SIDROPS
Internet-Draft
Intended status: Standards Track
Expires: November 14, 2021

D. Ma
ZDNS
H. Yan
CNCERT
M. Aelmans
Juniper Networks

May 13, 2021

RPKI validated cache Update in SLURM over HTTPs (RUSH) draft-madi-sidrops-rush-04

Abstract

This document defines a method for transferring RPKI validated cache update information in JSON object format over HTTPs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 14, 2021.

Copyright Notice

Table of Contents

1. Introduction

This document defines a mechanism called "RPKI validated cache Update in SLURM [RFC 8416] over HTTPs (RUSH)", for the use of SLURM in updating RPKI cache data over HTTP [RFC7540] using HTTPs [RFC2818] URIs (and therefore TLS [RFC8446] security for integrity and confidentiality). Integration with HTTPs provides a secure transport for distributing cache data, which is in alignment with SLURM file format in order to take advantage of using one same API for a cache server to do both remote update and local override.

The RPKI validated cache in this document refers to the validated data of assertion information certified by corresponding RPKI signed objects such as ROA [RFC6482] and BGPsec router certificate [RFC8209], which are transferred from the RPKI cache server to routers by RTR protocol [RFC8210] for the use of the RPKI. SLURM offers a standardized method for describing RPKI cache data in JSON format [RFC8259], and SLURM is designed to carry out incremental update.

Note that RUSH merely focuses on a standardized transport and data format of the RPKI cache data. RUSH has nothing to do with synchronization at the RUSH end system, that is, more sophisticated functions such as automatic re-synchronization and access control is out of this scope and MAY be left to private implementation.

2. Terminology

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

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. RUSH Usecase

o Cache Distribution
RUSH can be used to distribute a RPKI validated cache within a single ASN or network, for example a confederation composed of a number of ASes. A small site or enterprise network MAY also use RUSH by synchronizing with a third-party RPKI cache provider over external networks.

o Local Control over Networks
Network operators MAY want to inject SLURM Assertions/Filters via an API offered by RPKI validator/cache. RUSH is therefore able to carry out such local control signals inside an administrative bailiwick in a secure manner.

To summarize, RUSH MUST be used in scenarios where the authenticity of SLURM files can be assured when carried over multiple administrative domains. Alternatively, RUSH SHOULD be used inside an administrative domain to provide extra security by the virtue of pre-configured trust anchors.

4. RUSH Operations

4.1. Use of SLURM

RUSH uses SLURM file format to indicate the intended update. A SLURM file consists of a single JSON object containing some members. Among others, "validationOutputFilters" [Section 3.3 of [RFC8416]] and "locallyAddedAssertions" [Section 3.4 of [RFC8416]] are defined to describe actions of deleting some of existing data items and adding new data items respectively.

Note that RUSH re-uses the JSON members of SLURM object, not implying the very actions are taken locally to any extent. Typically, RUSH takes place over networks remotely while take effects to the cache in question locally.

The RUSH-aware HTTPs server/client MUST be prepared to parse SLURM object.

4.2. Use of HTTPs as Transport

HTTPs is employed by RUSH to transfer RPKI validated cache update information as expressed as a SLURM object. A new data type is therefore defined to identify SLURM object in HTTPs message body.

The RUSH-aware HTTPs server/client MUST be prepared to process media type "application/json-slurm".

4.3. RUSH Example

Figure 1 shows an example of using RUSH to carry out RPKI validated cache by HTTP POST method.

```
POST /rpki-cache HTTP/2
Host: rpki.example.com
Content-Type : application/json-slurm
Content-Length:964
<964 bytes represented by the following json string>
{
       "slurmVersion": 1,
       "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"
           },
           {
             "SKI": "Zm9v",
             "comment": "Key matching Router SKI"
           },
           {
             "asn": 64497,
             "SKI": "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>",
              "routerPublicKey": "<some base64 public key>"
            }
          ]
        }
  }
```

Figure 1.Example of an HTTP message for use of RUSH

5. IANA Considerations

Type name: application

Subtype name: json-slurm

Subtype name: json-slurm

Optional parameters: N/A

Encoding considerations: This is a JSON object.

Security considerations: N/A

Interoperability considerations: [RFC8416]

Published specification:

Applications that use this media type:

Systems that want to exchange RPKI cache data update information in SLURM file format [RFC8416] over HTTP.

Person&email address to contact for further information: Di Ma <madi@zdns.cn>

Intended usage: COMMON

Restrictions on usage: N/A

Author: Di Ma <madi@zdns.cn>

Change controller: IESG

6. Security Considerations

Note that RPKI offers signed-object-oriented security, which is not provided by RUSH any longer. There are some security issues must be handled properly as per different usecases as described in Section 3.

Cache Identity: RUSH is designed to carry out RPKI cache data update from one to another, with out-of-band trust established between those cache servers. That is, the scope of RUSH usage is convergent. Cache subscription management might be employed to implement cache identification and verification. The RPKI cache server security and the trust model for the interaction between cache servers is out of the scope of this document.

Transport Security: Updating RPKI validated cache over HTTPs relies on the security of the underlying HTTPs transport. Implementations utilizing HTTP/2 benefit from the TLS profile defined in Section 9.2 of [RFC7540].

Data Integrity: An HTTPS connection provides transport security for the interaction between cache servers, but it does not provide data integrity detection. An adversary that can control the cache used by the subscriber can affect that subscriber's view of the RPKI.

7. Acknowledgments

TBD

8. References

8.1. 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>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[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>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8416] Ma, D., Mandelberg, D., and T. Bruijnzeels, "Simplified Local Internet Number Resource Management with the RPKI (SLURM)", RFC 8416, DOI 10.17487/RFC8416, August 2018, <https://www.rfc-editor.org/info/rfc8416>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Informative References

[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>.

Authors' Addresses

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

Email: madi@zdns.cn

Hanbing Yan
CNCERT

Email: yhb@cert.org.cn

Melchior Aelmans
Juniper Networks
Boeing Avenue 240
Schiphol-Rijk 1119 PZ
The Netherlands

Email: maelmans@juniper.net