APNIC

Terry Sweetser

Been doing this “Internet thing” since 1989.

Former APNIC Community Trainer, CTO, Founder, Engineering Manager, etc

APNIC Training Delivery Manager for South Asia and Oceania

Nationality: Australian

Languages: English

about.me/terry.sweetser

Dave Phelan

Involved in the ISP/MSP/Infra Game for a LONG time.

Network Engineer and Infrastructure.
If it goes in a Rack or connects to a rack

Senior Network Analyst/Technical Trainer

Nationality: British(But you would never know)

Languages: English

Elly Tawhai

Former sysadmin at UQ

Working at APNIC since Sept 2000

Senior Internet Resource Analyst (Oceania sub-region)/Liaison Officer (Pacific)

APNIC Academy

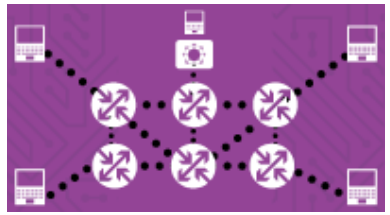
Do you have an account?

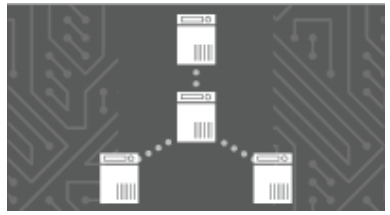
Online Courses

Self-paced courses

Webinar courses

Virtual Labs





Technical Assistance

Adli Wahid
APNIC
Subject Matter Expert
CERT/CSIRT, Security, Information Security

Jessica Wei
APNIC
Subject Matter Expert
BGP, IPv6, Internet Routing, MPLS

Jamie Gillespie
APNIC
Subject Matter Expert
Security, CERT/CSIRT, Information Security

Book an Expert

Upcoming Events

APNIC: APT Mongolia IPv6 Deployment Workshop
15-Jun-2022 — 3 days
Online — Workshop
INVITE ONLY

Reverse DNS Tutorial
21-Jun-2022 — 4 hours
Online — Tutorial
LOGIN TO REGISTER

Introduction to SDN/OpenFlow Tutorial
21-Jun-2022 — 4 hours
Online — Tutorial
LOGIN TO REGISTER

Introduction to SDN/OpenFlow Tutorial
22-Jun-2022 — 4 hours
Online — Tutorial
LOGIN TO REGISTER

OUR AGENDA

RPKI “Routing Security” in 60 minutes or less.

Signing Your ROAs.

Lab Work.

Our goals today:

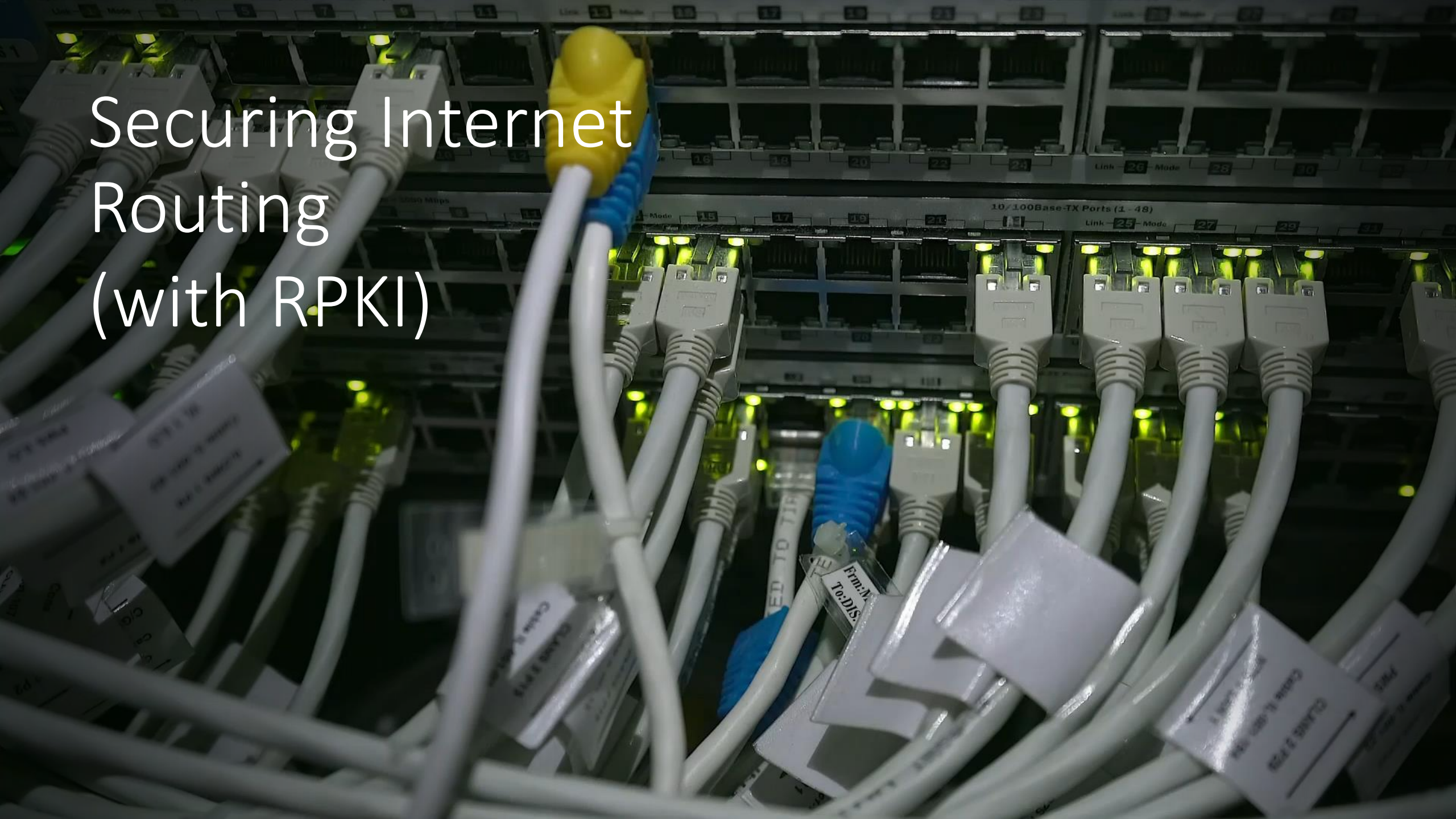
Introduce RPKI

Sign some ROAs

Look at ROV

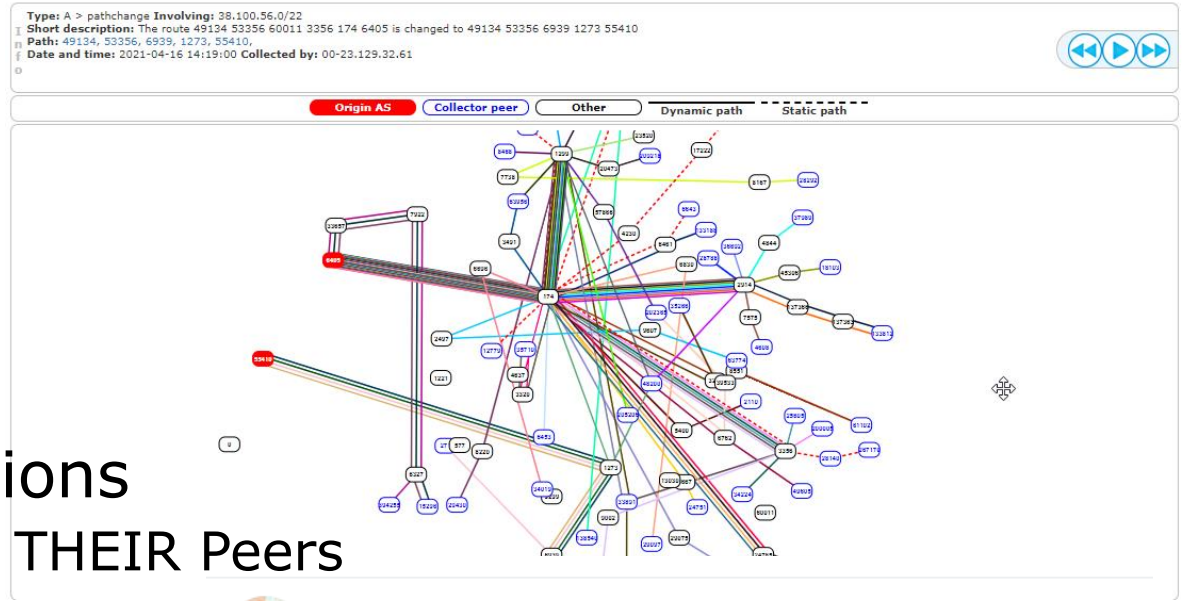
Discuss Deployment

Securing Internet Routing (with RPKI)



Headlines

- AS55410 Leaks ~30k Prefixes – **16 April 2021**
  - Approx 4k ASN Affected
    - Many with No Route Objects
    - Only ~4k Prefixes had ROA
  - Main Upstream leakers
    - AS9498(Bharti Airtel) and AS1273 (Vodafone UK)
  - Spread mostly VIA IX connections
    - Some of which re-propagated to THEIR Peers



Radar by Qrator
@Qrator_Radar

April 16, 2021 - AS55410 - VIL-AS-AP (Vodafone Idea) - hijacked 37739 prefixes - countries affected 164 - ASNs affected 4012 - duration 1:30:00

Doug Madory
@DougMadory

Large BGP routing leak out of India this morning.

AS55410 mistakenly announced over 30,000 BGP prefixes causing a 13x spike in inbound traffic to their network according to @kentikinc netflow data.

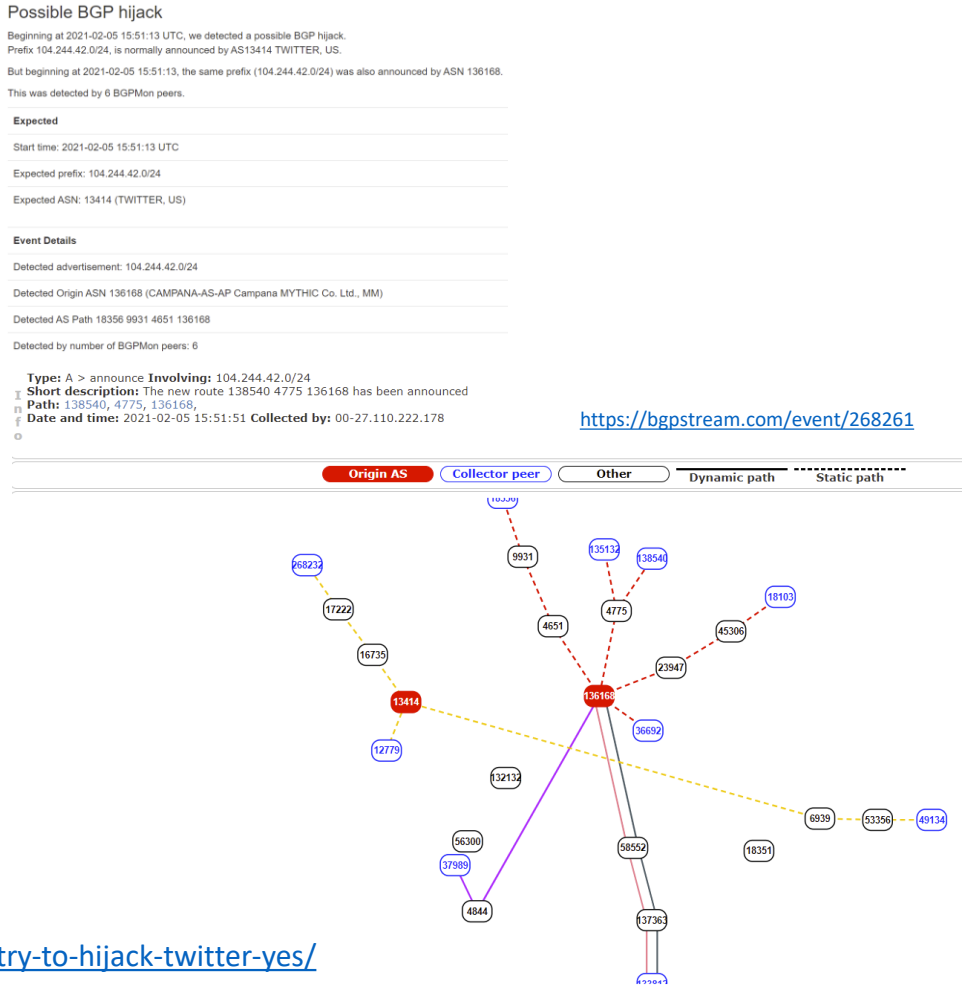
https://bgpstream.com/event/271479 https://bgpstream.com/event/271478

Headlines

- AS136168 attempts to hijack Twitter (AS13414) – **05 Feb 2021**
  - MM Military orders blocking of Twitter/Instagram
    - AS136168 originated 104.244.42.0/24
      - Out of the 91xIPv4 and 3XIPv6 prefixes Twitter/AS13414 originates?

```
~ dig twitter.com +short
104.244.42.193
```

    - Good:
      - Only 6 peers (AS36692, AS4844, AS4775, AS23947,AS132132, AS58552) accepted the announcement
      - Probably other networks doing some IRR based filtering
    - Bad:
      - Why weren't the above 6 peers filtering inbound?
      - Why didn't Twitter create ROAs for their prefixes?
      - More detailed analysis: https://www.manrs.org/2021/02/did-someone-try-to-hijack-twitter-yes/



