SIDROPS
Internet-Draft
Intended status: Informational
Expires: December 28, 2017

D. Ma
ZDNS
S. Kent
BBN

June 26, 2017

Requirements for Resource Public Key Infrastructure (RPKI) Relying Parties
draft-madi-sidrops-rp-00

Abstract

This document provides a single reference point for requirements for Relying Party (RP) software for use in the Resource Public Key Infrastructure (RPKI). It cites requirements that appear in several RPKI RFCs, making it easier for implementers to become aware of these requirements that are segmented with orthogonal functionalities.

Table of Contents

1. Introduction 2
2. Fetching and Caching RPKI Repository Objects 3
2.1. TAL Acquisition and Processing 4
2.2. Locating RPKI Objects Using Authority and Subject Information Extensions 4
2.3. Dealing with Key Rollover 4
2.4. Dealing with Algorithm Transition 4
2.5. Strategies for Efficient Cache Maintenance 5
3. Certificate and CRL Processing 5
3.1. Verifying Resource Certificate and Syntax 5
3.2. Certificate Path Validation 5
3.3. CRL Processing 5
4. Processing RPKI Repository Signed Objects 6
4.1. Basic Signed Object Syntax Checks 6
4.2. Syntax and Validation for Each Type of Signed Object . . 6
4.2.1. Manifest 6
4.2.2. ROA 6
4.2.3. Ghostbusters 7
4.2.4. Verifying BGPsec Router Certificate 7
4.3. How to Make Use of Manifest Data 7
4.4. What to Do with Ghostbusters Information 8
5. Delivering Validated Cache to BGP Speakers 8
6. Security considerations 8
7. IANA Considerations 8
8. Acknowledgements 8
9. References 8
9.1. Normative References 8
9.2. Informative References 10
Authors' Addresses 10

1. Introduction

The RPKI RP software is used by network operators and others to acquire and verify Internet Number Resource (INR) data stored in the RPKI repository system. RPKI data, when verified, allow an RP to verify assertions about which Autonomous Systems (ASes) are authorized to originate routes for IP address prefixes. RPKI data also establishes binding between public keys and BGP routers, and indicates the AS numbers that each router is authorized to represent.

Noting that the essential requirements imposed on RPs are scattered throughout numerous RFC documents that are protocol specific or provide best practices, as follows:

RFC 6481 (Repository Structure)
RFC 6482 (ROA format)
RFC 6486 (Manifests)
RFC 6487 (Certificate and CRL profile)
RFC 6488 (RPKI Signed Objects)
RFC 6489 (Key Rollover)
RFC 6810 (RPKI to Router Protocol)
RFC 6916 (Algorithm Agility)
RFC 7730 (Trust Anchor Locator)
RFC 7935 (Algorithms)
RFC XXXX (Router Certificates)[ID.sidr-bgpsec-pki-profiles]

This makes it hard for an implementer to be confident that he/she has addressed all of these generalized requirements. Besides, software engineering calls for how to segment the RP system into components with orthogonal functionalities, so that those components could be distributed across the operational timeline of the user. Taxonomy of generalized RP requirements is going to help have 'RP role' well framed.

To consolidate RP requirements in one document, with pointers to all the relevant RFCs, this document outlines a set of baseline requirements imposed on RPs and provides a single reference point for requirements for RP software for use in the RPKI, as segmented with orthogonal functionalities:

- Fetching and Caching RPKI Repository Objects
- Processing Certificates and CRLs
- Processing RPKI Repository Signed Objects
- Delivering Validated Cache Data to BGP Speakers

This document will be update to reflect new or changed requirements as these RFCs are updated, or new RFCs are written.

2. Fetching and Caching RPKI Repository Objects

RP software uses synchronization mechanisms supported by targeted repositories (e.g., [rsync]) to download all RPKI changed data objects in the repository system and cache them locally. The software validates the RPKI data and uses it to generate authenticated data identifying which ASes are authorized to originate routes for address prefixes, and which routers are authorized to sign BGP updates on behalf of ASes.

2.1. TAL Acquisition and Processing

In the RPKI, each relying party (RP) chooses its own set of trust anchors (TAs). Consistent with the extant INR allocation hierarchy, the IANA and/or the five RIRs are obvious candidates to be default TAs for the RP.

An RP does not retrieve TAs directly. A set of Trust Anchor Locators (TALs) is used by each RP to retrieve and verify the authenticity of each trust anchor.

TAL acquisition and processing are specified in Section 3 of [RFC7730].

