Global Routing Operations
Internet-Draft
Intended status: Informational
Expires: October 26, 2020

J. Snijders
NTT
M. Stucchi
Independent
M. Aelmans
Juniper Networks

April 24, 2020

RPKI Autonomous Systems Cones: A Profile To Define Sets of Autonomous Systems Numbers To Facilitate BGP Filtering
draft-ietf-grow-rpki-as-cones-02

Abstract

This document describes a way to define groups of Autonomous System numbers in RPKI [RFC6480]. We call them AS-Cones. AS-Cones provide a mechanism to be used by operators for filtering BGP-4 [RFC4271] announcements.

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 main goal of the Resource Public Key Infrastructure (RPKI) system [RFC6480] is to support improved security for the global routing system. This is achieved through the use of information stored in a distributed repository system comprised of signed objects. A

commonly used object type is the Route Object Authorisation (ROAs), which describe the relation between a prefix and its originating ASNs.

There is however no method for an operator to assert the routes for its customer networks, making it difficult to use the information carried by RPKI to create meaningful BGP-4 filters without relying on RPSL [RFC2622] as-sets.

This document introduces a new attestation object, called an AS-Cone. An AS-Cone is a digitally signed object with the goal to enable operators to define a set of customer or downstream ASNs that can be found as "right adjacencies", or transit customer networks, facilitating the construction of prefix filters for a given ASN, thus making routing more secure.

The goal of AS-Cones is to be able to recursively define all the originating ASNs that define the customer base of a given ASN, including all the transit relationships. This means that through AS-Cones, it is possible to create a tree of all the neighbour relationships for the customers of a given Autonomous System.

2. Format of AS-Cone objects

AS-Cones are composed of two types of distinct objects:

- Policy definitions; and
- The AS-Cones themselves.

These objects are stored in ASN.1 format and are digitally signed according to the same rules and conventions applied for RPKI ROA Objects ([RFC6482]).

2.1. Policy definition object

A policy definition object contains a list of the upstream and peering relationships for a given Autonomous System that need an AS-Cone to be used for filtering. For each relationship, either an AS-Cone or a plain Autonomous System Number is referenced to indicate which networks will be announced to the other end of the relationship using BGP.

The default behaviour for a neighbour, if the relationship is not explicitly described in the policy, is to only accept the networks originated by the ASN. This means that a stub ASN neither has to set up any AS-Cone, description, nor policy.

The Policy Definition object contains a field called "ContactEmail" containing the E-Mail address for which all the communication related to this policy definition should be sent to.

Only one AS-Cone or Autonomous System Number can be supplied for a given relationship. If more than one AS-Cone needs to be announced in the relationship, then it is mandatory to create a third AS-Cone that includes those two. If more than one ASN needs to be referenced, then an AS-Cone for the relationship needs to be created.

2.1.1. Naming convention for Policy definition objects

A Policy object is referenced using the Autonomous System number it refers to, preceded by the string "AS".

2.1.2. ASN.1 format of a Policy Definition object

```
ASNPolicy DEFINITIONS ::=
BEGIN
Neighbours ::= SEQUENCE OF Neighbour

Neighbour ::= SEQUENCE
{
ASN INTEGER (1..42949672965),
ASCone  VisibleString
}

Version ::= INTEGER
LastModified ::= GeneralizedTime
Created ::= GeneralizedTime
ContactEmail ::= PrintableString(SIZE (1..75))
END
```

ASN.1 format of a Policy definition object

2.1.3. Naming convention for neighbour relationships

When referring to a neighbour relationship contained in a Policy definition object, the following convention should be used:

ASX:ASY

Where X is the number of the ASN holder and Y is the number of the ASN intended to use the AS-Cone object to generate a filter.

2.2. AS-Cone definition object

An AS-Cone contains a list of the downstream customer ASNs and AS-Cones of a given ASN. The list is used to create filter lists by the networks providing transit to or having a peering relationship with the ASN.

An AS-Cone can reference another AS-Cone.

2.2.1. Adding entries in an AS-Cone object

When an entry is added, it is in the Unverified status, and its "Verified" variable is set to 0.

If an ASN is added as an entry, it becomes directly visible and usable in building prefix lists, and a notification is sent to the E-mail address contained in the "ContactEmail" field of the AS-Cone Policy Object for that Autonomous System Number. The holder of the Autonomous System Number can acknowledge the notification, in which case the "Verified" field is switched to the value of 1.

If an AS-Cone is added to the object, a notification is sent to the E-Mail address contained in the "ContactEmail" field of the AS-Cone object that is being added. If the "ContactEmail" field is blank, the notification is sent to the E-mail address contained in the "ContactEmail" field of the AS-Cone Policy Object of the ASN of which the AS-Cone is part of. Only when an acknowledgement from the holder of the object is obtained, the "Verified" field is changed to a value of 1, and the AS-Cone becomes visible.

The value of the "Verified" field is fundamental for the creation of appropriate prefix filtering rules as described later.

2.2.2. Removal of entries from an AS-Cone object

The owner of an AS-Cone can remove any entry from its object without requesting any permission from the holders of the entries being removed.

The holder of an entry in a third party AS-Cone can remove the entry by performing authentication based on the E-mail address contained in the "ContactEmail" field of the resource itself. The RIRs MUST provide means to perform this authentication via an auth code, an API, or other means. The removal of an entry SHOULD be immediate upon successful authentication.

2.2.3. Naming convention for AS-Cone objects

AS-Cones MUST have a unique name for the ASN they belong to. Names are composed of ASCII strings up to 255 characters long and cannot contain spaces.

