Network Working Group
Internet-Draft
Obsoletes: 7730 (if approved)
Intended status: Standards Track
Expires: January 27, 2019

G. Huston
APNIC
S. Weiler
W3C/MIT
G. Michaelson
APNIC
S. Kent
Unaffiliated
T. Bruijnzeels
NLnet Labs

July 26, 2018

Resource Public Key Infrastructure (RPKI) Trust Anchor Locator
draft-ietf-sidrops-https-tal-04

Abstract

This document defines a Trust Anchor Locator (TAL) for the Resource Public Key Infrastructure (RPKI). This document obsoletes RFC 7730 by adding support for HTTPS URIs in a TAL.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a Trust Anchor Locator (TAL) for the Resource Public Key Infrastructure (RPKI) [RFC6480]. This format may be used to distribute trust anchor material using a mix of out-of-band and online means. Procedures used by Relying Parties (RPs) to verify RPKI signed objects SHOULD support this format to facilitate interoperability between creators of trust anchor material and RPs. This document obsoletes [RFC7730] by adding support for HTTPS URIs in a TAL.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Trust Anchor Locator

2.1. Trust Anchor Locator Format

This document does not propose a new format for trust anchor material. A trust anchor in the RPKI is represented by a self-signed X.509 Certification Authority (CA) certificate, a format commonly used in PKIs and widely supported by RP software. This document

specifies a format for data used to retrieve and verify the authenticity of a trust anchor in a very simple fashion. That data is referred to as the TAL.

The motivation for defining the TAL is to enable selected data in the trust anchor to change, without needing to effect redistribution of the trust anchor per se. In the RPKI, certificates contain extensions that represent Internet Number Resources (INRs) [RFC3779]. The set of INRs associated with an entity acting as a trust anchor is likely to change over time. Thus, if one were to use the common PKI convention of distributing a trust anchor to RPs in a secure fashion, then this procedure would need to be repeated whenever the INR set for the entity acting as a trust anchor changed. By distributing the TAL (in a secure fashion), instead of distributing the trust anchor, this problem is avoided, i.e., the TAL is constant so long as the trust anchor's public key and its location do not change.

The TAL is analogous to the TrustAnchorInfo data structure specified in [RFC5914], which is on the Standards Track. That specification could be used to represent the TAL, if one defined an rsync or HTTPS URI extension for that data structure. However, the TAL format was adopted by RPKI implementors prior to the PKIX trust anchor work, and the RPKI implementer community has elected to utilize the TAL format, rather than define the requisite extension. The community also prefers the simplicity of the ASCII encoding of the TAL, versus the binary (ASN.1) encoding for TrustAnchorInfo.

The TAL is an ordered sequence of:

1. a URI section,

2. a "<CRLF>" or "<LF>" line break,

3. a subjectPublicKeyInfo [RFC5280] in DER format [X.509], encoded in Base64 (see Section 4 of [RFC4648]). To avoid long lines, "<CRLF>" or "<LF>" line breaks MAY be inserted into the Base64-encoded string.

where the URI section is comprised or one of more of the ordered sequence of:

1.1. either an rsync URI [RFC5781], or an HTTPS URI [RFC7230]

1.2. a "<CRLF>" or "<LF>" line break.

2.2. TAL and Trust Anchor Certificate Considerations

Each URI in the TAL MUST reference a single object. It MUST NOT reference a directory or any other form of collection of objects.

The referenced object MUST be a self-signed CA certificate that conforms to the RPKI certificate profile [RFC6487]. This certificate is the trust anchor in certification path discovery [RFC4158] and validation [RFC5280] [RFC3779].

The validity interval of this trust anchor SHOULD reflect the anticipated period of stability of the particular set of INRs that are associated with the putative trust anchor.

The INR extension(s) of this trust anchor MUST contain a non-empty set of number resources. It MUST NOT use the "inherit" form of the INR extension(s). The INR set described in this certificate is the set of number resources for which the issuing entity is offering itself as a putative trust anchor in the RPKI [RFC6480].

The public key used to verify the trust anchor MUST be the same as the subjectPublicKeyInfo in the CA certificate and in the TAL.

