APNIC

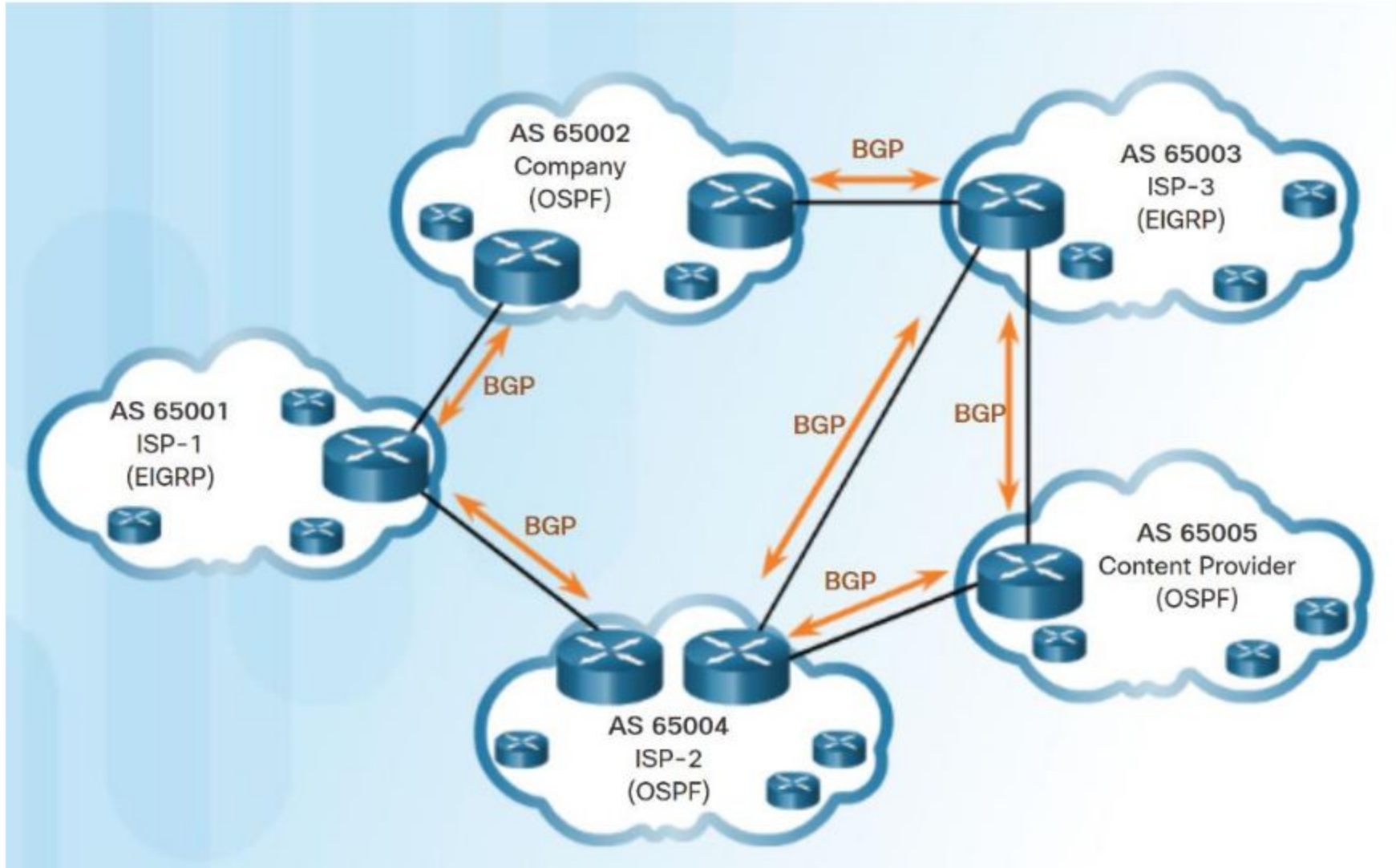
Securing Internet Routing (with RPKI)

Agenda

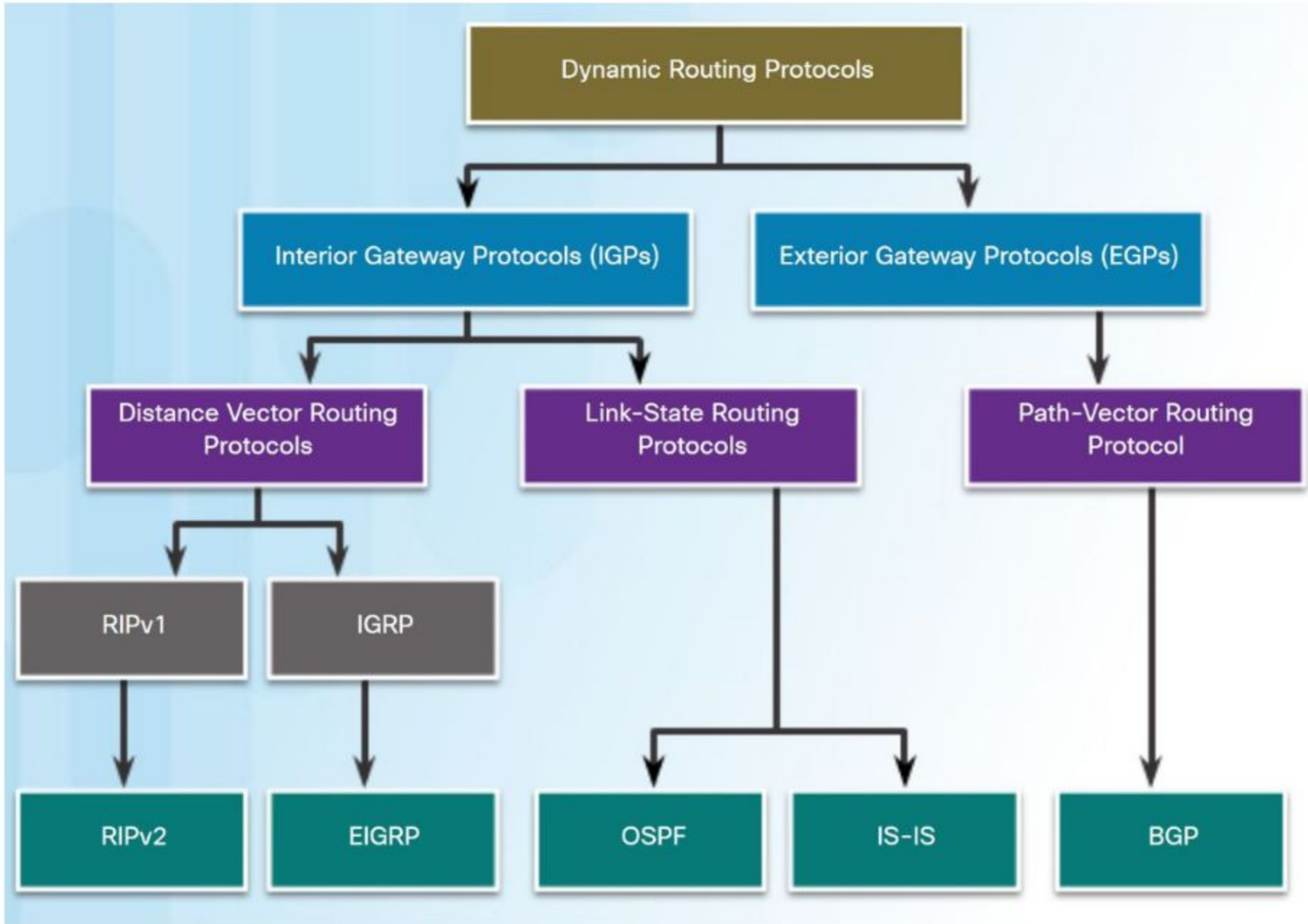
An introduction to the following:

- Internet Routing System
- Border Gateway Protocol(BGP) Hijacks
- Resource Public Key Infrastructure (RPKI)
  - Increase awareness of RPKI (what is RPKI and its benefits)
  - The importance of creating ROAs and deploying RPKI
  - How to deploy RPKI
  - How to check ROAs

