Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 30, 2018

A. Azimov
E. Uskov
Qrator Labs
R. Bush
Internet Initiative Japan
K. Patel
Arrcus
J. Snijders
NTT
R. Housley
Vigil Security
June 28, 2018

A Profile for Autonomous System Provider Authorization
draft-azimov-sidrops-aspa-profile-00

Abstract

This document defines a standard profile for Autonomous System Provider Authorization in the Resource Public Key Infrastructure. An Autonomous System Provider Authorization is a digitally signed object that provides a means of verifying that a Customer Autonomous System holder has authorized a Provider Autonomous System to be its upstream provider and for the Provider to send prefixes received from the Customer Autonomous System in all directions including providers and peers.

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

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2018.

Copyright Notice

Table of Contents

1. Introduction

The primary purpose of the Resource Public Key Infrastructure (RPKI) is to improve routing security. (See [RFC6480] for more information.) As part of this infrastructure, a mechanism is needed to verify that a Provider AS (PAS) has permission from a Customer AS (CAS) holder to send routes in all directions. The digitally signed Autonomous System Provider Authorization (ASPA) object provides this verification mechanism.

The ASPA uses the template for RPKI digitally signed objects [RFC6488], which defines a Cryptographic Message Syntax (CMS) [RFC5652] wrapper for the ASPA content as well as a generic validation procedure for RPKI signed objects. As ASPAs need to be validated with RPKI certificates issued by the current infrastructure, we assume the mandatory-to-implement algorithms in [RFC6485], or its successor.

To complete the specification of the ASPA (see Section 4 of [RFC6488]), this document defines:

1. The object identifier (OID) that identifies the ASPA signed object. This OID appears in the eContentType field of the encapContentInfo object as well as the content-type signed attribute within the signerInfo structure).

2. The ASN.1 syntax for the ASPA content, which is the payload signed by the CAS. The ASPA content is encoded using the ASN.1 [X680] Distinguished Encoding Rules (DER) [X690].

3. The steps required to validate an ASPA beyond the validation steps specified in [RFC6488]).

2. The ASPA Content Type

The content-type for an ASPA is defined as id-cct-ASPA, which 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 structure as well as the content-type signed attribute within the signerInfo structure (see [RFC6488]).

3. The ASPA eContent

The content of an ASPA identifies the Customer AS (CAS) as well as the Provider AS (PAS) that is authorized to further propagate announcements received from the customer. If customer has multiple providers, it issues multiple ASPAs, one for each provider AS. An ASPA is formally defined as:

```
ct-ASPA CONTENT-TYPE ::=
    { ASProviderAttestation IDENTIFIED BY id-ct-ASPA }

id-ct-ASPA OBJECT IDENTIFIER ::= { id-ct TBD }

ASProviderAttestation ::= SEQUENCE {
    version [0] ASPAVersion DEFAULT v0,
    AFI  AddressFamilyIdentifier,
    customerASID  ASID,
    providerASID  ASID }

ASPAVersion ::= INTEGER  { v0(0) }

AddressFamilyIdentifier ::= INTEGER

ASID ::= INTEGER
```

Note that this content appears as the eContent within the encapContentInfo as specified in [RFC6488].

3.1. version

The version number of the ASProviderAttestation MUST be v0.

3.2. AFI

The AFI field contains Address Family Identifier for which the relation between customer and provider ASes is authorized. Presently defined values for the Address Family Identifier field are specified in the IANA’s Address Family Numbers registry [IANA-AF].

3.3. customerASID

The customerASID field contains the AS number of the Autonomous System that authorizes an upstream provider (listed in the providerASId) to propagate prefixes in the specified address family other ASes.

3.4. providerASID

The providerASID contains the AS number that is authorized to further propagate announcements in the specified address family received from the customer.

4. ASPA Validation

Before a relying party can use an ASPA to validate a routing announcement, the relying party MUST first validate the ASPA object itself. To validate an ASPA, the relying party MUST perform all the validation checks specified in [RFC6488] as well as the following additional ASPA-specific validation step.

- The autonomous system identifier delegation extension [RFC3779] is present in the end-entity (EE) certificate (contained within the ASPA), and the customer AS number in the ASPA is contained within the set of AS numbers specified by the EE certificate’s autonomous system identifier delegation extension.

5. ASN.1 Module for the ASPA Content Type

```
RPKI-ASPA-2018
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
       pkcs-9(9) smime(16) modules(0) id-mod-rpki-aspa-2018(TBD2) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS

CONTENT-TYPE
FROM CryptographicMessageSyntax-2010  -- RFC 6268
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
       pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) } ;

ContentSet CONTENT-TYPE ::= { ct-ASPA, ... }

--
-- ASPA Content Type
--

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

id-ct OBJECT IDENTIFIER ::= { id-smime 1 }

id-ct-ASPA OBJECT IDENTIFIER ::= { id-ct TBD }

ct-ASPA CONTENT-TYPE ::=
    { TYPE ASProviderAttestation IDENTIFIED BY id-ct-ASPA }

ASProviderAttestation ::= SEQUENCE {
    version [0] ASPAVersion DEFAULT v0,
    AFI  AddressFamilyIdentifier,
    customerASID  ASID,
    providerASID  ASID }

ASPAVersion ::= INTEGER  { v0(0) }

AddressFamilyIdentifier ::= INTEGER

ASID ::= INTEGER

END
```

6. IANA Considerations

Please add the id-mod-rpki-aspa-2018 to the SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0) registry (https://www.iana.org/assignments/smi-numbers/smi-numbers.xml#security-smime-0) as follows:

Decimal	Description	Specification
TBD2	id-mod-rpki-aspa-2018	[ThisRFC]

Please add the ASPA to the SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1) registry (https://www.iana.org/assignments/smi-numbers/smi-numbers.xml#security-smime-1) as follows:

Decimal	Description	Specification
TBD	id-ct-ASPA	[ThisRFC]

Please add the ASPA to the RPKI Signed Object registry (https://www.iana.org/assignments/rpki/rpki.xhtml#signed-objects) as follows:

Name	OID	Specification
ASPA	1.2.840.113549.1.9.16.1.TBD	[ThisRFC]

7. Security Considerations

8. Acknowledgments

9. References

9.1. Normative References

[IANA-AF] IANA, "Address Family Numbers", <http://www.iana.org/numbers.html>.

[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>.

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

[RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, DOI 10.17487/RFC6485, February 2012, <https://www.rfc-editor.org/info/rfc6485>.

[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>.

[X680] ITU-T, "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, 2015.

[X690] ITU-T, "Information Technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, 2015.

9.2. Informative References

[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>.

Authors' Addresses

Alexander Azimov
Qrator Labs

Email: aa@qrator.net

Eugene Uskov
Qrator Labs

Email: eu@qrator.net

Randy Bush
Internet Initiative Japan

Email: randy@psg.com

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Job Snijders
NTT Communications
Theodorus Majofskistraat 100
Amsterdam 1065 SZ
The Netherlands

Email: job@ntt.net

Russ Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA

Email: housley@vigilsec.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

A. Azimov
E. Bogomazov
Qrator Labs
R. Bush
Internet Initiative Japan
K. Patel
Arrcus, Inc.
J. Snijders
NTT

October 22, 2018

Verification of AS_PATH Using the Resource Certificate Public Key Infrastructure and Autonomous System Provider Authorization

draft-azimov-sidrops-aspa-verification-01

Abstract

This document defines the semantics of an Autonomous System Provider Authorization object in the Resource Public Key Infrastructure to verify the AS_PATH attribute of routes advertised in the Border Gateway Protocol.