Headlines

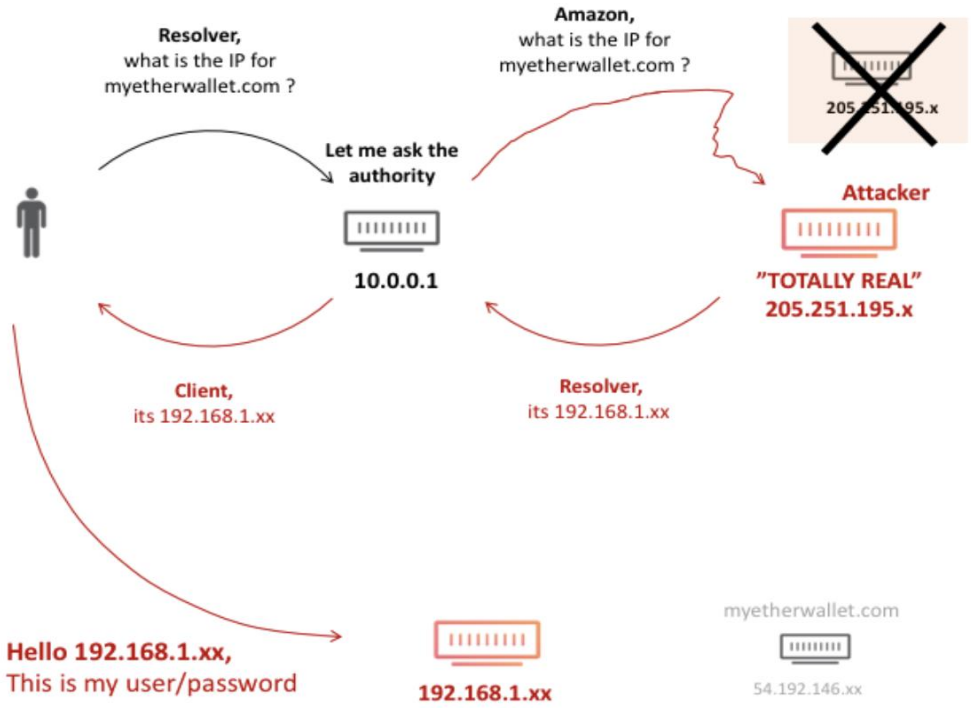
- Amazon (AS16509) Route53 hijack – **April2018**
  - AS10279 (eNET) originated more specifics (/24s) of Amazon Route53's prefix (205.251.192.0/21)

    205.251.192.0/24 ……. 205.251.199.0/24

    https://ip-ranges.amazonaws.com/ip-ranges.json
  - Its peers, like AS6939 (HE), shared these routes with 100s of their own peers…
  - The motive?
    - During the period, DNS servers in the hijacked range only responded to queries for myetherwallet.com
    - Responded with addresses associated with AS41995/AS48693

Headlines

- Route53 hijack (contd…)
  - Resolvers querying any Route53 managed names, would ask the authoritative servers controlled through the BGP hijack
    - *Possibly, used an automated cert issuer to get a cert for* myetherwallet.com
  - use _*THEIR*_ crypto to end-users to see everything (including passwords)



https://blog.cloudflare.com/bgp-leaks-and-crypto-currencies

Headlines

- YouTube (AS36561) Incident - **Feb 2008**
  - ~ 2 hours
  - AS17557 (PTCL) announced 208.65.153.0/24 (208.65.152.0/22)
    - Propagated by AS3491 (PCCW)

Why do we keep seeing these?

- Because NO ONE is in charge?
  - No single authority model for the Internet
  - No reference point for what's right in routing

Why do we keep seeing these?

- Routing works by RUMOUR
  - Tell what you know to your neighbors, and Learn what your neighbors know
  - Assume everyone is correct (and *honest*)
    - Is the originating network the rightful owner?

Why do we keep seeing these?

- Routing is VARIABLE
  - The view of the network depends on where you are
    - Different routing outcomes at different locations
  - ~ no reference view to compare the local view ☹

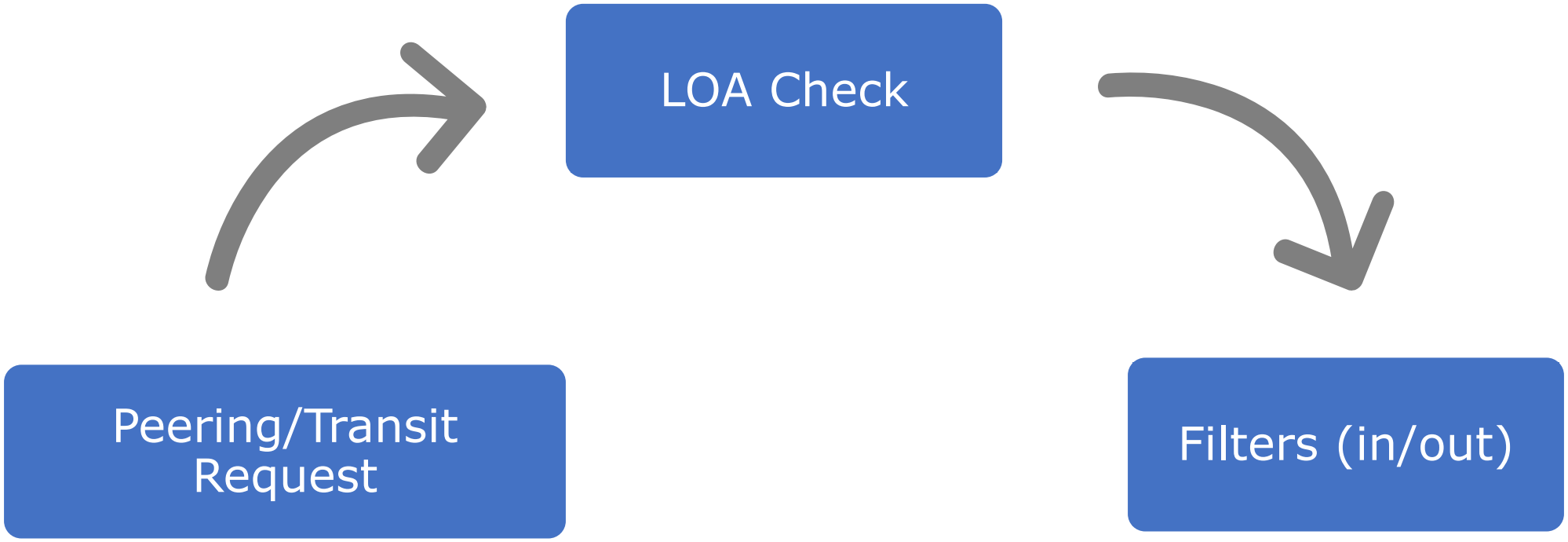
Why do we keep seeing these?

- Routing works in REVERSE
  - Outbound advertisement affects inbound traffic
  - Inbound (*Accepted*) advertisement influence outbound traffic

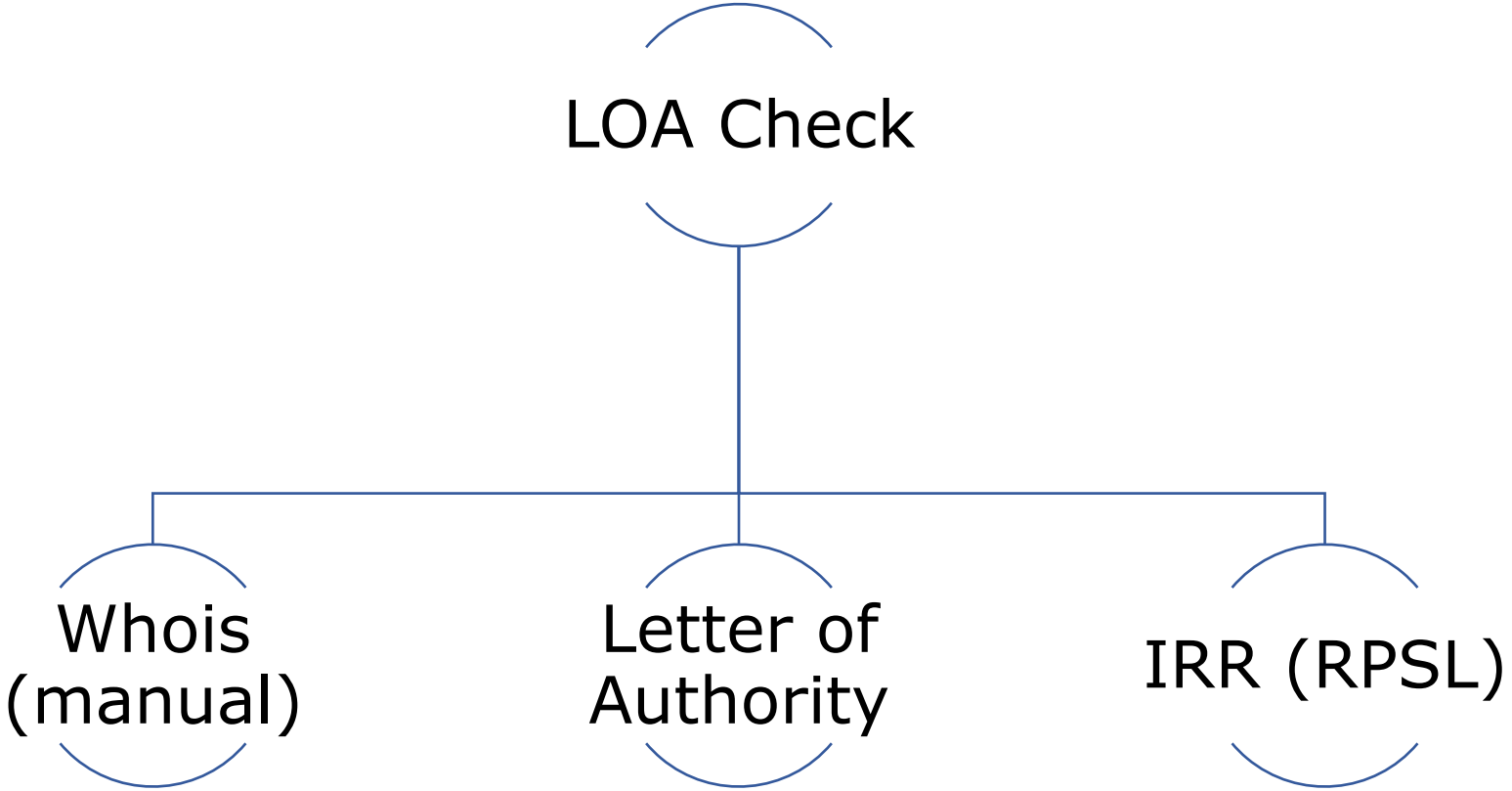
How do we address these?

- **Good Hygiene ~ Filter Filter Filter!**
  - your peers, upstream(s), and customers
    - Prefix filters/Prefix limit
    - AS-PATH filters/AS-PATH limit
    - RFC 8212 – BGP default reject or something similar

Current practice



Tools & Techniques



Tools & Techniques

- Look up **whois**
  - verify holder of a resource

```
~ whois -h whois.apnic.net 202.125.96.0
% [whois.apnic.net]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

% Information related to '202.125.96.0 - 202.125.96.255'

% Abuse contact for '202.125.96.0 - 202.125.96.255' is 'training@apnic.net'

inetnum:        202.125.96.0 - 202.125.96.255
netname:        APNICTRAINING-AP
descr:          Prefix for APNICTRAINING LAB DC
country:        AU
admin-c:        AT480-AP
tech-c:         AT480-AP
status:         ALLOCATED NON-PORTABLE
mnt-by:         MAINT-AU-APNICTRAINING
mnt-irt:        IRT-APNICTRAINING-AU
last-modified:  2016-06-17T00:17:28Z
source:         APNIC

irt:            IRT-APNICTRAINING-AU
address:        6 Cordelia Street
address:        South Brisbane
address:        QLD 4101
e-mail:         training@apnic.net
abuse-mailbox:  training@apnic.net
admin-c:        AT480-AP
tech-c:         AT480-AP
auth:           # Filtered
mnt-by:         MAINT-AU-APNICTRAINING
last-modified:  2013-10-31T11:01:10Z
source:         APNIC
```

```
role:           APNIC Training
address:        6 Cordelia Street
address:        South Brisbane
address:        QLD 4101
country:        AU
phone:          +61 7 3858 3100
fax-no:         +61 7 3858 3199
e-mail:         training@apnic.net
admin-c:        JW3997-AP
tech-c:         JW3997-AP
nic-hdl:        AT480-AP
mnt-by:         MAINT-AU-APNICTRAINING
last-modified:  2017-08-22T04:59:14Z
source:         APNIC

% Information related to '202.125.96.0/24AS131107'

route:          202.125.96.0/24
descr:          Prefix for APNICTRAINING LAB DC
origin:         AS131107
mnt-by:         MAINT-AU-APNICTRAINING
country:        AU
last-modified:  2016-06-16T23:23:00Z
source:         APNIC
```

Tools & Techniques

- Ask for a **Letter of Authority**
  - Absolve from any liabilities

APNIC

Asia Pacific Network Information Centre
APNIC Pty Ltd
ABN: 42 081 528 010

6 Cordelia Street
PO Box 3646
South Brisbane
QLD 4101 AUSTRALIA

URL www.apnic.net
Enquiries helpdesk@apnic.net
Accounts billing@apnic.net
Phone +61 7 3858 3100
Fax + 61 7 3858 3199

31/03/2018

Letter of Authorization

To whom it may concern,

APNIC Training (AS45192) runs a lab network to reproduce technical problems faced by members to help troubleshoot specific issues.

This letter serves as an authorization for APNIC Infra (AS4608) to advertise the following address blocks:

202.125.96.0/24

As a representative of APNIC Training team, that is the owner of the subnet and ASN, I hereby declare that I am authorized to sign this LOA.