Internet Routing

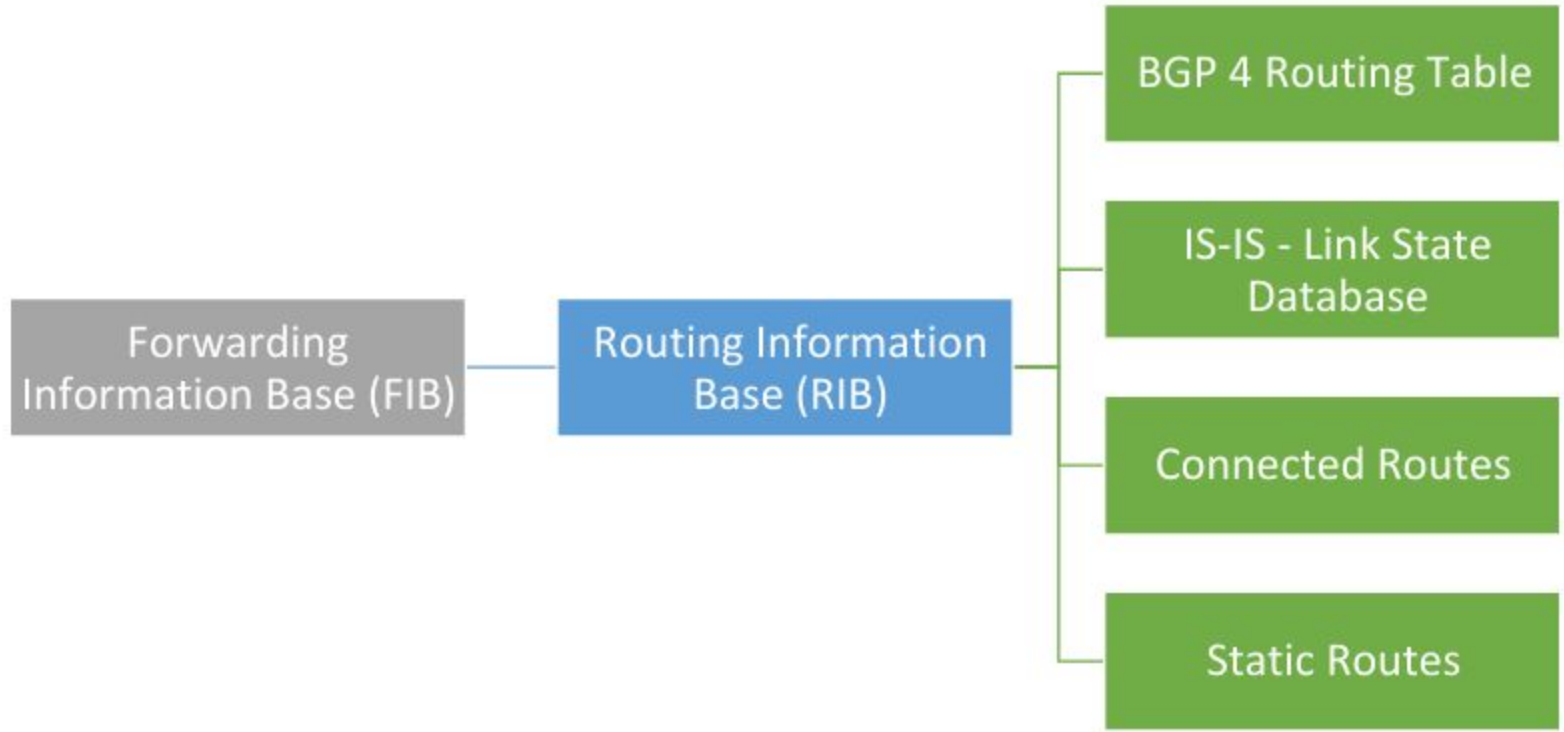


Screen shot taken from "3.5.3.4 Packet Tracer - Configure and Verify eBGP.pka" example from Connecting Networks Cisco Net academy course


Dynamic Routing Protocols
Interior Gateway Protocols (IGPs)
Exterior Gateway Protocols (EGPs)
Distance Vector Routing Protocols
Link-State Routing Protocols
Path-Vector Routing Protocol
RIPv1
IGRP
RIPv2
EIGRP
OSPF
IS-IS
BGP

