Workgroup: SIDROPS
Internet-Draft:
draft-ietf-sidrops-cms-signing-time-00
Updates: 6488 (if approved)
Published: 24 July 2023
Intended Status: Standards Track
Expires: 25 January 2024
Authors: J. Snijders T. Harrison
Fastly APNIC

On the use of the CMS signing-time attribute in RPKI Signed Objects

Abstract

RFC 6488 standardized a template for specifying Signed Objects that can be validated using the RPKI. Since the publication of that document, a new additional protocol for distribution of RPKI repositories was developed (RFC 8182), and new insights arose with respect to querying and combining the different distribution mechanisms. This document describes how Publishers and Relying Parties can use the CMS signing-time attribute to optimize seamless transitions from RRDP to RSYNC. Additionally, this document updates RFC 6488 by mandating the presence of the CMS signing-time attribute and disallowing the binary-signing-time attribute.

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.

Status of This Memo

Table of Contents

1. Introduction

[RFC6488] standardized a template for specifying Signed Objects that can be validated using the RPKI. Since the publication of that document, a new additional protocol for distribution of RPKI repositories was developed [RFC8182], and new insights arose with respect to querying and combining the different distribution mechanisms. This document describes how Publishers and Relying Parties can use the CMS signing-time [RFC5652] attribute to optimize seamless transitions from RRDP to RSYNC. Additionally, this document updates [RFC6488] by mandating the presence of the CMS signing-time attribute and disallowing the binary-signing-time attribute.

2. Optimizing Seamless transitions from RRDP to RSYNC

To avoid needless re-transfers of unchanged files in consecutive RSYNC synchronizations, [I-D.timbru-sidrops-publication-server-bcp] recommends the use of so-called 'deterministic' (normalized) timestamps for files: as long as a file's contents are unchanged, Publishers SHOULD ensure the file's last modification timestamp also doesn't change. This document advances the aforementioned concept by describing a synchronization strategy through which needless transfers are also avoided when RSYNC is used for the first time, in instances where previous retrieval via RRDP has occurred, by leveraging that RRDP data.

As described in [I-D.ietf-sidrops-prefer-rrdp], RP implementations are expected to first try synchronization via RRDP, and if and only if that fails - for some reason (malformed XML, expired TLS certificate, TCP connection timeout, etc.) - attempt to synchronize via RSYNC instead.

In the RSYNC synchronization protocol, a file's last modification timestamp (from here on 'mod-time') and filesize are used to determine whether the general-purpose RSYNC synchronization algorithm needs to be executed for the file. This is the default mode for both GPL [rsync] and [openrsync]. If the sender's copy of the file and the receiver's copy of the file both have the same mod-time and filesize, the files are assumed to contain the same content, and are skipped for the purposes of synchronization. Ensuring consistency with respect to mod-time for both senders and receivers helps to reduce the cost of RSYNC retrieval, in terms of bandwidth, disk operations, and CPU instructions.

In order to reduce the burden of the RSYNC synchronization (following a RRDP failure), Publishers and Relying Parties SHOULD adhere to the following guidelines.

2.1. Guidance for Publishers

When serializing RPKI Signed Objects to a filesystem hierarchy for RSYNC consumption, the mod-time of the file containing the Signed Object MUST be set to the CMS signing-time contained within the Signed Object.

2.2. Guidance for Relying Parties

When serializing RPKI Signed Objects retrieved via RRDP to a filesystem hierarchy, the mod-time of the file containing the Signed Object MUST be set to the CMS signing-time contained within the Signed Object.

If an RP uses RRDP to synthesize a filesystem hierarchy for the repository, then synchronizing from the publisher to the corresponding directory directly is an option. Alternatively, the RP may synchronize to a new (empty) directory while using the '--compare-dest=DIR' rsync feature, to avoid having to retrieve files that are already available by way of the synthesized filesystem hierarchy. The DIR variable SHOULD point at the directory containing previously fetched and validated RPKI data (in its original form, to ensure the filesize parameter matches).

Quoted from the GPL rsync man page:

> This option instructs rsync to use DIR on the destination machine as an additional hierarchy to compare destination files against doing transfers (if the files are missing in the destination directory). If a file is found in DIR that is identical to the sender's file, the file will NOT be transferred to the destination directory. This is useful for creating a sparse backup of just files that have changed from an earlier backup.

Quoted from the openrsync man page:

> Use directory as an alternate base directory to compare files against on the destination machine. If file in directory is found and identical to the sender's file, the file will not be transferred.

3. Presence of CMS signing-time attribute in the field

Analysing an archive [rpkiviews] containing valid RPKI Signed Objects discovered via the five RIR Trust Anchors in the last eight weeks (2023-04-14 to 2023-06-06), 100% of Signed Objects contain a CMS signing-time attribute. [NOTE: a job is running to analyse the milions of objects going back to 2022-06-06 - might take a few more days to parse all that data]

The above means that already today, all Certificate Authorities produce Signed Objects which contain a CMS signing-time attribute. Thus, making the CMS signing-time attribute mandatory would not make any existing CA operations non-compliant.

As of 3 June, 2023, for 25.8% of Signed Objects the CMS signing-time timestamp exactly matches the file's mod-time observed via RSYNC. This means that it is already the case that RPs would see a significant reduction in the amount of processing required in RSYNC if they adopted the strategy outlined in Section 2.2.

In the above-mentioned period of time, zero Signed Objects were discovered with a CMS binary-signing-time [RFC6019] attribute.

Therefore, disallowing the CMS binary-signing-time attribute would not make any existing CA operations non-compliant.

