Network Working Group
Internet-Draft
Updates: 6841, 8182 (if approved)
Intended status: Standards TrackInternet Initiative Japan & Arrcus, Inc.
Expires: October 27, 2020

T. Bruijnzeels
NLnet Labs
R. Bush
G. Michaelson
APNIC

April 25, 2020

Resource Public Key Infrastructure (RPKI) Repository Requirements draft-sidrops-bruijnzeels-deprecate-rsync-01

Abstract

This document formulates a plan of a phased transition to a state where RPKI repositories and Relying Party software performing RPKI Validation will use the RPKI Repository Delta Protocol (RRDP) [RFC8182] as the only mandatory to implement access protocol.

In short this plan consists of the following phases.

In phase 0, today's deployment, RRDP is supported by most, but not all Repositories, and most but not all RP software.

In the proposed phase 1 RRDP will become mandatory to implement for Repositories, in addition to rsync. This phase can start as soon as this document is published.

Once the proposed updates are implemented by all Repositories phase 2 will start. In this phase RRDP will become mandatory to implement for all RP software, and rsync must no longer be used.

Measurements will need to be done to help determine when it will be safe to transition to the final phase of this plan. During this phase Repositories will no longer be required to provide rsync access for RPKI validation purposes. However, they may still provide rsync access for direct access to files for other purposes, if desired, at a best effort basis.

Although this document currently includes descriptions and updates to RFCs for each of these phases, we may find that it will be beneficial to have separate documents for the plan, and each phase, so that it might be more clear to all when the updates to RFCs take effect.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements notation

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.

2. Motivation

The Resource Public Key Infrastructure (RPKI) [RFC6480] as originally defined uses rsync as its distribution protocol, as outlined in [RFC6481]. Later, the RPKI Repository Delta Protocol (RRDP) [RFC8182] was designed to provide an alternative. In order to facilitate incremental deployment RRDP has been deployed as an additional optional protocol, while rsync was still mandatory to implement.

A number of issues observed with rsync motivated the design of RRDP, e.g.:

- rsync is CPU and memory heavy, and easy to DoS
- rsync library support is lacking
- rsync makes it somewhat difficult to publish sets of object atomically

RRDP was designed to leverage HTTPS CDN infrastructure to provide RPKI Repository content in a resilient way, while reducing the load on the Repository server. It supports that updates are published as atomic deltas, which can help prevent most of the issues described in section 6 of [RFC6486].

For a longer discussion please see section 1 of [RFC8182].

In conclusion: we believe that RRDP is the better solution. Therefore, this document outlines a transition plan where RRDP

becomes mandatory to implement, and rsync becomes optional and eventually deprecated.

3. Plan

Changing the RPKI infrastructure to rely on RRDP instead of rsync is a delicate operation. There is current deployment of Certification Authorities, Repository Servers and Relying Party software which relies on rsync, and which may not yet support RRDP.

Therefore we need to have a plan that ultimately updates the relevant RFCs, but which uses a phased approach combined with measurements to limit the operational impact of doing this to (almost) zero.

The general outline of the plan is as follows. We will describe each step in more detail below.

Phase	Description
0	RPKI repositories support rsync, and optionally RRDP
1	RPKI repositories support both rsync and RRDP
2	All RP software prefers RRDP
3	RPKI repositories support RRDP, and optionally rsync

3.1. Phase 0 - RPKI repositories support rsync, and optionally RRDP

This is the situation at the time of writing this document. Relying Parties can prefer RRDP over rsync today, but they need to support rsync until all RPKI repositories support RRDP. Therefore all repositories should support RRDP at their earliest convenience.

3.2. Phase 1 - RPKI repositories support both rsync and RRDP

During this phase we will make RRDP mandatory to support for Repository Servers, and measure whether the deployed Repository Servers have been upgraded to do so, in as far as they don’t support RRDP already.

3.2.1. Current Support for RRDP in Repository Software

The currently known support for RRDP for repositories is as follows:

Repository Implementation	Support for RRDP
afrinic	yes
apnic	yes
arin	yes
lacnic	planned
ripe ncc	yes
Dragon Research Labs	yes(1,2)
krill	yes(1)

(1) in use at various National Internet Registries, as well as other resource holders under RIRs. (2) not all organizations using this software have upgraded to using RRDP.

3.2.2. Updates to RFC 6481

During this phase the updates are applied to section 3 of [RFC6481].

OLD:

- The publication repository SHOULD be hosted on a highly available service and high-capacity publication platform.

- The publication repository MUST be available using rsync [RFC5781] [RSYNC]. Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

NEW:

- The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.

- The publication repository MUST be available using rsync [RFC5781] [RSYNC]. The rsync server SHOULD be hosted on a highly available platform.

- Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

3.2.3. Measurements

We can find out whether all RPKI repositories support RRDP by running (possibly) modified Relying Party software that keeps track of this.

When it is found that Repositories do not yet support RRDP, outreach should be done to them individually. Since the number of Repositories is fairly low, and it is in their interest to run RRDP because it addresses availability concerns, we have confidence that we will find these Repositories willing to make changes.

3.3. Phase 2 - All RP software prefers RRDP

Once all Repositories support RRDP we can proceed to make RRDP mandatory to implement for Relying Party software.

