Network Working Group
Internet-Draft
Updates: 6841, 8182 (if approved)
Intended status: Standards TrackInternet Initiative Japan & Arrcus, Inc.
Expires: August 26, 2021

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

February 22, 2021

Resource Public Key Infrastructure (RPKI) Repository Requirements
draft-ietf-sidrops-prefer-rrdp-00

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.

The first objective is to make RRDP the preferred access protocol, and require rsync as a fallback option only. This will greatly reduce the operational burden and concerns for RPKI repository operators.

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 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.

Furthermore, this document currently includes an early discussion of a future objective, which would be to change the RPKI standards such that names in RPKI objects are no longer tightly coupled to rsync. By using transport independent names and validation, we will obtain the agility needed to phase out rsync altogether and/or introduce other future access protocols.

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.

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
- we cannot ensure that RPs get atomic sets of updated objects

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 while RRDP is not perfect, and we may indeed need future work to improve on 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

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.

3.1.1. Updates to RFC 8182

Repositories which support RRDP MUST ensure that RRDP resources are available to Relying Parties (section 3.3 of [RFC8182]). Furthermore, the RRDP repository MUST include all current repository objects. Because of this the choice of falling back to alternative

repository access mechanisms was left as a local policy choice of RP software.

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.

The following update is therefore applicable to section 3.4.5 "Considerations Regarding Operational Failures in RRDP" of [RFC8182]:

OLD: 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]).

NEW: 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]).

3.1.2. Updates to RFC 6481

As noted above section 3.3 of [RFC8182] already stipulates that RRDP files MUST be made available by repositories which support RRDP. In other words the RRDP service must be treated as a critical service wherever it is supported.

During this phase the updates are applied to section 3 of [RFC6481], to make this abundantly clear:

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 MAY be available using the RPKI Repository Delta Protocol [RFC8182]. If RPDP is provided, it 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. 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

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.

3.2.2. 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.

3.3.1. 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 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 this protocol is unsuccessful.

3.3.2. Rsync URIs as object identifiers

Rsync URIs are used in the RPKI to name objects and hierarchies, and they are as such very useful when doing RPKI object validation, as well as for error reporting on validation issues.

Note that RRDP includes rsync URIs in its structure. See section 3.5 of [RFC8182]. Theoretically, RRDP servers could include any rsync URI. However, Relying Party software knows which RRDP server to is expected to include the rsync URIs for RPKI objects issued under any

given CA certificate, because of the id-ad-rpkiNotify SIA extenion, see section 3.2 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.

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.

4. Future Objective: Remove the dependency on rsync

Note that, while we discuss this here, we would probably do well to separate this section into a separate follow-up document.

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

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

The most pragmatic way to deal with rsync URIs in the RPKI would be to continue to use them as namespaces, but no longer require that rsync is available. Much like how https based namespaces are used in XML.

4.1.1. 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.

Note that this means that RP software is still required to try to fall back to rsync if RRDP is unavailable, but it may find that the rsync repository is not available.

4.2. Transport agnostic RPKI object names

We could develop a new naming scheme for RPKI objects. Perhaps based on Universal Resource Names ([RFC8141]). Doing so, would allow us to use names which are independent from retrieval mechanisms, and thus they could be less confusing in some regards, and provide more agility with regards to future changes in those mechanisms. However, this would require that many updates are made to existing RFCs. An incomplete list:

- RFC6487 New names would have be allowed in the SIA, or perhaps an X509 extension, could be used. But, the latter would have a direct impact on the deployability of updated CA certificates - RP software would reject these certificates if the extension is marked as critical by the CA and not understood by the RP.

- RFC6492 New names (in whatever form) would need to be included certificate sign requests sent to a parent CA. The parent CA will need to include a 'cert_url', indicating where an issued certificate is published, in a different format.

- RFC8181 The RPKI publication protocol is based rsync URIs, and it assumes that publishers have access to a specific directory in rsync space. This would need to be changed.

- RFC8183 This RFC defines the identity exchange between an RPKI CA and Publication Server. The server's response includes an 'sia_base', in the form of an rsync directory, under which a CA is supposed to name its objects.

- RFC8182 The RRDP protocol uses rsync URIs for compatibility with rsync as a retrieval method. This would need to be updated.

Obviously this needs more discussion.

The exercise would not be trivial. But, arguably doing this work will not become easier by postponing it, and once done would leave the RPKI better positioned to use alternative access methods in future as well.

5. 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.

5.1. 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)

(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.

5.2. Current 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	1.2.0	02/2021
OctoRPKI	yes	1.0.0	02/2019
rcynic	yes	?	?
RIPE NCC RPKI Validator 2.x	yes	2.18	07/2015
RIPE NCC RPKI Validator 3.x	yes	3.0	03/2018
Routinator	yes	0.6.0	09/2019
rpki-client	ongoing	?	?
RPSTIR2	yes	2.0	04/2020

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.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

TBD

8. Acknowledgements

TBD

9. References

9.1. 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>.

[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>.

[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>.

9.2. Informative References

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

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