4. Considerations and Alternative Approaches

A slightly different approach that has been suggested is to normalize file mod-times based on the Signed Object's embedded End-Entity (EE) X.509 notBefore timestamp value. A downside of that approach is that CAs might backdate the notBefore timestamp to increase the validity window of the Signed Object, which in turn decreases insight for RPKI operators as to when exactly the Signed Object purportedly came into existence.

Along similar lines, the notBefore timestamp may be set in the future. Setting the mod-time of a file to a future date may be unintuitive for users, and some programs (e.g. make) will warn on encountering files with such mod-times.

There is also an increased chance of two distinct objects published to the same path having the same mod-time and filesize under this approach, due to CAs setting the notBefore timestamp to some stable value for a given object and reissuance often not changing the file size (e.g. where a prefix or a max-length value is changed in a ROA). In such a situation, if the receiver has the first copy of a file, RSYNC retrieval will skip the second copy of the file, and the synchronization operation for the associated repository will result in a "failed fetch", per section 6.6 of [RFC9286], due to an inconsistency between the file's hash and the hash listed in the associated manifest. That in turn necessitates further retrieval operations on the part of the receiver. The chance of two distinct objects being issued with the same mod-time and filesize when CMS signing-time is used to set the mod-time is much smaller, since it requires that those distinct objects be issued in very close succession.

Another downside of using notBefore is that Publishers would need to deserialize both the CMS envelope and the X.509 EE certificate contained therein to extract a timestamp, instead of merely parsing the CMS envelope.

Ensuring the mod-time is set to the CMS signing-time gives RPKI operators a headstart when using tools like [ls], in the sense that the mod-time aligns with the purported time of object issuance.

The CMS signing-time attribute has proven useful in researching and tracking delays in various layers of the RPKI [PAM23]. Mandating the CMS signing-time to be present might aid future researchers studying the RPKI ecosystem.

The --checksum option to rsync disables the mod-time and filesize comparison check in favour of a check based on a whole-file checksum. This check is slower than the mod-time and filesize check, but (in instances where the file content has not changed) faster than the general-purpose RSYNC synchronization algorithm. Since ensuring consistency between the mod-time and filesize on both sides of the transaction is straightforward, there is no particular reason to pursue an approach based on --checksum.

5. Update to RFC 6488

This section updates [RFC6488] to make the CMS signing-time attribute mandatory and disallow the presence of the CMS binary-signing-time attribute.

In section 2.1.6.4 the paragraph starting with "The signedAttrs element MUST be present and ..." and ending in "Other signed attributes MUST NOT be included." is replaced with the following text:

> The signedAttrs element MUST be present and MUST include the content-type, message-digest, and signing-time attributes [RFC5652]. Other signed attributes MUST NOT be included.

In section 2.1.6.4.3 the first sentence "The signing-time attribute MAY be present." is replaced with the following text:

> The signing-time attribute MUST be present.

Section 2.1.6.4.4 is removed in its entirety.

6. Security Considerations

This document has no Security Considerations.

7. IANA Considerations

This document has no IANA actions.

8. Acknowledgements

The authors would like to thank Ties de Kock for their helpful review of this document.

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>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[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, <https://www.rfc-editor.org/info/rfc6488>.

[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

[I-D.ietf-sidrops-prefer-rrdp] Bruijnzeels, T., Bush, R., and G. G. Michaelson, "Resource Public Key Infrastructure (RPKI) Repository Requirements", Work in Progress, Internet-Draft, draft-ietf-sidrops-prefer-rrdp-02, 23 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-prefer-rrdp-02>.

[I-D.timbru-sidrops-publication-server-bcp] Bruijnzeels, T., de Kock, T., Hill, F., and T. Harrison, "RPKI Publication Server Best Current Practices", Work in Progress, Internet-Draft, draft-timbru-sidrops-publication-server-bcp-01, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-timbru-sidrops-publication-server-bcp-01>.

[ls] IEEE and The Open Group, "ls - The Open Group Base Specifications Issue 7", 2018, <https://pubs.opengroup.org/onlinepubs/9699919799/utilities/ls.html>.

[openrsync] Jeker, C., Obser, F., and K. Dzonsons, "openrsync", 2023, <https://www.openrsync.org/>.

[PAM23] Fontugne, R., Phokeer, A., Pelsser, C., Vermeulen, K., and R. Bush, "RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes", February 2023, <https://www.iijlab.net/en/members/romain/pdf/romain_pam23.pdf>.

[RFC6019] Housley, R., "BinaryTime: An Alternate Format for Representing Date and Time in ASN.1", RFC 6019, DOI 10.17487/RFC6019, September 2010, <https://www.rfc-editor.org/info/rfc6019>.

[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>.

[rpki-client] Jeker, C., Snijders, J., Dzonsons, K., and T. Buehler, "rpki-client", June 2023, <https://www.rpki-client.org/>.

[rpkitouch] Snijders, J., "rpkitouch", June 2023, <https://github.com/job/rpkitouch>.

[rpkiviews] Snijders, J., "rpkiviews", June 2023, <http://www.rpkiviews.org/>.

[rsync] Tridgell, A., Mackerras, P., and W. Davison, "rsync", 2022, <https://rsync.samba.org/>.

Appendix A. Implementation status - RFC EDITOR: REMOVE BEFORE PUBLICATION

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

*For the Publication side of the house: [rpkitouch]

*For the Relying Party side of the house: OpenBSD [rpki-client]

Authors' Addresses

Job Snijders
Fastly
Amsterdam
Netherlands

Email: job@fastly.com

Tom Harrison
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane QLD 4101
Australia

Email: tomh@apnic.net