3.3.1. RRDP support in Relying Party software

The currently known support for RRDP in Relying Party software is as follows:

Relying Party Implementation	RRDP	version	since
FORT	yes	?	?
OctoRPKI	yes	?	?
rcynic	yes	?	?
RIPE NCC RPKI Validator 2.x	yes	?	?
RIPE NCC RPKI Validator 3.x	yes	?	?
Routinator	yes	0.6.0	Sep 2019
rpki-client	no	?	?
RPSTIR	yes	?	?

The authors kindly request Relying Party software implementers to let us know in which version of their tool support for RRDP was introduced, and when that version was released.

3.3.2. Updates to RFC 8182

From this phase onwards the updates are applied to section 3.4.1 of [RFC8182].

OLD: When a Relying Party performs RPKI validation and learns about a valid certificate with an SIA entry for the RRDP protocol, it SHOULD use this protocol as follows.

NEW: When a Relying Party performs RPKI validation and learns about a valid certificate with an SIA entry for the RRDP protocol, it MUST use this protocol. It MUST NOT depend on object retrieval for this certificate over rsync for validation, although it MAY still use rsync access for other purposes under the understanding that availability is not guaranteed.

3.3.3. Measurements

Although the tools may support RRDP, users will still need to install updated versions of these tools in their infrastructure. Any Repository operator can measure this transition by observing access to their RRDP and rsync repositories respectively.

But even after new versions have been available, it is expected that there will be long, low volume, tail of users who did not upgrade and still depend on rsync.

It is hard to quantify here now, what would be an acceptable moment to conclude that it's safe to move to the next phase and make rsync optional. A parallel to the so-called DNS Flag Day comes to mind.

3.4. Phase 3 - RPKI repositories support RRDP, and optionally rsync

The end goal of this phase is that there will be no operational dependencies on rsync for Repositories, although they MAY still choose to operate rsync at a best effort basis.

3.4.1. Updates to RFC 6481

From this phase onwards these updates are applied to section 3 of [RFC6481] as it was updated during Phase 2 described above:

OLD:

- The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.

- The publication repository MUST be available using rsync [RFC5781] [RSYNC]. The rsync server SHOULD be hosted on a highly available platform.

- Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

NEW:

- The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.

- The publication repository MAY be available using rsync [RFC5781] [RSYNC].

- Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

4. Rsync URIs as object identifiers

If and when RPKI Repositories no longer need to support rsync, this begs the question whether rsync should still be used in URIs used in RPKI objects.

[RFC6481] defines a profile for the Resource Certificate Repository Structure. In this profile objects are identified through rsync URIs. E.g. a CA certificate has an Subject Information Access descriptor which uses an rsync URI to identify its manifest [RFC6486]. The manifest enumerates the relative names and hashes for all objects published under the private key of the CA certificate. The full rsync URI identifiers for each object can be resolved relative to the manifest URI.

Though it would be possible in principle to build up an RPKI tree hierarchy of objects based on key identifiers and hashes [RFC8488], most Relying Party implementations have found it very useful to use rsync URIs for this purpose. Furthermore, these identifiers make it much easier to name object in case of validation problems, which help operators to address issues.

For these reasons, RRDP still includes rsync URIs in the definition of the publish, update and withdraw elements in the snapshot and delta files that it uses. See section 3.5 of [RFC8182]. Thus, objects retrieved through RRDP can be mapped easily to files and URIs, similar to as though rsync would have been used to retrieve them.

Even though objects are no longer guaranteed to be available over rsync, we still use rsync as the mandatory scheme in the CRL Distribution Points, Authority Information Access, and Subject Information Access defined in [RFC6487]. Changing this would introduce breaking changes which make deployment very hard indeed: we

would need to invent an alternative naming scheme, which would need to be supported by all Relying Parties, before Certification Authorities can issue any certificate or RPKI signed objects using these schemes.

Furthermore, it is very convenient to have direct access to RPKI objects using rsync for troubleshooting, debugging and research purposes. Therefore Repository operators MAY still choose to make an rsync repository available for these purposes.

5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

TBD

7. Acknowledgements

TBD

8. 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>.

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, DOI 10.17487/RFC5781, February 2010, <https://www.rfc-editor.org/info/rfc5781>.

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

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

[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, <https://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, <https://www.rfc-editor.org/info/rfc6487>.

[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>.

[RFC8182] Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, "The RPKI Repository Delta Protocol (RRDP)", RFC 8182, DOI 10.17487/RFC8182, July 2017, <https://www.rfc-editor.org/info/rfc8182>.

[RFC8488] Muravskiy, O. and T. Bruijnzeels, "RIPE NCC's Implementation of Resource Public Key Infrastructure (RPKI) Certificate Tree Validation", RFC 8488, DOI 10.17487/RFC8488, December 2018, <https://www.rfc-editor.org/info/rfc8488>.

Authors' Addresses

Tim Bruijnzeels
NLnet Labs

Email: tim@nlnetlabs.nl
URI: https://www.nlnetlabs.nl/

Randy Bush
Internet Initiative Japan & Arrcus, Inc.

Email: randy@psg.com

George Michaelson
APNIC

Email: ggm@apnic.net
URI: http://www.apnic.net