Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: December 10, 2018

T. Bruijnzeels
NLnet Labs
C. Martinez
LACNIC

June 8, 2018

RPKI signed object for TAL
draft-ietf-sidrops-signed-tal-01

Abstract

Trust Anchor Locators (TALs) [I-D.ietf-sidrops-https-tal] are used by Relying Parties in the RPKI to locate and validate Trust Anchor certificates used in RPKI validation. This document defines an RPKI signed object [RFC6488] for a Trust Anchor Locator (TAL) that can be used by Trust Anchors to perform a planned migration to a new key, allowing Relying Parties to discover the new key up to one year after the migration occurred.

Status of This Memo

This Internet-Draft will expire on December 10, 2018.

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. Introduction

Trust Anchor Locators (TALs) [I-D.ietf-sidrops-https-tal] are used by Relying Parties in the RPKI to locate and validate Trust Anchor certificates used in RPKI validation. This document defines an RPKI signed object [RFC6488] for a Trust Anchor Locator (TAL) that can be used by Trust Anchors to perform a planned migration to a new key, allowing Relying Parties to discover the new key up to one year after the migration occurred. (Note "one year" is arbitrary, and may be changed in a future version of this document)

Note that [RFC5011] describes Automated Updates of DNS Security (DNSSEC) Trust Anchors and can provide some useful insight here as well. However, concepts like a set of Trust Anchors, standby Trust Anchors, and TTLs are not applicable to the RPKI. Therefore, an alternative approach based on already existing concept of the Trust Anchor Locator [I-D.ietf-sidrops-https-tal], and top-down validation of an RPKI Trust Anchor certificate tree, where objects are retrieved from the RPKI repositories, is appropriate.

3. Signed TAL definition

The Signed TAL makes use of the template for RPKI digitally signed objects [RFC6488], which defines a Crytopgraphic Message Syntax (CMS) [RFC5652] wrapper for the Signed TAL content as well as a generic validation procedure for RPKI signed objects. Therefore, to complete the specification of the Signed TAL (see Section 4 of [RFC6488]), this document defines:

- The OID defined in Section 3.1 that identifies the signed object as being a Signed TAL. (This OID appears within the eContentType in the encapContentInfo object as well as the content-type signed attribute in the signerInfo object).

- The ASN.1 syntax for the Signed TAL eContent defined in Section 3.2. (This is the payload that specifies the AS being authorized to originate routes as well as the prefixes to which the AS may originate routes.)

- Additional steps to the validation steps specified in [RFC6488] required to validate Signed TALs, defined in Section 3.3.

3.1. The Signed TAL Content Type

This document requests an OID for signed-Tal as follows:

```
signed-Tal OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
           rsadsi(113549) pkcs(1) pkcs9(9) 16 id-smime (1) TBD }
```

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

3.2. The Signed TAL eContent

The content of a Signed TAL is ASN.1 encoded using the Distinguished Encoding Rules (DER) [X.690], and is defined as follows:

```
SignedTAL ::= SEQUENCE {
   version           [0] INTEGER DEFAULT 0,
   activationTime        GeneralizedTime,
   certificateURIs       SEQUENCE SIZE (1..MAX) OF CertificateURI,
   subjectPublicKeyInfo  SubjectPublicKeyInfo }

CertificateURI ::= IA5String

SubjectPublicKeyInfo ::= IA5String
```

3.2.1. version

The version number of the SignedTAL MUST be 0.

3.2.2. activationTime

This field contains the time when this TAL is intended to replace any previously known TAL for this Trust Anchor.

3.2.3. certificateURIs

This field is equivalent to the URI section in section 2.1 of [I-D.ietf-sidrops-https-tal]. It MUST contain at least one CertificateURI element. Each CertificateURI element contains the IA5String representation of either an rsync URI [RFC5781], or an HTTPS URI [RFC7230].

3.2.4. subjectPublicKeyInfo

This field is equivalent to the subjectPublicKeyInfo section in section 2.1 of [I-D.ietf-sidrops-https-tal].

3.3. Signed TAL Validation

To determine whether a Signed TAL is valid, the RP MUST perform the following steps in addition to those specified in [RFC6488]:

- The eContentType OID matches the OID described in Section 3.1

- The Signed TAL appears as the product of a Trust Anchor CA certificate.

