A profile for RPKI Signed Lists of Prefixes

Abstract

This document defines a "RPKI Prefix List", a Cryptographic Message Syntax (CMS) protected content type for use with the Resource Public Key Infrastructure (RPKI) to carry the complete list of prefixes which an Autonomous System (AS) may originate to all or any of its routing peers. The validation of a RPKI Prefix List confirms that the holder of the listed ASN produced the object, and that this list is a current, accurate and complete description of address prefixes that may be announced into the routing system originated by this AS.

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 19 April 2024.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with
1. Introduction

This document defines an "RPKI Prefix List", a Cryptographic Message Syntax (CMS) [RFC5652] [RFC6268] protected content type to carry a list of IP prefixes and an Autonomous System Number. The list of prefixes describes the maximal set of prefixes that the Autonomous System MAY announce to any of its routing peers. The content is signed by the holder of the RPKI private key associated with the listed ASN.
RPKI Prefix Lists allow other RPKI-validating remote routing entities to audit the collection of announcements that have the subject ASN as the originating AS. Any prefixes originated by this AS not contained in a validated RPKI Prefix List SHOULD be regarded as invalid, but ultimately their consequent handling by the local routing entity that performed the audit function is a matter of local policy.

The intent of this object is to offer a RPKI-based successor to the [RFC2622] 'route-set' class objects used in Internet Routing Registries (IRRs). The semantics of the route-set and the RPKI Prefix List are similar. The difference is that the RPKI signature allows a relying party to be assured of the currency and authenticity of the Prefix List as a complete enumeration of all prefixes that may be announced as originating by this AS if the object can be validated by the RPKI.

1.1. 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.

2. RPKI Signed Prefix List Profile

RPKI Prefix List objects follow the Signed Object Template for the RPKI [RFC6488].

2.1. EE Certificates

The Certification Authority (CA) MUST sign only one PrefixList with each EE certificate and MUST generate a new key pair for each new PrefixList or PrefixList Opt-Out Listing. This type of EE certificate is termed a "one-time-use" EE certificate (see Section 3 of [RFC6487]).

2.2. Object Filenames

A guideline for naming PrefixList is that the file name chosen in the repository be a value derived from the public key of the EE certificate. One such method of generating a publication name is described in Section 2.1 of [RFC4387]; convert the 160-bit hash of a EE's public key value into a 27-character string using a modified form of Base64 encoding, with an additional modification as proposed in Section 5, table 2, of [RFC4648].
3. eContentType

3.1. The PrefixList eContentType

The eContentType for a PrefixList is defined as id-ct-rpkiSignedPrefixList, with Object Identifier (OID) 1.2.840.113549.1.9.16.1.TBD.

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

4. eContent

4.1. The PrefixList eContent

The content of a PrefixList is a list of IP address prefixes and a single ASN. A PrefixList is formally defined as follows:
RpkiSignedPrefixList-2023

{ iso(1) member-body(2) us(840) rsadsi(113549)
  pkcs(1) pkcs9(9) smime(16) mod(0)
  id-mod-rpkiSignedPrefixList-2023(TBD) }

DEFINITIONS EXPLICIT TAGS ::= BEGIN

IMPORTS
  CONTENT-TYPE
FROM CryptographicMessageSyntax-2010 -- in [RFC6268]
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
  pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) };

cr-rpkiSignedPrefixList CONTENT-TYPE ::= {
  TYPE RpkiSignedPrefixList
  IDENTIFIED BY id-ct-rpkiSignedPrefixList }

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

RpkiSignedPrefixList ::= SEQUENCE {
  version [0]     INTEGER DEFAULT 0,
  asID            ASID,
  prefixList      SEQUENCE (SIZE(0..MAX)) OF AddressFamilyIPAddress }

ASID ::= INTEGER (1..4294967295)

AddressFamilyIPAddress ::= SEQUENCE {
  addressFamily ADDRESS-FAMILY.&afi ({AddressFamilySet}),
  prefix        ADDRESS-FAMILY.&Prefix ({AddressFamilySet}{@addressFamily}) }

ADDRESS-FAMILY ::= CLASS {
  &afi          OCTET STRING (SIZE(2)) UNIQUE,
  &Prefix
} WITH SYNTAX { AFI &afi PREFIX-TYPE &Prefix }

AddressFamilySet ADDRESS-FAMILY ::= { addressFamilyIPv4 | addressFamilyIPv6 }

addressFamilyIPv4 ADDRESS-FAMILY ::= { AFI afi-IPv4 PREFIX-TYPE AddressesIPv4
  addressFamilyIPv6 ADDRESS-FAMILY ::= { AFI afi-IPv6 PREFIX-TYPE AddressesIPv6

afi-IPv4 OCTET STRING ::= '0001'H
afi-IPv6 OCTET STRING ::= '0002'H

AddressesIPv4 ::= Prefix{32}
AddressesIPv6 ::= Prefix{128}

Prefix {INTEGER: len} ::= SEQUENCE {
address       BIT STRING (SIZE(0..len)) }

END
4.1.1. Version

The version number of the RpkiSignedPrefixList MUST be 0.

4.1.2. asID

The Autonomous System Number contained here MUST be a contained within the set of AS Identifier resources listed by the EE certificate carried in the CMS certificates field.