In order for AS-Cones to be unique in the global routing system, their string name is preceded by the AS number of the ASN they are part of, followed by ":". For example, AS-Cone "EuropeanCustomers" for ASN 65530 is represented as "AS65530:EuropeanCustomers" when referenced from a third party.

2.2.4. ASN.1 format of an AS-Cone

```
ASCone DEFINITIONS ::=
BEGIN
Entities ::= SEQUENCE OF Entity

Entity CHOICE
{
        ASN INTEGER (1..4294967295),
        OtherASCone VisibleString
        Verified ::= BOOLEAN
}

Version ::= INTEGER
LastModified ::= GeneralizedTime
Created ::= GeneralizedTime
ContactEmail ::= PrintableString(SIZE (1..75))
END
```

ASN.1 format of an AS-Cone

3. Validating an AS-Cone

In order to validate a full AS-Cone, a network operator MUST have access to the validated cache of an RPKI validator software containing all the Policy definition and AS-Cone objects. Validation occurs following the description in [RFC6488]:

In order to validate a full AS-Cone, an operator SHOULD perform the following steps:

1. For every downstream ASN, the operator verifies if a related policy definition (see Section 2.1) file exists. If no object exists, the status of the AS-Cone is "Unknown". If instead it

exists, it proceeds to collect a list of ASNs for the cone by looking at the following data, in exact order:

1. A policy for the specific relationship, in the form of ASX:ASY, where ASX is the downstream ASN, and ASY is the ASN of the operator validating the AS-Cone;

2. If there is no specific definition for the relationship, the ASX:Default policy;

If none of the two definitions above exists, then the operator should only consider the ASN of its downstream to be added to the list.

2. These objects can either point to:

    1. An AS-Cone; or

    2. An ASN

3. If the definition points to an AS-Cone, the operator looks for the object referenced, which should be contained in the validated cache;

4. If the validated cache does not contain the referenced object, then the validation moves on to the next downstream ASN;

5. If the validated cache contains the referenced object, the validation process evaluates every entry in the AS-Cone. For each entry:

    1. If there is a reference to an ASN, then the operator adds the ASN to the list for the given AS-Cone;

    2. If there is a reference to another AS-Cone, the validating process should recursively process all the entries in that AS-Cone first, with the same principles contained in this list.

    Since the goal is to build a list of ASNs announcing routes in the AS-Cone, then if an ASN or an AS-Cone are referenced more than once in the process, their contents should only be added once to the list. This is intended to avoid endless loops, and in order to avoid cross-reference of AS-Cones.

6. When all the AS-Cones referenced in the policies have been recursively iterated, and all the originating ASNs have been taken into account, the operator can then build a full prefix-

list with all the prefixes originated in its AS-Cone. This can be done by querying the RPKI validator software for all the networks originated by every ASN referenced in the AS-Cone.

4. Types of validation for AS-Cones

AS-Cones can be validated in 4 different ways:

Loose Validation. This is the method described in the procedure above;

Opportunistic Validation. This is similar to Loose validation, but it discards all the ASNs for which the "Validated" fields have a value of 0. The intent is to remove from the prefix list all the ASNs that haven't validated their entry in the customer cone for the operator;

Almost-Strict validation. In this method, whenever an entry with the "Validated" field set to 0 is found, the entire sub-tree (the AS-Cone) in which it is contained is discarded.

Strict Validation. In this method, only the entries with the "Validated" field set to 1 are considered. If even a single entry has a "Validated" field set to 0, the whole AS-Cone is discarded.

It is important to note that no AS-Cone with the "Validated" field set to 0 is going to be visible at any time, so they are automatically discarded. This protects AS-Cone holders from being considered customers of a third party without their consent.

5. Recommendations for use of AS-Cones at Internet Exchange points

When an operator is a member of an internet exchange point, it is recommended for it to create at least a Default policy.

In case of a peering session with a route server, the operator could publish a policy pointing to the ASN of the route server. A route server operator, then, could build strict prefix filtering rules for all the participants, and offer it as a service to its members.

For internet exchange points operators, the recommendation is to use Strict Filtering as explained in the previous section.

6. Publication of AS-Cones as IRR objects

AS-Cones are very similar to AS-Set RPSL Objects, so they could also be published in IRR Databases as AS-Set objects. Every ASN contained in an AS-Cone, and all the AS-Cones referenced should be considered

as member: attributes. The naming convention for AS-Cones (ASX:AS-Cone) should be maintained, in order to keep consistency between the two databases.

7. Security Considerations

TBW

8. IANA Considerations

This memo includes no request to IANA.

9. Contributors

The following people contributed significantly to the content of the document: Greg Skinner.

10. Acknowledgments

The authors would like to thank Randy Bush, Nick Hilliard and Aftab Siddiqui.

11. References

11.1. 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>.

[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>.

[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>.

11.2. Informative References

[RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, DOI 10.17487/RFC2622, June 1999, <https://www.rfc-editor.org/info/rfc2622>.

[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>.

[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>.

Authors' Addresses

Job Snijders
NTT Ltd.
Theodorus Majofskistraat 100
Amsterdam 1065 SZ
The Netherlands

Email: job@ntt.net

Massimiliano Stucchi
Independent

Email: max@stucchi.ch

Melchior Aelmans
Juniper Networks
Boeing Avenue 240
Schiphol-Rijk 1119 PZ
The Netherlands

Email: maelmans@juniper.net