- This Trust Anchor CA has published only one Signed TAL object in its repository, and this object appears on the Manifest as the only entry using the ".tal" extension (see [RFC6481]). In case more than one Signed TAL object is found, all such objects MUST be considered invalid.

- The EE certificate of this Signed TAL describes its Internet Number Resources (INRs) using the "inherit" attribute

- The decoded TAL content conforms to the format defined in Section 3.2.

If the above procedure indicates that the manifest is invalid, then the Signed TAL MUST be discarded and treated as though no Signed TAL were present.

4. Signed TAL Generation

A TA MAY choose to generate a single Singed TAL object to publish in its TA certificate publication point(s) in the RPKI. The TA MUST perform the following steps to generate the Signed TAL:

- Generate a key pair for a "one-time-use" EE certificate to use for the Signed TAL

- Generate a one-time-use EE certificate for the Signed TAL

- This EE certificate MUST have an SIA extension access description field with an accessMethod OID value of id-ad-signedobject, where the associated accessLocation references the publication point of the Sigend TAL as an object URL.

- As described in [RFC6487], an [RFC3779] extension is required in the EE certificate used for this object. However, because the resource set is irrelevant to this object type, this certificate MUST describe its Internet Number Resources (INRs) using the "inherit" attribute, rather than explicit description of a resource set.

- This EE certificate MUST have a "notBefore" time that is before the moment that the Signed TAL will be published.

- This EE certificate MUST have a "notAfter" time that reflects the intended time that this Signed TAL will be published. If the EE certificate for a Signed TAL is expired, it MUST no longer be publish, but of course it MAY be replaced by a newly generated Signed TAL object with similar content and an updated "notAfter" time.

- The Signed TAL MUST have an "activationTime" that reflects when Relying Parties MUST use use this new TAL in place of any previously known TAL for this Trust Anchor.

5. Signed TAL Publication

A TA MAY publish a single Signed TAL object directly under its CA repository publication points. The TA MUST NOT publish multiple Signed TAL objects at any time. It is RECOMMENDED that a TA publishes a Signed TAL object for its current key and CA certificate publication URIs at all times.

A non-normative guideline for naming this object is that the filename chosen for the signed TAL in the publication repository be a value derived from the public key part of the entity's key pair, using the algorithm described for CRLs in section 2.2 of [RFC6481] for generation of filenames. The filename extension of ".tal" MUST be used to denote the object as a signed TAL. Note that this is in-line with filename extensions defined in section 7.2 of [RFC6481]

6. Performing a planned Key Roll as a Trust Anchor

A Signed TAL SHOULD be used to communicate a planned key roll by a Trust Anchor. From the Trust Anchor perspective a planned key roll consists of the following steps:

- Prepare a new Trust Anchor key and CA certificate, see Section 6.1

- Publish the new CA certificate, see Section 6.2

- Verify the validity of the new CA certificate, see Section 6.3

- Publish the objects under the current key under the new key, see Section 6.4

- Verify that the validity of objects under the new key, see Section 6.5

- Publish a Signed TAL as the only object under the current key, see Section 6.6

- Delete the current key, see Section 6.7

6.1. Prepare a new Trust Anchor key and CA certificate

The Trust Anchors MUST a new key pair and generate a new TA Certificate. For the Subject Information Access (see section 4.8.8.1 of [RFC6487]) this MUST use URIs that will be used by the new key to publish objects. These URIs MUST be uniqe for use by this new key only. The Internet Number Resources on this new certificate MUST be equivalent to those found on the current certificate.

6.2. Publish the new CA certificate

The new CA certificate MUST be published under one or more new Certificate URIs for use by this new key only.

6.3. Verify the validity of the new CA certificate

The Trust Anchor MUST generate a new (unsigned) TAL file [I-D.ietf-sidrops-https-tal] and verify with RP software that the new Trust Anchor certificate can be retrieved from all locations and that it matches the subjectPublicKeyInfo

6.4. Publish the objects under the current key under the new key

ALL current signed certificates and other objects, with the exception of the CRL, Manifest and existing Signed TAL, must be re-issued by the new key and published under the new publication point(s).

It is RECOMMENDED that a new Signed TAL object is generated and published, listing the Certificate URIs for this new key, the subjectPublicKeyInfo of this new key, and using an "activationTime" that is effective immediately. Note that Relying Parties will not discover this new Signed TAL object until they have effectively switched over from the current key.