Tashi Phuntsho
Training Delivery Manager

Email: tashi@apnic.net
Phone: +61 7 3858 3114

Tools & Techniques

- Look up (or ask to enter) details in **internet routing registries** (IRR)
  - describes route origination and inter-AS routing policies

```
tashi@tashi ~> whois -h whois.radb.net 61.45.248.0/24
route:          61.45.248.0/24
descr:          APNICTRAINING-DC
origin:         AS135533
mnt-by:         MAINT-AS4826
changed:        noc@vocus.com.au 20160702
source:         RADB

route:          61.45.248.0/24
descr:          Prefix for APNICTRAINING LAB - AS135533
origin:         AS135533
mnt-by:         MAINT-AU-APNICTRAININGLAB
country:        AU
last-modified:  2017-10-19T01:36:37Z
source:         APNIC
```

```
tashi@tashi ~> whois -h whois.radb.net AS17660
aut-num:        AS17660
as-name:        BT-Bhutan
descr:          Divinetworks for BT
admin-c:        DUMY-RIPE
tech-c:         DUMY-RIPE
status:         OTHER
mnt-by:         YP67641-MNT
mnt-by:         ES6436-RIPE
created:        2012-11-29T10:31:33Z
last-modified:  2018-09-04T15:26:24Z
source:         RIPE-NONAUTH
remarks:        ****************************
remarks:        * THIS OBJECT IS MODIFIED
remarks:        * Please note that all data that is generally regarded as personal
remarks:        * data has been removed from this object.
remarks:        * To view the original object, please query the RIPE Database at:
remarks:        * http://www.ripe.net/whois
remarks:        ****************************

aut-num:        AS17660
as-name:        DRUKNET-AS
descr:          DrukNet ISP
descr:          Bhutan Telecom
descr:          Thimphu
country:        BT
org:            ORG-BTL2-AP
import:         from AS6461   action pref=100;     accept ANY
export:         to AS6461     announce AS-DRUKNET-TRANSIT
import:         from AS2914   action pref=150;     accept ANY
export:         to AS2914     announce AS-DRUKNET-TRANSIT
import:         from AS6453   action pref=100;     accept ANY
export:         to AS6453     announce AS-DRUKNET-TRANSIT
```

Tools & Techniques

- IRR
  - *Helps auto generate network (prefix/as-path) filters using RPSL tools*
    - Filter out route advertisements not described in the registry

```
~ bgpq3 -bl PEERv4-IN AS17660
PEERv4-IN = [
    45.64.248.0/22,
    103.7.252.0/22,
    103.7.254.0/23,
    103.245.240.0/22,
    103.245.242.0/23,
    119.2.96.0/19,
    119.2.96.0/20,
    202.89.24.0/21,
    202.144.128.0/19,
    202.144.128.0/23,
    202.144.144.0/20,
    202.144.148.0/22
];
~ bgpq3 -6bl PEERv4-IN AS17660
PEERv4-IN = [
    2405:d000::/32,
    2405:d000:7000::/36
];
```

```
~ bgpq3 -S APNIC -bl PEERv4-IN AS17660
PEERv4-IN = [
    45.64.248.0/22,
    103.245.240.0/22,
    103.245.242.0/23,
    119.2.96.0/19
];
~ bgpq3 -S APNIC -Jl PEERv4-IN AS17660
policy-options {
replace:
 prefix-list PEERv4-IN {
    45.64.248.0/22;
    103.245.240.0/22;
    103.245.242.0/23;
    119.2.96.0/19;
 }
}
```

```
~ bgpq3 -3f 17660 -l BT-IN AS-DRUKNET-TRANSIT
no ip as-path access-list BT-IN
ip as-path access-list BT-IN permit ^17660(_17660)*$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(18024|18025|38004|59219)$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(132232|134715|135666|137925)$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(137994)$
```

```
~ bgpq3 -3f 38195 -l SUPERLOOP-IN AS-SUPERLOOP
no ip as-path access-list SUPERLOOP-IN
ip as-path access-list SUPERLOOP-IN permit ^38195(_38195)*$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(681|4647|4749|4785)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(4841|4858|5091|5740)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(6404|6461|7280|7469)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7477|7490|7578|7585)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7604|7628|7631|7699)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(8360|8444|9249|9290)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9313|9438|9463|9479)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9499|9544|9549|9661)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9795|9797|10143|10145)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(10310|11031|11054|12041)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(12189|13331|13414|13720)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(14148|15133|15562|15967)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(16164|17158|17457|17462)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17477|17498|17732|17766)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17812|17819|17829|17889)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17906|17907|17983|17985)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17991|18000|18110|18201)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(18231|18291|18292|18349)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(18385|18407|18549|18701)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(19385|19397|20473|21534)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(21859|22097|22363|23156)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23197|23352|23667|23677)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23686|23747|23858|23913)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23935|24007|24008|24033)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24065|24093|24098|24129)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24231|24233|24238|24242)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24322|24341|24380|24459)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24570|25605|25665|27232)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(29457|30081|30103|30109)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(30215|30762|31732|32771)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(36351|37993|38068|38172)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38220|38263|38269|38298)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38451|38534|38541|38570)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38716|38719|38726|38809)$
```

Tools & Techniques

- Problem(s) with IRR
  - No single authority model
    - How do I know if a RR entry is genuine and correct?
    - How do I differentiate between a current and a lapsed entry?
  - Many RRs
    - If two RRs contain conflicting data, which one do I trust and use?
  - Incomplete data - Not all resources are registered in an IRR
    - If a route is not in a RR, is the route invalid or is the RR just missing data?
  - Scaling
    - How do I apply IRR filters to upstream(s)?

Tools & Techniques

- Automating network filters (IRR filters) - Caution
  - IRR filters only as good as the correctness of the IRR entries
    - Might require manual overrides and offline verification of resource holders
    - Good idea to use specific sources (`-S` in `bgpq3`, `-s` in `rtconfig`) when generating filters, assuming mirrors are up to date

Back to basics – identify GOOD

- Could we use a digital signature to convey the *authority to use*?
  - Private key to *sign* the *authority*, and
  - Public key to *validate* the *authority*

- ~ If the holder of the resource has the private key, it can sign/authorize the use of the resource

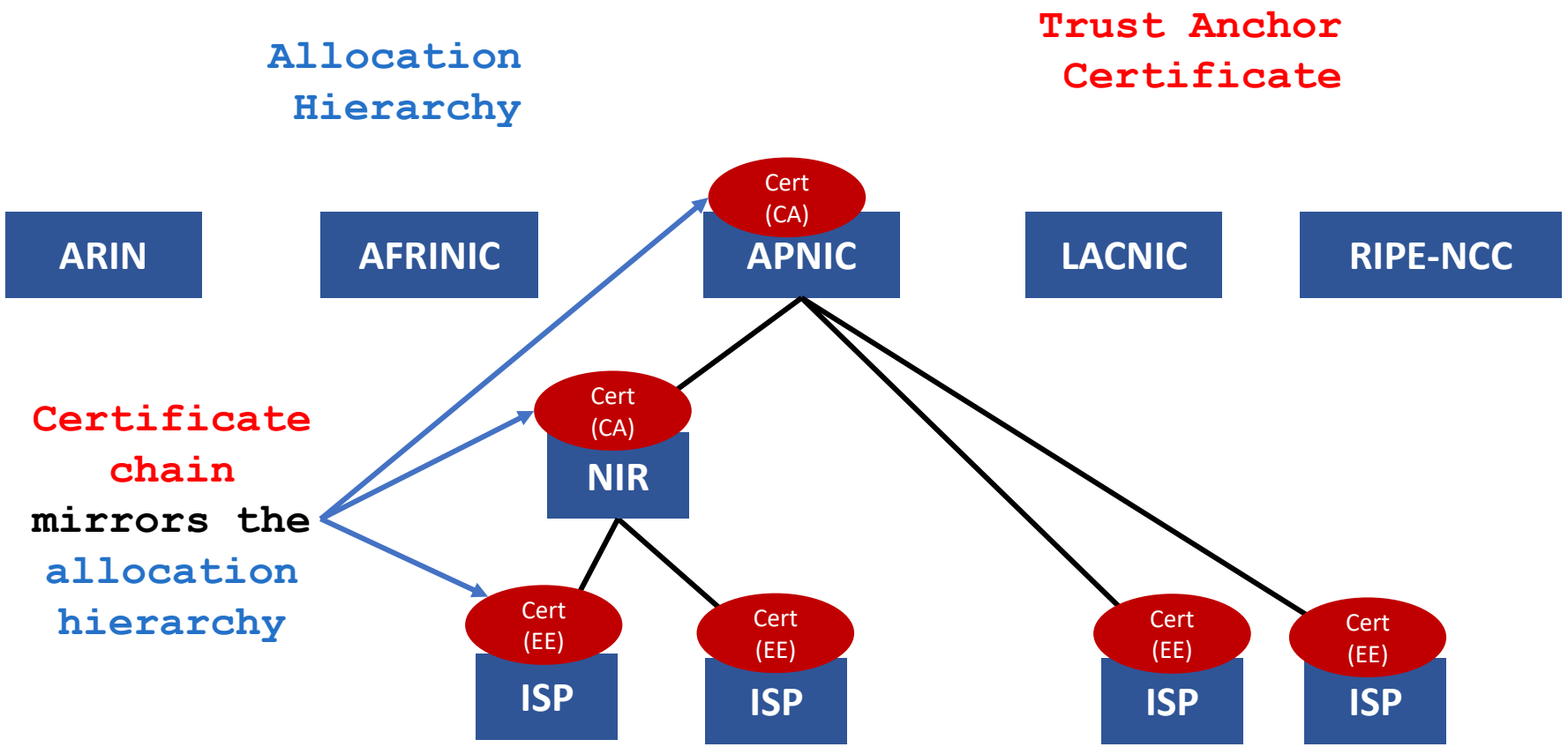
How about trust?

- How do we build a chain of trust in this framework??
  - Follow the resource allocation/delegation hierarchy

```
IANA → RIRs → NIRs/LIRs → End Holders
         |
         v
    End Holders
```

    - To describe the address allocation using digital certificates

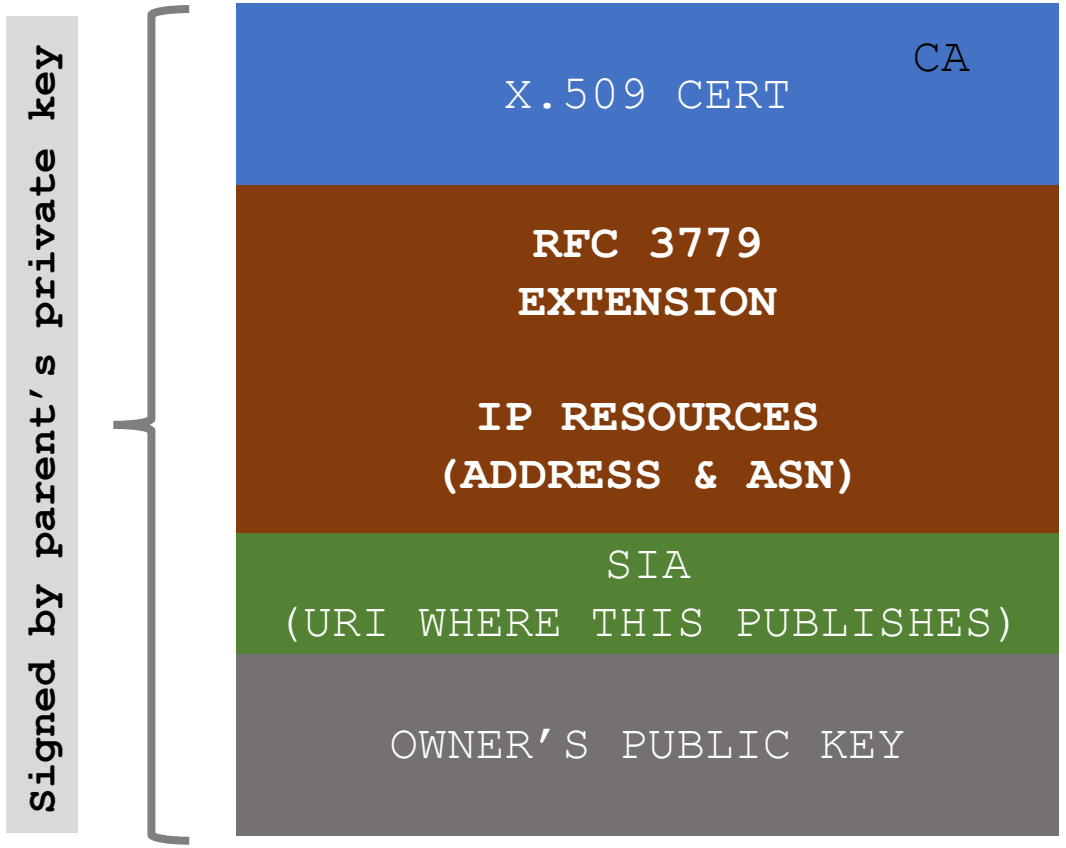
RPKI Chain of Trust



RPKI Chain of Trust

- RIRs hold a self-signed root certificate for all the resources they have in the registry
  - they are the *Trust Anchor* for the system
- The root certificate signs the resource certificates for end-holder allocations
  - binds the resources to the end-holders public key
- Any attestations signed by the end-holder's private key, can now be validated up the chain of trust

RPKI profile ~ Resource Certificates



- RFC 3779 extensions – binds a list of resources (**IPv4/v6,ASN**) to the subject of the certificate (private key holder)
- SIA (subject information access) contains a URI that identifies the publication point of the objects signed by the subject of the cert.

Resource Certificates

- When an address holder A (*IRs) allocates resources (IP address/ASN) to B (end holders)
  - ***A** issues a resource certificate that binds the allocated address with **B's** public key, all signed by **A's** (CA) private key*
  - *The resource certificate proves the holder of the private key (**B**) is the legitimate holder of the number resource!*

Route Origin Authorization (ROA)

- (B) can now sign *authorities* using its private key
  - which can be validated by any third party against the TA

- For routing, the address holder can *authorize* a network (ASN) to *originate* a route, and **sign** this permission with its private key (~ROA)

Route Origin Authorization (ROA)

- Digitally signed object
  - Binds list of prefixes and the nominated ASN
  - *can be verified cryptographically*

Prefix	203.176.32.0/19
Max-length	/24
Origin ASN	**AS17821**

- ** *Multiple ROAs can exist for the same prefix*

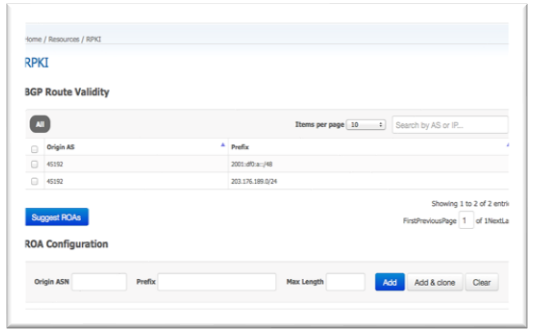
What can RPKI do?