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

The Border Gateway Protocol (BGP) was designed with no mechanisms to validate BGP attributes. Two consequences are BGP Hijacks and BGP Route Leaks [RFC7908]. BGP extensions are able to partially solve these problems. For example, ROA-based Origin Validation [RFC6483] can be used to detect and filter accidental mis-originations, and [I-D.ymbk-idr-bgp-eotr-policy] can be used to detect accidental route leaks. While these upgrades to BGP are quite useful, they still rely on transitive BGP attributes, i.e. AS_PATH, that can be manipulated by attackers.

BGPSec [RFC8205] was designed to solve the problem of AS_PATH validation. Unfortunately, strict cryptographic validation brought unaffordable computational overhead for BGP routers. BGPSec also proved to be vulnerable to downgrade attacks that can nullify all the work of AS_PATH signing. As a result, to abuse the AS_PATH or any

other signed transit attribute, an attacker merely needs to downgrade to 'old' BGP-4.

An alternative approach was introduced with soBGP [I-D.white-sobgp-architecture]. Instead of strong cryptographic AS_PATH validation, it was suggested to create an AS_PATH security function based on a shared database of ASN adjacencies. While such an approach has reasonable computational cost, the two side adjacencies don't provide a way to automate anomaly detection without high adoption rate - an attacker can easily up a one-way adjacency. SO-BGP suggested sharing data about adjacencies using additional BGP messages, which is recursively complex thus significantly increasing adoption complexity. In addition, the general goal to verify all AS_PATHs was not achievable given the indirect adjacencies at internet exchange points.

Instead of the general goal of checking AS_PATH correctness, this document focuses on solving real-world operational problems - automatic detection of malicious hijacks and route leaks. To achieve this goal a new AS_PATH verification procedure is defined which is able to automatically detect invalid (malformed) AS_PATHs in announcements that are received from customers and peers. This procedure uses a shared signed database of customer-to-provider relationships that is built using a new RPKI object - Autonomous System Provider Authorization (ASPA). This technique provides benefits for the participants even in a state of early adoption.

2. Anomaly Propagation

Both route leaks and hijacks have similar effects on ISP operations - they redirect traffic, resulting in increased latency, packet loss, or possible MiTM attacks. But the level of risk depends significantly on the propagation of these BGP anomalies. For example, a hijack that is propagated only to customers may concentrate traffic in a particular ISP's customer cone; while if the anomaly is propagated through peers, upstreams, or reaches Tier-1 networks, thus distributing globally, traffic may be redirected at the level of entire countries and/or global providers.

The ability to constrain propagation of BGP anomalies to upstreams and peers, without requiring support from the source of the anomaly (which is critical if source has malicious intent), should significantly improve the security of inter-domain routing and solve the majority of problems.

3. Autonomous System Provider Authorization

As described in [RFC6480], the RPKI is based on a hierarchy of resource certificates that are aligned to the Internet Number Resource allocation structure. Resource certificates are X.509 certificates that conform to the PKIX profile [RFC5280], and to the extensions for IP addresses and AS identifiers [RFC3779]. A resource certificate is a binding by an issuer of IP address blocks and Autonomous System (AS) numbers to the subject of a certificate, identified by the unique association of the subject's private key with the public key contained in the resource certificate. The RPKI is structured so that each current resource certificate matches a current resource allocation or assignment.

ASPAs are digitally signed objects that bind a selected AFI Provider AS number to a Customer AS number (in terms of BGP announcements not business), and are signed by the holder of the Customer AS. An ASPA attests that a Customer AS holder (CAS) has authorized a particular Provider AS (PAS) to propagate the Customer's IPv4/IPv6 announcements onward, e.g. to the Provider's upstream providers or peers. The ASPA record profile is described in [I-D.azimov-sidrops-aspa-profile].

4. Customer-Provider Verification Procedure

This section describes an abstract procedure that checks that pair of ASNs (AS1, AS2) is included in the set of signed ASPAs. The semantics of its usa are defined in next section. The procedure takes (AS1, AS2, ROUTE_AFI) as input parameters and returns three types of results: "valid", "invalid" and "unknown".

A relying party (RP) must have access to a local cache of the complete set of cryptographically valid ASPAs when performing customer-provider verification procedure.

1. Retrieve all cryptographically valid ASPAs in a selected AFI with a customer value of AS1. This selection forms the set of "candidate ASPAs."

2. If the set of candidate ASPAs is empty, then the procedure exits with an outcome of "unknown."

3. If there is at least one candidate ASPA where the provider field is AS2, then the procedure exits with an outcome of "valid."

4. Otherwise, the procedure exits with an outcome of "invalid."

Since an AS1 may have different set providers in different AFI, it should also have different set of corresponding ASPAs. In this case,

the output of this procedure with input (AS1, AS2, ROUTE_AFI) may have different output for different ROUTE_AFI values.

5. AS_PATH Verification

The AS_PATH attribute identifies the autonomous systems through which an UPDATE message has passed. AS_PATH may contain two types of components: ordered AS_SEQes and unordered AS_SETs, as defined in [RFC4271].

The value of each AS_SEQ component can be described as set of pairs {(AS(I), prepend(I)), (AS(I-1), prepend(I-1))...}. In this case, the sequence {AS(I), AS(I-1),...} represents different ASNs, that packet should pass towards the destination. When a route is received from a customer or a literal peer, each pair (AS(I-1), AS(I)) MUST belong to customer-provider or sibling relationship. If there are other types of relationships, it means that the route was leaked or the AS_PATH attribute was malformed. The goal of the above procedure is to check the correctness of this statement.

For 32-bit AS number compatible BGP speakers, if a route from ROUTE_AFI address family is received from a customer or peer, its AS_PATH MUST be verified as follows:

1. If the closest AS in the AS_PATH is not the receiver’s neighbor ASN then procedure halts with the outcome "invalid";

2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)), and customer-provider verification procedure (Section 4) with parameters (AS(I-1), AS(I), ROUTE_AFI) returns "invalid" then the procedure also halts with the outcome "invalid";

3. If the AS_PATH has at least one AS_SET segment then procedure halts with the outcome "unverifiable";

4. Otherwise, the procedure halts with an outcome of "valid".

For BGP speakers that are not 32-bit AS compatible, the above procedure is slightly different. In point 2 if at least one AS(I-1), AS(I) is equal to AS_TRANS(23456), the corresponding pair must be passed without check using the customer-provider verification procedure.

If the output of the AS_PATH verification procedure is "invalid" the LOCAL_PREF SHOULD be set to 0 or the route MAY be dropped. If an "invalid" route has no alternative route(s) and it is propagated to other ASes despite the above, it MUST be marked with the GRACEFUL_SHUTDOWN community to avoid possible stable oscillations,

when an unchecked route received from a provider becomes preferred over an invalid route received from a customer. This also allows customers to detect malformed routes received from upstream providers.

If the output of the AS_PATH verification procedure is 'unverifiable' it means that AS_PATH can't be fully verified. Such routes should be treated with caution and SHOULD be processed the same way as "invalid" routes. This policy goes with full correspondence to [I-D.kumari-deprecate-as-set-confed-set].

The above AS_PATH verification procedure is able to check routes received from customers and peers. The ASPA mechanism combined with BGP Roles [I-D.ietf-idr-bgp-open-policy] and ROA-based Origin Validation [RFC6483] provide a fully automated solution to detect and filter hijacks and route leaks, including malicious ones.

