Constraining RPKI Trust Anchors
draft-snijders-constraining-rpki-trust-anchors

Job Snijders
Fastly / OpenBSD
job@fastly.com

What is “constraining RPKI Trust Anchors”?

An approach for RPKI Relying Parties to impose locally configured *constraints* on cryptographic products subordinate to publicly-trusted Trust Anchors, as implemented in OpenBSD's rpki-client validator.

The ability to constrain a Trust Anchor operator's effective signing authority to a limited set of Internet Number Resources allows Relying Parties to **enjoy the potential benefits of assuming trust - within a bounded scope**.

Trust Anchors can't really constrain themselves.

Authority for any resource is just certificate issuance away!

1.1. Definitions

Assumed Trust

In the RPKI hierarchical structure, a Trust Anchor is an authority for which trust is assumed and not derived. Assumed trust means that violation of that trust is out-of-scope for the threat model.

Derived Trust

Derived Trust can be automatically and securely computed with subjective logic. In the context of the RPKI, trust is derived according to the rules for validation of RPKI Certificates and Signed Objects.

Constraints

The locally configured union set of IP prefixes, IP address ranges, AS identifiers, and AS identifier ranges for which the Relying Party operator anticipates CA operators under a given Trust Anchor to issue cryptographic products.

A mini policy language was created

The mini policy language uses two keywords: **allow** and **deny**

```
# cat example.constraints
allow 0.0.0.0/0
deny 10.0.0.0/8
deny 172.16.0.0/12
```

Causes the validator to reject End-Entity certificates which contain 10.0.0.0/24 in the RFC 3779 extensions

Simple example

```
# cat /etc/rpki/ripe.constraints
deny 0.0.0.0/0
deny ::/0

# rpki-client -t /etc/rpki/ripe.tal 2>&1 | grep violates | head -5
rpki-client:
rpki.ripe.net/repository/DEFAULT/b4/566ef5-d13d-474c-a299-298fbe7dc168/1/ZCCXyxSxofpWvFbPyfQfVjyn
h4U.roa: resource violates ripe.constraints: 194.9.4.0/23
rpki-client:
rpki.ripe.net/repository/DEFAULT/e9/f3891f-4dc7-4f4c-b519-6107eb7c48d2/1/KWIxqPkuvRu-x-V0dydEMVWL
BEI.roa: resource violates ripe.constraints: 185.204.197.0/24
rpki-client:
rpki.ripe.net/repository/DEFAULT/e0/b3c215-d866-42d0-a6c3-20677d80e838/1/NRqEkvjaiHLhP7YwHQ36JqAy
dNk.roa: resource violates ripe.constraints: 185.75.120.0/22
rpki-client:
rpki.ripe.net/repository/DEFAULT/e0/b3c215-d866-42d0-a6c3-20677d80e838/1/GuBACEaaLKhTCHhpo7PwnylC
eoo.roa: resource violates ripe.constraints: 185.75.120.0/22
rpki-client:
rpki.ripe.net/repository/DEFAULT/59/1759da-c66e-43c8-881d-6db090214035/1/fzSUNgxMmiJxsae3BEa4N-LT
cSs.roa: resource violates ripe.constraints: 194.35.104.0/22
```

Operational considerations

- ARIN has no IPv6 transfer policy
- AFRINIC has no IPv4 transfer policy
- AFRINIC has no IPv6 transfer policy
- AFRINIC has no ASN transfer policy
- LACNIC has no IPv6 transfer policy
- LACNIC has no ASN transfer policy
- APNIC has no IPv6 transfer policy
- RIPE NCC has no other RIR to transfer IPv6 space from or to
- None of the RIRs manage RFC1918 / RFC6598 / RFC 6890 / etc IP space
- None of the RIRs manage RFC 5398 / RFC 6996 / etc ASN identifiers

Lot of research - constructed a 5 way filter

```
# wc -l *.constraints
    631    afrinic.constraints
    984    apnic.constraints
    980    arin.constraints
    754    lacnic.constraints
    987    ripe.constraints
```