- Authoritatively proof:
  - Who is the legitimate owner of an address, and
  - Identify which ASNs have the permission from the holder to originate the address

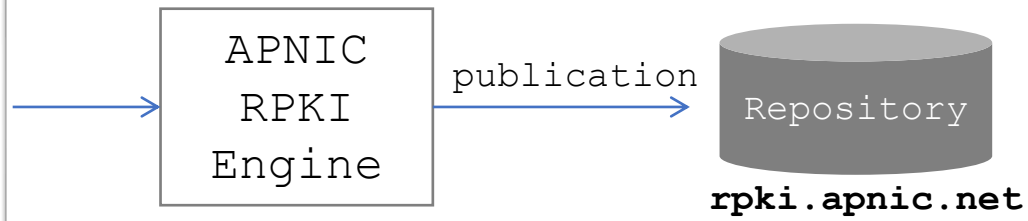
- Can help:
  - prevent **route hijacks/mis-origination/misconfiguration**

RPKI Components

- **Issuing Party** – Internet Registries (*IRs)
  - Certificate Authority (CA) that issues resource certificates to end-holders
  - Publishes the objects (ROAs) signed by the resource certificate holders



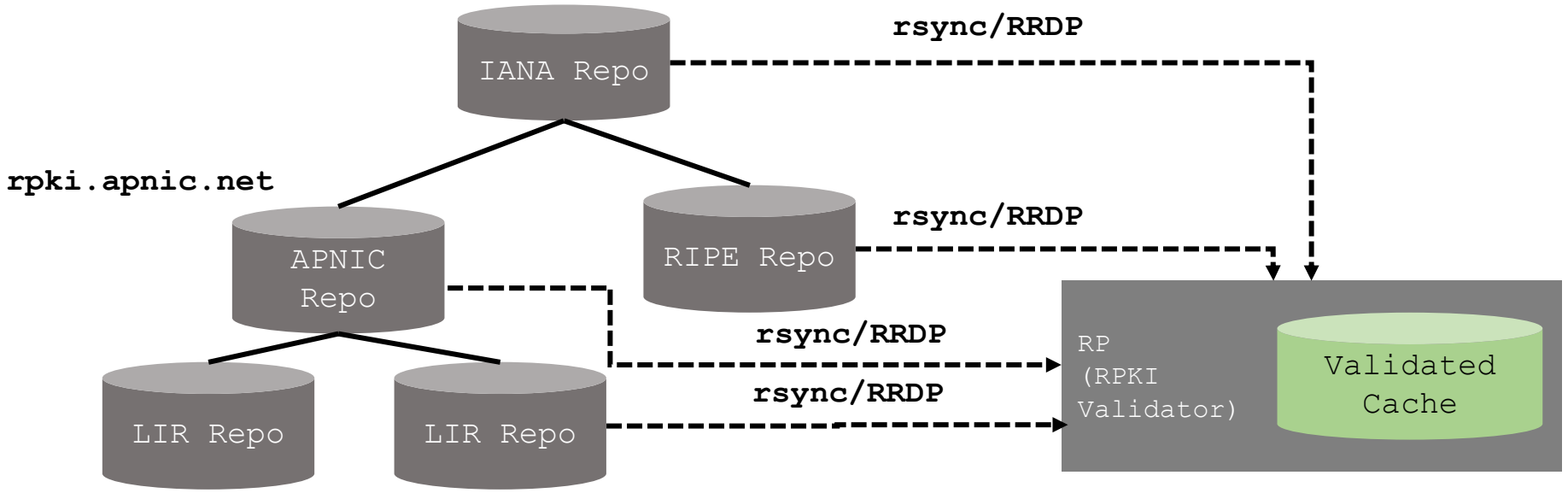
MyAPNIC GUI



rpki.apnic.net

RPKI Components

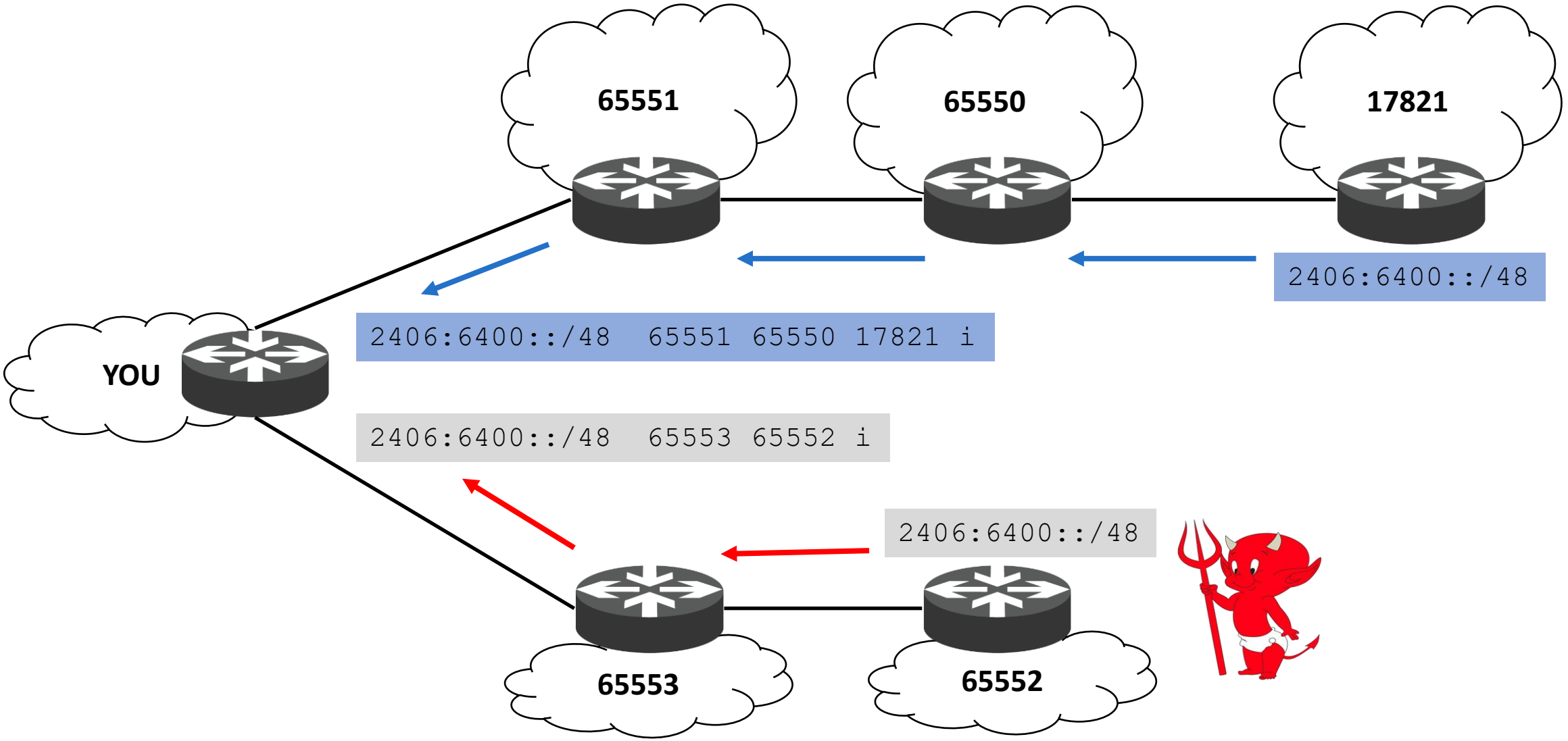
- **Relying Party (RP)**
  - RPKI Validator that gathers data (ROA) from the distributed RPKI repositories
  - Validates each entry's signature against the TA to build a "*Validated cache*"



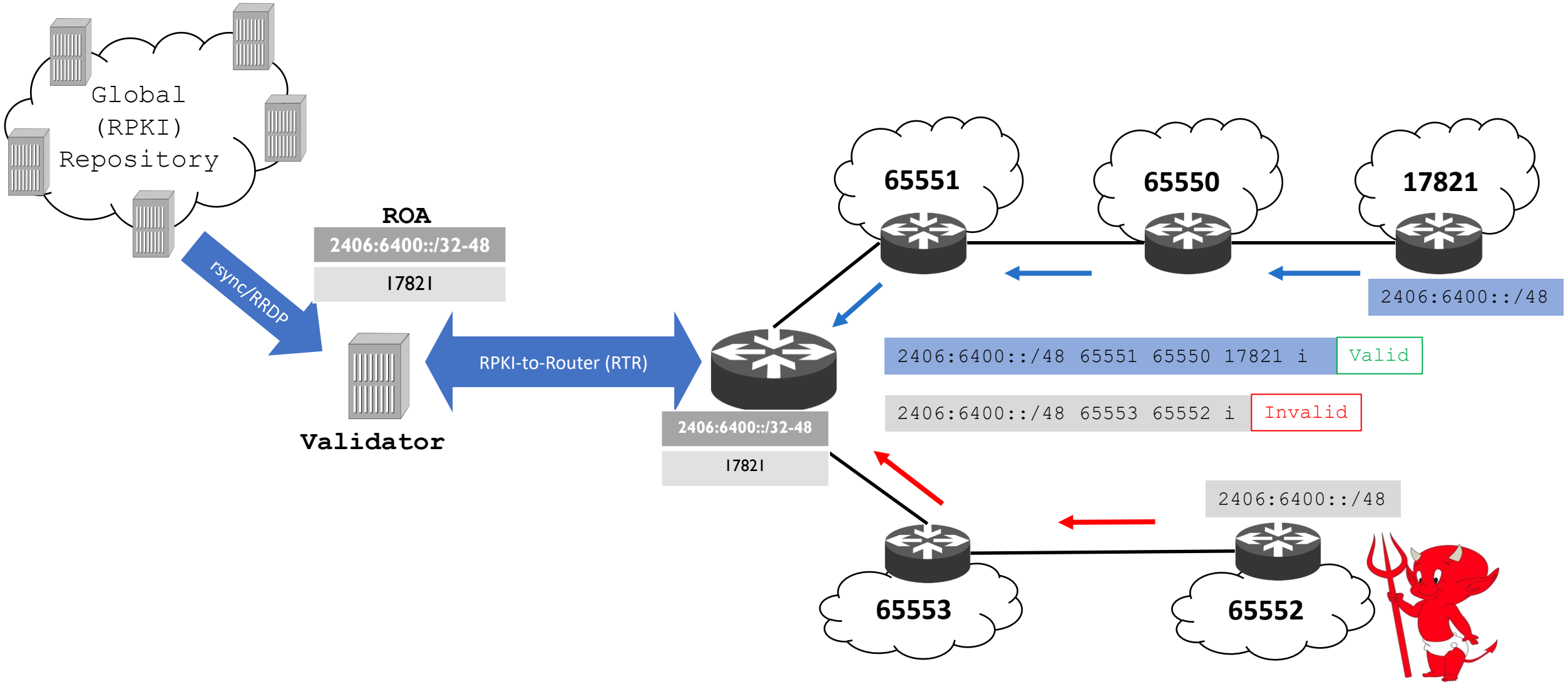
RPKI Service Models

- Hosted model:
  - The RIR (APNIC) runs the CA functions on members' behalf
    - Manage keys, repo, etc.
    - Generate certificates for resource delegations
- Delegated model:
  - Member becomes the CA (delegated by the parent CA) and operates the full RPKI system
    - JPNIC, TWNIC, CNNIC (IDNIC in progress)

Route Origin Validation (ROV)



Route Origin Validation (ROV)



Route Origin Validation

- Router fetches ROA information from the validated RPKI cache
  - *Crypto stripped by the validator*
- BGP checks each received BGP update against the ROA information and labels them

Validation States

- **Valid**
  - the prefix (prefix length) and AS pair found in the database.
- **Invalid**
  - prefix is found, but origin AS is wrong, OR
  - the prefix length is longer than the maximum length
- **Not Found/Unknown**
  - No valid ROA found
    - Neither valid nor invalid (perhaps not created)

Validation States

ROA

ASN	Prefix	Max Length
65420	10.0.0.0/16	18

BGP Routes

ASN	Prefix	RPKI State
65420	10.0.0.0/16	VALID
65420	10.0.128.0/17	VALID
65421	10.0.0.0/16	INVALID
65420	10.0.10.0/24	INVALID
65430	10.0.0.0/8	NOT FOUND

Acting on Validation states

- Tag
  - If you have downstream customers or run a route server (IXP)
  - Ex:

```
[Valid (ASN:65XX0), Not Found (ASN:65XX1), Invalid (ASN:65XX2)]
```