6. Disavowal of Provider Authorizaion

An ASPA is a positive attestation that an AS holder has authorized its provider to redistribute received routes to the provider's providers and peers. This does not preclude the provider AS from redistribution to its other customers. By creating an ASPA where the provider AS is 0, the customer indicates that no provider should further announce its routes. Specifically, AS 0 is reserved to identify provider-free networks, Internet exchange meshes, etc.

An ASPA with a provider AS of 0 is a statement by the customer AS that the its routes should not be received by any relying party AS from any of its customers or peers.

By convention, an ASPA with a provider AS of 0 should be the only ASPA issued by a given AS holder; although this is not a strict requirement. A provider 0 ASPA may coexist with ASPAs that have different provider AS values; though in such cases, the presence or absence of the provider AS 0 ASPA does not alter the AS_PATH verification procedure.

7. Siblings (Complex Relations)

There are peering relationships which can not be described as strictly simple peer-peer or customer-provider; e.g. when both parties are intentionally sending prefixes received from each other to their peers and/or upstreams.

In this case, two symmetric ASPAs records {(AS1, AS2), (AS2, AS1)} must be created by AS1 and AS2 respectively.

8. Security Considerations

ASPA issuers should be aware of the verification implication in issuing an ASPA - an ASPA implicitly invalidates all routes passed to upstream providers other than the provider ASs listed in the collection of ASPAs. It is the Customer AS's duty to maintain a correct set of ASPAs.

While the ASPA provides a check of an AS_PATH for routes received from customers and peers, it doesn't provide full support for routes that are received from upstream providers. So, this mechanism guarantees detection of both malicious and accidental route leaks and provides partial support for detection of malicious hijacks: upstream transit ISPs will still be able to send hijacked prefixes with malformed AS_PATHs to their customers.

9. Acknowledgments

The authors wish to thank authors of [RFC6483] since its text was used as an example while writing this document.

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>.

[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.azimov-sidrops-aspa-profile] Azimov, A., Uskov, E., Bush, R., Patel, K., Snijders, J., and R. Housley, "A Profile for Autonomous System Provider Authorization", draft-azimov-sidrops-aspa-profile-00 (work in progress), June 2018.

[I-D.ietf-idr-bgp-open-policy] Azimov, A., Bogomazov, E., Bush, R., Patel, K., and K. Sriram, "Route Leak Prevention using Roles in Update and Open messages", draft-ietf-idr-bgp-open-policy-02 (work in progress), January 2018.

[I-D.kumari-deprecate-as-set-confed-set]
Kumari, W. and K. Sriram, "Deprecation of AS_SET and AS_CONFED_SET in BGP", draft-kumari-deprecate-as-set-confed-set-12 (work in progress), July 2018.

[I-D.white-sobgp-architecture]
White, R., "Architecture and Deployment Considerations for Secure Origin BGP (soBGP)", draft-white-sobgp-architecture-02 (work in progress), June 2006.

[I-D.ymbk-idr-bgp-eotr-policy]
Azimov, A., Bogomazov, E., Bush, R., and K. Patel, "Route Leak Detection and Filtering using Roles in Update and Open messages", draft-ymbk-idr-bgp-eotr-policy-02 (work in progress), March 2018.

[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>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[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>.

[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>.

[RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, DOI 10.17487/RFC6483, February 2012, <https://www.rfc-editor.org/info/rfc6483>.

[RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <https://www.rfc-editor.org/info/rfc7908>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

Authors' Addresses

Alexander Azimov
Qrator Labs

Email: aa@qrator.net

Eugene Bogomazov
Qrator Labs

Email: eb@qrator.net

Randy Bush
Internet Initiative Japan

Email: randy@psg.com

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Job Snijders
NTT Communications
Theodorus Majofskistraat 100
Amsterdam 1065 SZ
The Netherlands

Email: job@ntt.net

Internet Engineering Task Force (IETF)
Internet-Draft
Updates: 6811, 8097 (if approved)
Intended status: Standards Track
Expires: April 26, 2019

O. Borchert
D. Montgomery
USA NIST

October 23, 2018

RPKI Validation State Unverified
draft-borchert-sidrops-rpki-state-unverified-00

Abstract

In case operators decide not to evaluate BGP route prefixes according to RPKI origin validation, none of the available states as specified in RFC 6811 do properly represent this decision. This document introduces "Unverified" as well-defined validation state which allows to properly identify route prefixes as not evaluated according to RPKI route origin validation.

Copyright Notice

Table of Contents

1. Introduction

Prefix origin validation provides well-defined validation states. Though, there are instances in which no evaluation of a route prefix is performed, not through RPKI route origin validation [RFC6811], signaling via the extended community string as specified in [RFC8097], or operator configuration. In these circumstances RFC 6811 specifies the implementation SHOULD initialize the validation state of such route to "NotFound". Here, the absence of a well-defined validation state for a route prefix not evaluated, requires the usage of a state otherwise reserved as outcome of the evaluation of such. This "waters" down the meaning of the used state. The specification of a proper validation state that allows identifying non-evaluated routes, becomes of essence once an operator decides to write policies on the validation state "NotFound". A route prefix labeled "NotFound" cannot be considered same as an unverified route prefix.

Hence, this document updates RFC 6811 and RFC 8097 by adding the proposed validation state "Unverified".

1.1. Terminology

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. Suggested Reading

It is assumed that the reader understands BGP [RFC4271], the RPKI [RFC6480], Route Origin Authorizations (ROAs) [RFC6482], RPKI-based Prefix Validation [RFC6811], BGP Prefix Origin Validation State Extended Community [RFC8097], Clarifications to BGP Origin Validation Based on Resource Public Key Infrastructure (RPKI) [RFC8481]

3. Initializing route prefixes

This document introduces the validation state "Unverified" to be used for route prefixes that are not evaluated through either operator configuration, RPKI route origin validation, or other means such as receiving a signaled validation state via the extended community string. To allow proper initialization the following state is introduced:

- Unverified: Specifies the state of a route prefix on which no evaluation has been performed.

3.1. Update to RFC 6811

RFC 6811 specifies that:

> If validation is not performed on a Route, the implementation SHOULD initialize the validation state of such a route to "NotFound".

This document specifies that:

If no evaluation of a route prefix is performed in any form, the implementation MUST initialize the validation state of such a route to "Unverified".

This removes the necessity to initialize the route with any of the states "Valid", "Invalid", or "NotFound" and therefore does not "water-down" the meaning of such.

3.2. Update to RFC 8097

As specified in RFC 8097:

> If the router is configured to support the extensions defined in this document" - (RFC 8097) - ", it SHOULD attach the origin validation state extended community to BGP UPDATE messages sent to IBGP peers by mapping the computed validation state in the last octet of the extended community.

The missing part here is what to do with route prefixes not evaluated and no validation state was assigned. At this point the only solution is to omit the extended community for such routes. If the usage of the extended community would have been negotiated during the BGP OPEN MESSAGE the receiver would be able to determine that the sender did not evaluate the route in any form. But this is not the case, so a receiver does not know if the sender is RPKI capable and chose not to attach the origin validation state to the BGP UPDATE or the route did not have any validation state assigned.

Hence, this document specifies for all routes that are labeled as "Unverified" to attach the "unverified" state extended community to BGP UPDATE messages send to IBGP peers by mapping the computed validation state in the last octet of the extended community.

AS specified in the table below, this document adds the value "unverified = 3" to the list of acceptable values.

