IRR & RPKI feature parity

Job Snijders, Massimiliano Stucchi
job@ntt.net, mstucchi@ripe.net

IETF 101

What is this about

- Creating more feature parity between IRR and RPKI
- Making provisioning operations easier
- Enabling South America

An IRR route object: the atom

```
$ whois -h rr.ntt.net 192.147.168.0/24

route:          192.147.168.0/24
descr:          Job Snijders
origin:         AS15562
notify:         job@instituut.net
mnt-by:         MAINT-JOB
changed:        job@ntt.net 20161003
source:         NTTCOM
```

(only the bold lines are relevant in the process)

Generating a prefix filter

```
job@vurt ~$ whois -h rr.ntt.net '!gAS15562'
A212
165.254.255.132/32 165.254.255.26/32
165.254.255.0/25 165.254.255.144/28
165.254.255.133/32 192.147.168.0/24
165.254.255.160/28 165.254.255.149/32
209.24.0.0/16 204.42.254.192/26
165.254.255.0/24 67.221.245.0/24
C
job@vurt ~$
```

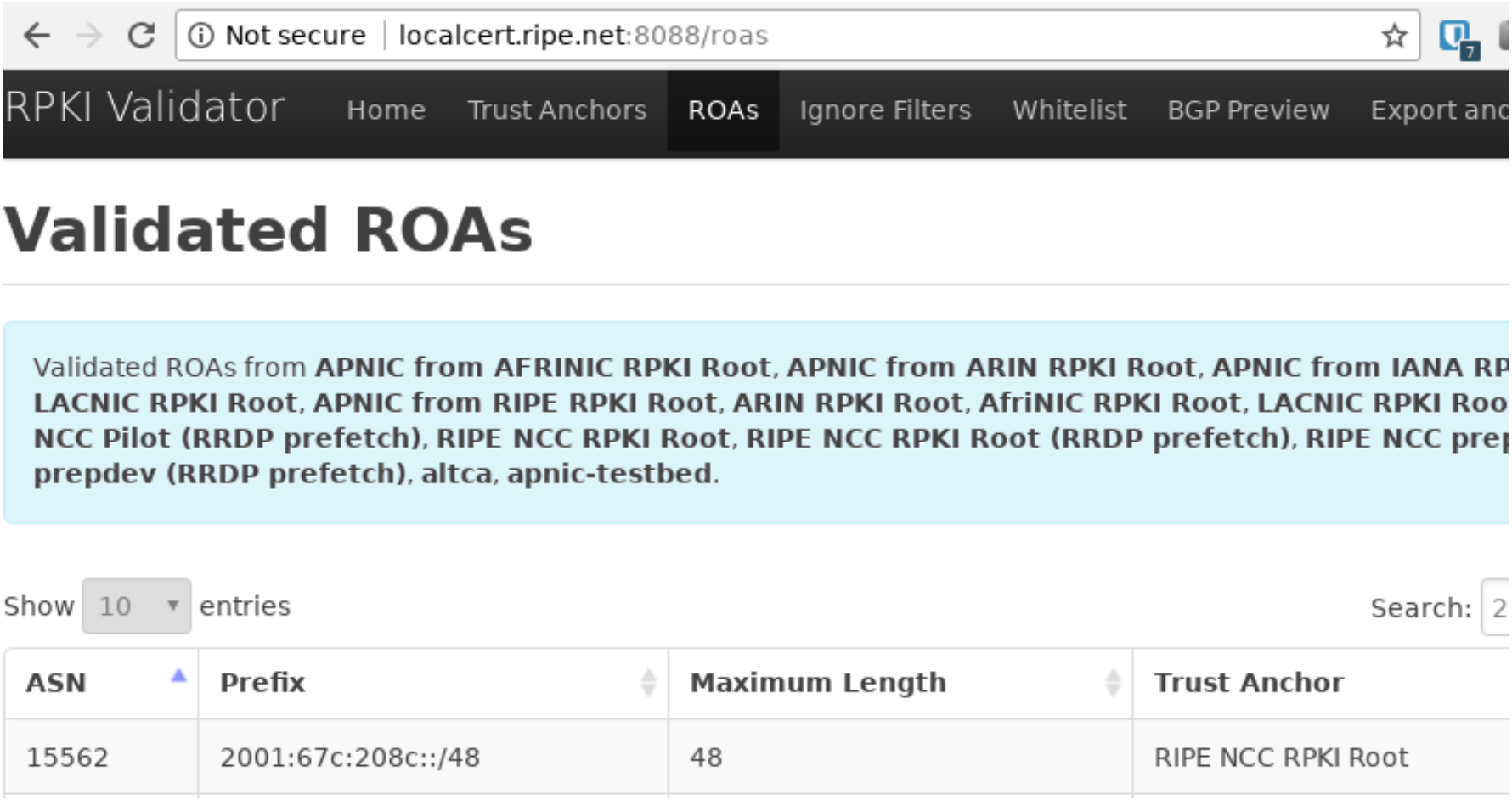
Same example, with bgpq3

```
job@vurt ~$ bgpq3 -h rr.ntt.net -l AS15562-in AS15562
no ip prefix-list AS15562-in
ip prefix-list AS15562-in permit 67.221.245.0/24
ip prefix-list AS15562-in permit 165.254.255.0/24
ip prefix-list AS15562-in permit 165.254.255.0/26
ip prefix-list AS15562-in permit 165.254.255.26/32
ip prefix-list AS15562-in permit 165.254.255.64/26
ip prefix-list AS15562-in permit 165.254.255.132/32
ip prefix-list AS15562-in permit 165.254.255.133/32
ip prefix-list AS15562-in permit 165.254.255.144/28
ip prefix-list AS15562-in permit 165.254.255.149/32
ip prefix-list AS15562-in permit 165.254.255.160/28
ip prefix-list AS15562-in permit 192.147.168.0/24
ip prefix-list AS15562-in permit 204.2.30.0/23
ip prefix-list AS15562-in permit 204.42.254.192/26
ip prefix-list AS15562-in permit 209.24.0.0/16
```