2.2. Locating RPKI Objects Using Authority and Subject Information Extensions

The RPKI repository system is a distributed one, consisting of multiple repository instances. Each repository instance contains one or more repository publication points. An RP discovers publication points using the SIA and AIA extensions from (validated) certificates.

Section 5 of [RFC6481] specifies how an RP locates all RPKI objects by using the SIA and AIA extensions. Detailed specifications of SIA and AIA extensions in a resource certificate are described in section 4 of [RFC6487].

2.3. Dealing with Key Rollover

An RP takes the key rollover period into account with regard to its frequency of synchronization with RPKI repository system.

RP requirements in dealing with key rollover are described in section 3 of [RFC6489].

2.4. Dealing with Algorithm Transition

The set of cryptographic algorithms used with the RPKI is expected to change over time. Each RP is expected to be aware of the milestones established for the algorithm transition and what actions are required at every juncture.

RP requirements for dealing with algorithm transition are specified in section 4 of [RFC6916].

2.5. Strategies for Efficient Cache Maintenance

Each RP is expected to maintain a local cache of RPKI objects. The cache needs to be as up to date and consistent with repository publication point data as the RP's frequency of checking permits.

The last paragraph of section 5 of [RFC6481] provides guidance for maintenance of a local cache.

3. Certificate and CRL Processing

The RPKI make use of X.509 certificates and CRLs, but it profiles these standard formats [RFC6487]. The major change to the profile established in [RFC5280] is the mandatory use of a new extension to X.509 certificate [RFC3779].

3.1. Verifying Resource Certificate and Syntax

Certificates in the RPKI are called resource certificates, and they are required to conform to the profile [RFC6487]. An RP is required to verify that a resource certificate adheres to the profile established by [RFC6487]. This means that all extensions mandated by [RFC6487] must be present and value of each extension must be within the range specified by this RFC. Moreover, any extension excluded by [RFC6487] must be omitted.

Section 7.1 of [RFC6487] gives the procedure that the RP should follow to verify resource certificate and syntax.

3.2. Certificate Path Validation

In the RPKI, issuer can only assign and/or allocate public INRs belong to it, thus the INRs in issuer's certificate are required to encompass the INRs in the subject's certificate. This is one of necessary principles of certificate path validation in addition to cryptographic verification i.e., verification of the signature on each certificate using the public key of the parent certificate).

Section 7.2 of [RFC6487] gives the procedure that the RP should follow to perform certificate path validation.

3.3. CRL Processing

The CRL processing requirements imposed on CAs and RP are described in [RFC6487]. CRLs in the RPKI are tightly constrained; only the AuthorityKeyIndetifier and CRLNumber extensions are allowed, and they MUST be present. No other CRL extensions are allowed, and no CRLEntry extensions are permitted. RPs are required to verify that

these constraints have been met. Each CRL in the RPI MUST be verified using the public key from the certificate of the CA that issued the CRL.

In the RPKI, RPs are expected to pay extra attention when dealing with a CRL that is not consistent with the Manifest associated with the publication point associated with the CRL.

Processing of a CRL that is not consistent with a manifest is a matter of local policy, as described in the fourth paragraph of Section 6.6 of [RFC6486].

4. Processing RPKI Repository Signed Objects

4.1. Basic Signed Object Syntax Checks

Before an RP can use a signed object from the RPKI repository, the RP is required to check the signed object syntax.

Section 3 of [RFC6488] lists all the steps that the RP is required to execute in order to validate the top level syntax of a repository signed object.

Note that these checks are necessary, but not sufficient. Additional validation checks must be performed based on the specific type of signed object.

4.2. Syntax and Validation for Each Type of Signed Object

4.2.1. Manifest

To determine whether a manifest is valid, the RP is required to perform manifest-specific checks in addition to those specified in [RFC6488].

Specific checks for a Manifest are described in section 4 of [RFC6486]. If any of these checks fails, indicating that the manifest is invalid, then the manifest will be discarded and treated as though no manifest were present.

4.2.2. ROA

To validate a ROA, the RP is required perform all the checks specified in [RFC6488] as well as the additional ROA-specific validation steps. The IP address delegation extension [RFC3779] present in the end-entity (EE) certificate (contained within the ROA), must encompass each of the IP address prefix(es) in the ROA.

More details for ROA validation are specified in section 2 of [RFC6482].

4.2.3. Ghostbusters

The Ghostbusters Record is optional; a publication point in the RPKI can have zero or more associated Ghostbuster Records. If a CA has at least one Ghostbuster Record, RP is required to verify that this Ghostbusters Record conforms to the syntax of signed object defined in [RFC6488].