The value on the protocol

Value	Meaning
0	Lookup result = "valid"
1	Lookup result = "not found"
2	Lookup result = "invalid"
3	Lookup result = "unverified"

3. Usage Considerations

The well-defined validation state "Unverified" allows to distinguish between evaluated routes and non-evaluated routes. This allows the operator to create policies to treat such route prefixes different from route prefixes labeled with one of the validation states "Valid", "NotFound", or "Invalid".

4. Security Considerations

This document introduces no new security concerns beyond what is described in [RFC6811] and [RFC8097]

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. 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>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC8097] Mohapatra, P., Patel, K., Scudder, J., Ward, D., and R. Bush, "BGP Prefix Origin Validation State Extended Community", RFC 8097, DOI 10.17487/RFC8097, March 2017, <https://www.rfc-editor.org/info/rfc8097>.

[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>.

8.2. Informative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[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>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <https://www.rfc-editor.org/info/rfc6482>.

[RFC8481] Bush, R., "Clarifications to BGP Origin Validation Based on Resource Public Key Infrastructure (RPKI)", RFC 8481, DOI 10.17487/RFC8481, September 2018, <https://www.rfc-editor.org/info/rfc8481>.

Acknowledgements

The authors would like to acknowledge the valuable review and suggestions from K. Sriram on this document.

Authors' Addresses

Oliver Borchert
National Institute of Standards and Technology (NIST)
100 Bureau Drive
Gaithersburg, MD 20899
United States of America

Email: oliver.borchert@nist.gov

Douglas Montgomery
National Institute of Standards and Technology (NIST)
100 Bureau Drive
Gaithersburg, MD 20899
United States of America

Email: dougm@nist.gov

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

T. Bruijnzeels
NLnet Labs
C. Martinez
LACNIC
R. Austein
Dragon Research Labs
October 16, 2018

RPKI Signed Object for Trust Anchor Keys
draft-ietf-sidrops-signed-tal-02

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 for Trust Anchor Keys (TAK), that can be used by Trust Anchors to signal their set of current keys and the location(s) of the accompanying CA certiifcates to Relying Parties, as well as changes to this set in the form of revoked keys and new keys, in order to support both planned and unplanned key rolls without impacting RPKI validation.

Status of This Memo

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. Overview

Trust Anchor Locators (TALs) [I-D.ietf-sidrops-https-tal] are used by Relying Parties in the RPKI to locate and validate Trust Anchor (TA) certificates used in RPKI validation. However, until now there has been no formal way of notifying Relying Parties (RP) of updates to a TAL. Such updates may be needed in particular in case a Trust Anchor needs to perform a planned, or unplanned, key roll.

This document defines a new RPKI signed object that can be used to document the current set of keys and the loctaion(s) of the accompanying CA certificates, as well as any changes to this set. This allows RPs to be notified automatically of such changes, and enables Trust Anchors to pre-stage a number of operational keys so that planned and unplanned key rolls can be performed without risking the invalidation of the RPKI tree under the TA. We call this object the Trust Anchor Keys (TAK) object.

When Relying Parties (RPs) are first bootstrapped, they use any current TAL to discover a key and location(s) of the TA certificate(s) for a TA. The RP can then retreive and validating the TA certificate, and subsequently validate the manifest [RFC6486] and CRL [section 5 of @!RFC6487]. However, before processing any other objects it will then first validate the TAK object, if present. All enumarated new keys (and locations) are then added to a new list of current TA keys for this TA. The RP will then recursively fetch and validate the TA certificates, manifest, CRL and TAK objects for each of these keys. As a part of this process the RP will also compile a list of revoked keys enumarated by any of the validly signed TAK objects. As the final step the RP will then filter out any revoked TA keys from its new set. This new set now replaces the previous set.

If the key used to start this process is still considered current, then validation continues. But if the key was revoked, then validation is restarted using one of the remaining keys in the set.

This process allows Trust Anchors to operate a set of N current keys, where any key can effectively revoke any or all of the other keys to perform either a planned, or an unplanned, key roll. This also

allows Trust Anchors to produce long lived TAK objects as forward pointers to RPs, and retire its old key when doing a key roll.

While the generic process is quite involved, the amount of work needed to support an envisioned normal key roll is fairly limited. Under normal circumstances a TA will typically have two current keys, so that is can perform an emergency roll over in case one of the keys is lost. This means that the RP will need to validate two TAK objects. However, typically these files will agree that both keys are current and validation continues.

When a key roll is executed a TA will remove one old key, and introduce one new (back-up) key. The RP will remove the old key from its set, and it will not be queried again, and it will add the new key and its TA certifcate location(s).

Only in a situation where an RP is very outdated can it be expected that the RP will have to discover several chained TAK object. But, since it will remove the outdated TALs in this process, this presents a one time cost only.

Note that in theory a TA can revoke all of its keys and make itself obsolete. In practice however, a well operated TA will have measures in place to prevent this. Furthermore they can protect themselves against key loss to adversaries through the use of such as the use of a Hardware Security Module (HSM) to protect keys. Protecting against this mis-operation would incur complexity and guesswork on the RPs. Therefore it is believed that it is best to keep the process straightforward, and offer a solution for the more likely issues of loss of a key, e.g. because an HSM or card set is broken, and planned key rolls.

3. TAK Object definition

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

- The OID defined in Section 3.1 that identifies the signed object as being a TAK. (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 TAK eContent defined in Section 3.2.

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

3.1. The TAK Object Content Type

This document requests an OID for TAK objects 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 TAK Object eContent

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

```
TAK ::= SEQUENCE {
   version   INTEGER DEFAULT 0,
   current   ::= SEQUENCE SIZE (1..MAX) OF CurrentKey,
   revoked   ::= SEQUENCE OF SubjectPublicKeyInfo
}

CurrentKey ::= SEQUENCE {
   certificateURIs       SEQUENCE SIZE (1..MAX) OF CertificateURI,
   subjectPublicKeyInfo  SubjectPublicKeyInfo
}

CertificateURI ::= IA5String

SubjectPublicKeyInfo ::= SEQUENCE {
     algorithm            AlgorithmIdentifier,
     subjectPublicKey     BIT STRING
}
```

3.2.1. version

The version number of the TAK object MUST be 0.

3.2.2. current

This field defines the set of current keys (CurrentKey) according to the signer of this Signed TALs object.

3.2.2.1. CurrentKey

This field defines a current TA Key, equivalent to [I-D.ietf-sidrops-https-tal]. This structure contains a sequence of one or more URIs and a SubjectPublicKeyInfo.

3.2.2.1.1. 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].

3.2.2.1.2. subjectPublicKeyInfo

This field contains a SubjectPublicKeyInfo [section 4.1.2.7 or @!RFC5280] in DER format [X.690].

3.2.3. revoked

This field contains the list of keys, identified by SubjectPublicKeyInfo, that are no longer to be used according to the signer of this document.

3.3. TAK Object Validation

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

- o The eContentType OID matches the OID described in Section 3.1

- o The TAK object appears as the product of a Trust Anchor CA certificate.

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

- o The EE certificate of this TAK object describes its Internet Number Resources (INRs) using the "inherit" attribute

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

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

4. Maintaining multiple TA keys

As described in Section 6 a TA will most likely choose to operate two keys at any one time in order to be prepared for an emergency key roll. When a TA operates multiple keys, each key MUST use its own CA repository publication point as described in [RFC6481]. The CRL and Manifest [RFC6486] for each of these keys will be unique to each key, but the TA MUST ensure that equivalent CA certificates and RPKI signed objects are issued under each key. Note that this is similar to how such certificates and RPKI signed objects are re-issued as part of a lower level CA key roll, described in section 4 of [RFC6489].

