SIDR
Internet-Draft
Intended status: Standards Track
Expires: August 9, 2018

D. Ma
ZDNS
D. Mandelberg
Unaffiliated
T. Bruijnzeels
RIPE NCC

February 5, 2018

Simplified Local internet nUmber Resource Management with the RPKI
draft-ietf-sidr-slurm-05

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. However, ISPs may want to establish a local view of the RPKI to control its own network while making use of RPKI data. The mechanisms described in this document provide a simple way to enable INR holders to establish a local, customized view of the RPKI, overriding global RPKI repository data as needed.

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 [RFC6811], and validation of the path of a route is described in [RFC8205].)

However, an RPKI relying party may want to override some of the information expressed via putative TAs and the certificates downloaded from the RPKI repository system. For instances, [RFC6491] recommends the creation of ROAs that would invalidate public routes for reserved and unallocated address space, yet some ISPs might like to use BGP and the RPKI with private address space ([RFC1918], [RFC4193], [RFC6598]) or private AS numbers ([RFC1930], [RFC6996]). Local use of private address space and/or AS numbers is consistent with the RFCs cited above, but such use cannot be verified by the global RPKI. This motivates creation of mechanisms that enable a network operator to publish a variant of RPKI hierarchy (for its own use and that of its customers) at its discretion. Additionally, a network operator might wish to make use of a local override capability to protect routes from adverse actions [RFC8211], until the results of such actions have been addressed. The mechanisms developed to provide this capability to network operators are hereby called Simplified Local internet nUmber Resource Management with the RPKI (SLURM).

SLURM allows an operator to create a local view of the global RPKI by generating sets of assertions. For Origin Validation [RFC6811], 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 [RFC8210]. For BGPsec [RFC8205], an assertion is a tuple of {AS number, subject key identifier, router public key} as used by rpki-rtr version 1. (For the remainder of this document, these assertions are called Origin Validation assertions and BGPsec assertions, respectively.)

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. RPKI RPs with SLURM

SLURM provides a simple way to enable RPs to establish a local, customized view of the RPKI, by overriding RPKI repository data if needed. To that end, an RP with SLURM filters out (removes from consideration for routing decisions) any assertions in the RPKI that

are overridden by local Origin Validation assertions and BGPsec assertions.

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.

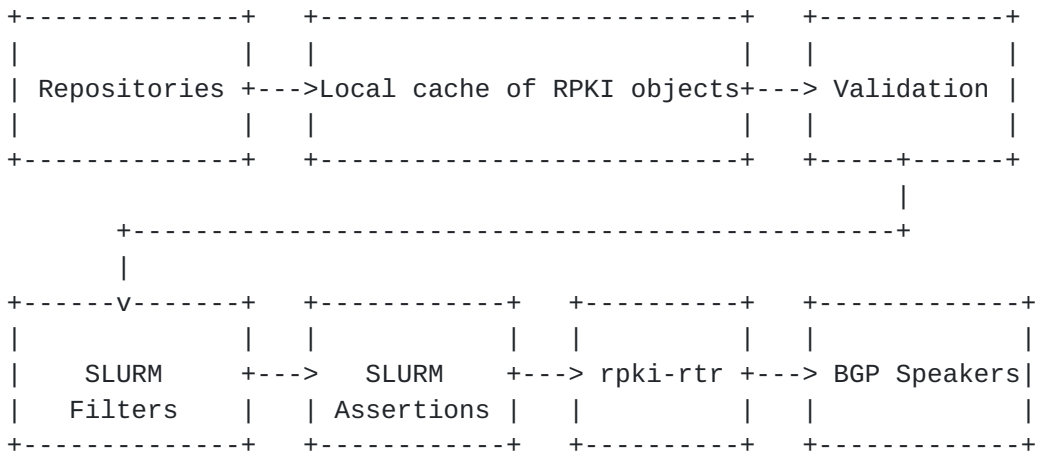

Figure 1: SLURM's Position in the Relying Party Stack

3. SLURM File and Mechanisms

3.1. Use of JSON

This document describes responses in the JSON [RFC7159]format. JSON members that are not defined here MUST NOT be used in SLURM Files. Relying Parties MUST consider any deviations from the specification an error. Future additions to the specifications in this document MUST use an incremented value for the "slurmVersion" member.

3.2. SLURM File Overview

A SLURM file consists of:

- A SLURM Version indication that MUST be 1

- A slurmTarget element (Section 3.3) consisting of:

    * Zero or more target elements. In this version of SLURM, there are two types of values for the target: ASN or FQDN. If more than one target line is present, all targets must be acceptable to the RP.