- Modify preference values – RFC7115

```
[Valid > Not Found > Invalid]
```

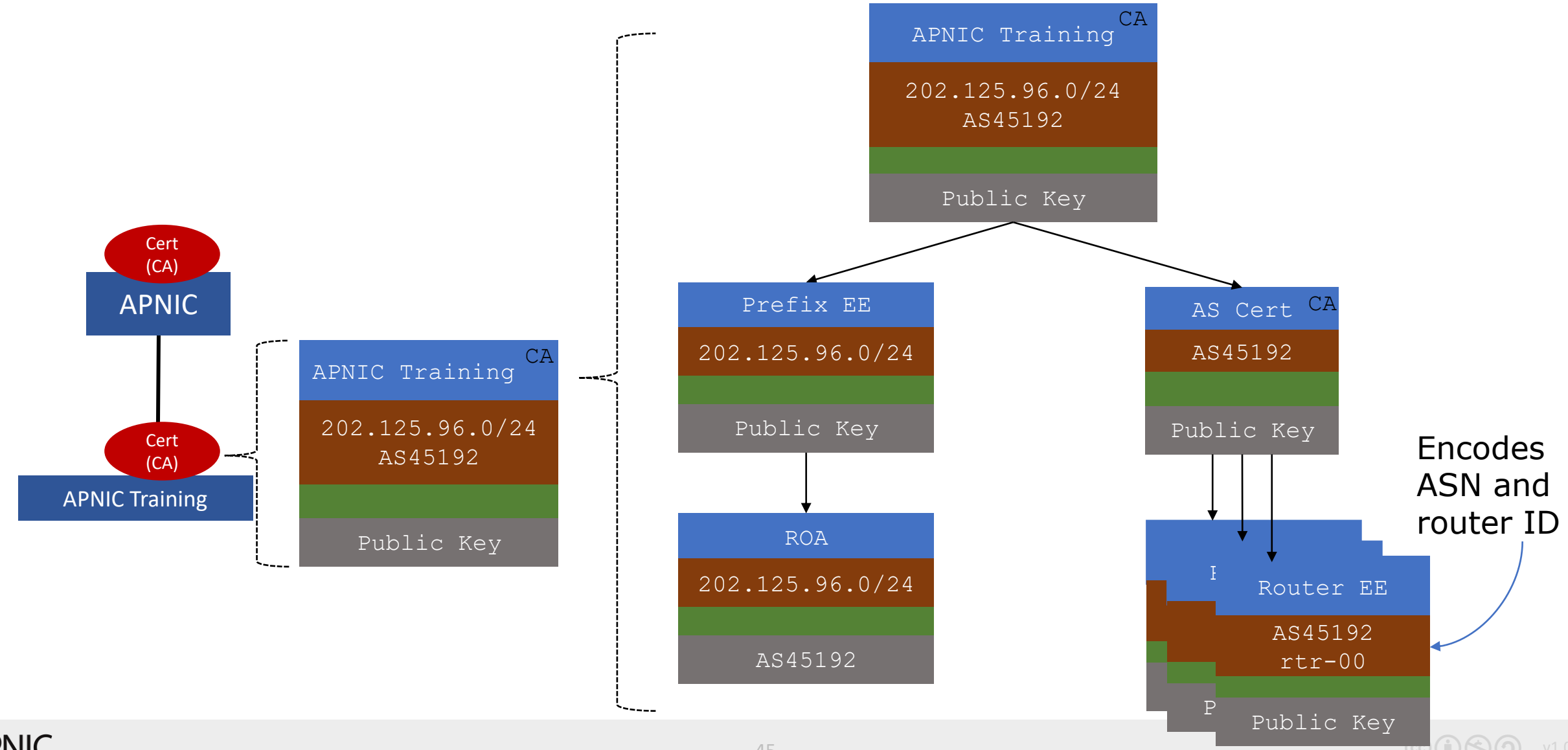
- Drop Invalids

```
IPv4 ~ 6K
IPv6 ~ 3K
```

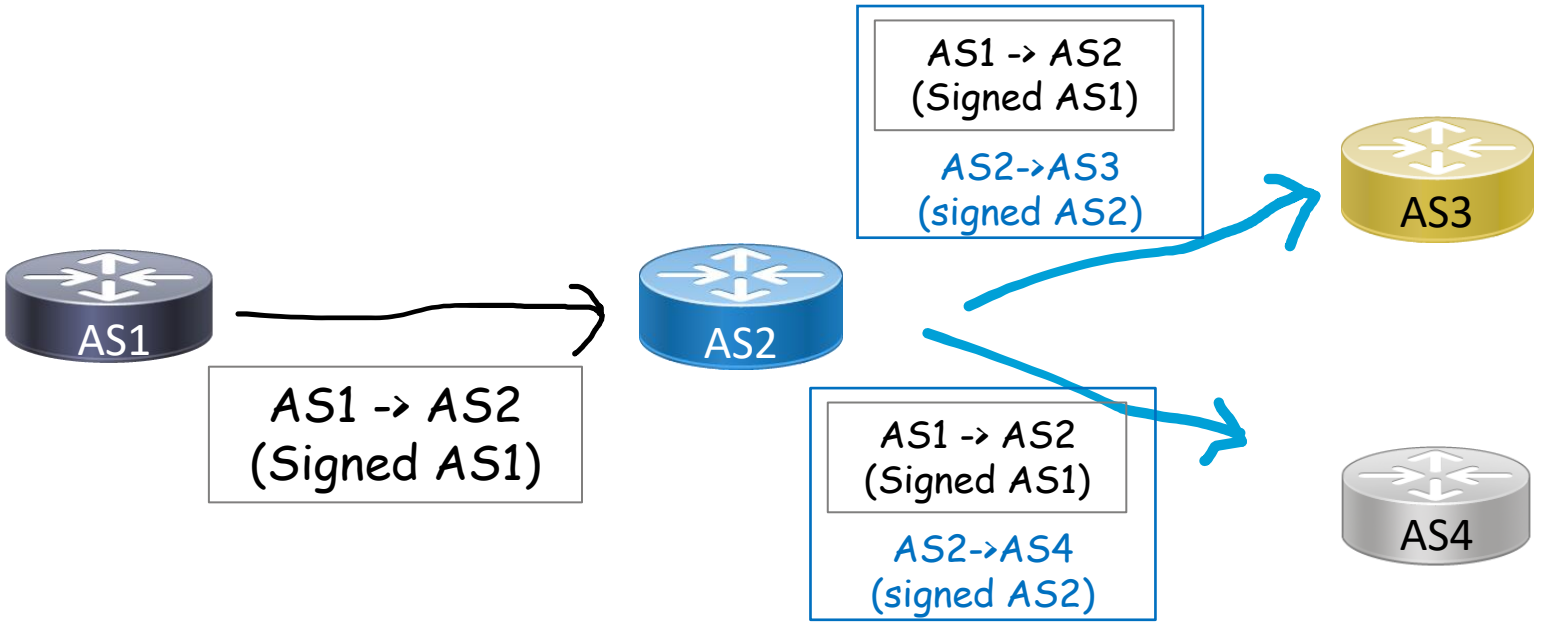
Are ROAs enough?

- What if I forge the origin AS in the AS path?
  - Would be accepted as good – pass origin validation!
- Which means, we need to secure the AS path as well
  - AS path validation (per-prefix)
- We can use RPKI certificates for this

AS keys (per-router keys)



BGPsec (RFC8205)



- AS1 router crypto signs the message to AS2
- AS2 router signs the message to AS3 and AS4, encapsulating AS1's message

- A BGPsec speaker validates the received update by checking:
  - If there is a ROA that describes the prefix and origin AS
  - If the received AS path can be validated as a chain of signatures (for each AS in the AS path) using the AS keys

So why NOT BGPsec?

- Cannot have partial adoption
  - Cannot jump across non-participating networks
- More HW resources
  - CPU - high crypto overhead to validate signatures, and
  - Memory
    - Updates in BGPsec would be per prefix
    - New attributes carrying signatures and certs/key IDs for every AS in the AS path
- No clarity on how to distribute the collection of certificates required to validate the signatures

ASPA - AS Provider Authorization

- …..draft but promising
- ASPA is digitally signed object that binds
  - a Set of Provider ASNs to a Customer ASN (for a specific AFI),
- For Routing, the ASPA is an attestation
  - that the AS holder (Customer ASN) has authorized the set of provider ASNs to propagate its announcements onwards….

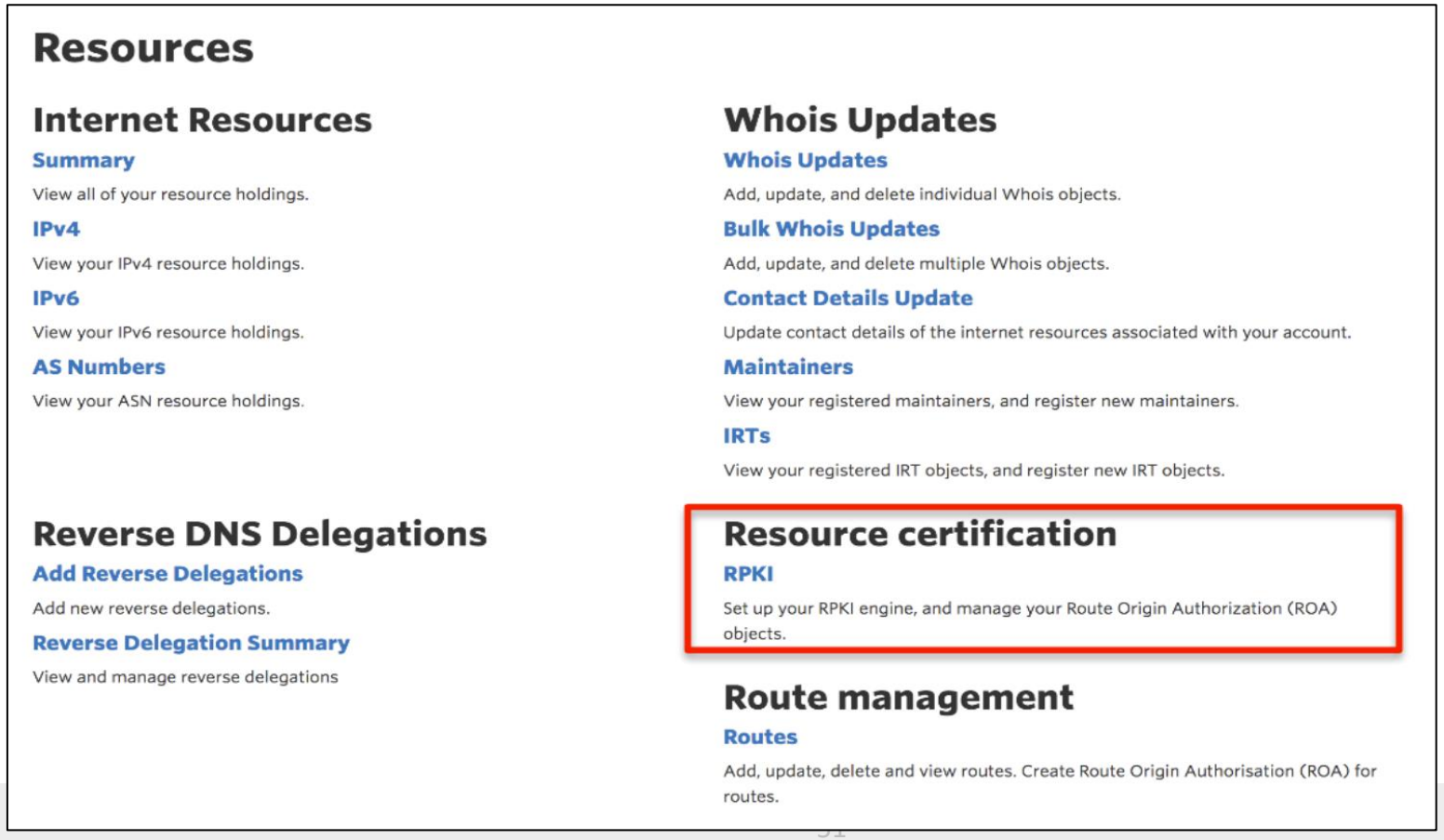
ASPA Validation/Verification ~ simplified

- For a received route (v4/v6):
  - If there is no valid ASPA for the Customer AS (AS(0)) UNKNOWN
  - If there is an ASPA with the customer AS, and:
    - if AS(I) in the AS_PATH attribute (AS_SEQ{AS(I), AS(I-1)}) is in the SPAS VALID
    - Else, INVALID

Implementation

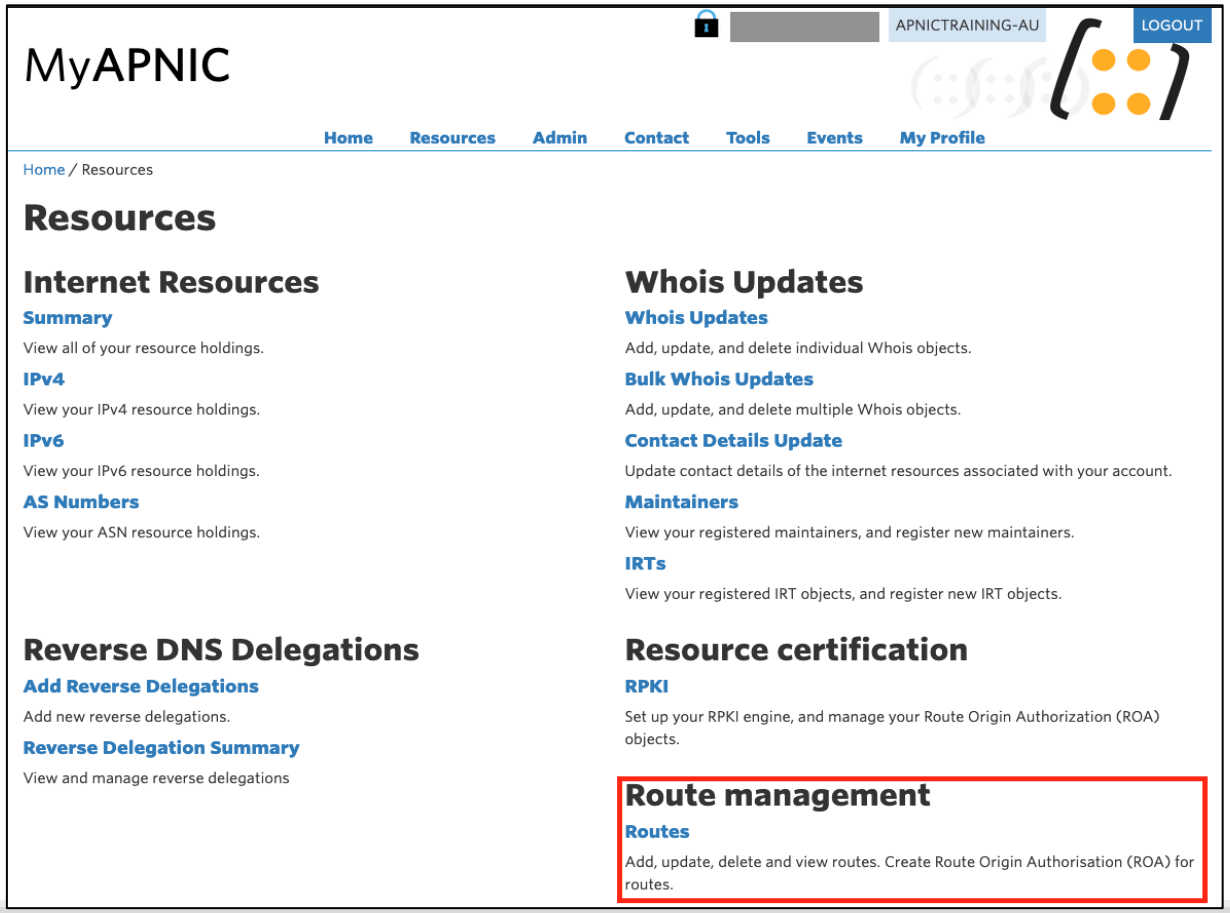
1. sign & publish your ROA

- Login MyAPNIC
  - Need to activate the RPKI engine to create ROAs
  - Go to **Resources → Resource certification → RPKI** (see image below)



