Network Working Group
Internet-Draft
Updates: 6841, 8182 (if approved)
Intended status: Standards TrackInternet Initiative Japan & Arrcus, Inc.
Expires: 26 June 2023

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

23 December 2022

Resource Public Key Infrastructure (RPKI) Repository Requirements draft-ietf-sidrops-prefer-rrdp-02

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 preferred access protocol, and require rsync as a fallback option only.

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.

Phase 2 will start once the proposed updates are implemented by all compliant Repositories. In this phase RRDP will become mandatory to implement for all compliant RP software, and rsync will be required as a fallback option only.

It should be noted that 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 one or more separate documents for these phases, so that it might be more clear to all when the updates to RFCs take effect.

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/.

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.

While rsync has been very useful in the initial deployment of RPKI, a number of issues observed with it motivated the design of RRDP, e.g.:

* rsync is CPU and memory heavy on the server side, and easy to DoS
* rsync library support is lacking, complicating RP efficiency and error logging

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 updates being 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 while RRDP is not perfect, and we may indeed need future work to improve it, it is an improvement over using rsync in the context of RPKI. Therefore, this document outlines a transition plan where RRDP becomes mandatory to implement, and the operational dependency on rsync is reduced to that of a fallback option.

3. Plan to prefer RRDP

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

Table 1

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. Therefore all repositories should support RRDP at their earliest convenience.

3.1.1. Updates to RFC 8182

Section 3.4.5 of [RFC8182] has the following on "Considerations Regarding Operational Failures in RRDP":

Relying Parties could attempt to use alternative repository access mechanisms, if they are available, according to the accessMethod element value(s) specified in the SIA of the associated certificate (see Section 4.8.8 of [RFC6487]).

The use of the lower case 'could' in this sentence has led some older versions of RP implementations to conclude that any fallback from RRDP to rsync as an alternative access mechanism is a local choice. However, following discussions on this subject it has become clear that there is a preference to instruct RP software to make use of all possible data sources. The main motivation being that because of RPKI object security using a secondary source of data can never lead to a worse outcome in terms of validation.

Per this document text mentioned above is replaced by the following:

Relying Parties MUST attempt to use alternative repository access mechanisms, if they are available, according to the accessMethod element value(s) specified in the SIA of the associated certificate (see Section 4.8.8 of [RFC6487]).

Note that there is a risk that the rsync repository, as the alternative access mechanism, becomes overloaded in case all Relying Parties fall back to it at roughly the same time due to an issue with RRDP. Therefore it is RECOMMENDED that Relying Parties use a retry strategy and/or random jitter time before falling back to rsync. But, the fallback to rsync MUST NOT be postponed for more than 1 hour.

3.1.2. Updates to RFC 6481

Section 3.3 of [RFC8182] stipulates that RRDP files MUST be made available by repositories which support RRDP. In other words [RFC8182] expects that RRDP repository availability is treated as a critical service wherever it is supported.

Per this document the following bullet point is added to the considerations listed in in section 3 of [RFC6481]:

* The publication repository MAY be available using the RPKI Repository Delta Protocol [RFC8182]. If RPDP is provided, it SHOULD be hosted on a highly available platform.

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. Updates to RFC 6481

In this phase the bullet point update to section 3 of [RFC6481] mentioned above, where it was said the publication repository MAY be available using the RPKI Repository Delta Protocol is replaced by:

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

3.2.2. 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. But note that RP software is not prohibited from implementing this support sooner. At the time of this writing all known RP software supports RRDP, although it is not known to the authors whether all of them have RRDP enabled and use it as the preferred protocol.

3.3.1. Updates to RFC 8182

From this phase onwards the first paragraph of section 3.4.1 of [RFC8182] is replaced by the following:

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 with preference.

Relying Parties MUST NOT attempt to fetch objects using alternate access mechanisms, if object retrieval through this protocol is successful.

However, as stipulated in section 3.4.5, Relying Parties MUST attempt to use alternative repository access mechanisms, if object retrieval through RRDP is unsuccessful.

3.3.2. 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 a long, low volume, tail of users who did not upgrade and still depend on rsync.

4. Appendix - Implementation Status

Note that this section is included for tracking purposes during the discussion phase of this document and is not intended to be included in an RFC.

4.1. Current RRDP Support 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	ongoing
ripe ncc	yes
Dragon Research Labs	yes(1,2)
krill	yes(1)

Table 2

(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.

4.2. Current RRDP Support in Relying Party software

All current versions of known Relying Party software support RRDP:

Relying Party Implementation	support	version	since
DRL	yes	?	?
FORT	yes	1.2.0	02/2021
OctoRPKI	yes	1.0.0	02/2019
Routinator	yes	0.6.0	09/2019
rpki-client	yes	0.7.0	04/2021
RPSTIR2	yes	2.0	04/2020

Table 3

But, support for RRDP does not necessarily mean that it is also enabled and preferred over rsync by default. The authors kindly request that RP implementors provide the following information:

Relying Party Implementation	prefer	version	since
DRL	?	?	?
FORT	yes	?	?
OctoRPKI	?	?	?
Routinator	yes	0.6.0	09/2019
rpki-client	?	?	?
RPSTIR2	?	?	?

Table 4

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>.

[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>.

[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>.

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