https://github.com/openbsd/src/tree/master/etc/rpki

```
commit b44517b3478be6ff687445f7ca922657a2fd0bc4
Author: job <job@openbsd.org>
Date:   Thu Dec 14 09:13:00 2023 +0000

    For historical reasons, APNIC ended up with a v6 block for IX assignments
    carved out of a larger block assigned to RIPE NCC

    OK tb@

diff --git etc/rpki/apnic.constraints etc/rpki/apnic.constraints
index 420b86f0cc9..dd97bd48a9d 100644
--- etc/rpki/apnic.constraints
+++ etc/rpki/apnic.constraints
@@ -8,6 +8,9 @@ allow 2001:a000::/20
 allow 2001:b000::/20
 allow 2400::/12

+# IX Assignments
+allow 2001:7fa::/32
+
 # AFRINIC Internet Number Resources cannot be transferred
 # From https://www.iana.org/assignments/ipv4-address-space/
 deny 41.0.0.0/8
```

```
# AFRINIC IPv4 resources cannot be transferred to APNIC
# From https://www.iana.org/assignments/ipv4-address-space/
deny 41.0.0.0/8
deny 102.0.0.0/8
deny 105.0.0.0/8
deny 154.0.0.0/16
deny 154.16.0.0/16
deny 154.65.0.0 - 154.255.255.255
deny 196.0.0.0 - 196.1.0.255
deny 196.1.4.0/24
deny 196.1.7.0 - 196.1.63.255
deny 196.1.71.0/24
deny 196.1.74.0 - 196.1.103.255
deny 196.1.115.0 - 196.1.133.255
deny 196.1.137.0/24
deny 196.1.143.0 - 196.1.159.255
deny 196.1.176.0 - 196.1.255.255
deny 196.2.2.0/23
deny 196.2.8.0 - 196.2.255.255
deny 196.3.14.0/23
deny 196.3.57.0 - 196.3.64.255
deny 196.3.90.0/24
deny 196.3.92.0 - 196.3.94.255
```

```
# Private use IPv4 & IPv6 addresses and ASNs
deny 0.0.0.0/8                # RFC 1122 Local Identification
deny 10.0.0.0/8               # RFC 1918 private space
deny 100.64.0.0/10            # RFC 6598 Carrier Grade NAT
deny 127.0.0.0/8              # RFC 1122 localhost
deny 169.254.0.0/16           # RFC 3927 link local
deny 172.16.0.0/12            # RFC 1918 private space
deny 192.0.2.0/24             # RFC 5737 TEST-NET-1
deny 192.88.99.0/24           # RFC 7526 6to4 anycast relay
deny 192.168.0.0/16           # RFC 1918 private space
deny 198.18.0.0/15            # RFC 2544 benchmarking
deny 198.51.100.0/24          # RFC 5737 TEST-NET-2
deny 203.0.113.0/24           # RFC 5737 TEST-NET-3
deny 224.0.0.0/4              # Multicast
deny 240.0.0.0/4              # Reserved
deny 23456                    # RFC 4893 AS_TRANS
deny 64496 - 64511            # RFC 5398
deny 64512 - 65534            # RFC 6996
deny 65535                    # RFC 7300
deny 65536 - 65551            # RFC 5398
deny 65552 - 131071           # IANA Reserved
deny 4200000000 - 4294967294 # RFC 6996
deny 4294967295               # RFC 7300

# APNIC supports IPv4 and ASN transfers: allow the complement of what is denied
allow 0.0.0.0/0
allow 1 - 4199999999
```

IANA IPv6 Global Unicast Address Assignments Registry Update

David Dong david.dong at iana.org
Sat Nov 2 01:16:42 UTC 2024

- Previous message (by thread): Weekly Global IPv4 Routing Table Report
- Next message (by thread): Microsoft NOC Contact - ASN Association Request
- **Messages sorted by:** [date] [thread] [subject] [author]

Hi,

The IANA IPv6 Global Unicast Address Assignments registry has been updated to reflect the allocation of the following block to APNIC:

2410::/12 APNIC 2024-11-01

You can find the registry at:

https://www.iana.org/assignments/ipv6-unicast-address-assignments/

The allocation was made in accordance with the Policy for Allocation of IPv6 Blocks to Regional Internet Registries:

https://www.icann.org/resources/pages/allocation-ipv6-rirs-2012-02-25-en

Best Regards,

David Dong
IANA Services Sr. Specialist

1 file changed +2 -1 lines changed

etc/rpki/apnic.constraints +2 -1

```
@@ -1,4 +1,4 @@
-#       $OpenBSD: apnic.constraints,v 1.6 2024/04/17 14:31:59 job Exp $
+#       $OpenBSD: apnic.constraints,v 1.7 2024/11/02 09:43:12 job Exp $

 # From https://www.iana.org/assignments/ipv6-unicast-address-assignments
 allow 2001:200::/23
@@ -9,6 +9,7 @@ allow 2001:8000::/19
 allow 2001:a000::/20
 allow 2001:b000::/20
 allow 2400::/12
+allow 2410::/12

 # IX Assignments
 allow 2001:7fa::/32
```

Other approaches

Single root trust anchor managed by NRO

The experiment: https://labs.apnic.net/nro-ta/

A monitoring archive: https://nro-tal.rpkiviews.org/

I really really like this

Blockers for a (NRO) single trust anchor

Revision of the RPKI Validation Algorithm

https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rpki-validation-update

Two implementations underway:

- rpki-prover
- rpki-client

Next steps?

- Request WG Adoption?
- Publish via Independent Stream Editor (ISE)?