4.1. Prepare a new TA key

The Trust Anchors MUST generate 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.

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

As decribed above, the TA MUST issue and publish equivalent CA certificates and RPKI signed objects under this new key.

It is RECOMMENDED that the TA now generates a new TAL [I-D.ietf-sidrops-https-tal] and verifies that the new Trust Anchor certificate can be retrieved from all locations, and that it generates the same results when it is used for top-down validation instead of (any of) the current TA key(s).

Note that the TA MAY choose to make this TAL available to Relying Parties, in particular to those that do not support TAK objects, and for inclusion in the distribution of RP software in order to minimise the overhead in bootstrapping fresh installations.

4.2. Publishing for Multiple TA Keys

If a TA uses a single remote publication server for its keys using the RPKI publication protocol [RFC8181], then it MUST include all <publish/> and <withdraw/> PDUs for the products of each of its keys

in a single query in order to ensure that they will reflect the same content at all times.

If a TA uses multiple publication servers then it is by definition inevitable that the content of different keys will be out of sync at times. In such cases the TA SHOULD ensure that the duration of these moments are limited to the shortest possible time. Furthermore the following should be observed:

- It is strongly RECOMMENDED that TAs do not issue any RPKI Signed Objects, such as ROAs [RFC6482], but limit their operations to maintaining a CRL, Manifest and CA certificates only. If an organisation maintaining a TA has an operational need for such objects then it is strongly RECOMMENDED that they operate a separate non-TA CA as a child of their TA for these operations. If this approach is used the remaining issues regarding temporary inconsistencies between multiple TA key repository publication points is greatly reduced.

- In cases where a CA certificate is revoked completely, or replaced by a certifcate with a reduced set of resources, these changes will not take effect fully until all the TA keys repository publication points have been updated. Given that TA key operations are normally performed infrequently we don’t expect that this is a problem. I.e. if the revocation or shrinking of an issued CA certificate is staged for days, or weeks anyway, then experiencing a delay of several minutes for the repository publication points to all be updated is fairly insignificant.

- In cases where a CA certificate is replaced by a certifcate with an extend set of resources the TA MUST inform the receiving CA only after all its repository publication points have been updated. This ensures that the receiveing CA will not issue any products that could be invalid if an RP uses a TA key just before the CA certificate was due to be updated.

5. TAK Object Generation and Publication

A TA MAY choose to use TAK objects to communicate its set of current, and revoked keys. If a TA chooses to use TAK objects, then it SHOULD generate and publish TAK objects under each of its current keys. An exception to this rule exists when a TA has lost permanent access to one of its keys or the accompanying repository publication point. In such cases however, the key in question MUST be revoked as described below in Section 6.

A non-normative guideline for naming this object is that the filename chosen for the Signed TAL Object 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 ".tak" MUST be used to denote the object as a TAK. Note that this is in-line with filename extensions defined in section 7.2 of [RFC6481]

In order to generate the TAK Objects, the TA MUST perform the following actions:

- o The TA MUST generate a key pair for a "one-time-use" EE certificate to use for the TAK

- o The TA MUST generate a one-time-use EE certificate for the TAK

- o 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 TAK as an object URL.

- o 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.

- o This EE certificate MUST have a "notBefore" time that matches, or predates the moment that the TAK will be published.

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

- o The same set of current keys (see Section 3.2.2) MUST be included on each TAK object for each current key.

- o The TAK object MUST include all revoked keys (see Section 3.2.3) that became revoked while the key signing the TAK in question was current.

6. Performing TA Key Rolls

6.1. Opting in to Key Rolls

6.1.1. Trust Anchor

For simplicitly let's start with a situation where a TA has only one key. The TA wants to start using TAK objects to perform key rolls in future, so it introduces a TAK object under its single key 'A'. The repository structure looks as follows (irrelevant details omitted):

```
+--------------------+
|        A.MFT       |
+--------------------+
| A.CRL      <hash>  |
| A.TAK      <hash>  |
| C1-A.CER   <hash>  |
| C2-A.CER   <hash>  |
+--------------------+

+--------------------+
|        A.CRL       |
+--------------------+
| revocations..      |
+--------------------+

+--------------------+
|        A.TAK       |
+--------------------+
| current: A         |
| revoked: none      |
+--------------------+

+--------------------+
|      C1-A.CER      |
+--------------------+
| resources: C1 res  |
| subject:   C1 name |
| pub key:   C1 key  |
| SIA:       C1 SIAs |
| AKI:       A       |
+--------------------+

+--------------------+
|      C2-A.CER      |
+--------------------+
| resources: C2 res  |
| subject:   C2 name |
| pub key:   C2 key  |
| SIA:       C2 SIAs |
| AKI:       A       |
+--------------------+
```

So, the TA publishes a CRL and MFT under its key A, listing a TAK object and in this case two certificates issued to children 'C1' and 'C2' signed using key A. The TAK object lists key 'A' as the only current key, and has no revoked keys.

6.1.2. Relying Parties

Relying Parties who have a TAL for key 'A' configured will discover the TAK object. If the RP does not support this object, it will reject this object but continue to validate the remaining RPKI tree as usual. If the RP does support TAK objects it will conclude that key 'A' is the one and only current key, and will proceed to validate the remaining RPKI tree as usual.

6.2. Pre-stage a New Key

6.2.1. Trust Anchor

Now the TA prestages a new key 'B' and produces equivalent CA certificates for children 'C1' and 'C2', i.e. the resources, subject name, public key and SIA etc are all equivalent, but these certificates are signed under key 'B'. (See Section 4 for a more thorough description of this). The TAK object for key 'B' recognises both keys 'A' and 'B' as current.

The repostory structure and TAK object for key B are then as follows:

```
+--------------------+
|        B.MFT       |
+--------------------+
| B.CRL      <hash>  |
| B.TAK      <hash>  |
| C1-B.CER   <hash>  |
| C2-B.CER   <hash>  |
+--------------------+

+--------------------+
|        B.CRL       |
+--------------------+
| revocations..      |
+--------------------+

+--------------------+
|        B.TAK       |
+--------------------+
| current: A, B      |
| revoked: none      |
+--------------------+

+--------------------+
|      C1-B.CER      |
+--------------------+
| resources: C1 res  |
| subject:   C1 name |
| pub key:   C1 key  |
| SIA:       C1 SIAs |
| AKI:       B       |
+--------------------+

+--------------------+
|      C2-B.CER      |
+--------------------+
| resources: C2 res  |
| subject:   C2 name |
| pub key:   C2 key  |
| SIA:       C2 SIAs |
| AKI:       B       |
+--------------------+
```

When the TA is certain that the content for key 'B' is correct, it can also update the TAK object for key 'A' to include 'B':

```
+--------------------+
|        A.TAK       |
+--------------------+
| current: A, B      |
| revoked: none      |
+--------------------+
```

One way to do this is by generating a TAL [I-D.ietf-sidrops-https-tal] for key B and verifying that validation using this yields the same results as validation using the TAL for key A would. However, note, that it is preferred that this is done as part of an automated process that is sufficiently well tested, and that the contents of the repositories for keys 'A' and 'B' are updated as a single delta if the publication protocol [RFC8181] is used (see also: Section 5).

6.2.2. Relying Parties

Relying Parties who have a TAL for key 'A' configured will discover the TAK object. If the RP does not support this object, it will reject this object but continue to validate the remaining RPKI tree as usual. If the RP does support TAK objects it will conclude that there are now two keys 'A' and 'B', and no revoked keys that it should be aware of. Since key 'A' is still current, the RP will continue to validate the RPKI tree structure using the repository for key 'A', ignoring the non-TAK objects in the repository for key 'B'.