Routing Table

Image by Stephan Fuchs from Pixabay

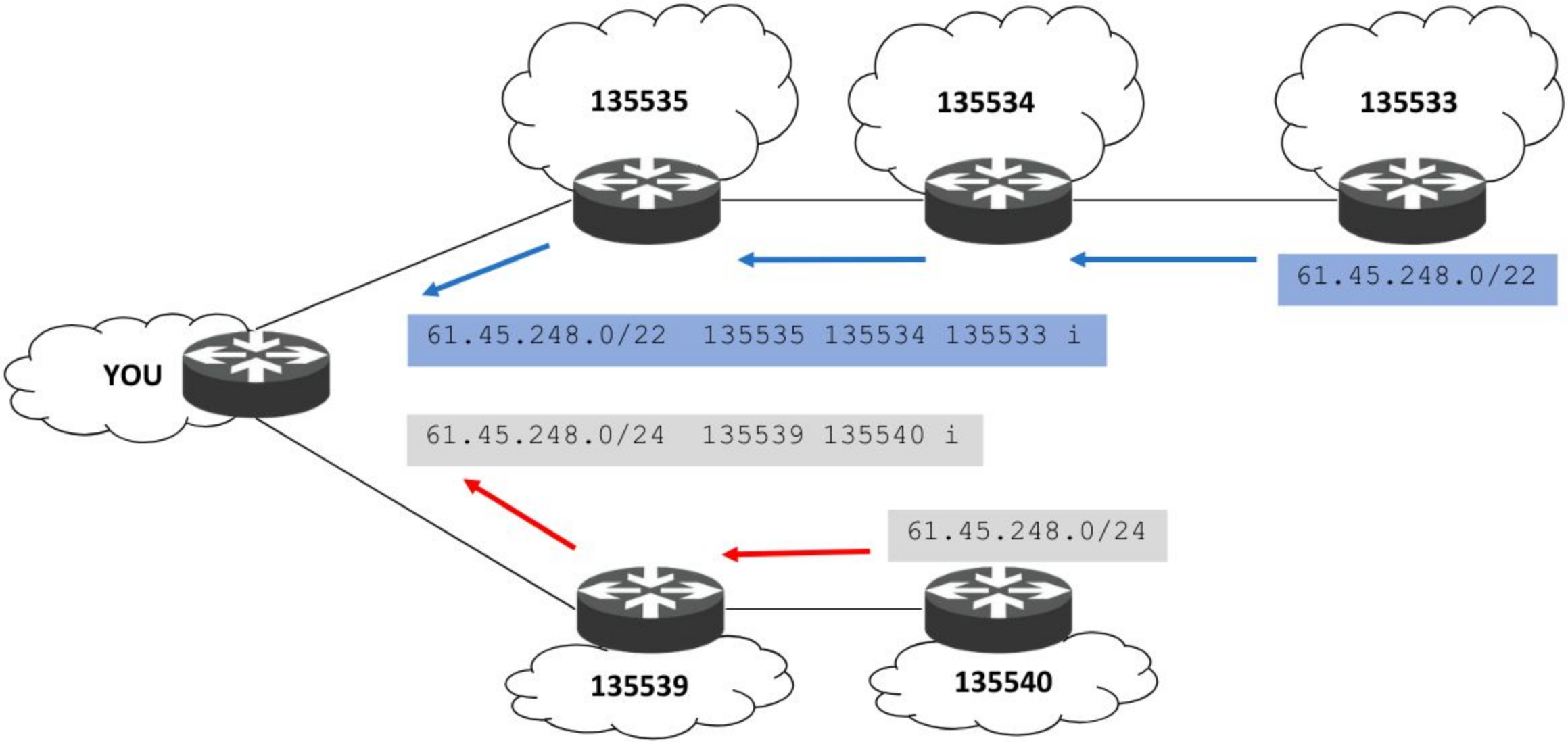


IP Route Lookup

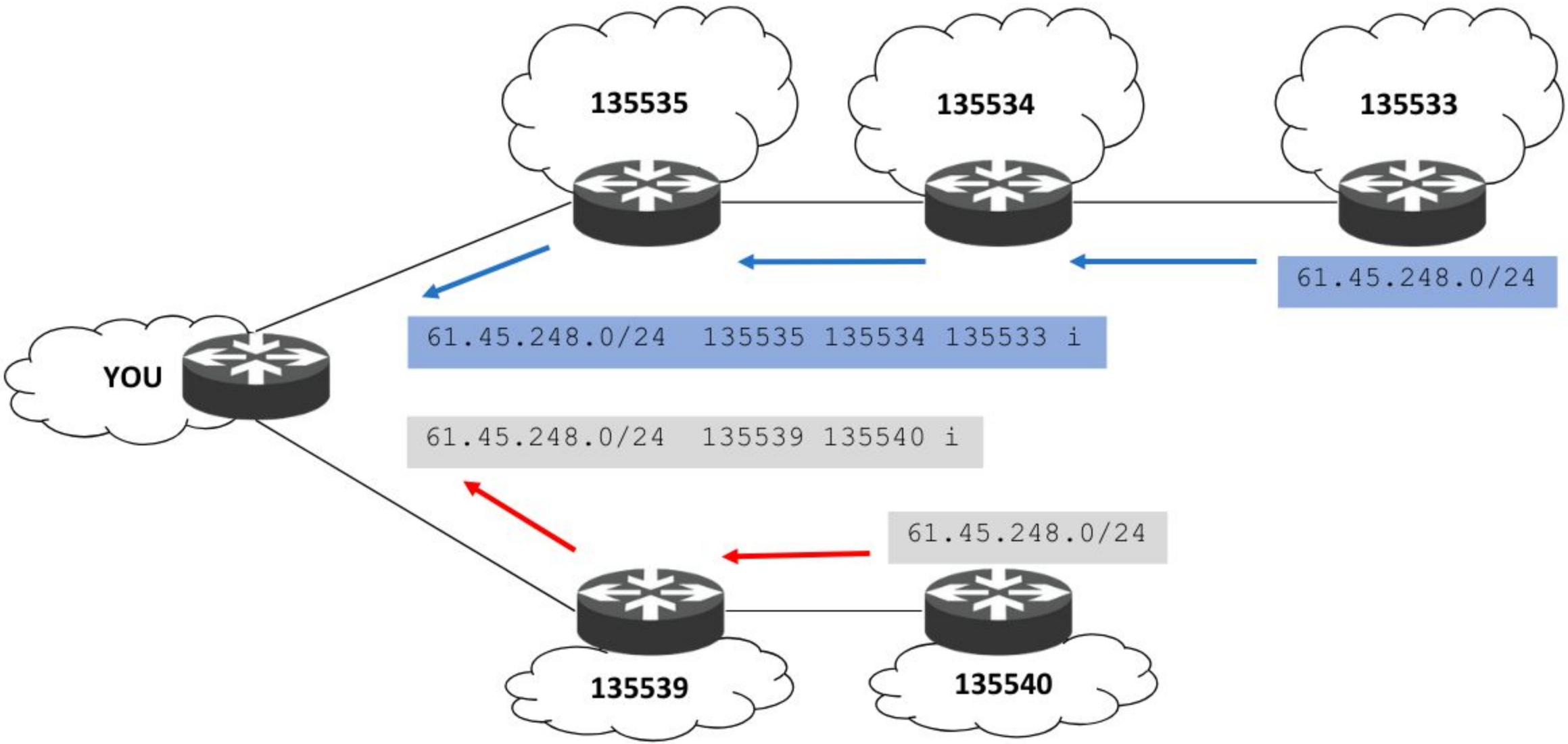
Based on destination IP address - “longest match” routing

More specific prefix preferred over less specific prefix

IP Route Lookup



IP Route Lookup

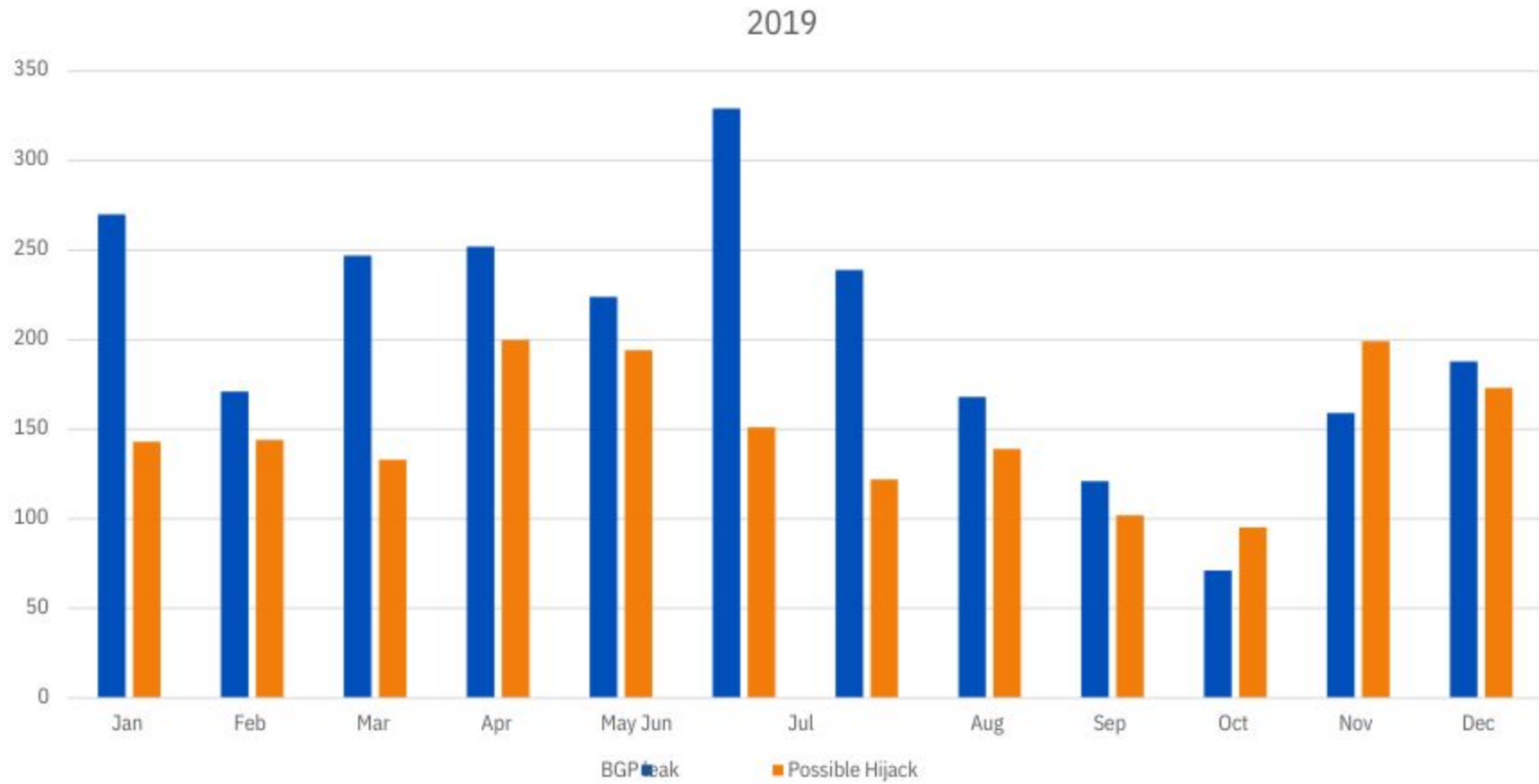


BGP Hijack 101

- Announce a more specific path
- Announce an IP address space that is owned by someone else

Williams, R. (2015).*street signs being stolen*[Image]. Retrieved from https://media.apnarm.net.au/media/images/2015/02/06/IQT_06-02-2015_NEWS_05_STOLENSIGNS1_t1880.jpg