- Validation Output Filters (Section 3.4), consisting of:

    * An array of zero or more Prefix Filters, described in Section 3.4.1

    * An array of zero or more BGPSec Filters, described in Section 3.4.2

- Locally Added Assertions (Section 3.5), consisting of:

    * An array of zero or more Prefix Assertions, described in Section 3.5.1

    * An array of zero or more BGPSec Assertions, described in Section 3.5.2

In the envisioned typical use case, a relying party uses both Validation Output Filters 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.

The following JSON structure with JSON members represents a SLURM file that has no filters or assertions:

```
{
  "slurmVersion": 1,
  "slurmTarget": [],
  "validationOutputFilters": {
    "prefixFilters": [],
    "bgpsecFilters": []
  },
  "locallyAddedAssertions": {
    "prefixAssertions": [],
    "bgpsecAssertions": []
  }
}
```

Empty SLURM File

3.3. SLURM Target

A SLURM filer MUST specify a "slurmTarget" element that identifies the environment in which the SLURM file is intended to be used. The "slurmTarget" element MAY have an empty array as its value, which means "applies to all". The meaning of the "slurmTarget" element, if present, is determined by the user. If a "slurmTarget" element is present, a relying party SHOULD verify that the target is an acceptable value, and reject this SLURM file if the "slurmTarget" element is not acceptable. Each "slurmTarget" element contains merely one "asn" or one "hostname". An explanatory "comment" MAY be included in each "slurmTarget" element so that it can be shown to users of the RP software.

For instance, a large ISP may want some of its ASes to establish a local view of RPKI while the others not. Accordingly, this ISP needs to make its RPs aware of this distinction for different BGP speakers by adding ASN(s) to SLURM file target. Such a target value is an ASN expressed in number.

```
"slurmTarget": [
  {
    "asn": 65536
    "comment": "This file is intended for BGP speakers in AS 65536"
  }
]
```

slurmTarget example 1

Also, for instance, an organization may share one trusted third-party SLURM file source. For the local control, or in the case of Emergency Response Team Coordination, the SLURM file source may generate a SLURM file that is to be applied to only one specific RP. This file can take advantage of the "target" element to restrict the ASes that will accept and use the file. Accordingly, the SLURM file source needs to indicate which RP(s) should make use of the file by adding the domain name(s) of the RP(s) to the SLURM file target. Such a target value is a server name expressed in FQDN.

```
"slurmTarget": [
  {
    "hostname": "rpki.example.com"
    "comment": "This file is intended for RP server rpki.example.com"
  }
]
```

slurmTarget example 2

3.4. Validation Output Filters

3.4.1. Validated ROA Prefix Filters

The RP can configure zero or more Validated ROA Prefix Filters (Prefix Filters in short). Each Prefix Filter can contain either an IPv4 or IPv6 prefix and/or an AS number. It is RECOMMENDED that an explanatory comment is included with each Prefix Filter, so that it can be shown to users of the RP software.

Any Validated ROA Prefix (VRP, [RFC6811]) that matches any configured Prefix Filter MUST be removed from the RP's output.

A Validated ROA Prefix is considered to match with a Prefix Filter if one of the following cases applies:

1. If the Prefix Filter contains an IPv4 or IPv6 Prefix only, the VRP is considered to match the filter if the VRP Prefix is equal to or subsumed by the Prefix Filter.

2. If Prefix Filter contains an AS number only, the VRP is considered to match the filter if the VRP ASN matches the Prefix Filter ASN.

3. If Prefix Filter contains both an IPv4 or IPv6 prefix AND an AS Number, the VRP is considered to match if the VRP Prefix is equal to or subsumed by the Prefix Filter AND the VRP ASN matches the Prefix Filter ASN

The following JSON structure represents an array of "prefixFilters" with an element for each use case listed above:

```
"prefixFilters": [
  {
    "prefix": "192.0.2.0/24",
    "comment": "All VRPs encompassed by prefix"
  },
  {
    "asn": 64496,
    "comment": "All VRPs matching ASN"
  },
  {
    "prefix": "198.51.100.0/24",
    "asn": 64497,
    "comment": "All VRPs encompassed by prefix, matching ASN"
  }
]
```

prefixFilters examples

3.4.2. BGPsec Assertion Filters

The RP can configure zero or more BGPSec Assertion Filters (BGPSec Filters in short). Each BGPSec Filter can contain an AS number and/or a Router SKI.

The Router SKI is the Base64 [RFC4648] encoding of a router certificate's Subject Key Identifier, as described in [RFC8209] and [RFC6487]. This is the value of the ASN.1 OCTET STRING without the ASN.1 tag or length fields.

Furthermore it is RECOMMENDED that an explanatory comment is included with each BGPSec Filter, so that it can be shown to users of the RP software.

Any BGPSec Assertion that matches any configured BGPSec Filter MUST be removed from the RPs output.