What about RPKI?

- A RPKI ROA kind of looks like a route object
- It has a “prefix” and an “origin”
- RPKI is trustworthy data, we know for sure that the owner of the IP space created the ROA

Provisioning use case for RPKI data?



Simple example

```
job@vurt ~$ ftp -VM -o - \
   http://localcert.ripe.net:8088/export.json \
   | jq '.roas[] | select(.asn | contains("AS15562"))? | .prefix' \
   | uniq
"2001:67c:208c::/48"
job@vurt ~$
```

Grouping ASNs: AS-SETs

```
$ whois -h rr.ntt.net AS15562:AS-SNIJDERS
as-set:          AS15562:AS-SNIJDERS
members:         AS15562        # Me
members:         AS57436        # Samer
members:         AS-KING        # Thomas King
members:         AS-NETHER # Jared
tech-c:          DUMY-RIPE
admin-c:         DUMY-RIPE
notify:          job@instituut.net
org:             ORG-SNIJ1-RIPE
mnt-by:          SNIJDERS-MNT
created:         2018-01-16T17:54:54Z
last-modified:   2018-01-16T17:58:36Z
source:          RIPE
```

Programmatic access to AS-SETS

```
$ whois -h rr.ntt.net '!iAS15562:AS-SNIJDERS,1'
A130
AS15562 AS202539 AS205591 AS205593 AS206479
AS206499 AS206551 AS234 AS267 AS31451 AS41731
AS49697 AS51861 AS57436 AS60003 AS61438
C
```

```
$ irrtree AS15562:AS-SNIJDERS
Processed: 0 objects (Elapsed Time: 0:00:00)
IRRTree (1.1.3) report for 'AS15562:AS-SNIJDERS'
(IPv4), using rr.ntt.net at 2018-01-24 16:23
AS15562:AS-SNIJDERS (16 ASNs, 25 pfxs)
 +-- AS-KING (12 ASNs, 8 pfxs)
 |   +-- AS205591 (2 pfxs)
 |   +-- AS206499 (2 pfxs)
 |   +-- AS49697 (2 pfxs)
 |   +-- AS51861 (1 pfxs)
 |   +-- AS60003 (1 pfxs)
 +-- AS-NETHER (2 ASNs, 4 pfxs)
 |   +-- AS267 (2 pfxs)
 |   +-- AS234 (2 pfxs)
 +-- AS15562 (12 pfxs)
 +-- AS57436 (1 pfxs)
```

Summary so far:

- An AS-SET is resolved into all its member ASNs
- For each ASN we do a reverse lookup to find all route-objects where the ASN is the “`origin:`”
- The total list of prefixes from the above 2 steps is the input into `bgpq3` and ends up on routers

Problems with AS-SETs

- What if IRR AS-SET exists in multiple IRRs?
  - AS-STEALTH exists in **both** RIPE and RADB
  - The two are **not** managed by the same company
- How to discover what AS-SET to use?
  - Ask people in the service order form?
  - Look at PeeringDB?
  - (Virtually nobody looks at import:/export: lines)

PeeringDB currently is probably the best source for this data :-/

My wishlist

- Ease of discovery
  - given ASN X – what list of downstreams should I use in my provisioning system?
- Guarantees that only the owner of the ASN could’ve created that list
- Unilateral declarations (just like AS-SETs)
- Per adjacent ASN granularity:
  - AS 15562 may announce a different set of downstreams to NTT than to GTT

Work in progress: “AS Cones”

Materials here:

https://github.com/bgp/draft-ss-grow-rpki-as-cones