Stream: Internet Engineering Task Force (IETF)
RFC: 9697
Updates: 8182
Category: Standards Track
Published: November 2024
ISSN: 2070-1721
Authors: J. Snijders, *Fastly*
T. de Kock, *RIPE NCC*

RFC 9697
Detecting RPKI Repository Delta Protocol (RRDP) Session Desynchronization

Abstract

This document describes an approach for Resource Public Key Infrastructure (RPKI) Relying Parties to detect a particular form of RPKI Repository Delta Protocol (RRDP) session desynchronization and how to recover. This document updates RFC 8182.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9697.

Copyright Notice

Table of Contents

1. Introduction

The Resource Public Key Infrastructure (RPKI) Repository Delta Protocol (RRDP) [RFC8182] is a one-way synchronization protocol for distributing RPKI data in the form of *differences* (deltas) between sequential repository states. Relying Parties (RPs) apply a contiguous chain of deltas to synchronize their local copy of the repository with the current state of the remote Repository Server. Delta files for any given session_id and serial number are expected to contain an immutable record of the state of the Repository Server at that given point in time, but this is not always the case.

This document describes an approach for RPs to detect a form of RRDP session desynchronization where the hash of a delta for a given serial number and session_id have mutated from the previous Update Notification File and how to recover.

1.1. Requirements Language

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. Immutability of RRDP Files

Section 3.1 of [RFC8182] describes how discrete publication events such as the addition, modification, or deletion of one or more repository objects *can* be communicated as immutable files, highlighting advantages for publishers, such as the ability to precalculate files and make use of caching infrastructure.

While the global RPKI is understood to present a loosely consistent view, depending on timing, updating, and fetching (see Section 6 of [RFC7115]), different caches having different data for the same RRDP session at the same serial violates the principle of least astonishment.

If an RRDP server over time serves differing data for a given session_id and serial number, distinct RP instances (depending on the moment they connected to the RRDP server) would end up with divergent local repositories. Comparing only the server-provided session_id and latest serial number across distinct RP instances would not bring such divergence to light.

The RRDP specification [RFC8182] alludes to immutability being a property of RRDP files, but it doesn't make it clear that immutability is an absolute requirement for the RRDP protocol to work well.

3. Detection of Desynchronization

Relying Parties can implement a mechanism to keep a record of the serial and hash attribute values in delta elements of the previous successful fetch of an Update Notification File. Then, after fetching a new Update Notification File, the Relying Party should compare if the serial and hash values of previously seen serials match those in the newly fetched file. If any differences are detected, this means that the Delta files were unexpectedly mutated, and the RP should proceed to Section 4.

3.1. Example

This section contains two versions of an Update Notification File to demonstrate an unexpected mutation. The initial Update Notification File is as follows:

```
<notification xmlns="http://www.ripe.net/rpki/rrdp" version="1"
              session_id="fe528335-db5f-48b2-be7e-bf0992d0b5ec"
              serial="1774">
  <snapshot uri="https://rrdp.example.net/1774/snapshot.xml"
    hash="4b5f27b099737b8bf288a33796bfe825fb2014a69fd6aa99080380299952f2e2"/>
  <delta serial="1774"
    hash="effac94afd30bbf1cd6e180e7f445a4d4653cb4c91068fa9e7b669d49b5aaa00"
    uri="https://rrdp.example.net/1774/delta.xml" />
  <delta serial="1773"
    hash="731169254dd5de0ede94ba6999bda63b0fae9880873a3710e87a71bafb64761a"
    uri="https://rrdp.example.net/1773/delta.xml" />
  <delta serial="1772"
    hash="d4087585323fd6b7fd899ebf662ef213c469d39f53839fa6241847f4f6ceb939"
    uri="https://rrdp.example.net/1772/delta.xml" />
</notification>
```

Figure 1

Based on the above Update Notification File, an RP implementation could record the following state:

```
fe528335-db5f-48b2-be7e-bf0992d0b5ec
1774 effac94afd30bbf1cd6e180e7f445a4d4653cb4c91068fa9e7b669d49b5aaa00
1773 731169254dd5de0ede94ba6999bda63b0fae9880873a3710e87a71bafb64761a
1772 d4087585323fd6b7fd899ebf662ef213c469d39f53839fa6241847f4f6ceb939
```