Create & publish your ROA

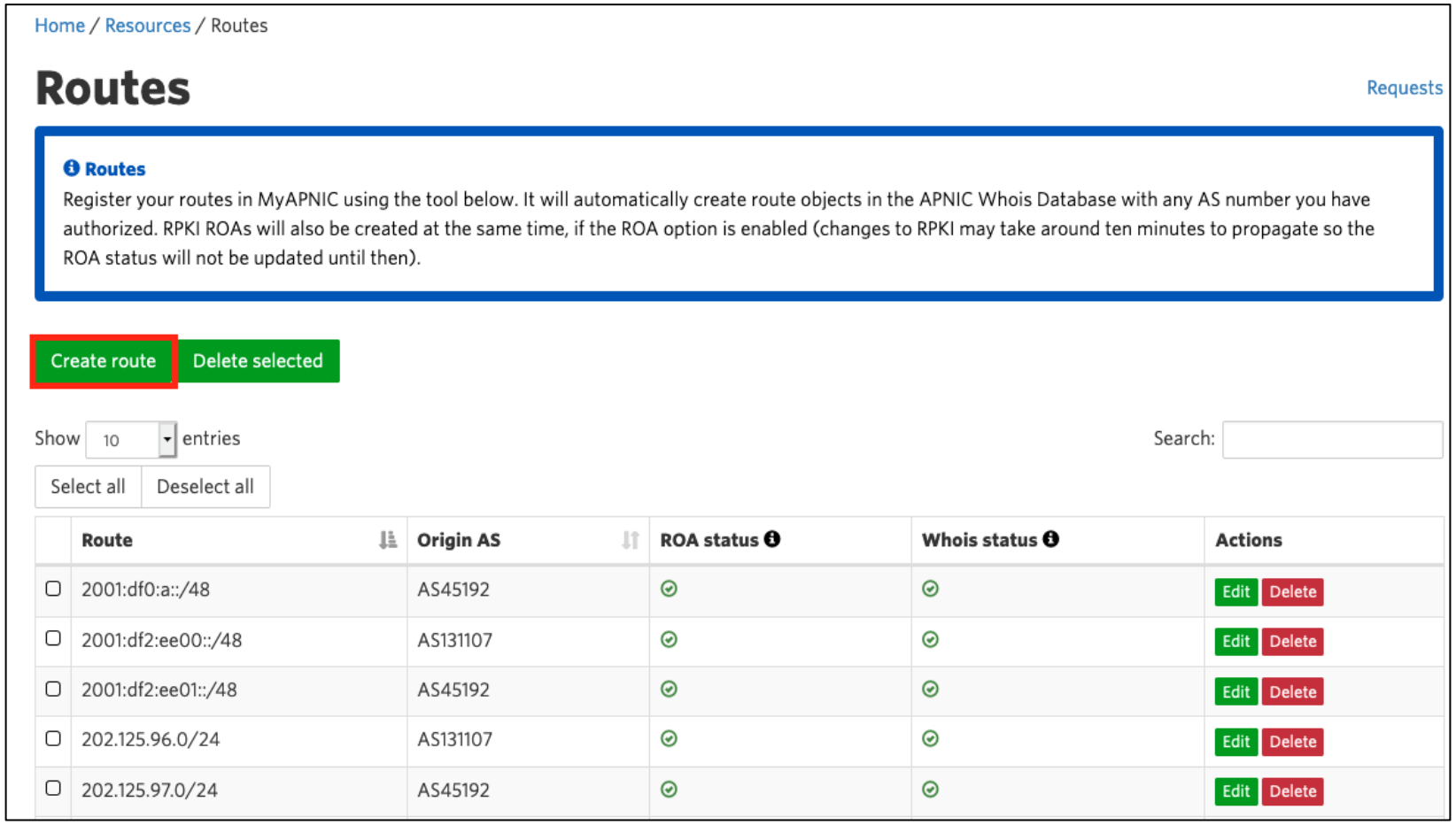
- Then go to the Routes page
  - Go to **Resources → Route Management → Routes** (see image below)



https://www.apnic.net/wp-content/uploads/2017/12/ROUTE_MANAGEMENT_GUIDE.pdf

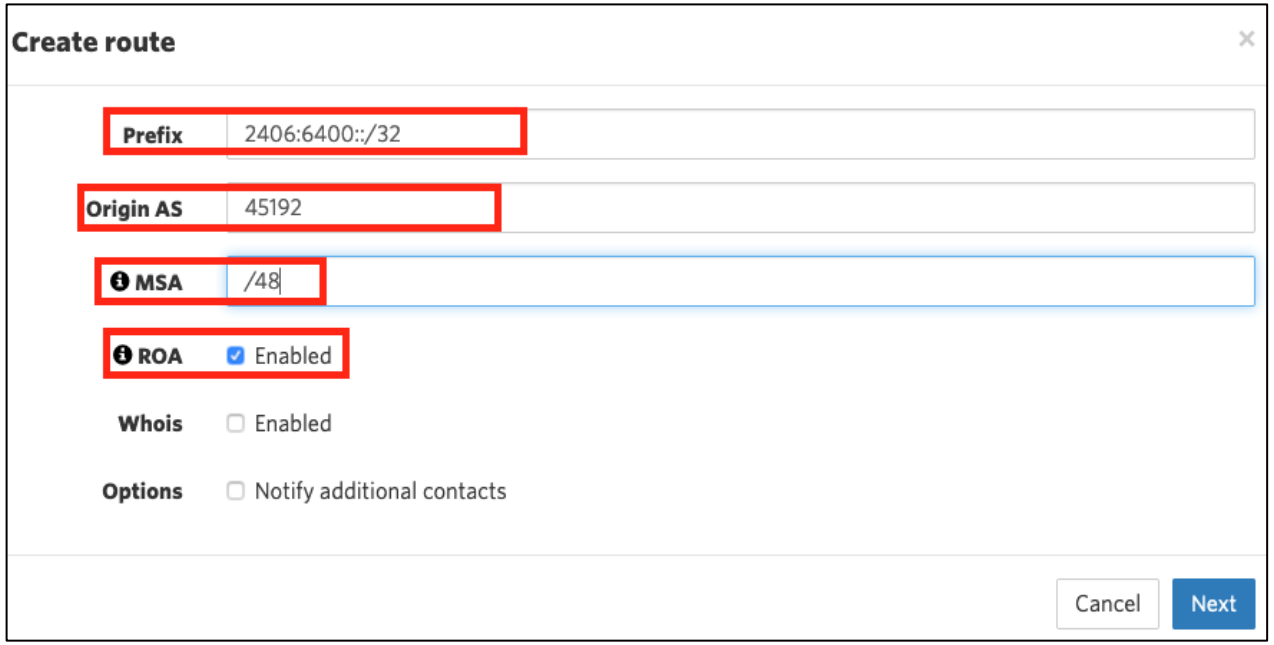
Create (publish) your ROA

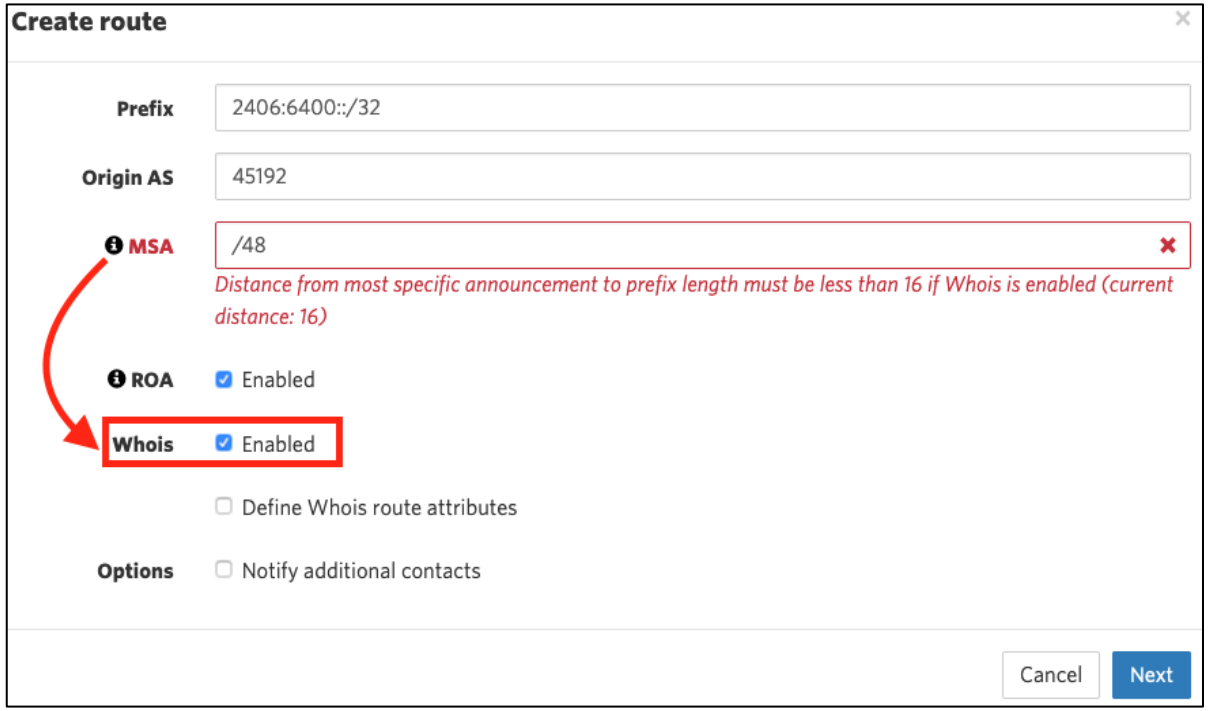
- Select **`Create route`** (as shown below)