The result will be the same for Relying Parties who have a TAL for key 'B' configured, because both keys are equivalent at this time.

6.3. Planned Key Revocation

6.3.1. Trust Anchor

The TA has now decided that key 'A' must be revoked. It still has access to this key and the repository, so it can perform a planned key roll. In addition to revoking key 'A', the TA will also generate new key 'C' to ensure that it has at least two current keys at all times for redundancy.

Keys 'B' and 'C' will become current keys on the TAK objects for all keys: 'A', 'B' and 'C'. Key 'A' will become part of the revoked keys on the TAK objects for keys 'A' and 'B'. Note that it is not needed to list key 'A' as revoked on the TAK file for key 'C', because RPs will only learn about key 'C' at the same time as learning about the revocation of key 'A' (see also below).

The TA will publish a long-lived TAK file and MFT and CRL only for key 'A' and publish these objects as waypointers for RPs that have a TAL pointing at key 'A' before destroying key 'A'.

The resulting structure for key 'A' will be as follows:

```
+--------------------+
|        A.MFT       |
+--------------------+
| A.CRL      <hash>  |
| A.TAK      <hash>  |
+--------------------+

+--------------------+
|        A.CRL       |
+--------------------+
| revocations..      |
+--------------------+

+--------------------+
|        A.TAK       |
+--------------------+
| current: B, C      |
| revoked: A         |
+--------------------+
```

The resulting structures for keys 'B' and 'C' will be as follows:

```
+--------------------+          +--------------------+
|       B.MFT        |          |       C.MFT        |
+--------------------+          +--------------------+
| B.CRL      <hash>  |          | B.CRL      <hash>  |
| B.TAK      <hash>  |          | B.TAK      <hash>  |
| C1-B.CER   <hash>  |          | C1-C.CER   <hash>  |
| C2-B.CER   <hash>  |          | C2-C.CER   <hash>  |
+--------------------+          +--------------------+

+--------------------+          +--------------------+
|       B.CRL        |          |       C.CRL        |
+--------------------+          +--------------------+
| revocations..      |          | revocations..      |
+--------------------+          +--------------------+

+--------------------+          +--------------------+
|       B.TAK        |          |       C.TAK        |
+--------------------+          +--------------------+
| current: B, C      |          | current: B, C      |
| revoked: A         |          | revoked: <none>    |
+--------------------+          +--------------------+

+--------------------+          +--------------------+
|     C1-B.CER       |          |     C1-C.CER       |
+--------------------+          +--------------------+
| resources: C1 res  |          | resources: C1 res  |
| subject:   C1 name |          | subject:   C1 name |
| pub key:   C1 key  |          | pub key:   C1 key  |
| SIA:       C1 SIAs |          | SIA:       C1 SIAs |
| AKI:       B       |          | AKI:       C       |
+--------------------+          +--------------------+

+--------------------+          +--------------------+
|     C2-B.CER       |          |     C2-B.CER       |
+--------------------+          +--------------------+
| resources: C2 res  |          | resources: C2 res  |
| subject:   C2 name |          | subject:   C2 name |
| pub key:   C2 key  |          | pub key:   C2 key  |
| SIA:       C2 SIAs |          | SIA:       C2 SIAs |
| AKI:       B       |          | AKI:       B       |
+--------------------+          +--------------------+
```

In addition to this the TA SHOULD reach out to RP vendors so that they can update the TAL included in the RP software distribution to use key 'B'.

6.3.2. Relying Parties

Relying Parties who have a TAL for key 'A' configured will discover the TAK object. If the RP does not support this object, it will reject this object but continue to validate the remaining RPKI tree as usual. In this case that means that validation will stop, because there are no more objects under key 'A'. Therefore it is important that RPs that do not support TAK files are updated to use the TAL for key 'B' through some other process.

If the RP uses a TAL for key 'A' and it supports TAK objects, it will discover that the TAL for key 'A' has keys 'B' and 'C' as current, and revokes itself. It will then proceed to process keys 'B' and 'C' and find TALs which list the same current keys. So, it will now replace its notion of the current key set for this TA based on its TAL (key 'A') with what it learned. To keep things simple the RP will now conclude that it should re-start validation using a remaining current key, in this case key either 'B' or 'C' may be used.

If the RP already had a TAL for key 'B' and it supports TAK objects, or it simply started with key 'B' because it added it to its set of current keys when this key was pre-staged (see Section 6.2), it will learn that key 'A' is revoked and therefore will not attempt to verify the TAK file for key 'A'. It will also learn about key 'C' and inspect this key's TAL, and discover that only keys 'B' and 'C' are considered current. Since it started the validation process with a key that is still current, it can proceed to validate the RPKI tree using the repository under key 'B'.

6.4. Unplanned revocation

6.4.1. Trust Anchor

Now keys 'B' and 'C' are current. The TA may have intended to revoke key 'B', essentially rolling over to key 'C' and a new key 'D', but let us suppose that the TA lost access to key 'C'. In this case the TA will simply revoke key 'C' instead, and still introduce a new key 'D'.

The major difference with the process described above for planned rolls, is that now the TA will not be able to update the TAK object, MFT or CRL for key 'C'. However, because all TAL objects for current keys are evaluated before tree validation is performed, it is safe to leave these objects in a repository. Keys 'B' and 'D' will simply mark key 'C' as being revoked.

If an RP still has a TAL pointing at key 'C' it will discover that key 'D' is added, and that key 'B' has been revoked through the TAK object published for keys 'B' and 'D'. At least, as long as the the MFT and TAK EE certificates have not expired, and the CRL and MFT are not stale.

If the TA is absolutely sure that the TAL for key 'C' never shipped with any RP distribution, then it would also be safe to delete the repository key 'C' altogether. RPs will learn that 'C' is revoked, and therefore will not even attempt to download the TAK object. However, it is hard to be certain of this and there this is NOT RECOMMENDED.

7. 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 relied 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. Until such time, TAs SHOULD continue to generate unsigned TAL files [I-D.ietf-sidrops-https-tal], and indicate which should be considered their current TAL, and communicate them to RPs through other means.

However, once a 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 key rolls will work is by making them a part of normal operations. Determining when this moment arrives is by definition out of scope for this document, as it should be based on operational experience.

8. IANA Considerations

8.1. OID

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

Decimal	Description	References
TBD	Trust Anchor Keys	[section 3.1]

8.2. 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
.tak	Trust Anchor Keys	[this document]

9. Security Considerations

TBD

10. Acknowledgements

The authors wish to thank Martin Hoffmann for a thourough review of this document.

11. References

11.1. 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-05 (work in progress), October 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>.

[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>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <https://www.rfc-editor.org/info/rfc6482>.

[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>.

[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>.

[RFC6489] Huston, G., Michaelson, G., and S. Kent, "Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI)", BCP 174, RFC 6489, DOI 10.17487/RFC6489, February 2012, <https://www.rfc-editor.org/info/rfc6489>.

[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>.

[RFC8181] Weiler, S., Sonalker, A., and R. Austein, "A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", RFC 8181, DOI 10.17487/RFC8181, July 2017, <https://www.rfc-editor.org/info/rfc8181>.

[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.

