Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 24, 2021

J. Snijders
Fastly

February 20, 2021

RPKI Signed Checklists
draft-spaghetti-sidrops-rpki-rsc-03

Abstract

This document defines a Cryptographic Message Syntax (CMS) profile for a general purpose listing of checksums (a 'checklist'), for use with the Resource Public Key Infrastructure (RPKI). The objective is to allow an attestation, in the form of a listing of one or more checksums of arbitrary digital objects (files), to be signed "with resources", and for validation to provide a means to confirm a specific Internet Resource Holder produced the signed checklist. The profile is intended to provide for the signing of a checksum listing with an arbitrary set of Internet Number Resources.

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

Copyright Notice

Table of Contents

1. Introduction

This document defines a Cryptographic Message Syntax (CMS) [RFC5652] profile for a general purpose listing of checksums (a 'checklist'), for use with the Resource Public Key Infrastructure (RPKI) [RFC6480]. The objective is to allow an attestation, in the form of a listing of one or more checksums of arbitrary files, to be signed "with resources", and for validation to provide a means to confirm a given Internet Resource Holder produced the RPKI Signed Checklist (RSC).

The profile is intended to provide for the signing of a checksum listing with an arbitrary set of Internet Number Resources.

RSC files are expected to facilitate Bring Your Own IP (BYOIP) authentication, inter-domain interconnection provisioning, and resource holdership verification processes.

The RSC concept borrows heavily from RTA [I-D.ietf-sidrops-rpki-rta], Manifests [RFC6486], and OpenBSD's [signify] utility. The main difference between RSC and RTA is that an RTA enables multiple signers to attest a single anonymous digital object through a checksum of its content, while an RSC allows a single signer to attest the checksums of multiple named digital objects. This difference is expected to represent a simplification for implementers and operators.

2. RSC Profile and Distribution

RSC follows the Signed Object Template for the RPKI [RFC6488] with one exception. Because RSCs MUST NOT be distributed through the global RPKI repository system, the Subject Information Access (SIA) extension is omitted from the RSC's X.509 EE certificate.

What constitutes suitable transport for RSC files is deliberately unspecified. It might be a USB stick, a web interface secured with conventional HTTPS, PGP-signed email, a T-shirt printed with a QR code, or a carrier pigeon.

3. The RSC ContentType

The ContentType for an RSC is defined as rpkiSignedChecklist, and has the numerical value of 1.2.840.113549.1.9.16.1.TBD.

This OID MUST appear both within the eContentType in the encapContentInfo object as well as the ContentType signed attribute in the signerInfo object (see [RFC6488]).

4. The RSC eContent

The content of an RSC indicates that an a checklist for arbitrary named digital objects has been signed "with resources". An RSC is formally defined as:

```
   RpkiSignedChecklist-2021
     { iso(1) member-body(2) us(840) rsadsi(113549)
       pkcs(1) pkcs9(9) smime(16) mod(0) TBD }

   DEFINITIONS EXPLICIT TAGS ::=
   BEGIN

   IMPORTS
     CONTENT-TYPE, Digest, DigestAlgorithmIdentifier
     FROM CryptographicMessageSyntax-2009 -- in [RFC5911]
       { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
         pkcs-9(9) smime(16) modules(0) id-mod-cms-2004-02(41) }

     ASIdOrRange, IPAddressFamily
     FROM IPAddrAndASCertExtn -- in [RFC3779]
       { iso(1) identified-organization(3) dod(6) internet(1)
         security(5) mechanisms(5) pkix(7) mod(0)
         id-mod-ip-addr-and-as-ident(30) } ;

   ct-rpkiSignedChecklist CONTENT-TYPE ::=
       { TYPE RpkiSignedChecklist IDENTIFIED BY
         id-ct-signedChecklist }

   id-ct-signedChecklist OBJECT IDENTIFIER ::=
       { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
         pkcs-9(9) id-smime(16) id-ct(1) TBD }

   RpkiSignedChecklist ::= SEQUENCE {
     version  [0]          INTEGER DEFAULT 0,
     resources             ResourceBlock,
     digestAlgorithm       DigestAlgorithmIdentifier,
     checkList             SEQUENCE SIZE (1..MAX) OF FileAndHash }

   FileAndHash ::= SEQUENCE {
     file            IA5String OPTIONAL,
     hash            Digest }

   ResourceBlock ::= SEQUENCE {
       asID         [0]       AsList OPTIONAL,
       ipAddrBlocks [1]       IPList OPTIONAL }
       -- at least one of asID or ipAddrBlocks MUST be present
       ( WITH COMPONENTS { ..., asID PRESENT} |
         WITH COMPONENTS { ..., ipAddrBlocks PRESENT } )

   AsList ::= SEQUENCE (SIZE(1..MAX)) OF ASIdOrRange

   IPList ::= SEQUENCE (SIZE(1..MAX)) OF IPAddressFamily

   END
```

4.1. version

The version number of the RpkiSignedChecklist MUST be 0.