The payload of this signed object is a (severely) profiled vCard. An RP is required to verify that the payload of Ghostbusters conforms to format as profiled in [RFC6493].

4.2.4. Verifying BGPsec Router Certificate

A BGPsec Router Certificate is a resource certificate, so it is required to comply with [RFC6487]. Additionally, the certificate must contain an AS Identifier Delegation extension, and must not contain an IP Address Delegation extension. The validation procedure used for BGPsec Router Certificates is identical to the validation procedure described in Section 7 of [RFC6487], but using the constraints applied come from specification of section 7 of [ID.sidr-bgpsec-pki-profiles].

Note that the cryptographic algorithms used by BGPsec routers are found in [ID.sidr-bgpsec-algs]. Currently, the algorithms specified in [ID.sidr-bgpsec-algs] and [RFC7935] are different. BGPsec RPs will need to support algorithms that are used to validate BGPsec signatures as well as the algorithms that are needed to validate signatures on BGPsec certificates, RPKI CA certificates, and RPKI CRLs.

4.3. How to Make Use of Manifest Data

For a given publication point, the RP ought to perform tests to determine the state of the Manifest at the publication point. A Manifest can be classified as either valid or invalid, and a valid Manifest is either current and stale. An RP decides how to make use of a Manifest based on its state, according to local (RP) policy.

If there are valid objects in a publication point that are not present on a Manifest, [RFC6486] does not mandate specific RP behavior with respect to such objects. However, most RP software ignores such objects and this document recommends that this behavior be adopted uniformly.

In the absence of a Manifest, an RP is expected to accept all valid signed objects present in the publication point. If a Manifest is stale (see [RFC6486]) and an RP has no way to acquire a more recent Manifest, the RP is expected to (TBD).

4.4. What to Do with Ghostbusters Information

An RP may encounter a stale Manifest or CRL, or an expired CA certificate or ROA at a publication point. An RP is expected to use the information from the Ghostbusters record to contact the maintainer of the publication point where any stale/expired objects were encountered. The intent here is to encourage the relevant CA and/or repository manager to update the slate or expired objects.

5. Delivering Validated Cache to BGP Speakers

On a periodic basis, BGP speakers within an AS request updated validated origin AS data and router/ASN data from the RP's cache. The RP passes this information to BGP speakers to enable them to verify the authenticity of routing announcements. The specification of the protocol designed to deliver validated cache data from an RP to a BGP Speaker is provided in [RFC6810].

6. Security considerations

TBD

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

The authors thank David Mandelberg and Wei Wang for their review, feedback and editorial assistance in preparing this document.

9. References

9.1. Normative References

[ID.sidr-bgpsec-algs]
Turner, S., "BGPsec Algorithms, Key Formats and Signature Formats", work-in-progress, <draft-ietf-sidr-bgpsec-algs>.

[ID.sidr-bgpsec-pki-profiles]
Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", work-in-progress, <draft-ietf-sidr-bgpsec-pki-profiles>.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <http://www.rfc-editor.org/info/rfc3779>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <http://www.rfc-editor.org/info/rfc6481>.

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

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <http://www.rfc-editor.org/info/rfc6486>.

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

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <http://www.rfc-editor.org/info/rfc6488>.

[RFC6489] Huston, G., Michaelson, G., and S. Kent, "Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI)", BCP 174, RFC 6489, DOI 10.17487/RFC6489, February 2012, <http://www.rfc-editor.org/info/rfc6489>.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, DOI 10.17487/RFC6493, February 2012, <http://www.rfc-editor.org/info/rfc6493>.

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

[RFC6916] Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI)", BCP 182, RFC 6916, DOI 10.17487/RFC6916, April 2013, <http://www.rfc-editor.org/info/rfc6916>.

[RFC7730] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 7730, DOI 10.17487/RFC7730, January 2016, <http://www.rfc-editor.org/info/rfc7730>.

[RFC7935] Huston, G. and G. Michaelson, Ed., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure", RFC 7935, DOI 10.17487/RFC7935, August 2016, <http://www.rfc-editor.org/info/rfc7935>.

9.2. Informative References

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <http://www.rfc-editor.org/info/rfc6480>.

[rsync] "rsync web page", <http://rsync.samba.org/>.

Authors' Addresses

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

Email: madi@zdns.cn

Stephen Kent
BBN
10 Moulton St
Cambridge, MA 02138-1119
USA

Email: kent@alum.mit.edu