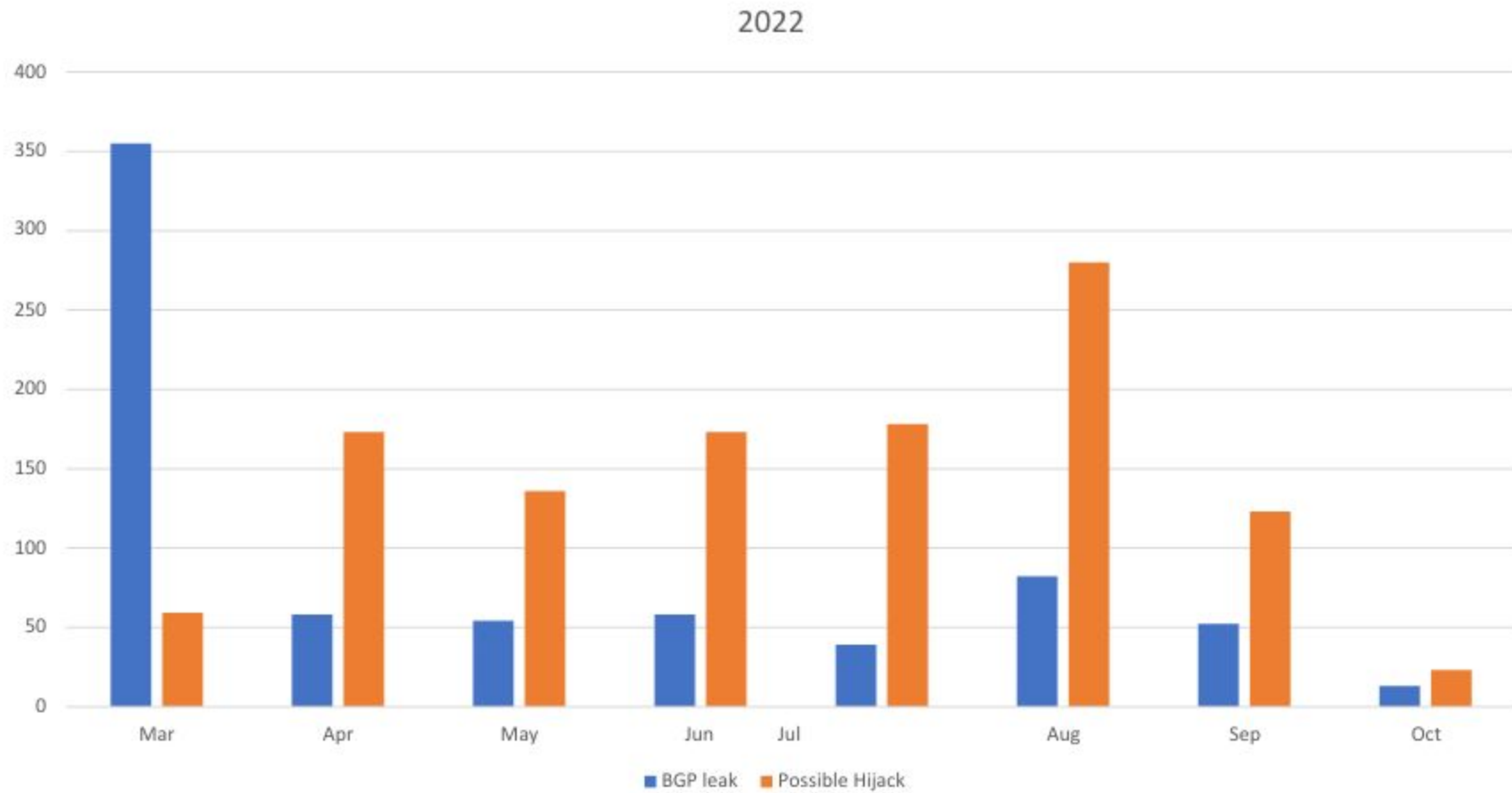
Hijacks and Leaks in 2019



https://bgpstream.com

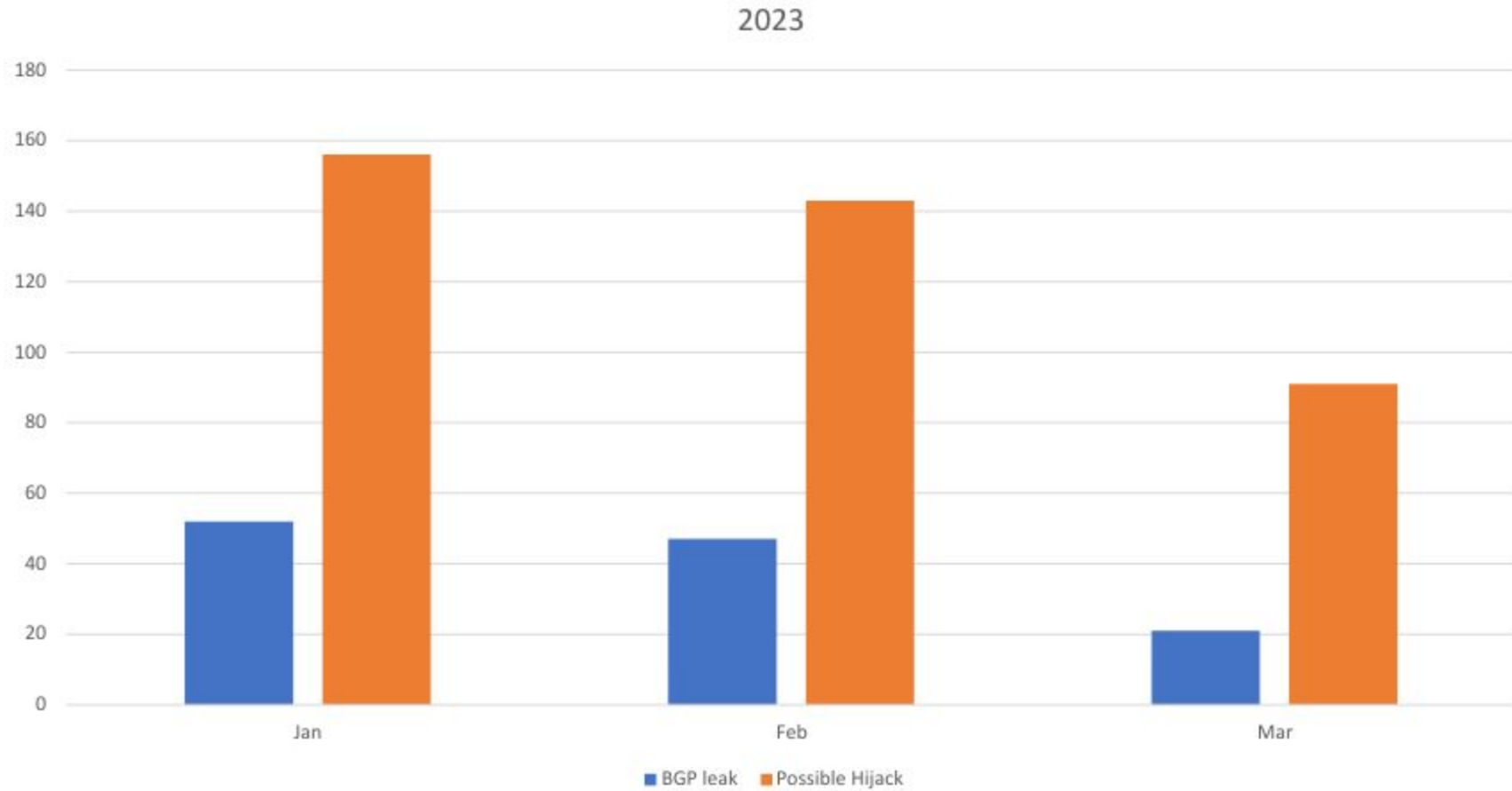
http://blog.catchpoint.com/2020/04/09/one-year-bgp-security

