

Workgroup: Internet Engineering Task Force  
Internet-Draft:  
draft-kwvanhove-sidrops-rpki-tree-hints-01  
Published: 13 December 2021  
Intended Status: Standards Track  
Expires: 16 June 2022  
Authors: K.W. van Hove

University of Twente

## **Tree Hints for the Resource Public Key Infrastructure (RPKI)**

### **Abstract**

In the Resource Public Key Infrastructure (RPKI), holders of IP address space can become a Certification Authority (CA), optionally hosting their repository. They can also delegate (part of) their resources to subordinate CAs, who in turn may do the same. This CA hierarchy forms a tree structure. Relying Party (RP) software walks this tree and determines the current valid objects. An underlying assumption is that this tree is a reasonable size, and that the information can be processed within reasonable time. This assumption is not guaranteed to hold. This document describes two new extensions, "maxDescendants" and "maxVrps", that add constraints for use in RP processing that ensure this assumption holds.

### **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 16 June 2022.

### **Copyright Notice**

Copyright (c) 2021 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 respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
  - [1.1. Requirements Language](#)
- [2. Scope](#)
- [3. Resource Certificate Extensions](#)
  - [3.1. maxDescendants](#)
  - [3.2. maxVrps](#)
- [4. Validation](#)
- [5. IANA Considerations](#)
- [6. Security Considerations](#)
- [7. References](#)
  - [7.1. Normative References](#)
- [Appendix A. Example Resource Certificate Author's Address](#)

## 1. Introduction

In the RPKI, holders of IP address space can host their own repositories and act as their own CA. They have full control over that repository and any objects signed by their CA. They may, for example, sign one or more certificates that hold a subset of the resources from the parent certificate. These certificates may reference publication points in the same repository or different ones. These new certificates can, in turn, do the same, ad infinitum. The nested structure of CAs forms a tree structure. The root of these trees are defined by the Trust Anchor Locators (TALs) [RFC8630]. RP software is assumed to walk this tree, visit every node, and retrieve all objects (e.g. Manifests [RFC6486], Route Origin Authorizations [RFC6482], Ghostbuster records [RFC6493], other certificates [RFC6481], etc.). RP software collects all information from the objects and processes it. It is important to note that RP software needs to visit every repository and consider every object CAs put on manifests. If it would exclude any repository or CA, then a BGP advertisement that should be valid can become invalid. For example, if a ROA for the prefix 2001:DB8::/32 and AS64496 is included, but the ROA for 2001:DB8:123::/48 and AS64497 (from another CA) is not, then a BGP speaker performing ROV validation may falsely reject the latter, more specific, announcement.

For RP software to fully walk the tree, the tree needs to be finite and reasonably sized. However, the size of the tree can only be determined while traversing the tree - RP software cannot verify these properties in advance. A malicious CA could, for example, create its children in an ad-hoc fashion while RP software is discovering it, thereby violating the implicit assumption that the tree is finite. That specific behaviour can be countered by RP software by setting a maximum depth for a certificate chain. However, at 10 children per child, the number of repositories would already reach 1111111111 ( $10^0 + 10^1 + \dots + 10^9$ ) after a modest 10 levels. With other strategies, such as serving gigabytes of data and simulating a very low bandwidth, a malicious repository can violate our second assumption that the tree is reasonably sized. Using malicious repository content, any CA can cause the process to take so unreasonably long that RP software does not finish processing in a reasonable amount of time (possibly years). The size and structure of nodes in the RPKI tree varies. For example, a NIR may have a legitimate need for hundreds of child-CAs, while a regular CA under the same parent does not. This diversity makes heuristics unsuitable for detecting this issue adequately, only discarding the malicious repository and its children, without heavily restricting the freedom of the structure of RPKI or causing false positives capping future growth.

Likewise, there may be valid reasons for splitting a prefix into many subprefixes, or authorising subprefixes for many autonomous system numbers (ASNs), but allowing any party to add limitless prefix-ASN pairs may overflow BGP Origin Validation tables. Setting a fixed limit may be problematic in these cases.

The new certificate extensions, "maxDescendants" and "maxVrps", are added to mitigate this issue by providing RP software prior knowledge about the tree limits before walking the tree.

### **1.1. Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

## **2. Scope**

The scope of "maxDescendants" and "maxVrps" is to provide guidance for RP software regarding the expected structure of the tree, as well as impose requirements on other aspects of the CA and its repository. Following the information in "maxDescendants" and "maxVrps" is RECOMMENDED. However, local policy MAY prevail.