The trust anchor MUST contain a stable key. This key MUST NOT change when the certificate is reissued due to changes in the INR extension(s), when the certificate is renewed prior to expiration, or for any reason other than a key change.

Because the public key in the TAL and the trust anchor MUST be stable, this motivates operation of that CA in an offline mode. Thus, the entity that issues the trust anchor SHOULD issue a subordinate CA certificate that contains the same INRs (via the use of the "inherit" option in the INR extensions of the subordinate certificate). This allows the entity that issues the trust anchor to keep the corresponding private key of this certificate offline, while issuing all relevant child certificates under the immediate subordinate CA. This measure also allows the Certificate Revocation List (CRL) issued by that entity to be used to revoke the subordinate CA certificate in the event of suspected key compromise of this online operational key pair that is potentially more vulnerable.

The trust anchor MUST be published at a stable URI. When the trust anchor is reissued for any reason, the replacement CA certificate MUST be accessible using the same URI.

Because the trust anchor is a self-signed certificate, there is no corresponding CRL that can be used to revoke it, nor is there a manifest [RFC6486] that lists this certificate.

If an entity wishes to withdraw a self-signed CA certificate as a putative trust anchor, for any reason, including key rollover, the entity MUST remove the object from the location referenced in the TAL.

Where the TAL contains two or more URIs, then the same self- signed CA certificate MUST be found at each referenced location. In order to increase operational resilience, it is RECOMMENDED that the domain name parts of each of these URIs resolve to distinct IP addresses that are used by a diverse set of repository publication points, and these IP addresses be included in distinct Route Origin Authorizations (ROAs) objects signed by different CAs.

2.3. Example

```
rsync://rpki.example.org/rpki/hedgehog/root.cer
https://rpki.example.org/rpki/hedgehog/root.cer
```


MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAovWQL2lh6knDx
GUG5hbtCXvvh4AOzjhDkSHlj22gn/1oiM9IeDATIwP44vhQ6L/xvuk7W6
Kfa5ygmqQ+xOZOwTWPcrUbqaQyPNxokuivzyvqVZVDecOEqs78q58mSp9
nbtxmLRW7B67SJCBSzfa5XpVyXYEgYAjkk3fpmefU+AcxtxvvHB5OVPIa
BfPcs80ICMgHQX+fphvute9XLxjfJKJWkhZqZ0v7pZm2uhkcPx1PMGcrG
ee0WSDC3fr3erLueagpiLsFjwwpX6F+Ms8vqz45H+DKmYKvPSstZjCCq9
aJ0qANT9OtnfSDOS+aLRPjZryCNyvvBHxZXqj5YCGKtwIDAQAB

3. Relying Party Use

In order to use the TAL to retrieve and validate a (putative) trust anchor, an RP SHOULD:

1. Retrieve the object referenced by (one of) the URI(s) contained in the TAL.

2. Confirm that the retrieved object is a current, self-signed RPKI CA certificate that conforms to the profile as specified in [RFC6487].

3. Confirm that the public key in the TAL matches the public key in the retrieved object.

4. Perform other checks, as deemed appropriate (locally), to ensure that the RP is willing to accept the entity publishing this self-signed CA certificate to be a trust anchor. These tests apply to the validity of attestations made in the context of the RPKI relating to all resources described in the INR extension of this certificate.

An RP SHOULD perform these functions for each instance of TAL that it is holding for this purpose every time the RP performs a resynchronization across the local repository cache. In any case, an RP also SHOULD perform these functions prior to the expiration of the locally cached copy of the retrieved trust anchor referenced by the TAL.

In the case where a TAL contains multiple URIs, an RP MAY use a locally defined preference rule to select the URI to retrieve the self-signed RPKI CA certificate that is to be used as a trust anchor. Some examples are:

- Using the order provided in the TAL
- Selecting the URI randomly from the available list
- Creating a prioritized list of URIs based on RP-specific parameters, such as connection establishment delay

If the connection to the preferred URI fails, or the retrieved CA certificate public key does not match the TAL public key, the RP SHOULD retrieve the CA certificate from the next URI, according to the local preference ranking of URIs.