Figure 2

A new version of the Update Notification File is published as follows:

```
<notification xmlns="http://www.ripe.net/rpki/rrdp" version="1"
              session_id="fe528335-db5f-48b2-be7e-bf0992d0b5ec"
              serial="1775">
  <snapshot uri="https://rrdp.example.net/1775/snapshot.xml"
    hash="cd430c386deacb04bda55301c2aa49f192b529989b739f412aea01c9a77e5389"/>
  <delta serial="1775"
    hash="d199376e98a9095dbcf14ccd49208b4223a28a1327669f89566475d94b2b08cc"
    uri="https://rrdp.example.net/1775/delta.xml" />
  <delta serial="1774"
    hash="10ca28480a584105a059f95df5ca8369142fd7c8069380f84ebe613b8b89f0d3"
    uri="https://rrdp.example.net/1774/delta.xml" />
  <delta serial="1773"
    hash="731169254dd5de0ede94ba6999bda63b0fae9880873a3710e87a71bafb64761a"
    uri="https://rrdp.example.net/1773/delta.xml" />
</notification>
```

Figure 3

Using its previously recorded state (see Figure 2), the RP can compare the hash values for serials 1773 and 1774. For serial 1774, compared to the earlier version of the Update Notification File, a different hash value is now listed, meaning an unexpected delta mutation occurred.

4. Recovery After Desynchronization

Following the detection of RRDP session desynchronization, in order to return to a synchronized state, RP implementations **SHOULD** issue a warning and **SHOULD** download the latest Snapshot File and process it as described in Section 3.4.3 of [RFC8182].

See Section 6 for an overview of risks associated with desynchronization.

5. Changes to RFC 8182

The following paragraph is added to Section 3.4.1 of [RFC8182], "Processing the Update Notification File", after the paragraph that ends "The Relying Party **MUST** then download and process the Snapshot File specified in the downloaded Update Notification File as described in Section 3.4.3."

NEW

> If the session_id matches the last known session_id, the Relying Party **SHOULD** compare whether hash values associated with previously seen files for serials match the hash values of the corresponding serials in the newly fetched Update Notification File. If any differences are detected, this means that files were unexpectedly mutated (see [RFC9697]). The Relying Party **SHOULD** then download and process the Snapshot File specified in the downloaded Update Notification File as described in Section 3.4.3.

6. Security Considerations

Due to the lifetime of RRDP sessions (often measured in months), desynchronization can persist for an extended period if undetected.

Caches in a desynchronized state pose a risk by emitting a different set of Validated Payloads than they would otherwise emit with a consistent repository copy. Through the interaction of the desynchronization and the *failed fetch* mechanism described in Section 6.6 of [RFC9286], Relying Parties could spuriously omit Validated Payloads or emit Validated Payloads that the Certification Authority intended to withdraw. As a result, due to the desynchronized state, route decision making processes might consider route announcements intended to be marked valid as "unknown" or "invalid" for an indeterminate period.

Missing Validated Payloads negatively impact the ability to validate BGP announcements using mechanisms such as those described in [RFC6811] and [ASPA].

Section 6.6 of [RFC9286] advises RP implementations to continue to use cached versions of objects, but only until such time as they become stale. By detecting whether the remote Repository Server is in an inconsistent state and then immediately switching to using the latest Snapshot File, RPs increase the probability to successfully replace objects before they become stale.

7. IANA Considerations

This document has no IANA actions.

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>.

[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>.

8.2. Informative References

[ASPA] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J., and K. Sriram, "BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-verification-19, 27 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-verification-19>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC7115] Bush, R., "Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)", BCP 185, RFC 7115, DOI 10.17487/RFC7115, January 2014, <https://www.rfc-editor.org/info/rfc7115>.

[RFC9286] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 9286, DOI 10.17487/RFC9286, June 2022, <https://www.rfc-editor.org/info/rfc9286>.

Acknowledgements

During the hallway track at RIPE 86, Ties de Kock shared the idea for detecting this particular form of RRDP desynchronization, after which Claudio Jeker, Job Snijders, and Theo Buehler produced an implementation based on rpki-client. Equipped with tooling to detect this particular error condition, in subsequent months it became apparent that unexpected delta mutations in the global RPKI repositories do happen from time to time.

The authors wish to thank Theo Buehler, Mikhail Puzanov, Alberto Leiva, Tom Harrison, Warren Kumari, Behcet Sarikaya, Murray Kucherawy, Éric Vyncke, Roman Danyliw, Tim Bruijnzeels, and Michael Hollyman for their careful review and feedback on this document.

Authors' Addresses

Job Snijders
Fastly
Amsterdam
Netherlands
Email: job@fastly.com

Ties de Kock
RIPE NCC
Amsterdam
Netherlands
Email: tdekock@ripe.net