A BGPSec Assertion is considered to match with a BGPSec Filter if one of the following cases applies:

1. If the BGPSec Filter contains an AS number only, a BGPSec Assertion is considered to match if the Assertion ASN matches the Filter ASN.

2. If the BGPSec Filter contains a Router SKI only, a BGPSec Assertion is considered to match if the Assertion Router SKI matches the Filter Router SKI.

3. If the BGPSec Filter contains both an AS number AND a Router SKI, then a BGPSec Assertion is considered to match if both the Assertion ASN matches the Filter ASN and the Assertion Router SKI matches the Filter Router SKI.

The following JSON structure represents an array of "bgpsecFilters" with an element for each use case listed above:

```
"bgpsecFilters": [
  {
    "asn": 64496,
    "comment": "All keys for ASN"
  },
  {
    "routerSKI": "<Base 64 of some SKI>",
    "comment": "Key matching Router SKI"
  },
  {
    "asn": 64497,
    "routerSKI": "<Base 64 of some SKI>",
    "comment": "Key for ASN 64497 matching Router SKI"
  }
]
```

bgpsecFilters examples

3.5. Locally Added Assertions

3.5.1. ROA Prefix Assertions

Each relying party is locally configured with a (possibly empty) array of ROA Prefix Assertions. This array is added to the RP's output.

Each ROA Prefix Assertion MUST contain an IPv4 or IPv6 prefix, an AS number, optionally a MaxLength and optionally a comment that can be shown to users of the RP software.

The following JSON structure represents an array of "prefixAssertions" with an element for each use case listed above:

```
  "prefixAssertions": [
    {
      "asn": 64496,
      "prefix": "198.51.100.0/24",
      "comment": "My other important route"
    },
    {
      "asn": 64496,
      "prefix": "2001:DB8::/32",
      "maxPrefixLength": 48,
      "comment": "My other important de-aggregated routes"
    }
  ]
```

prefixAssertions examples

3.5.2. BGPSec Assertions

Each relying party is locally configured with a (possibly empty) array of BGPSec Assertions. This array is added to the RP's output.

Each BGPSec Assertion MUST contain an AS number, a Router SKI, the Router Public Key, and optionally a comment that can be shown to users of the RP software.

The Router SKI is the Base64 [RFC4648] encoding of a router certificate's Subject Key Identifier, as described in [RFC8209] and [RFC6487]. This is the value of the ASN.1 OCTET STRING without the ASN.1 tag or length fields.

The Router Public Key is the Base64 [RFC4648] encoding of a router public key's subjectPublicKeyInfo value, as described in [RFC8208]. This is the full ASN.1 DER encoding of the subjectPublicKeyInfo, including the ASN.1 tag and length values of the subjectPublicKeyInfo SEQUENCE.

The following JSON structure represents an array of "bgpsecAssertions" with one element as described above:

```
    "bgpsecAssertions": [
      {
        "asn": 64496,
        "comment" : "My known key for my important ASN",
        "SKI": "<some base64 SKI>",
        "publicKey": "<some base64 public key>"
      }
    ]
```

prefixAssertions examples

3.6. Example of a SLURM File with Filters and Assertions

The following JSON structure represents an example of a SLURM file that uses all the elements described in the previous sections:

```
{
  "slurmVersion": 1,
  "slurmTarget":[
    {
      "asn":65536
    },
    {
      "hostname":"rpki.example.com"
    }
  ],
  "validationOutputFilters": {
    "prefixFilters": [
      {
        "prefix": "192.0.2.0/24",
        "comment": "All VRPs encompassed by prefix"
      },
      {
        "asn": 64496,
        "comment": "All VRPs matching ASN"
      },
      {
        "prefix": "198.51.100.0/24",
        "asn": 64497,
        "comment": "All VRPs encompassed by prefix, matching ASN"
      }
    ],
    "bgpsecFilters": [
      {
        "asn": 64496,
        "comment": "All keys for ASN"
```

```
          },
          {
            "routerSKI": "Zm9v",
            "comment": "Key matching Router SKI"
          },
          {
            "asn": 64497,
            "routerSKI": "YmFy",
            "comment": "Key for ASN 64497 matching Router SKI"
          }
        ]
      },
      "locallyAddedAssertions": {
        "prefixAssertions": [
          {
            "asn": 64496,
            "prefix": "198.51.100.0/24",
            "comment": "My other important route"
          },
          {
            "asn": 64496,
            "prefix": "2001:DB8::/32",
            "maxPrefixLength": 48,
            "comment": "My other important de-aggregated routes"
          }
        ],
        "bgpsecAssertions": [
          {
            "asn": 64496,
            "comment" : "My known key for my important ASN",
            "SKI": "<some base64 SKI>",
            "publicKey": "<some base64 public key>"
          }
        ]
      }
    }
```

Full SLURM File

4. SLURM File Configuration

4.1. SLURM File Atomicity

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 file 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 Added Assertions. If neither addition occurs, then both routes could be in the unknown state [RFC6811]. 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.