4. HTTPS Considerations

Note that a Man in the Middle (MITM) cannot produce a CA certificate that would be considered valid according to the process described in Section 3. However, a MITM attack can be performed to prevent the Relying Party from learning about an updated CA certificate. Because of this, Relying Parties SHOULD do TLS certificate and host name validation when they fetch a CA certificate using an HTTPS URI on a TAL.

Relying Party tools SHOULD log any TLS certificate or host name validation issues found, so that an operator can investigate the cause. However, such validation issues are often due to configuration errors or a lack of a common TLS trust anchor. In these cases, it is better if the Relying Party retrieves the CA certificate regardless and performs validation on it. Therefore, the Relying Party MUST continue to retrieve the data in case of errors.

It is RECOMMENDED that Relying Parties and Repository Servers follow the Best Current Practices outlined in [RFC7525] on the use of HTTP over TLS (HTTPS) [RFC7230]. Relying Parties SHOULD do TLS certificate and host name validation using subjectAltName dNSName identities as described in [RFC6125]. The rules and guidelines defined in [RFC6125] apply here, with the following considerations:

- o Relying Parties and Repository Servers SHOULD support the DNS-ID identifier type. The DNS-ID identifier type SHOULD be present in Repository Server certificates.

- o DNS names in Repository Server certificates SHOULD NOT contain the wildcard character "*".

- o A Common Name (CN) field may be present in a Repository Server certificate's subject name but SHOULD NOT be used for authentication within the rules described in [RFC6125].

- o This protocol does not require the use of SRV-IDs.

- o This protocol does not require the use of URI-IDs.

Note, however, that this validation is done on a best-effort basis and serves to highlight potential issues, but CA certificate validation in relation to a TAL as described in Section 3 does not depend on this. Therefore, Relying Parties MAY deviate from the validation steps listed above.

5. Security Considerations

Compromise of a trust anchor private key permits unauthorized parties to masquerade as a trust anchor, with potentially severe consequences. Reliance on an inappropriate or incorrect trust anchor has similar potentially severe consequences.

This TAL does not directly provide a list of resources covered by the referenced self-signed CA certificate. Instead, the RP is referred to the trust anchor itself and the INR extension(s) within this certificate. This provides necessary operational flexibility, but it also allows the certificate issuer to claim to be authoritative for any resource. Relying parties should either have great confidence in the issuers of such certificates that they are configuring as trust anchors, or they should issue their own self-signed certificate as a trust anchor and, in doing so, impose constraints on the subordinate certificates.

6. Acknowledgements

This approach to trust anchor material was originally described by Robert Kisteleki.

The authors acknowledge the contributions of Rob Austein and Randy Bush, who assisted with drafting this document and with helpful review comments.

The authors acknowledge with work of Roque Gagliano, Terry Manderson, and Carlos Martinez Cagnazzo in developing the ideas behind the inclusion of multiple URIs in the TAL.

7. References

7.1. 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>.

[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>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[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>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[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>.

[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>.

[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>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[RFC7730] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 7730, DOI 10.17487/RFC7730, January 2016, <https://www.rfc-editor.org/info/rfc7730>.

[X.509] TU-T Recommendation X.509, "The Directory: Public-key and attribute certificate frameworks", October 2012.

7.2. Informative References

[RFC4158] Cooper, M., Dzambasow, Y., Hesse, P., Joseph, S., and R. Nicholas, "Internet X.509 Public Key Infrastructure: Certification Path Building", RFC 4158, DOI 10.17487/RFC4158, September 2005, <https://www.rfc-editor.org/info/rfc4158>.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <https://www.rfc-editor.org/info/rfc5914>.

[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>.

Authors' Addresses

Geoff Huston
APNIC

Email: gih@apnic.net
URI: https://www.apnic.net

Samuel Weiler
W3C/MIT

Email: weiler@csail.mit.edu

George Michaelson
APNIC

Email: ggm@apnic.net
URI: https://www.apnic.net

Stephen Kent
Unaffiliated

Email: kent@alum.mit.edu

Tim Bruijnzeels
NLnet Labs

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