Hijacks and Leaks in 2022



https://bgpstream.com

Hijacks and Leaks in 2023 (so far)



https://bgpstream.com

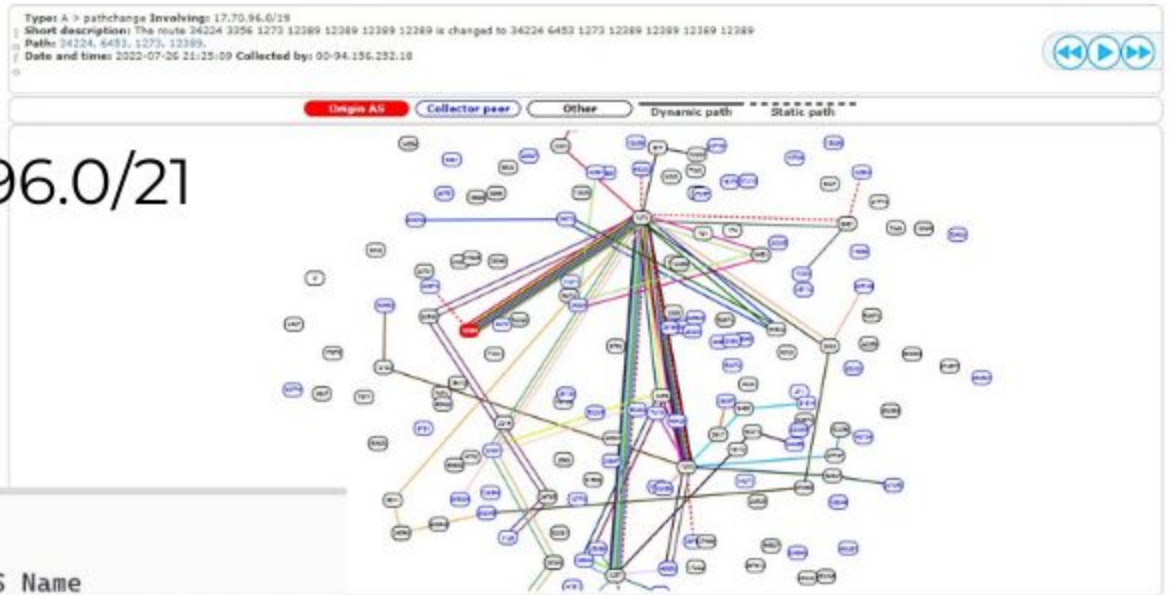
13th Mar 2023

Acknowledgements

- Slides/ideas from
  - Geoff Huston, APNIC
  - Randy Bush, IIJ Labs/Arrcus
  - MANRS, see https://www.manrs.org/
  - RPKI at APNIC, see https://www.apnic.net/rpki-at-apnic/

Headlines

- AS12389 Leaks Apple's Prefixes –26-27 July 2022
  - Russia's Rostelecomstarted announcing routes for part of Apple's (AS714) network (17.70.96.0/19)
  - Lasted more than 12 hours
  - To mitigate Apple announced 17.70.96.0/21
  - Turns out that Rostelcomowns (217.70.96.0/19)
  - Probably a fat finger mistake

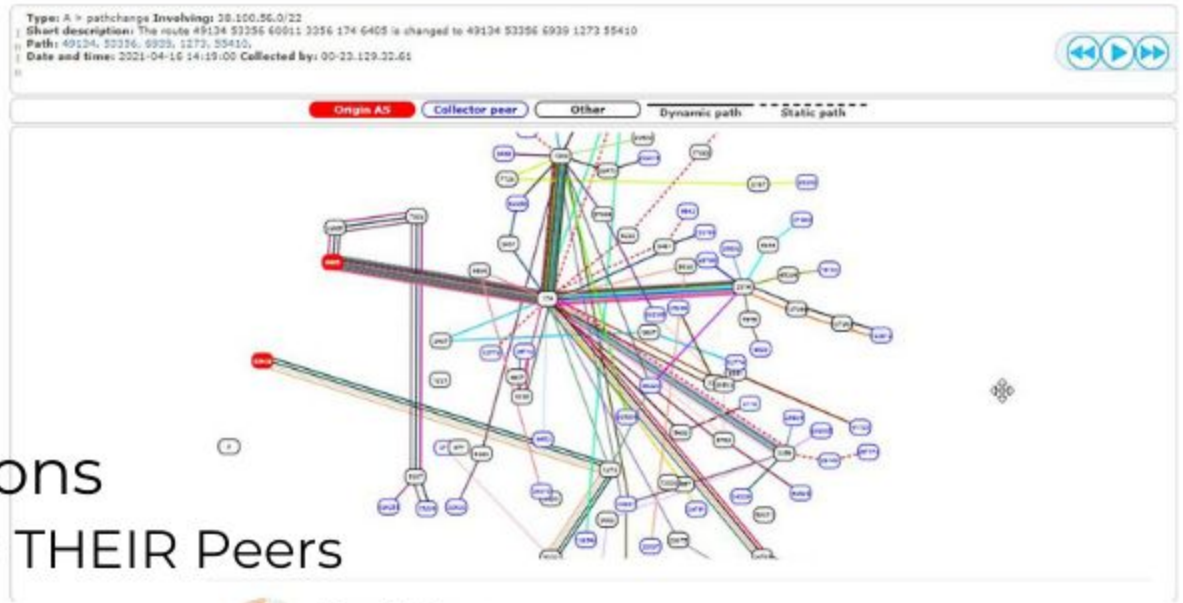



```
                    ~$ whois -h whois.cymru.com "-v 17.70.96.0"
Warning: RIPE flags used with a traditional server.
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
714     | 17.70.96.0       | 17.0.0.0/9          | US | arin     | 1990-04-16 | APPLE-ENGINEERING, US
                    :~$ whois -h whois.cymru.com "-v 217.70.96.0"
Warning: RIPE flags used with a traditional server.
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
12389   | 217.70.96.0      | 217.70.96.0/19      | RU | ripencc  | 2000-11-20 | ROSTELECOM-AS, RU
```

https://bgpstream.crosswork.cisco.com/event/293915

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 (AS6939)



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.

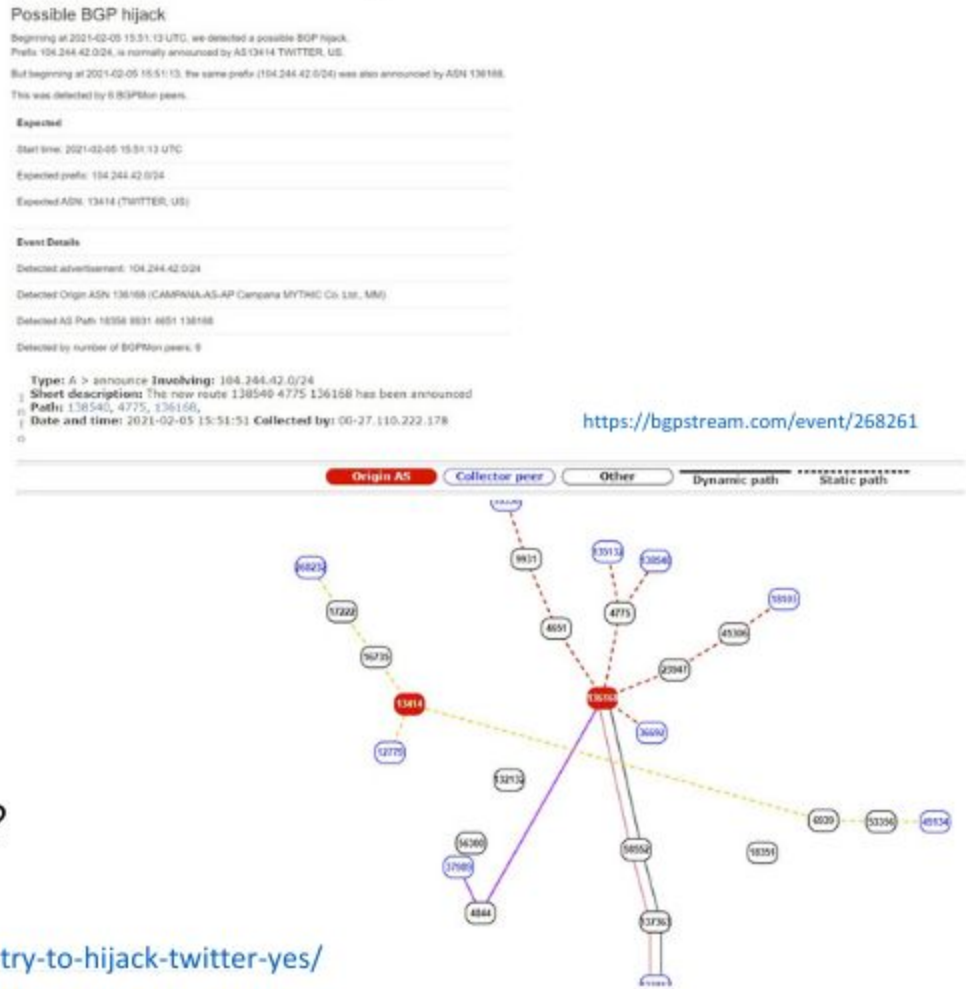
https://bgpstream.com/event/271479 https://bgpstream.com/event/271478