### 3. Resource Certificate Extensions

These extensions extend the already defined extensions for PKIX Resource Certificates as defined in RFC 6487 section 4.8 [[RFC6487](#)]. Both maxDescendants and maxVrps SHOULD appear at least once in each certificate chain. If these extensions are absent on a certificate, it means that this certificate imposes no additional limits.

#### 3.1. maxDescendants

This extension is a non-critical extension that defines the maximum cumulative amount of descendants under this CA. BGPsec Router Certificates [[RFC8209](#)] are not counted because they do not add child-objects to the validation tree. At a value of 0, an RP SHOULD NOT visit any of the child CAs listed on the manifest. At a value of N, an RP SHOULD at most visit N descendants of this CA. This does not affect the amount of EE certificates used for signing objects like manifests and ROAs - those are not affected by this limit. This does include children and children of children. If maxDescendants is defined at multiple levels in the certificate chain, then the strictest limit MUST prevail.

The maxDescendants extension contains the maximum amount of children the CA may at most have. This number SHOULD be lower than the effective maxDescendants value for this CA. A value higher than or equal to the effective maxDescendants value will cause the children to have an effective maxDescendants value equal to the effective maxDescendants value of this CA minus one.

#### 3.2. maxVrps

This extension is a non-critical extension that defines the maximum cumulative amount of Validated ROA Payloads (VRPs) [[RFC6811](#)] under this CA. At a value of 0, an RP SHOULD NOT accept any of the ROAs under this CA. At a value of N, an RP SHOULD create most accept N VRPs based on data from this CA and its descendants. This means that at a limit of 25, one can create five ROAs with different ASNs with each five prefixes, or one ROA with 25 prefixes, or any combination that ensures the VRP count stays less or equal to maxVrps. This includes data from ROAs at children and children of children. If maxDescendants is defined at multiple levels in the certificate chain, then the strictest limit MUST prevail.

### 4. Validation

In order to validate the limits, RP software constructs the chain of certificates from the current certificate up to the root. For each limit, RP software should check for each certificate in this chain whether that certificate defines a limit. Then the most strict limit of all limits present in the chain should be used as limit. During

evaluation the RP software checks whether any limits have been violated, and if so, stops processing below the violating branch of the tree. If a limit is absent from the entire chain, a reasonable default SHOULD be used. Root CAs SHOULD define all limits on certificates for third-parties.

## 5. IANA Considerations

This document registers the following RPKI extensions:

Name: maxDescendants

OID: xxx

Reference: [RFCxxxx] (this document)

Name: maxVrps

OID: xxx

Reference: [RFCxxxx] (this document)

## 6. Security Considerations

This document contains security enhancements for the tree discovery process in the RPKI protocol. maxDescendants and maxVrps can help prevent a number of denial of service attacks against RP instances.

There may be maxDescendants and maxVrps extensions published at the root level with very large allowances, thereby effectively negating the protections offered. The same precautions described in [\[RFC8630\]](#) apply here as well.

CAs should be careful with setting their maxDescendants limits. If the maxDescendants value times the amount of children of a CA is higher than the effective maxDescendants value of that CA, then one or more children may cause the maximum amount of children to be exceeded, even if none act malicious. This may cause routing data to not be retrieved. For example, take a CA A with three children: AA, AB, and AC. A has an effective maxDescendants of 10, and sets its maxDescendants value to 5, which thus applies to AA, AB, and AC. If both AA and AB decide to fully use their five children, for example by creating AAA, AAB, AAC, AACA, AACB, ABA, ABAA, ABAAA, ABAAAA, and ABAAAAA, then RP software may no longer check AC, as AA and AB together already hit the effective maxDescendants of A. Note that the retrieval order is not defined, thus different RP software may decide to first retrieve AA, AB, and AC, and exclude a different CA, for example ABAAAAA. This also applies to maxVrps.

This may lead to RP software not retrieving data from certain CAs, which can lead to partial data. The threat that comes with partial data is that, for example, a BGP advertisement that should be valid, may become invalid, as the ROA for the advertisement is missing, and the less-specific prefix does have a ROA that was retrieved. When choosing limits, careful consideration must be taken to ensure that malicious actors cannot disrupt RPKI, whilst the data from valid actors is still retrieved.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, DOI 10.17487/RFC6493, February 2012, <<https://www.rfc-editor.org/info/rfc6493>>.
- [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>>.
- [RFC8209] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", RFC 8209, DOI 10.17487/

RFC8209, September 2017, <<https://www.rfc-editor.org/info/rfc8209>>.