4.1.3. prefixList

This field contains a SEQUENCE of AddressFamilyIPAddress. The AddressFamilyIPAddress elements MUST be ordered in ascending order: first by numeric value of the addressFamily field, then by value of the prefix field.

4.1.3.1. Element AddressFamilyIPAddress

This field contains a SEQUENCE which contains one instance of addressFamily and one instance of prefix.

4.1.3.1.1. addressFamily

This field contains a OCTET STRING which is either '0001'H (IPv4) or '0002'H (IPv6).

4.1.3.1.2. prefix

This field contains a BIT STRING, its length bounded through the addressFamily field. The type is a BIT STRING, see Section 2.2.3.8 of [RFC3779] for more information.

5. PrefixList Validation

To validate an PrefixList, the RP MUST perform all the validation checks specified in [RFC6488]. In addition, the RP MUST perform the following validation steps:

1. The contents of the CMS eContent field MUST conform to all of the constraints described in Section 4.

2. The Autonomous System Identifier Delegation extension [RFC3779] MUST be present in the EE certificate contained in the CMS certificates field.

3. The AS identifier present in the RpkiSignedPrefixList eContent 'asID' field MUST be a subset of the AS Indentifiers present in the certificate extension.
4. The Autonomous System Identifier Delegation extension MUST NOT contain "inherit" elements.

5. The IP Address Delegation Extension [RFC3779] is not used in PrefixList, and MUST NOT be present in the EE certificate.

6. Operational Considerations

Multiple valid PrefixList objects could exist which contain the same asID. In such cases the union of address prefix members forms the complete set of members. It is highly RECOMMENDED that a compliant CA maintains a single PrefixList for a given asID.

If an AS holder publishes a PrefixList, then relying parties SHOULD assume that the list is complete for that originating AS, and the presence of any route with the same AS as the originating AS and an address prefix that is not included in the Prefix List implies that the route has been propagated within the routing system without the permission of the originating AS.

The construction of an 'allowlist' for a given EBGP session using RPKI PrefixList(s) compliments best practises [RFC7454] and rejecting RPKI-invalid BGP route announcements [RFC6811]. In other words, if a given BGP route is covered by a RPKI PrefixList, but also is "invalid" from a Route Origin Validation perspective, it is RECOMMENDED to reject the route announcement.

7. Security Considerations

Relying Parties are warned that the data in an PrefixList is self-asserted by the AS holder. There is no implied authority from any IP prefix holder that grants the AS permission to originate a route for any prefixes. Such an authority is separately conveyed in the RPKI as a ROA.

While a one-time-use EE certificate must only be used to generate and sign a single PrefixList object, CAs technically are not restricted from generating and signing multiple different PrefixList objects with a single key pair. Any PrefixList objects sharing the same EE certificate cannot be revoked individually.

8. IANA Considerations

8.1. SMI Security for S/MIME CMS Content Type
(1.2.840.113549.1.9.16.1)

IANA is requested to allocated the following in the "SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)" registry:
8.2. RPKI Signed Objects

IANA is requested to register two OIDs in the "RPKI Signed Objects" registry [RFC6488] as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>OID</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed PrefixList</td>
<td>1.2.840.113549.1.9.16.1.TBD</td>
<td>draft-spaghetti-sidrops-rpki-prefixlist</td>
</tr>
</tbody>
</table>

Table 2

8.3. RPKI Repository Name Schemes

IANA is requested to add the Signed PrefixList file extension to the "RPKI Repository Name Schemes" registry [RFC6481] as follows:

<table>
<thead>
<tr>
<th>Filename</th>
<th>RPKI Object</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>.pfx</td>
<td>Signed PrefixList</td>
<td>draft-spaghetti-sidrops-rpki-prefixlist</td>
</tr>
</tbody>
</table>

Table 3

8.4. SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)

IANA is requested to allocate the following in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>id-mod-rpkiSignedPrefixList-2023</td>
<td>draft-spaghetti-sidrops-rpki-prefixlist</td>
</tr>
</tbody>
</table>

Table 4

8.5. Media Types

IANA is requested to register the media type "application/rpki-prefixlist" in the "Media Types" registry as follows:

8.5.1. PrefixList Media Type

Type name: application
Subtype name: rpki-prefixlist
This media type is a signed object, as defined in [RFC6488], which contains a payload of a list of checksums as defined in draft-spaghetti-sidrops-rpki-prefixlist.

**Magic number(s):** N/A

**File extension(s):** .pfx

**Macintosh file type code(s):** N/A

**Person & email address to contact for further information:** Job Snijders (job@fastly.com)

**Intended usage:** COMMON

**Restrictions on usage:** N/A

**Author:** Job Snijders (job@fastly.com)

**Change controller:** IETF

9. References

9.1. Normative References


9.2. Informative References


Appendix A. Acknowledgements

The authors wish to thank Russ Housley for feedback.

Appendix B. Example payloads

B.1. Example PrefixList eContent Payload

Below an example of a DER encoded PrefixList eContent is provided with annotation following the '#' character.
Appendix C. Implementation status

This section is to be removed before publishing as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF.
Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

*Example .pfx files were created by Job Snijders with the use of asn1c and OpenSSL.

Authors' Addresses

Job Snijders
Fastly
Amsterdam
Netherlands

Email: job@fastly.com

Geoff Huston
APNIC
6 Cordelia St
South Brisbane QLD 4101
Australia

Email: gih@apnic.net