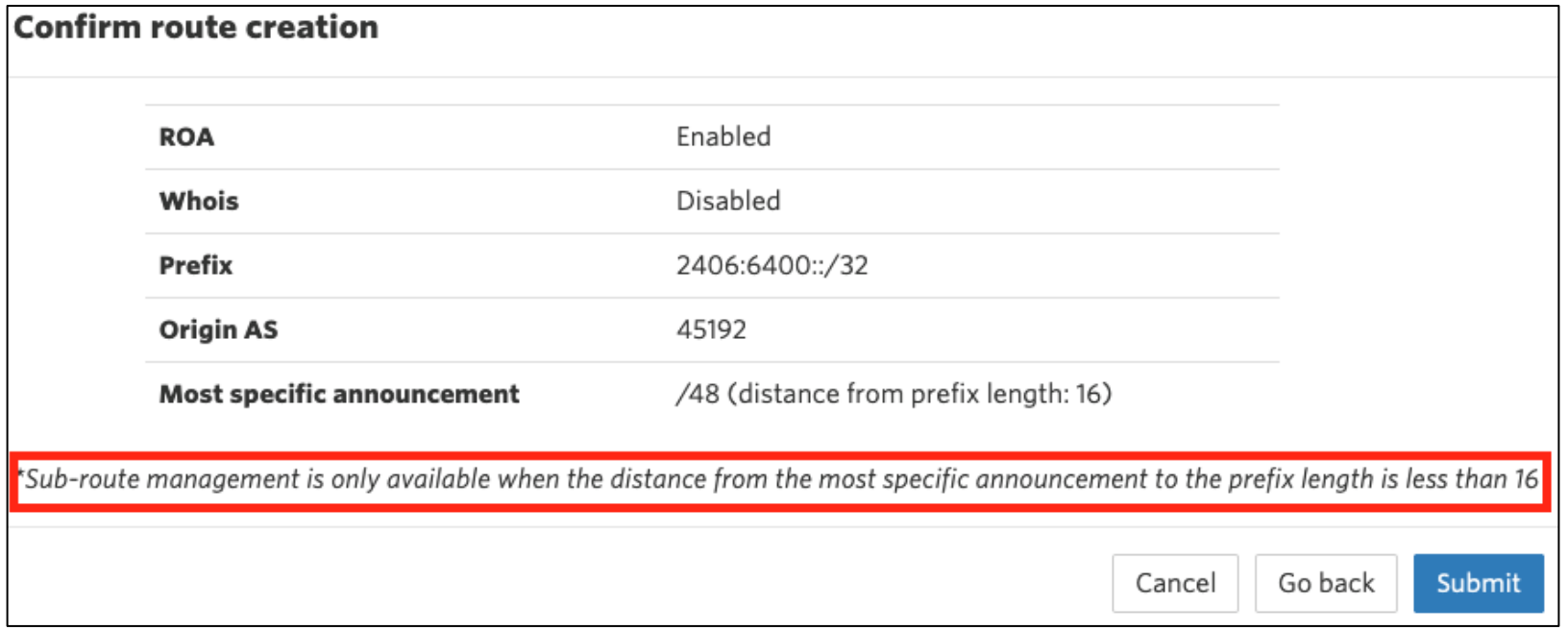
Create (publish) your ROA

- Example for **IPv6** below



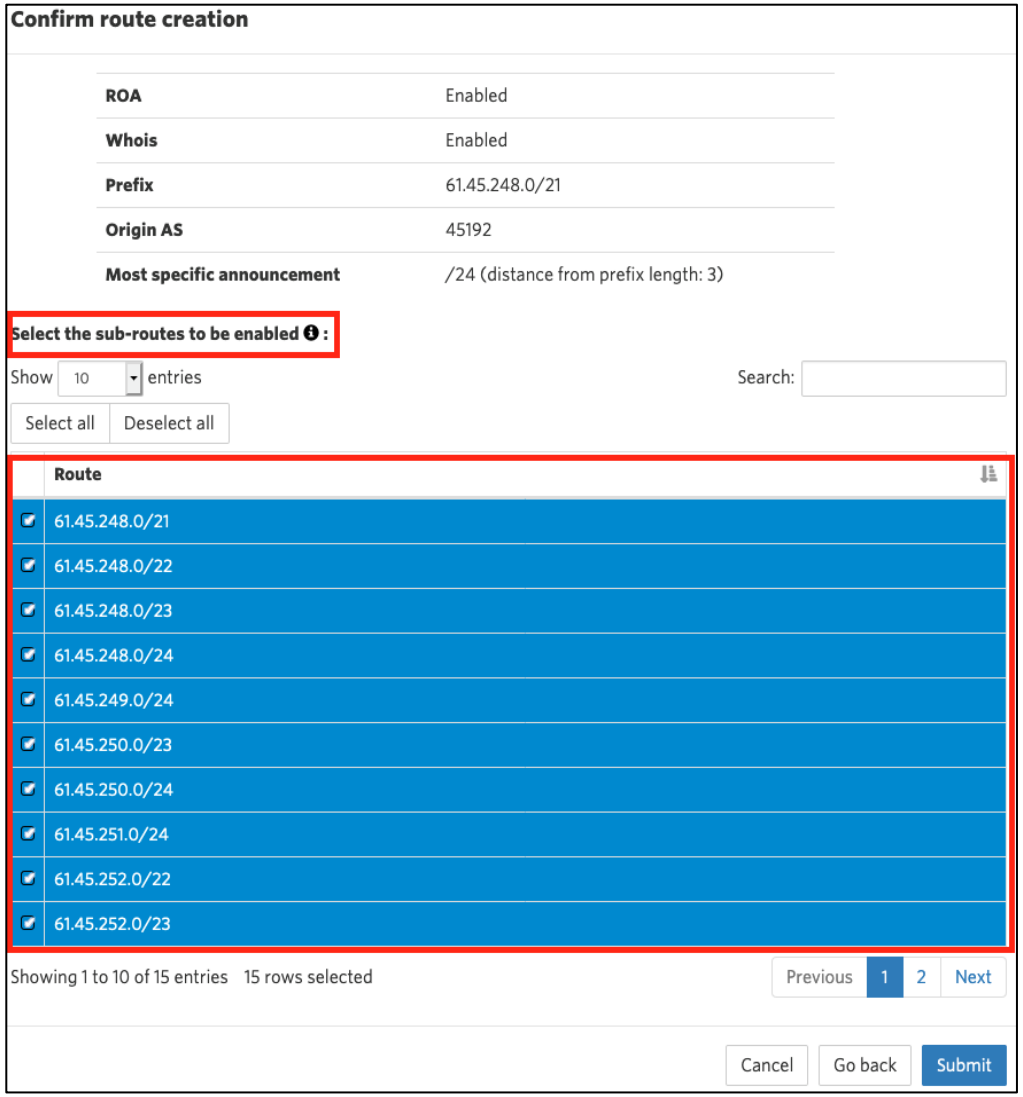


Create (publish) your ROA



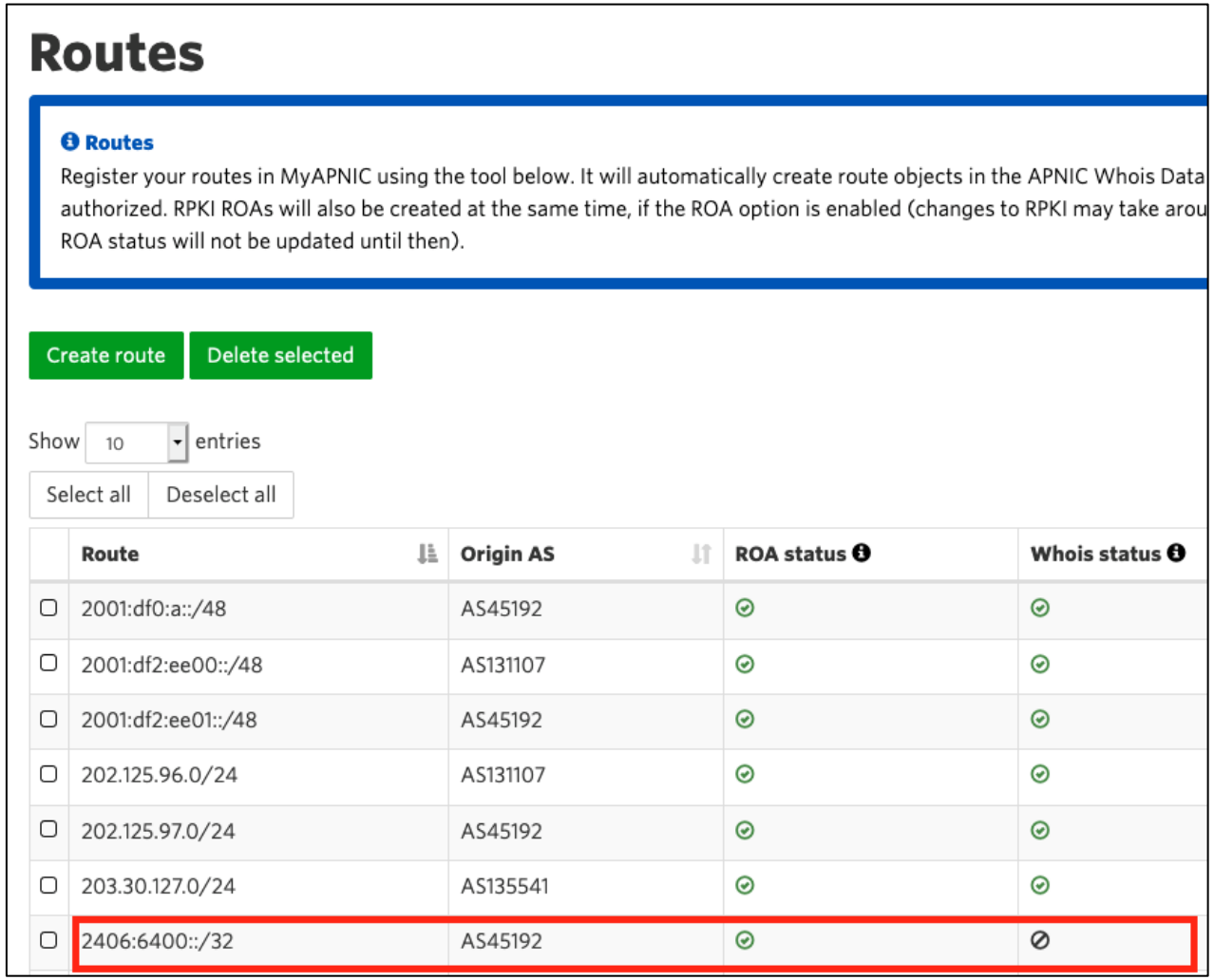
Create (publish) your ROA

- Example for **IPv4**



Create (publish) your ROA

- Your ROAs are ready!



Check your ROA

https://rpki-validator.ripe.net/roas

Validated ROAs

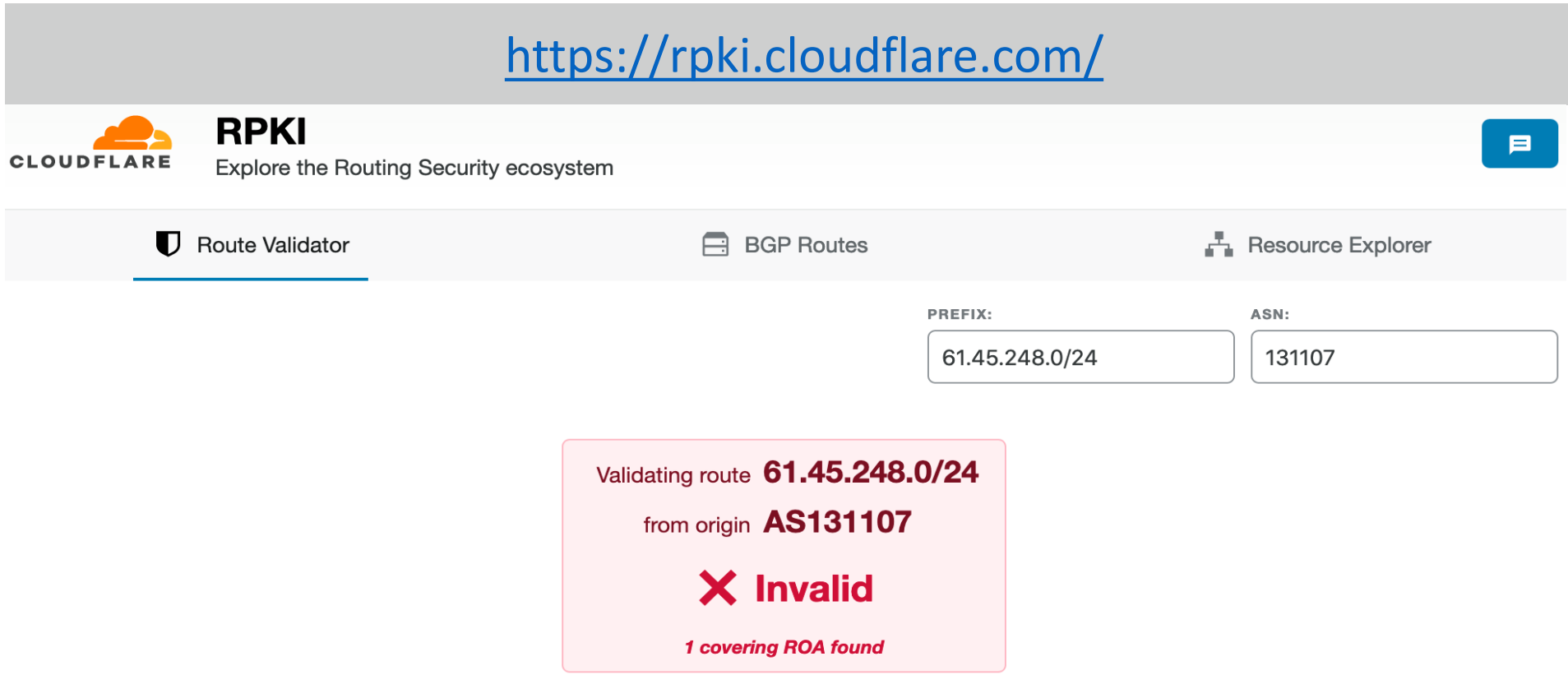
Show 10 entries

Search: 61.45.248.0

ASN	Prefix	Max Length	Trust Anchors	URI of ROA
135533	61.45.248.0/24	24	APNIC RPKI Root	

Check your ROA

https://rpki.cloudflare.com/



Check your ROA

https://bgp.he.net/

Announced By		
Origin AS	**Announcement**	**Description**
AS131107	2001:df2:ee00::/48 	testing

Check your ROA

```
# whois -h rr.ntt.net 2001:df2:ee00::/48
```

route6: 2001:df2:ee00::/48
descr: RPKI ROA for 2001:df2:ee00::/48
remarks: This route object represents routing data retrieved from the RPKI
remarks: The original data can be found here: https://rpki.gin.ntt.net/r/AS131107/2001:df2:ee00::/48
remarks: This route object is the result of an automated RPKI-to-IRR conversion process.
remarks: maxLength 48
origin: AS131107
mnt-by: MAINT-JOB
changed: job@ntt.net 20180802
source: RPKI # Trust Anchor: APNIC RPKI Root

Check your ROA

```
# whois -h whois.bgpmon.net 2001:df2:ee00::/48

Prefix:                 2001:df2:ee00::/48
Prefix description:     APNICTRAINING-DC
Country code:           AU
Origin AS:                      131107
Origin AS Name:         APNICTRAINING LAB DC
RPKI status:            ROA validation successful
First seen:             2016-06-30
Last seen:              2018-01-21
Seen by #peers:         97
```

```
# whois -h whois.bgpmon.net " --roa 131107 2001:df2:ee00::/48"

------------------------
ROA Details
------------------------
Origin ASN:       AS131107
Not valid Before: 2016-09-07 02:10:04
Not valid After:  2020-07-30 00:00:00  Expires in 2y190d9h34m23.2000000029802s
Trust Anchor:     rpki.apnic.net
Prefixes:         2001:df2:ee00::/48 (max length /48) 202.125.96.0/24 (max length /24)
```

ROA Considerations

- Max-length
  - Make sure the value covers your BGP announcements

- minimal ROAs
  - Reduce spoofed origin-AS attack surface
    - https://tools.ietf.org/html/draft-ietf-sidrops-rpkimaxlen-03
    - ROAs should cover only those prefixes announced in BGP

VALIDATION

ROV

Run (your own) RPKI Validator

- Lots of options:
  - Dragon Research RPKI toolkit - https://github.com/dragonresearch/rpki.net
  - RIPE Validator - https://github.com/RIPE-NCC/rpki-validator-3
  - Routinator - https://github.com/NLnetLabs/routinator/releases/tag/v0.7.1
  - OctoRPKI/GoRTR (Cloudflare's toolkit) - https://github.com/cloudflare/cfrpki
  - Fort (NIC Mexico's Validator) - https://nicmx.github.io/FORT-validator/

Validator considerations

- Securing the RTR session
  - Plain text (TCP)
    - run within your routing domain
  - Other auth options
    - SSH (v2)
    - MD5 auth
    - IPsec
    - TLS
    - TCP-AO

Validator considerations

- When RTR session fails
  - Based on the expire interval of ROA cache
    - JunOS/SR-OS: 3600s, IOS-XE: 300s (RFC min ~ 600s)
  - Defaults to NOT FOUND
    - Including Invalids
  - Hence, at least 2 x Validators (RTR sessions)