11.2. 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/

Rob Austein
Dragon Research Labs

Email: sra@hactrn.net

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 22, 2019

G. Michaelson
APNIC
T. Bruijnzeels
opennetlabs

October 19, 2018

Deployment of Reconsidered Validation in the Resource Public Key Infrastructure (RPKI)
draft-va-sidrops-deploy-reconsidered-00

Abstract

This document defines a deployment model for reconsidered validation [RFC8360] in the Resource Public Key Infrastructure (RPKI).

It stipulates that Relying Parties in the RPKI MUST support reconsidered validation by 1 July TBD-Year, and that Certificate Authorities MAY use the reconsidered validation OIDs in CA certificates that they issue from this date. Furthermore Certificate Authorities should monitor whether the set of resources in CA certificate they receive has shrunk, and make adjustments in the CA certificates and/or other RPKI objects when appropriate.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 22, 2019.

Table of Contents

1. Overview

This document defines a deployment model for reconsidered validation [RFC8360] in the Resource Public Key Infrastructure (RPKI).

Reconsidered validation differs from normal validation [RFC6487] in that under reconsidered rules the intersection of resources between a child certificate and the resources contained in the (chain of) parent certificate(s) is accepted. Any resources that are contained in the child certificate only result in a warning about these resources, rather than the rejection of that certificate. Thus reconsidered validation limits the impact of over-claims in the RPKI to the set of resources under dispute.

The applicability of reconsidered validation is signalled by the use of a distinct set of OIDs on a Resource Certificate [RFC8360]. Because of this reconsidered validation can only be deployed when a majority of Relying Party software is updated to support this new algorithm. This document stipulates that RP software MUST support [RFC8360] by 1 July TBD-Year. After 1 July TBD-Year Certificate

Authorities MAY start to use [RFC8360] in CA certificates that they issue.

Note that the use of reconsidered validation is restricted to CA Certificates only. Issuing Certificate Authorities may (be forced to) re-issue delegated CA certificates with shrunk resource pro-actively, and therefore it's at the CA certificate level that mismatches between resources actually included on such a certificate and the resources the recipient believes to be included on these certificates may occur.

On the other hand, ROA and BGPSec Router Certificate reconsidered validation still requires that all resources are also held by the path of parent certificates to these objects. In other words, using the reconsidered validation here is unneccessary.

Furthermore, Certificate Authorities should monitor pro-actively whether the set of resources in the CA certificate they received has been shrunk by their parent. Resource Certificates that they in turn issue that use resources no longer validly held by them should be shrunk or revoked. BGPSec Router Certificates or ROAs that use such resources should be removed.

1.1. Terminology

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. Phased Deployment of the Amended Certificate Profile

There is an existing BCP document that describes an algorithm agility procedure for the RPKI [RFC6919]. This procedure involves four distinct phases with requirements for CAs and RPs. During this process the entire RPKI tree is essentially duplicated, and two disctinct trees are maintained in paralel for some time, until the old tree can be withdrawn. The dates for each milestones are expected to be documented in a BCP.

In this case however, the amended validation process is very similar to the existing validation process. Moreover, as [RFC8360] describes there is no fundamental issue in having an RPKI tree in which a mix of regular [RFC6487] and amended [RFC8360] certificates can be found.

The use of the amended certificate profile communicates that over-claims for this particular certificate can occur, and if they do,

that their impact should be limited to the resources that are in over-claim. Sections 4.2.5 and 4.2.6 of [RFC8360] stipulate that such over-claims on the EE certificate would invalidate ROA and BGPSec Router Certificates.

In conclusion the amended certificate profile MUST only be used on CA certificates for CA organisations where an overclaim may accidentally occur, and MUST NOT be used anywhere else: e.g. on a TA CA certificate which by definition cannot overclaim, or on any specific attestation about resources other than a delegation to another CA, e.g. ROAs and BGPSec Router Certificates.

So, contrary to the process described in [RFC6919] there is no desired outcome here to completely replace an existing algorithm with a new algorithm. And consequently a different approach to the deployment phases is applicable here.

We recognise the following phases:

1. Relying Party software MUST support the amended profile

2. Operators MUST use updated Relying Party software;

3. Certificate Authorities MAY use the amended profile

As suggested in [RFC6919] the dates for each of these phases can be documented in this BCP:

1. 1 July TBD-Year

2. 1 July TBD-Year - 1 January TBD-Year +1

3. 1 January TBD-Year +1

2.1. Phase 1: Requirements for RP Software

Relying Party software MUST support [RFC8360] by 1 July TBD-Year.

2.2. Phase 2: Requirements for operators

Network operators MUST update their Relying Party software between 1 July TBD-Year and 1 January TBD-Year +1.

2.3. Phase 3: Requirements for Certificate Authorities

Trust Anchor CA certificates referenced in Trust Anchor Locator (TAL) files [RFC7730] MUST NOT make use of amended Resource Certificates defined in [RFC8360].

From 1 January 2020 Certificate Authorities MAY use amended Resource Certificates [RFC8360] for CA certificates that they issue to delegated Certificate Authorities. Certificate Authorities MUST NOT use the amended Resource Certificate profile for any other certificates they issue.

3. Avoid over-claiming CA certificates

Even though the amended profile limits the impact of resource over-claims on CA certificates, this does not mean that such over-claims are intended to become the norm. As we will describe in the following sections:

- o Issuing CAs should try to avoid invalidating delegated CAs
- o Delegated CAs should self-monitor and take action in case resources are removed

3.1. Avoid Invalidating Delegated CAs

3.1.1. Graceperiod and Check Intervals

If resources need to be removed from a delegated CA it is reasonable to observe a graceperiod that will allow a delegated CA (and recursively their delegated CAs if applicable) to clean up objects. A reasonable duration for this period depends on the following factors:

- o The frequency that a child CA can check with its parents about its CA certificate eligibility (Check Interval)
- o The depth of the tree of delegated CAs

We believe that it’s reasonable to require the child CAs MUST issue a Resource Class List Query [RFC6492] to their parent CA no less frequently than once per houri (Check Interval). It is not expected that the depth of delegated CA certificates will exceed 5 or 6 CA authorities. In conclusion a graceperiod of 24 hours seems reasonable.

3.1.2. Shrinking issued CA certificates

When a Certificate Authority finds that it needs to shrink the set of resources held by a delegated Certificate Authority, but still holds the resources to be removed on its own CA certificate, then it SHOULD give the delegated Certificate Authority up to 24 hours to request a shrunk CA certificate, e.g. through the provision protocol [RFC6492].

The CA SHOULD issue a new CA certificate immediately using a "notAfter" time that is set to whichever is soonest: 24 hours from now, or the "notAfter" time on the CA certificate held by this issuing CA. This will alert the delegated CA of both the limited lifetime of their current CA certificate, and which resources remain eligible after this time, when the delegated CA sends a Resource Class List Query [RFC6492].

If the Certificate Authority no longer holds the resources that are to be removed, or this 24 hour period has passed, then a shrunk CA certificate MUST be issued. Such shrunk certificate SHOULD use the amended Resource Certificate profile in order to limit the impact in the validation of objects issued by the subsidiary Certificate Authority.

3.2. Self monitoring and clean-up

CAs in the RPKI MUST monitor whether the CA certificate issued to them by their parent needs to be shrunk, for example by sending a Resource Class List Query [RFC6492] to their parent CA no less frequently than once per hour.

If the CA finds that a reduced resource set is in order, but their current certificate is still valid, and they have issued delegated CA certificates with the resources to be reduced to delegated CAs, then they SHOULD give these delegated CAs up to 24 hours, or the time until 1 hour before their own CA certificate "notAfter" time if this period is shorter, to request a shrunk CA certificate as described above.