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/

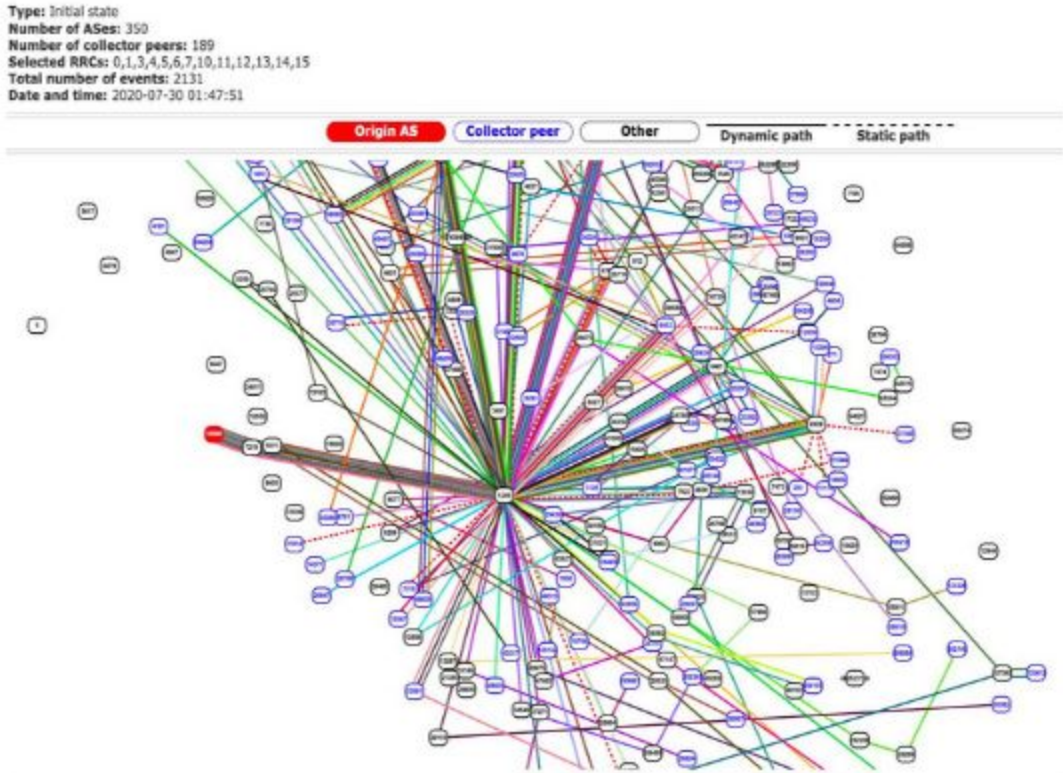


Headlines

- Optimizer strikes (again) –**30 July 2020**
  - AS10990 (Tulix) hijacks/leaks 200+ more specifics
    - Mostly North American ASNs
      - > 2 hours

https://radar.qrator.net/blog/as10990-routing-optimization-tale

```
104.230.0.0/18206313 6724 1299 7219 10990
104.230.64.0/18206313 6724 1299 7219 10990
107.184.0.0/16206313 6724 1299 7219 10990
107.185.0.0/16206313 6724 1299 7219 10990
107.189.192.0/19206313 6724 1299 7219 10990
107.189.224.0/19206313 6724 1299 7219 10990
```



https://bgpstream.com/event/245265.

  - AS1299 accepted/propagated the leaks:
    - Filters
    - ?

ROV?

Headlines

- Not so funny –1 April 2020
  - AS12389 (Rostelecom) hijacks/leaks 8K+ more specifics
    - Facebook, Cloudflare, AWS, Akamai, Google, Digital Ocean….
    - ~200 ASNs
  - Some peers accepted/propagated the leaks:
    - →AS20764 (Rascom) AS174 (Cogent) AS3356 (Level3)

Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 104.18.216.0/21	AS13335 - CLOUDFLARENET - [US]: 265 - 104.18.208.0/20 from 2020-04-01 19:33 to 2020-04-01 20:04 [high] 265 - 104.16.0.0/12 from 2020-04-01 19:33 to 2020-04-01 20:04 [high]	2020-04-01 19:33	0:31:00
Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 104.17.128.0/21	AS13335 - CLOUDFLARENET - [US]: 269 - 104.17.128.0/20 from 2020-04-01 19:33 to 2020-04-01 20:04 [high] 269 - 104.16.0.0/12 from 2020-04-01 19:33 to 2020-04-01 20:04 [high]	2020-04-01 19:33	0:31:00
Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 104.18.184.0/21	AS13335 - CLOUDFLARENET - [US]: 266 - 104.18.176.0/20 from 2020-04-01 19:33 to 2020-04-01 20:04 [high] 266 - 104.16.0.0/12 from 2020-04-01 19:33 to 2020-04-01 20:04 [high]	2020-04-01 19:33	0:31:00
Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 95.100.200.0/24	AS20940 - AKAMAI-ASN1 - [EU]: 327 - 95.100.200.0/22 from 2020-04-01 19:33 to 2020-04-01 20:04 [high] AS34164 - AKAMAI-LON - [GB]: 327 - 95.100.0.0/15 from 2020-04-01 19:33 to 2020-04-01 20:04 [high]	2020-04-01 19:33	0:31:00

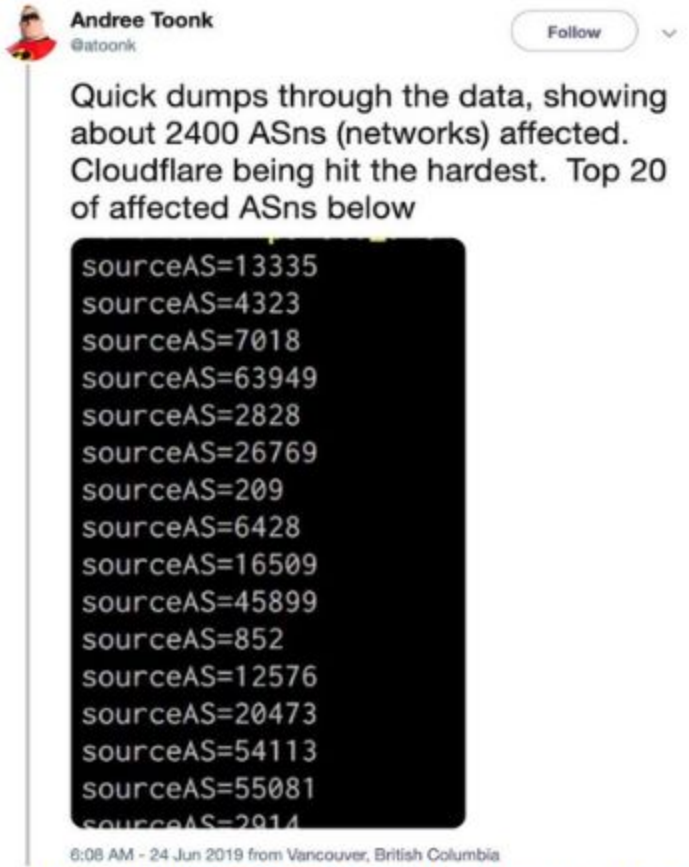
https://blog.qrator.net/en/how-you-deal-route-leaks_69/

Headlines

- BGP Optimizers impact Internet –**June 2019**
  - AS13335 hosted sites were not reachable during the leak
    - About 15% of their global traffic!! ~
    - 120mins

On Mon, Jun 24, 2019 at 3:57 AM [redacted] wrote:

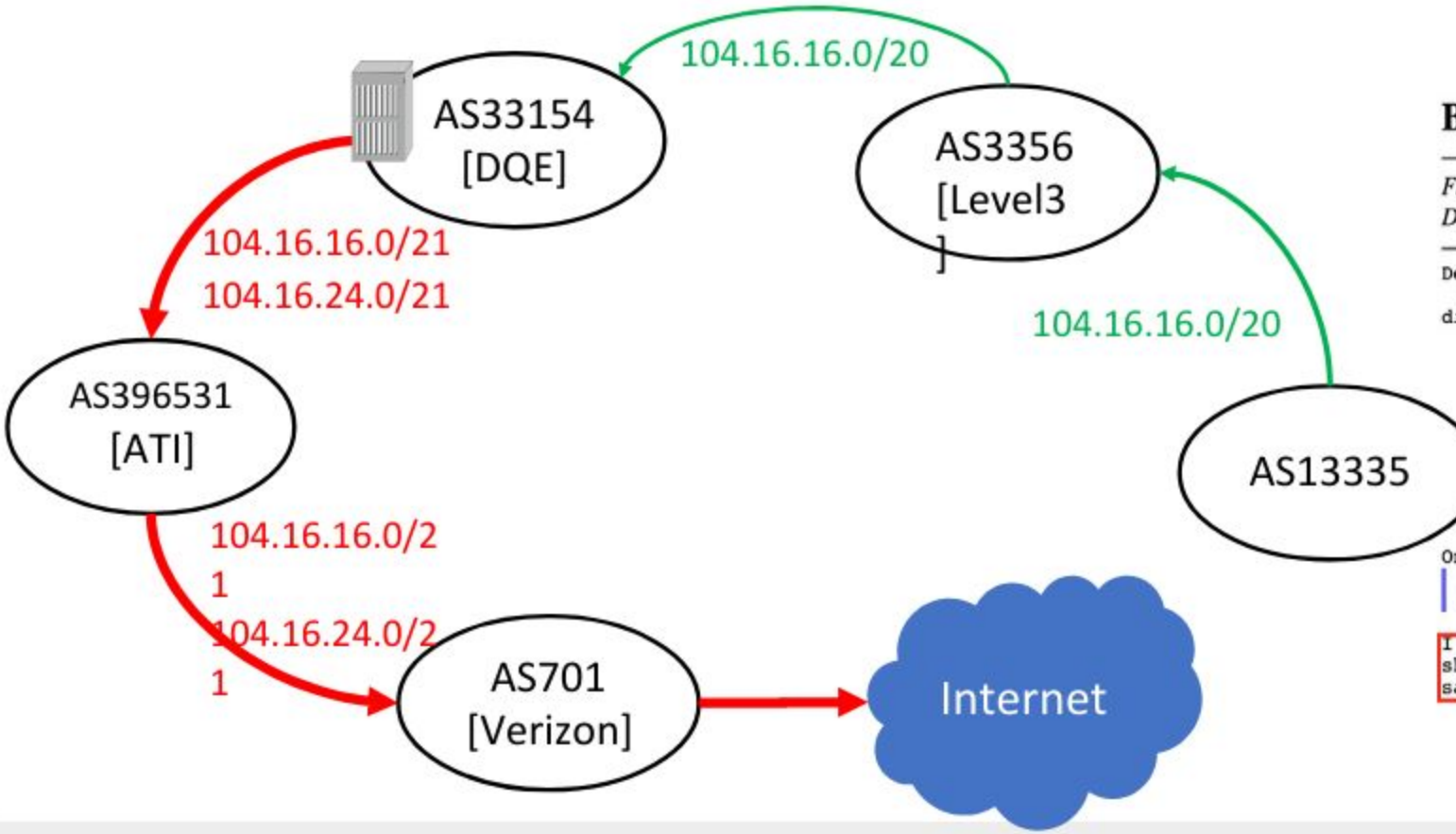
> Hello are there any issues with CloudFlare services now?



https://twitter.com/atoonk/status/1143143943531454464/photo/1

Headlines

- BGP Optimizers impact Internet (contd...)
  - What & How?



BGP Optimizers (Was: Validating possible BGP MITM attack)

From: Job Snijders <job () ntt net>
Date: Thu, 31 Aug 2017 22:06:49 +0200

Dear all,

disclaimer:

[The following is targetted at the context where a BGP optimizer generates BGP announcement that are ordinarily not seen in the Default-Free Zone. The OP indicated they announce a /23, and were unpleasantly surprised to see two unauthorized announcements for /24 more-specifics pop up in their alerting system. No permission was granted to create and announce these more-specifics. The AS_PATH for those /24 announcements was entirely fabricated. Original thread https://mailman.nanog.org/pipermail/nanog/2017-August/092124.html]

On Thu, Aug 31, 2017 at 11:13:18AM -0700, Andy Litzinger wrote:
> Presuming it was a route optimizer and the issue was ongoing, what would be the suggested course of action?

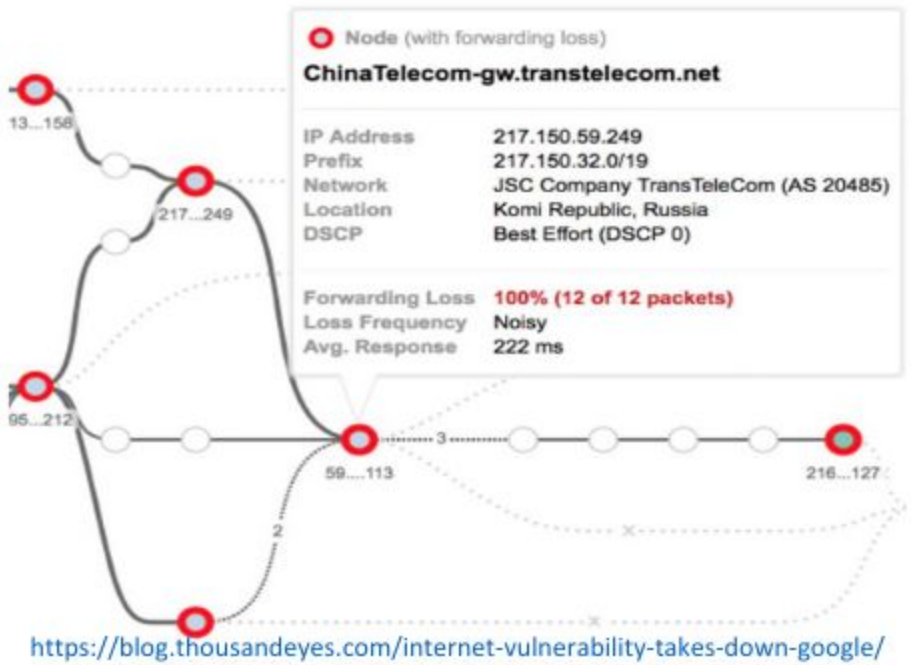
I strongly recommend to turn off those BGP optimizers, glue the ports shut, burn the hardware, and salt the grounds on which the BGP optimizer sales people walked.