6.5. Verify that the validity of objects under the new key

The Trust Anchor MUST verify that validation using the new TAL file generated in Section 6.3 results in the set of valid objects as when the current TAL file is used.

6.6. Publish a Signed TAL as the only object under the current key

The Trust Anchor MUST publish a new Signed TAL, CRL and Manifest as the only objects under the current, to be deleted, key. The "nextUpdate" values of the Manifest and CRL objects SHOULD use a date that is set at least one year into the future. (arbitrary value, open to suggestions). The "notValidAfter" date on the Manifest and Signed TAL EE certificate SHOULD use this same date. The Trust Anchor MUST ensure that this Signed TAL, CRL and Manifest remain available for download for this full period. Note that this is done to give RPs the opportunity to discover the new key up to one year after the key roll occurred.

6.7. Delete the current key

As the final step the current key, which has been replaced now, SHOULD be deleted. The new key can now be marked as the current key.

7. Relying Party Use

When an RP discovers a valid Signed TAL signed under a TA, and it notices that the "subjectPublicKeyInfo" has changed and/or the set of "Certificate URIs" has changed from the values it knew for this TA, and the "activationTime" is in the past, then the RP MUST accept these new values for this TA, abort the current top-down validation operation, and initiate a new top-down validation operation using the updated information.

Note that the Trust Anchor MUST have verified that all objects are available under the new key (Section 6.5) and that that the TA CA certificate can be retrieved and validated for all new URIs (Section 6.3).

8. Deployment Considerations

Including Signed TAL objects while RP tools do not support this standard will result in these RPs rejecting these objects. It is not expected that this will result in the invalidation of any other object under a Trust Anchor.

That said, the flagging mechanism introduced here can only be trusted on once a majority of RPs support it. Defining when that moment arrives is by definition something that cannot be established at the time of writing this document.

However, once the majority of RPs support this mechanism it would be RECOMMENDED that Trust Anchor operators perform key rolls regularly.

The most assured way to know that such planned rolls will work is by making them a part of normal operations.

9. Unplanned Key Roll operations

The mechanism described in this document is not applicable to uplanned key rolls. Unplanned key rolls could theoretically be supported by a mechanism where a new key is introduced before it's used, with the power to revoke the current key. This would have to be signalled from the new key, as the TA may have lost access to its current key.

However, this introduces a great amount of operational complexity as well as a new vulnerability: an adversary would need access to only one of these keys in order to compromise a TA.

With that in mind we believe, for now, that unplanned key rolls should not be covered here, and would need to be communicated to Relying Parties in some other out-of-band fashion.

10. Changing a Trust Anchor Certificate URIs

Earlier versions of this document included a description of how Signed TAL objects could be used to signal a change of Certificate URIs only; i.e. where the key is not changed.

However, Relying Parties that do not support the mechanism described in this document would not be able to learn about the changes in URIs. While for RPs that do support this mechanism a planned key roll will be a normal part of RPKI validation.

Therefore we believe that a planned key roll should be used in cases like this, and that the set of Certificate URIs for any given key must never be changed.

11. IANA Considerations

11.1. OID

IANA is to add the following to the "RPKI Signed Objects" registry:

Decimal	Description	References
TBD	signed-Tal	[section 3.1]

11.2. File Extension

IANA is to add an item for the Signed TAL file extension to the "RPKI Repository Name Scheme" created by [RFC6481] as follows:

Extension	RPKI Object	References
.tal	Signed TAL	[this document]

12. Security Considerations

TBD

13. Acknowledgements

TBD

14. References

14.1. Normative References

[I-D.ietf-sidrops-https-tal]
Huston, G., Weiler, S., Michaelson, G., Kent, S., and T. Bruijnzeels, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", draft-ietf-sidrops-https-tal-03 (work in progress), June 2018.

[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>.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <https://www.rfc-editor.org/info/rfc3779>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <https://www.rfc-editor.org/info/rfc5011>.

[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>.

[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>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/info/rfc6487>.

[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>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[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>.

[X.690] ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", 2002.

14.2. Informative References

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

Authors' Addresses

Tim Bruijnzeels
NLnet Labs

Email: tim@nlnetlabs.nl
URI: https://www.nlnetlabs.nl/

Carlos Martinez
LACNIC

Email: carlos@lacnic.net
URI: https://www.lacnic.net/