3. Router Configuration (IOS)

- Enable RTR on your routers
  - eBGP speakers (border/peering/transit)

- Know your platform defaults and knobs
  - Example: IOS-XE wont use Invalids for best path selection

```
router bgp 131107
 rpki server <validatorIP>
  transport tcp port <323/3323/8282>
  refresh-time <secs>
```

```
router bgp 131107
 bgp rpki server tcp <validatorIP> port <323/8282/3323> refresh <secs>
```

Configuration (IOS)

- Policies based on validation:

```
route-map ROUTE-VALIDATION permit 10
 match rpki valid
 set local-preference 200
!
route-map ROUTE-VALIDATION permit 20
 match rpki not-found
 set local-preference 100
!
route-map ROUTE-VALIDATION permit 30
 match rpki invalid
 set local-preference 50
!
```

OR

```
route-map ROUTE-VALIDATION deny 30
 match rpki invalid
```

Configuration (IOS)

- Apply the route-map to inbound updates

```
router bgp 131107
!---output omitted-----!
 address-family ipv4
  bgp bestpath prefix-validate allow-invalid
  neighbor X.X.X.169 activate
  neighbor X.X.X.169 route-map ROUTE-VALIDATION in
 exit-address-family
 !
 address-family ipv6
  bgp bestpath prefix-validate allow-invalid
  neighbor X6:X6:X6:X6::151 activate
  neighbor X6:X6:X6:X6::151 route-map ROUTE-VALIDATION in
 exit-address-family
!
```

Router Configuration (JunOS)

- Establishing session with the validator

```
routing-options {
   autonomous-system 131107;
   validation {
       group rpki-validator {
           session <validator-IP> {
               refresh-time 120;
               port <323/3323/8282>;
               local-address X.X.X.253;
           }
       }
   }
}
```

Configuration (JunOS)

- Define policies based on the validation states

```
policy-options {
   policy-statement ROUTE-VALIDATION {
       term valid {
           from {
               protocol bgp;
               validation-database valid;
           }
           then {
               local-preference 200;
               validation-state valid;
               accept;
           }
       }
       term unknown {
           from {
               protocol bgp;
               validation-database unknown;
           }
           then {
               local-preference 100;
               validation-state unknown;
               accept;
           }
       }
```

```
term invalid {
            from {
                protocol bgp;
                validation-database invalid;
            }
            then {
                local-preference 50;
                validation-state invalid;
                accept;
            }
        }
    }
}


OR
            then {
                validation-state invalid;
                reject;
             }
```

Router Configuration (JunOS)

- Apply the policy to inbound updates

```
protocols {
    bgp {
        group external-peers {
            #output-ommitted
            neighbor X.X.X.1 {
                import ROUTE-VALIDATION;
                family inet {
                    unicast;
                }
            }
        }
        group external-peers-v6 {
            #output-ommitted
            neighbor X6:X6:X6:X6::1 {
                import ROUTE-VALIDATION;
                family inet6 {
                    unicast;
                }
            }
        }
    }
```

RPKI Verification (IOS)

- IOS has only

```
#sh bgp ipv6 unicast rpki ?
  servers Display RPKI cache server information
  table   Display RPKI table entries


#sh bgp ipv4 unicast rpki ?
  servers Display RPKI cache server information
  table   Display RPKI table entries
```

RPKI Verification (IOS)

- Check the RTR session

```
#sh bgp ipv4 unicast rpki servers

BGP SOVC neighbor is X.X.X.47/323 connected to port 323
Flags 64, Refresh time is 120, Serial number is 1516477445, Session ID is 8871
InQ has 0 messages, OutQ has 0 messages, formatted msg 7826
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
 Prefixes 45661
 Connection attempts: 1
 Connection failures: 0
 Errors sent: 0
 Errors received: 0

Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: X.X.X.225, Local port: 29831
Foreign host: X.X.X.47, Foreign port: 323
```

RPKI Verification (IOS)

- Check the RPKI cache

```
#sh bgp ipv4 unicast rpki table
37868 BGP sovc network entries using 6058880 bytes of memory
39655 BGP sovc record entries using 1268960 bytes of memory

Network            Maxlen Origin-AS Source Neighbor
1.9.0.0/16         24     4788      0      202.125.96.47/323
1.9.12.0/24        24     65037     0      202.125.96.47/323
1.9.21.0/24        24     24514     0      202.125.96.47/323
1.9.23.0/24        24     65120     0      202.125.96.47/323
```

```
#sh bgp ipv6 unicast rpki table
5309 BGP sovc network entries using 976856 bytes of memory
6006 BGP sovc record entries using 192192 bytes of memory

Network             Maxlen Origin-AS Source Neighbor
2001:200::/32       32     2500      0      202.125.96.47/323
2001:200:136::/48   48     9367      0      202.125.96.47/323
2001:200:900::/40   40     7660      0      202.125.96.47/323
2001:200:8000::/35  35     4690      0      202.125.96.47/323
```

Check routes (IOS)

```
#sh bgp ipv4 unicast 202.144.128.0/19
BGP routing table entry for 202.144.128.0/19, version 3814371
Paths: (1 available, best #1, table default)
 Advertised to update-groups:
    2
 Refresh Epoch 15
 4826 17660
   49.255.232.169 from 49.255.232.169 (114.31.194.12)
     Origin IGP, metric 0, localpref 110, valid, external, best
     Community: 4826:5101 4826:6570 4826:51011 24115:17660
     path 7F50C7CD98C8 RPKI State valid
     rx pathid: 0, tx pathid: 0x0
```

```
#sh bgp ipv6 unicast 2402:7800::/32
BGP routing table entry for 2402:7800::/32, version 1157916
Paths: (1 available, best #1, table default)
 Advertised to update-groups:
    2
 Refresh Epoch 15
 4826
   2402:7800:10:2::151 from 2402:7800:10:2::151 (114.31.194.12)
     Origin IGP, metric 0, localpref 100, valid, external, best
     Community: 4826:1000 4826:2050 4826:2110 4826:2540 4826:2900 4826:5203
     path 7F50B266CBD8 RPKI State not found
     rx pathid: 0, tx pathid: 0x0
```

RPKI Verification (JunOS)

- Check the RPKI cache

```
>show validation session
Session                                 State Flaps   Uptime #IPv4/IPv6 records
X.X.X.46                                  Up      75 09:20:59 40894/6747


>show validation session 202.125.96.46
Session                                 State Flaps   Uptime #IPv4/IPv6 records
X.X.X.46                                  Up       75 09:21:18 40894/6747
```

RPKI Verification (JunOS)

- Check the RPKI cache

```
>show validation database
RV database for instance master

Prefix                Origin-AS       Session                   State  Mismatch
1.9.0.0/16-24              4788 202.125.96.46                valid
1.9.12.0/24-24            65037 202.125.96.46                valid
1.9.21.0/24-24            24514 202.125.96.46                valid
1.9.23.0/24-24            65120 202.125.96.46                valid

----------
2001:200::/32-32           2500 202.125.96.46                valid
2001:200:136::/48-48       9367 202.125.96.46                valid
2001:200:900::/40-40       7660 202.125.96.46                valid
2001:200:8000::/35-35      4690 202.125.96.46                valid
2001:200:c000::/35-35     23634 202.125.96.46                valid
2001:200:e000::/35-35      7660 202.125.96.46                valid
```

Would have been nice if per AF!

RPKI Verification (JunOS)

- Can filter per origin ASN

```
>show validation database origin-autonomous-system 45192
RV database for instance master


Prefix                Origin-AS       Session                  State        Mismatch
202.125.97.0/24-24       45192        202.125.96.46             valid
203.176.189.0/24-24      45192        202.125.96.46             valid
2001:df2:ee01::/48-48    45192        202.125.96.46             valid


 IPv4 records: 2
 IPv6 records: 1
```

Check routes (JunOS)

```
>show route protocol bgp 202.144.128.0

inet.0: 693024 destinations, 693024 routes (693022 active, 0 holddown, 2
hidden)
+ = Active Route, - = Last Active, * = Both

202.144.128.0/20 *[BGP/170] 1w4d 21:03:04, MED 0, localpref 110, from
202.125.96.254
                      AS path: 4826 17660 I, validation-state: valid
                    >to 202.125.96.225 via ge-1/1/0.0
```

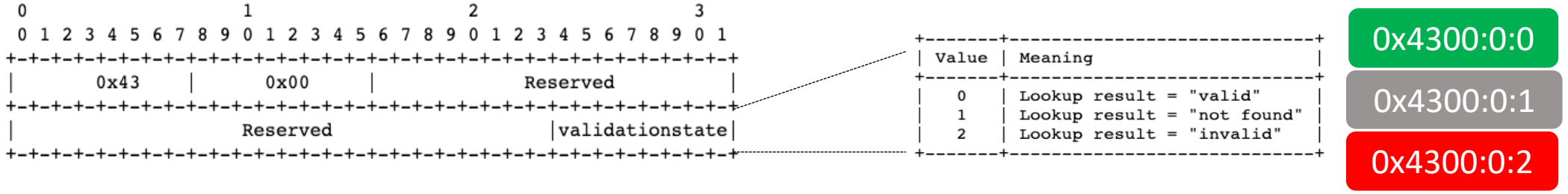
```
>show route protocol bgp 2001:201::/32

inet6.0: 93909 destinations, 93910 routes (93909 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

2001:201::/32      *[BGP/170] 21:18:14, MED 0, localpref 100, from
2001:df2:ee00::1
                      AS path: 65332 I, validation-state: unknown
                    >to fe80::dab1:90ff:fedc:fd07 via ge-1/1/0.0
```

Propagating RPKI states to iBGP peers

- To avoid every BGP speaker having an RTR session, and
- Ensure all BGP speakers have consistent information
  - Relies on non-transitive extended BGP community (RFC8097)



  - Sender (one with RTR session) attaches the extended community to Updates, and receiver derives the validation states from it
  - Must be enabled on both sender and receiver!

Propagating RPKI states (IOS)

- Sender (one with RTR session)

```
router bgp 131107
 bgp rpki server tcp <validator-IP> port <323/8282/3323> refresh 120
!---output omitted-----!
 address-family ipv4
  neighbor X.X.X.X activate
  neighbor X.X.X.X send-community both
  neighbor X.X.X.X announce rpki state
 exit-address-family
 !
 address-family ipv6
  neighbor X6:X6:X6:X6::X6 activate
  neighbor X6:X6:X6:X6::X6 send-community both
  neighbor X6:X6:X6:X6::X6 announce rpki state
 exit-address-family
!
```

Propagating RPKI states (IOS)

- Receiver (iBGP peer)

```
router bgp 131107
!---output omitted-----!
 address-family ipv4
  neighbor Y.Y.Y.Y activate
  neighbor Y.Y.Y.Y send-community both
  neighbor Y.Y.Y.Y announce rpki state
 exit-address-family
 !
 address-family ipv6
  neighbor Y6:Y6:Y6:Y6::Y6 activate
  neighbor Y6:Y6:Y6:Y6::Y6 send-community both
  neighbor Y6:Y6:Y6:Y6::Y6 announce rpki state
 exit-address-family
!
```

  - If `announce rpki state` is not configured for the neighbor, all prefixes received from the iBGP neighbor will be marked VALID!

Propagating RPKI states (JunOS)

- Sender (router with an RTR session)

```
policy-statement ROUTE-VALIDATION {
    term valid {
        from {
            protocol bgp;
            validation-database valid;
        }
        then {
            local-preference 200;
            validation-state valid;
            community add origin-validation-state-valid;
            accept;
        }
    }
    term invalid {
        from {
            protocol bgp;
            validation-database invalid;
        }
        then {
            local-preference 50;
            validation-state invalid;
            community add origin-validation-state-invalid;
            accept;
        }
    }
```

```
term unknown {
            from {
                protocol bgp;
                validation-database unknown;
            }
            then {
                local-preference 100;
                validation-state unknown;
                community add origin-validation-state-unknown;
                accept;
            }
        }
    }
}
```

Propagating RPKI states (JunOS)

- Receiver (iBGP peer)

```
policy-statement ROUTE-VALIDATION-1 {
    term valid {
        from community origin-validation-state-valid;
        then validation-state valid;
    }
    term invalid {
        from community origin-validation-state-invalid;
        then validation-state invalid;
    }
    term unknown {
        from community origin-validation-state-unknown;
        then validation-state unknown;
    }

}
```

Operational Considerations

- Default routes?
  - will match anything - Invalids

Operational Considerations

- iBGP State Propagation ~ Multivendor
  - Ex - IOS propagating states to JunOS iBGP peers
    `unknown iana 4300`
  - Options(*<JunOS 17.4R3, 18.2R3, 18.4R2*):
    - Either act on the states at the border, or
    - Match and tag them with custom communities before propagating

Other developments

- ROA with AS-0 origin (RFC6483/RFC7607)
  - Reserved by IANA for non-routed networks
  - Negative attestation: no valid ASN has been granted authority
    - Not to be routed (Ex: IXP LAN prefixes)
    - Overridden by another ROA (with an origin-AS other than AS-0)
    - APNIC TA ~ Nov 2018

Other developments

- Prop-132:
  - AS-0 ROA for unallocated/unassigned APNIC space
  - Similar to RFC6491 ~ for special use, reserved, unallocated IANA space
  - APNIC implementation done on 2 Sept 2020
    - Covers APNIC's undelegated IPv4 and IPv6 space
    - separate TAL

    https://blog.apnic.net/2020/09/02/policy-prop-132-as0-for-unallocated-space-deployed-in-service/

https://www.apnic.net/community/security/resource-certification/#routing

Any questions?

MANRS

- https://roa-stats.manrs.org/
- https://www.manrs.org/

Check Ups

- https://rpki.cloudflare.com/
- https://bgp.he.net/
- https://rpki-monitor.antd.nist.gov/

MikroTik

- https://help.mikrotik.com/docs/display/ROS/RPKI

Stay in Touch!

blog.apnic.net

apnic.net/social