses both output filtering and locally added assertions. In this case, the resulting assertions MUST be the same as if output filtering were performed before locally adding assertions. I.e., locally added assertions MUST NOT be removed by output filtering.

If a relying party chooses to use both SLURM and Suspenders [I-D.kent-sidr-suspenders], the SLURM mechanisms MUST be performed on the output of Suspenders.

6. IANA Considerations

TBD

7. Security Considerations

The mechanisms described in this document provide a network operator with additional ways to control its own network while making use of RPKI data. These mechanisms are applied only locally; they do not influence how other network operators interpret RPKI data. Nonetheless, care should be taken in how these mechanisms are employed.

8. Acknowledgements

The author would like to thank Stephen Kent for his guidance and detailed reviews of this document. Thanks go to Wesley Wang for the idea behind the target command, to Declan Ma for the idea behind use of multiple SLURM files, and to Richard Hansen for his careful reviews.

9. References

9.1. Informative References

[I-D.ietf-sidr-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPsec", draft-ietf-sidr-bgpsec-overview-06 (work in progress), January 2015.

[I-D.ietf-sidr-lta-use-cases]
Bush, R., "RPKI Local Trust Anchor Use Cases", draft-ietf-sidr-lta-use-cases-02 (work in progress), December 2014.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M., Kent, S., and M. Lepinski, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-08 (work in progress), April 2013.

[I-D.ietf-sidr-rpki-rtr-rfc6810-bis]
Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", draft-ietf-sidr-rpki-rtr-rfc6810-bis-03 (work in progress), March 2015.

[I-D.kent-sidr-suspenders]
Kent, S. and D. Mandelberg, "Suspenders: A Fail-safe Mechanism for the RPKI", draft-kent-sidr-suspenders-03 (work in progress), April 2015.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, March 1996.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, February 2012.

[RFC6491] Manderson, T., Vegoda, L., and S. Kent, "Resource Public Key Infrastructure (RPKI) Objects Issued by IANA", RFC 6491, February 2012.

[RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, January 2013.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, July 2013.

9.2. Normative References

[I-D.ietf-sidr-bgpsec-algs] Turner, S., "BGP Algorithms, Key Formats, & Signature Formats", draft-ietf-sidr-bgpsec-algs-09 (work in progress), January 2015.

[I-D.ietf-sidr-bgpsec-pki-profiles]
Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bgpsec-pki-profiles-10 (work in progress), January 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

Appendix A. Example SLURM File

```
SLURM 1.0

# This file is only intended to be used on a relying party running
# on rpki.example.com.
target hostname=rpki.example.com # this is a comment

# Reserve IP prefixes for local use.
del origination 10.0.0.0/24
del origination fd0b:dd1d:2dcc::/48

# Reserve AS numbers for local use.
del bgpsec 64512
del bgpsec 64513

# Allow either 64512 or 64513 to originate routes to 10.0.0.0/24.
add origination 10.0.0.0/24 64512
add origination 10.0.0.0/24 64513

# 64512 originates fd0b:dd1d:2dcc::/52 and sub-prefixes up to length
# 56.
add origination fd0b:dd1d:2dcc::/52-56 64512

# However, 64513 originates fd0b:dd1d:2dcc:42::/64.
add origination fd0b:dd1d:2dcc:42::/64 64513

# 64513 also originates fd0b:dd1d:2dcc:100::/52
add origination fd0b:dd1d:2dcc:100::/52 64513

# Authorize router keys to sign BGPsec paths on behalf of the
# specified ASes. Note that the Base64 strings used in this
# example are not valid SKIs or router public keys, due to line
# length restrictions in RFCs.
add bgpsec 64512 Zm9v VGhpcyBpcyBub3QgYSByb3V0ZXIgcHVibGljIGtleQ==
add bgpsec 64512 YmFy b3IgYSBmbG9jayBvZiBkdWNrcw==
add bgpsec 64513 YWJj bWF5YmUgYSBkaWZmZXJlbnQgYXZpYW4gY2Fycmllcj8=
```

Author’s Address

David Mandelberg
BBN Technologies
10 Moulton St.
Camridge, MA 02138
US

Email: david@mandelberg.org