[RFC8630] Huston, G., Weiler, S., Michaelson, G., Kent, S., and T. Bruijnzeels, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 8630, DOI 10.17487/RFC8630, August 2019, <<https://www.rfc-editor.org/info/rfc8630>>.

#### **Appendix A. Example Resource Certificate**

The following is the example resource certificate from [RFC 6487](#) [[RFC6487](#)] adapted with maxDescendants and maxVrps.

Certificate Name: 9JfgAEcq7Q-47IwMC5CJIJr6EJs.cer

Data:

Version: 3 (0x2)

Serial: 1500 (0x5dc)

Signature Algorithm: SHA256WithRSAEncryption

Issuer: CN=APNIC Production-CVPQsgUkLy7pOXdNeVWgvnFX\_0s

Validity

Not Before: Oct 25 12:50:00 2008 GMT

Not After : Jan 31 00:00:00 2010 GMT

Subject: CN=A91872ED

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (2048 bit)

Modulus (2048 bit):

00:bb:fb:4a:af:a4:b9:dc:d0:fa:6f:67:cc:27:39:  
34:d1:80:40:37:de:88:d1:64:a2:f1:b3:fa:c6:7f:  
bb:51:df:e1:c7:13:92:c3:c8:a2:aa:8c:d1:11:b3:  
aa:99:c0:ac:54:d3:65:83:c6:13:bf:0d:9f:33:2d:  
39:9f:ab:5f:cd:a3:e9:a1:fb:80:7d:1d:d0:2b:48:  
a5:55:e6:24:1f:06:41:35:1d:00:da:1f:99:85:13:  
26:39:24:c5:9a:81:15:98:fb:5f:f9:84:38:e5:d6:  
70:ce:5a:02:ca:dd:61:85:b3:43:2d:0b:35:d5:91:  
98:9d:da:1e:0f:c2:f6:97:b7:97:3e:e6:fc:c1:c4:  
3f:30:c4:81:03:25:99:09:4c:e2:4a:85:e7:46:4b:  
60:63:02:43:46:51:4d:ed:fd:a1:06:84:f1:4e:98:  
32:da:27:ee:80:82:d4:6b:cf:31:ea:21:af:6f:bd:  
70:34:e9:3f:d7:e4:24:cd:b8:e0:0f:8e:80:eb:11:  
1f:bc:c5:7e:05:8e:5c:7b:96:26:f8:2c:17:30:7d:  
08:9e:a4:72:66:f5:ca:23:2b:f2:ce:54:ec:4d:d9:  
d9:81:72:80:19:95:57:da:91:00:d9:b1:e8:8c:33:  
4a:9d:3c:4a:94:bf:74:4c:30:72:9b:1e:f5:8b:00:  
4d:e3

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

F4:97:E0:00:47:2A:ED:0F:B8:EC:8C:0C:0B:90:89:  
20:9A:FA:10:9B

X509v3 Authority Key Identifier:

keyid:09:53:D0:4A:05:24:2F:2E:E9:39:77:4D:79:  
55:86:BE:71:57:FF:4B

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

X509v3 Basic Constraints: critical

CA:TRUE



X509v3 CRL Distribution Points:

URI:rsync://rpki.apnic.net/repository/A3C38A24  
D60311DCAB08F31979BDBE39/CVPQSGUkLy7p0XdNe  
VWGvnFX\_0s.crl

Authority Information Access:

CA Issuers - URI:rsync://rpki.apnic.net/repos  
itory/8BDFC7DED5FD11DCB14CF4B1A703F9B7/CVP  
QSGUkLy7p0XdNeVWGvnFX\_0s.cer

X509v3 Certificate Policies: critical

Policy: 1.3.6.1.5.5.7.14.2

Subject Information Access:

CA Repository - URI:rsync://rpki.apnic.net/mem  
ber\_repository/A91872ED/06A83982887911DD81  
3F432B2086D636/

Manifest - URI:rsync://rpki.apnic.net/member\_r  
epository/A91872ED/06A83982887911DD813F432  
B2086D636/9JfgAEcq7Q-47IwMC5CJIJr6EJs.mft

sbgp-autonomousSysNum: critical

Autonomous System Numbers:

24021  
38610  
131072  
131074

sbgp-ipAddrBlock: critical

IPv4:

203.133.248.0/22  
203.147.108.0/23

maxDescendants:

16

maxVrps:

2048

**Author's Address**

Koen van Hove  
University of Twente

Email: [koen@koenvh.nl](mailto:koen@koenvh.nl)