4.2. Multiple SLURM Files

An implementation MAY support the concurrent use of multiple SLURM files. In this case, the resulting inputs to Validation Output Filters and Locally Added Assertions are the respective unions of the inputs from each file. The envisioned typical use case for multiple files is when the files have distinct scopes. For instance, operators of two distinct networks may resort to one RP system to frame routing decisions. As such, they probably deliver SLURM files to this RP respectively. Before an RP configures SLURM files from different sources it MUST make sure there is no internal conflict among the INR assertions in these SLURM files. To do so, the RP MUST check the entries of SLURM file with regard to overlaps of the INR assertions and report errors to the sources that created these SLURM files in question.

If a problem is detected with the INR assertions in these SLURM files, the RP MUST NOT use them, and SHOULD issue a warning as error report 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 <prefixAssertions> or <prefixFilters> in file Y and X is contained by any prefix in any <prefixAssertions> or <prefixFilters> 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 <bgpsecAssertions> or <bgpsecFilters> in file Y and X is used in any <bgpsecAssertions> or <bgpsecFilters> in file Z.

5. IANA Considerations

None

6. Security considerations

The mechanisms described in this document provide a network operator with additional ways to control use of RPKI data while preserving autonomy in address space and ASN management. 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. Note that it also is possible to use SLURM to (locally) manipulate assertions about non-private INRs, e.g., allocated address space that is globally routed. For example, a SLURM file may be used to override RPKI data that a network operator believes has been corrupted by an adverse action. Network operators who elect to use SLURM in this fashion should use extreme caution.

The goal of the mechanisms described in this document is to enable an RP to create its own view of the RPKI, which is intrinsically a security function. An RP using a SLURM file is trusting the assertions made in that file. Errors in the SLURM file used by an RP can undermine the security offered by the RPKI, to that RP. It could declare as invalid ROAs that would otherwise be valid, and vice versa. As a result, an RP must carefully consider the security implications of the SLURM file being used, especially if the file is provided by a third party.

Additionally, each RP using SLURM MUST ensure the authenticity and integrity of any SLURM file that it uses. Initially, the SLURM file may be pre-configured out of band, but if the RP updates its SLURM file over the network, it MUST verify the authenticity and integrity of the updated SLURM file. Yet the mechanism to update SLURM file to guarantee authenticity and integrity is out of the scope of this document.

7. Acknowledgements

The authors would like to thank Stephen Kent for his guidance and detailed reviews of this document. Thanks go to Wei Wang for the idea behind the target command, to Richard Hansen for his careful reviews, to Hui Zou and Chunlin An for their editorial assistance.

8. References

8.1. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, DOI 10.17487/RFC1930, March 1996, <https://www.rfc-editor.org/info/rfc1930>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <https://www.rfc-editor.org/info/rfc6482>.

[RFC6491] Manderson, T., Vegoda, L., and S. Kent, "Resource Public Key Infrastructure (RPKI) Objects Issued by IANA", RFC 6491, DOI 10.17487/RFC6491, February 2012, <https://www.rfc-editor.org/info/rfc6491>.

[RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April 2012, <https://www.rfc-editor.org/info/rfc6598>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <https://www.rfc-editor.org/info/rfc6996>.

8.2. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/info/rfc6487>.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <https://www.rfc-editor.org/info/rfc6810>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

[RFC8208] Turner, S. and O. Borchert, "BGPsec Algorithms, Key Formats, and Signature Formats", RFC 8208, DOI 10.17487/RFC8208, September 2017, <https://www.rfc-editor.org/info/rfc8208>.

[RFC8209] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", RFC 8209, DOI 10.17487/RFC8209, September 2017, <https://www.rfc-editor.org/info/rfc8209>.

[RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <https://www.rfc-editor.org/info/rfc8210>.

[RFC8211] Kent, S. and D. Ma, "Adverse Actions by a Certification Authority (CA) or Repository Manager in the Resource Public Key Infrastructure (RPKI)", RFC 8211, DOI 10.17487/RFC8211, September 2017, <https://www.rfc-editor.org/info/rfc8211>.

Authors' Addresses

Di Ma
ZDNS
4 South 4th St. Zhongguancun
Haidian, Beijing 100190
China

Email: madi@zdns.cn

David Mandelberg
Unaffiliated

Email: david@mandelberg.org
URI: https://david.mandelberg.org

Tim Bruijnzeels
RIPE NCC
Stationsplein 11
Amsterdam 1012 AB
Netherlands

Email: tim@ripe.net