        Resource Public Key Infrastructure (RPKI) Trust Anchor Locator

Abstract

   This document defines a Trust Anchor Locator (TAL) for the Resource
   Public Key Infrastructure (RPKI).

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 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6490.

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.

1.1. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" 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 likely will change over
   time. Thus, if one were to use the common PKI convention of

distributing a trust anchor to RPs in a secure fashion, this
procedure would need to be repeated whenever the INR set for the
trust anchor changed. By distributing the TAL (in a secure fashion)
instead of 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 [RFC5914]
adopted as a PKIX standard. That standard could be used to represent
the TAL, if one defined an rsync 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) An rsync URI [RFC5781],
2) A <CRLF> or <LF> line break, and
3) A subjectPublicKeyInfo [RFC5280] in DER format [X.509], encoded in
   Base64 (see Section 4 of [RFC4648]).

2.2. TAL and Trust Anchor Certificate Considerations

The rsync 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 for 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 off-line 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 off-line,
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
potentially more vulnerable online operational key pair.

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.

2.3. Example

    rsync://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 the URI 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 checks 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
   re-synchronization 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.

4. 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. For more information on this approach, see [TA-MGMT].

5. Acknowledgments

   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 earlier draft versions of this document and
   with helpful review comments.

6. References

6.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP
              Addresses and AS Identifiers", RFC 3779, June 2004.

   [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data
              Encodings", RFC 4648, October 2006.

   [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, May 2008.

   [RFC5781] 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, May 2008.

   [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for
              X.509 PKIX Resource Certificates", RFC 6487, February
              2012.

   [X.509]   ITU-T, "Recommendation X.509: The Directory -
              Authentication Framework", 2000.

6.2. 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, September 2005.

   [RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor
              Format", RFC 5914, June 2010.

   [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support
              Secure Internet Routing", RFC 6480, February 2012.

   [RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski,
              "Manifests for the Resource Public Key Infrastructure
              (RPKI)", RFC 6486, February 2012.

   [TA-MGMT] Reynolds, M. and S. Kent, "Local Trust Anchor Management
              for the Resource Public Key Infrastructure", Work in
              Progress, December 2011.

Authors' Addresses

   Geoff Huston
   APNIC

   EMail: gih@apnic.net
   URI: http://www.apnic.net

   Samuel Weiler
   SPARTA, Inc.
   7110 Samuel Morse Drive
   Columbia, Maryland 21046
   USA

   EMail: weiler@tislabs.com

   George Michaelson
   APNIC

   EMail: ggm@apnic.net
   URI: http://www.apnic.net

   Stephen Kent
   BBN Technologies
   10 Moulton St.
   Cambridge, MA 02138
   USA

   EMail: kent@bbn.com