The CA MUST now remove any other RPKI objects that it issued that reference any of the resources to be removed. If the CA issued ROAs that reference multiple prefixes, and some of these prefixes are not to be removed, then the CA SHOULD create new ROAs for these prefixes and use one ROA object per prefix rather than bundling multiple prefixes on a single ROA object.

If the CA no longer issues any CA certificates or RPKI objects referencing the resources to be removed, or it finds that its current CA certificate is no longer valid or will expire within 1 hour, then the CA MUST request a new CA certificate to be issued by their parent CA.

4. Example use of the amended profile with transfers

Consider the following starting situation where a Trust Anchor issues a resource certificate to Certificate Auhtority CA1, which in turn

issues a ROA and delegates some resources to CA2, which in turn also issues a ROA:

```
TA CA Certificate:
  Issuer:    TA
  Subject:   TA
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24, 198.51.100.0/24,
             2001:db8::/32, AS64496-AS64500

CA1 CA Certificate:
  Issuer:    TA
  Subject:   CA1
  Profile:   8360 (amended)
  Resources: 192.0.2.0/24, 198.51.100.0/24
             2001:db8::/32, AS64496-AS64500

CA1 ROA 1:
  Issuer:    CA1
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24

  ASN:       64496
  Prefixes:  192.0.2.0/24 (Max 24)

CA2 CA Certificate:
  Issuer:    CA1
  Subject:   CA2
  Profile:   8360 (amended)
  Resources: 198.51.100.0/24,
             2001:db8::/32, AS64496-AS64500

CA2 ROA 1:
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 2001:db8::/32

  ASN:       64496
  Prefixes:  2001:db8::/32 (Max 48)

CA2 ROA 2:
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 198.51.100.0/24

  ASN:       64496
  Prefixes:  198.51.100.0/24 (Max 24)
```

Now assume that the TA decides that CA1 should no longer hold the prefix 198.51.100.0/24. However, CA1 is offline for some reason and it does not check in with TA about its CA certificate eligibility.

After 24 hours TA will decided that it has waited long enough and it will now pro-actively issue an amended CA certificate for CA1. Because the amended profile is used for CA certificates the impact of this action is limited. CA2 has been unaware of the change, but only their ROA2 which is using the prefix is now invalidated:

```
TA CA Certificate:
  Issuer:    TA
  Subject:   TA
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24, 198.51.100.0/24,
             2001:db8::/32, AS64496-AS64500

CA1 CA Certificate:
  Issuer:    TA
  Subject:   CA1
  Profile:   8360 (amended)
  Resources: 192.0.2.0/24
             2001:db8::/32, AS64496-AS64500

CA1 ROA 1:
  Issuer:    CA1
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24

  ASN:       64496
  Prefixes:  192.0.2.0/24 (Max 24)

CA2 CA Certificate:
  Issuer:    CA1
  Subject:   CA2
  Profile:   8360 (amended)
  Rejected:  198.51.100.0/24
  Accepted:  2001:db8::/32, AS64496-AS64500

CA2 ROA 1:
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 2001:db8::/32

  ASN:       64496
  Prefixes:  2001:db8::/32 (Max 48)

CA2 ROA 2 (INVALID):
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 198.51.100.0/24

  ASN:       64496
  Prefixes:  198.51.100.0/24 (Max 24)
```

Now CA1 comes back online. It discovers that it lost the prefix 198.51.100.0/24. It will now re-issue the CA certificate issued to CA2 immediately:

```
TA CA Certificate:
  Issuer:    TA
  Subject:   TA
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24, 198.51.100.0/24,
             2001:db8::/32, AS64496-AS64500

CA1 CA Certificate:
  Issuer:    TA
  Subject:   CA1
  Profile:   8360 (amended)
  Resources: 192.0.2.0/24
             2001:db8::/32, AS64496-AS64500

CA1 ROA 1:
  Issuer:    CA1
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24

  ASN:       64496
  Prefixes:  192.0.2.0/24 (Max 24)

CA2 CA Certificate:
  Issuer:    CA1
  Subject:   CA2
  Profile:   8360 (amended)
  Resources: 2001:db8::/32, AS64496-AS64500

CA2 ROA 1:
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 2001:db8::/32

  ASN:       64496
  Prefixes:  2001:db8::/32 (Max 48)

CA2 ROA 2 (INVALID):
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 198.51.100.0/24

  ASN:       64496
  Prefixes:  198.51.100.0/24 (Max 24)
```

Finally CA2, who was trying to check in with CA1 even when it was unavailable, discovers that it lost the prefix '198.51.100.0/24'. It will therefore remove its ROA2:

```
TA CA Certificate:
  Issuer:    TA
  Subject:   TA
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24, 198.51.100.0/24,
             2001:db8::/32, AS64496-AS64500

CA1 CA Certificate:
  Issuer:    TA
  Subject:   CA1
  Profile:   8360 (amended)
  Resources: 192.0.2.0/24
             2001:db8::/32, AS64496-AS64500

CA1 ROA 1:
  Issuer:    CA1
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 192.0.2.0/24

  ASN:       64496
  Prefixes:  192.0.2.0/24 (Max 24)

CA2 CA Certificate:
  Issuer:    CA1
  Subject:   CA2
  Profile:   8360 (amended)
  Resources: 2001:db8::/32, AS64496-AS64500

CA2 ROA 1:
  Issuer:    CA2
  Subject:   R1
  Profile:   6487 (regular)
  Resources: 2001:db8::/32

  ASN:       64496
  Prefixes:  2001:db8::/32 (Max 48)
```

A few things to note:

- In this scenario CA1 was offline, and it was not performing the actions required to the occurance of an overclaiming CA certificate to remain for CA2 and CA2 was not aware of the coming change.

- o The use of the amened profile for reconsidered validation rules limited the impact of this operational problem to just those resources that were being removed.

- o Had CA2 not only monitored its CA certificate eligibility directly with its parent, but had they performed RPKI validation to monitor their own certificate and products. Then they would have removed their ROA2 sooner. Since CA1 was offline however, they would not have been able to request a shrunk CA certificate for themselves.

- o Had CA1 and CA2 both been online and TA observed the 24 hour grace period, then things would have been changed without the occurance of invalid objects or warnings. CA2 would have removed ROA2, and then would have requested a shrunk CA certificate for itself. CA1 would have noticed that it was safe to reques its own CA certificate to be shrunk. The CA depth here is 2, so this would have happened within 2 hours, well within the 24 hours limit.

5. RFC-EDITOR Considerations

The exact year value TBD-Year and TBD-Year +1 are to be defined in WG process and will be set before WGLC

6. Security Considerations

TBD

7. Acknowledgements

TBD

8. 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>.

[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>.

[RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, DOI 10.17487/RFC6492, February 2012, <https://www.rfc-editor.org/info/rfc6492>.

[RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, DOI 10.17487/RFC6919, April 2013, <https://www.rfc-editor.org/info/rfc6919>.

[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>.

[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>.

[RFC8360] Huston, G., Michaelson, G., Martinez, C., Bruijnzeels, T., Newton, A., and D. Shaw, "Resource Public Key Infrastructure (RPKI) Validation Reconsidered", RFC 8360, DOI 10.17487/RFC8360, April 2018, <https://www.rfc-editor.org/info/rfc8360>.

Authors' Addresses

George G. Michaelson
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane, QLD 4101
Australia

Email: ggm@apnic.net

Tim Bruijnzeels
Open Netlabs B.V.
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: timb@opennetlabs.nl