4.2. resources

The resources contained here are the resources used to mark the attestation, and MUST match the set of resources listed by the EE certificate carried in the CMS certificates field.

4.3. digestAlgorithm

The digest algorithm used to create the message digest of the attested digital object. This algorithm MUST be a hashing algorithm defined in [RFC7935].

4.4. checkList

This field is a sequence of FileAndHash objects. There is one FileAndHash entry for each arbitrary object referenced from the RSC. Each FileAndHash is an ordered pair consisting an optional name of the file containing the object, and the message digest of the digital object.

5. Operational Considerations

When working with objects of a plain-text nature (ASCII, UTF-8, HTML, Javascript, XML, etc) it is RECOMMENDED to distribute such objects in a lossless compressed form, and sign the compressed form. Wrapping plain-text objects in a compression envelope can help make those appear as a single octet string to any intermediate systems, which hopefully discourages in-transit modification of the file contents. The use of lossless compression can help avoid checksum verification errors.

6. RSC Validation

To validate an RSC the relying party MUST perform all the validation checks specified in [RFC6488] as well as the following additional RSC-specific validation steps.

o The message digest of each referenced digital object, using the digest algorithm specified in the the digestAlgorithm field, MUST be calculated and MUST match the value given in the messageDigest field of the associated FileAndHash.

o If a filename is present, the filename MUST NOT contain a '/' (slash) or '\' (backslash) character.

7. Security Considerations

Relying parties are hereby warned that the data in a RPKI Signed Checklist is self-asserted. These data have not been verified by the CA that issued the CA certificate to the entity that issued the EE certificate used to validate the Signed Checklist.

8. 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".

o A signer and validator implementation [rpki-rsc-demo] based on perl and OpenSSL was provided by Tom Harrison from APNIC.

o A validator implementation based on OpenBSD's rpki-client is expected to be published after IANA Early Allocation of the OIDs.

9. IANA Considerations

9.1. OID

The IANA has registered the OID for the RPKI Signed Checklist in the registry created by [RFC6488] as follows:

Name	OID	Specification
Checklists	1.2.840.113549.1.9.16.1.TBD	[RFC-TBD]

9.2. File Extension

The IANA has added an item for the signed Checklist file extension to the "RPKI Repository Name Scheme" created by [RFC6481] as follows:

Filename Extension	RPKI Object	Reference
.sig	Signed Checklist	[RFC-TBD]

9.3. Media Type

The IANA has registered the media type application/rpki-checklist as follows:

```
   Type name: application
   Subtype name: rpki-checklist
   Required parameters: None
   Optional parameters: None
   Encoding considerations: binary
   Security considerations: Carries an RPKI Signed Checklist
                            [RFC-TBD].
   Interoperability considerations: None
   Published specification: This document.
   Applications that use this media type: RPKI operators.
   Additional information:
     Content: This media type is a signed object, as defined
         in [RFC6488], which contains a payload of a list of
         checksums as defined above in this document.
     Magic number(s): None
     File extension(s): .sig
     Macintosh file type code(s):
   Person & email address to contact for further information:
     Job Snijders <job@fastly.com>
   Intended usage: COMMON
   Restrictions on usage: None
   Author: Job Snijders <job@fastly.com>
   Change controller: Job Snijders <job@fastly.com>
```

10. References

10.1. 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>.

[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>.

[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>.

[RFC7935] Huston, G. and G. Michaelson, Ed., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure", RFC 7935, DOI 10.17487/RFC7935, August 2016, <https://www.rfc-editor.org/info/rfc7935>.

[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>.

10.2. Informative References

[I-D.ietf-sidrops-rpki-rta]
Michaelson, G., Huston, G., Harrison, T., Bruijnzeels, T., and M. Hoffmann, "A profile for Resource Tagged Attestations (RTAs)", draft-ietf-sidrops-rpki-rta-00 (work in progress), January 2021.

[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>.

[rpki-rsc-demo]
Harrison, T., "A proof-of-concept for constructing and validating RPKI Signed Checklists (RSCs).", February 2021, <https://github.com/APNIC-net/rpki-rsc-demo>.

[signify] Unangst, T. and M. Espie, "signify - cryptographically sign and verify files", May 2014, <https://man.openbsd.org/signify>.

Appendix A. Acknowledgements

The author wishes to thank George Michaelson, Tom Harrison, Geoff Huston, Randy Bush, Stephen Kent, Matt Lepinski, Rob Austein, Ted Unangst, and Marc Espie for prior art. The author thanks Russ Housley for reviewing the ASN.1 notation and providing suggestions. The author would like to thank Nimrod Levy, Tom Harrison, Ben Maddison, and Tim Bruijnzeels for document review and suggestions.

Author's Address

Job Snijders
Fastly
Amsterdam
Netherlands

Email: job@fastly.com