Headlines

- Google prefix leaks –**Nov 2018**
  - Google services (G-Suite, Google search and Google analytics) affected by the leak
    - Traffic dropped at AS4809 (China Telecom)
    - ~ 74mins

Headlines

- Google prefix leaks (contd...)
  - How?
    - AS37282 (MainOne) leaked to AS4809 (CT) at IXPN, who leaked it to others like AS20485 (TransTelecom)



https://blog.thousandeyes.com/internet-vulnerability-takes-down-google/

Headlines

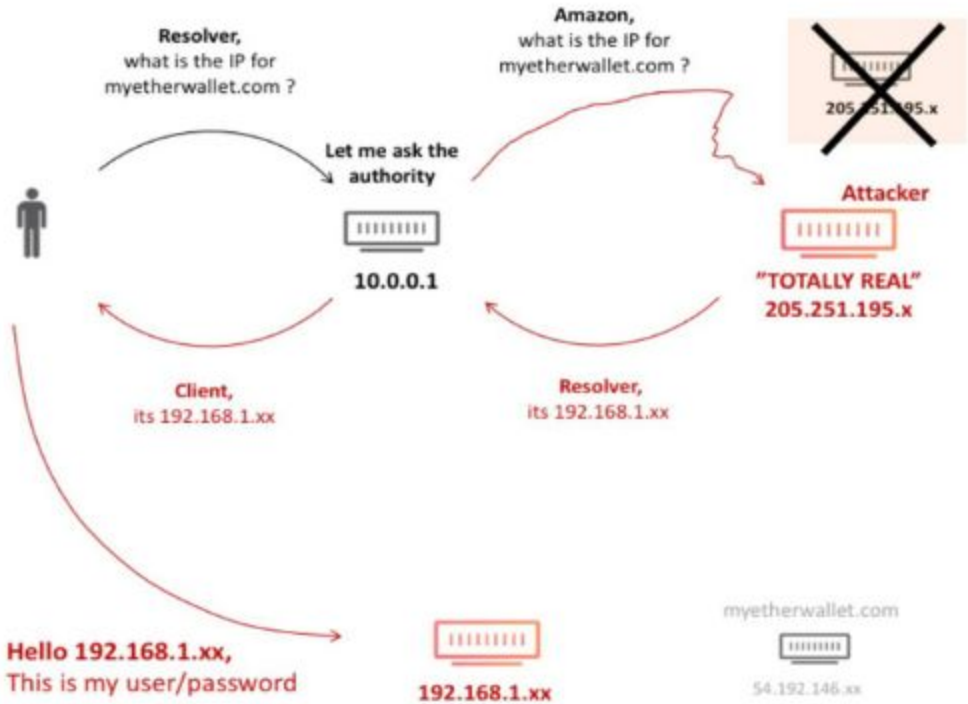
- Amazon (AS16509) Route53 hijack –**April2018**
  - AS10279 (eNET) originated more specifics (/24s) of Amazon Route53's prefix (205.251.192.0/21)

    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

- Google brings down Internet in Japan -**Aug 2017**
  - ~ 24 hours
  - Google (AS15169) leaked *>130K* prefixes to Verizon (AS701) in Chicago
    - Normally ~ 50 prefixes
    - ~25K of those were NTT OCN's (AS4713) more specifics
    - which was leaked onwards to KDDI and IIJ (and accepted)
  - Everyone who received the leaked more specifics, preferred the Verizon-Google path to reach NTT OCN!

Headlines

- Google leak (contd...)

```
trace from Tokyo, Japan to Inuyama, Japan at 04:44 Aug 24, 2017
 1  *
 2  202.177.203.50    xe-0-0-0.gw401.ty2.ap.equinix.com  Tokyo       Japan     0.717
 3  183.177.32.143    xe-1-1-1.gw402.ty1.ap.equinix.com  Tokyo       Japan     0.755
 4  143.90.232.25     25.143090232.odn.ne.jp             Tokyo       Japan     1.411
 5  143.90.161.73                                        Tokyo       Japan     2.757
 6  143.90.47.14      STOrs-01Te0-1-0-1.nw.odn.ad.jp     Tokyo       Japan     3.552
 7  210.252.167.230   230.210252167.odn.ne.jp            Tokyo       Japan     4.094
 8  *
 9  60.37.54.105      OCN (AS4713) CIDR BLOCK 70         Tokyo       Japan     4.088
10  125.170.97.85     OCN (AS4713) CIDR BLOCK 77                     Japan     4.017
11  125.170.97.74     OCN (AS4713) CIDR BLOCK 77         Ōsaka-shi   Japan    12.263
12  153.149.219.22    OCN (AS4713) CIDR BLOCK 93         Ōsaka-shi   Japan    12.362
13  153.146.148.18    OCN (AS4713) CIDR BLOCK 93         Tokyo       Japan     14.45
14  60.37.32.250      OCN (AS4713) CIDR BLOCK 70                     Japan    13.116
15  118.23.141.202    OCN (AS4713) CIDR BLOCK 86                     Japan    13.332
16  118.23.142.99     OCN (AS4713) CIDR BLOCK 86                     Japan    22.307
17  211.11.83.160     OCN (AS4713) CIDR BLOCK 23         Inuyama     Japan    15.672
```

Before leak (JP->JP)

After leak (JP->JP)

```
trace from Tokyo, Japan to Inuyama, Japan at 03:28 Aug 25, 2017
 1  *
 2  183.177.32.145    Equinix Asia Pacific               Tokyo       Japan     0.249
 3  210.130.154.37    IIJ IPv4 BLOCK ( AS2497 )          Tokyo       Japan     0.618
 4  58.138.102.109    tky001bb11.IIJ.Net                 Tokyo       Japan     0.877
 5  58.138.88.86      sjc002bb12.IIJ.Net                 San Jose    United States  97.797
 6  152.179.48.117    TenGigE0-3-0-8.GW6.SJC7.ALTER.NET  San Jose    United States  97.869
 7  *
 8  152.179.105.110   google-gw.customer.alter.net       Chicago     United States  337.19
 9  108.170.243.197   Google Inc.                        Chicago     United States  246.325
10  *
11  209.85.241.43     Google Inc.                                    United States  256.188
12  72.14.238.38      Google Inc.                        Vancouver   Canada         247.849
13  209.85.245.110    Google Inc.                        Vancouver   Canada         249.291
14  *
15  108.170.242.138   Google Inc.                        Tokyo       Japan          246.267
16  211.0.193.21      OCN (AS4713) CIDR BLOCK 21         Tokyo       Japan          246.351
17  122.1.245.65      OCN (AS4713) CIDR BLOCK 81         Tokyo       Japan          246.426
18  *
19  153.149.218.10    OCN (AS4713) CIDR BLOCK 93         Ōsaka-shi   Japan          256.027
20  125.170.96.38     OCN (AS4713) CIDR BLOCK 77                     Japan          255.683
21  *
22  60.37.32.250      OCN (AS4713) CIDR BLOCK 70                     Japan          254.989
23  118.23.141.202    OCN (AS4713) CIDR BLOCK 86                     Japan          254.526
24  *
25  211.11.83.160     OCN (AS4713) CIDR BLOCK 23         Inuyama     Japan          256.212
```

After leak (EU->EU)

```
trace from London, England to Nürnberg, Germany at 03:30 Aug 25, 2017
 1  *
 2  195.66.248.190    fe0-2.tr2.linx.net                 London      United Kingdom   0.327
 3  195.66.249.10     ge0-2-502.tr5.linx.net             London      United Kingdom   0.441
 4  195.66.249.13     ge0-2-501.tr4.linx.net             London      United Kingdom   0.477
 5  195.66.248.10     uunet-uk-transit.thn.linx.net      London      United Kingdom   0.507
 6  158.43.193.245    POS0-0.CR2.LND6.ALTER.NET          London      United Kingdom   0.497
 7  140.222.239.41    0.xe-0-0-0.IL1.NYC50.ALTER.NET     New York    United States  108.146
 8  146.188.4.197     xe-0-0-1.IL1.NYC41.ALTER.NET       New York    United States   75.719
 9  140.222.234.221   0.et-10-1-0.GW7.CHI13.ALTER.NET    Chicago     United States   94.793
10  152.179.105.110   google-gw.customer.alter.net       Chicago     United States  224.352
11  *
12  216.239.40.189    Google Inc.                        Northlake   United States  202.193
13  216.239.58.255    Google Inc.                                                   203.995
14  216.239.58.12     Google Inc.                                                   207.026
15  209.85.253.184    Google Inc.                        Luxembourg  Luxembourg     212.944
16  209.85.252.215    Google Inc.                                                   213.112
17  108.170.252.71    Google Inc.                                                   213.265
18  72.14.222.53      Google Inc.                                    Germany        212.061
19  188.111.165.169   Vodafone GmbH                                  Germany        227.077
20  178.7.138.112     Vodafone D2 GmbH                   Nürnberg    Germany        234.226
```

https://dyn.com/blog/large-bgp-leak-by-google-disrupts-internet-in-japan/

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

Why do we keep seeing these?

- As always, there is no E-bit (evil!)
  - RFC3514

    - A bad routing update does not identify itself as BAD
    - All we can do is identify GOOD updates
    - But how do we identify what is GOOD???

Why should we worry?

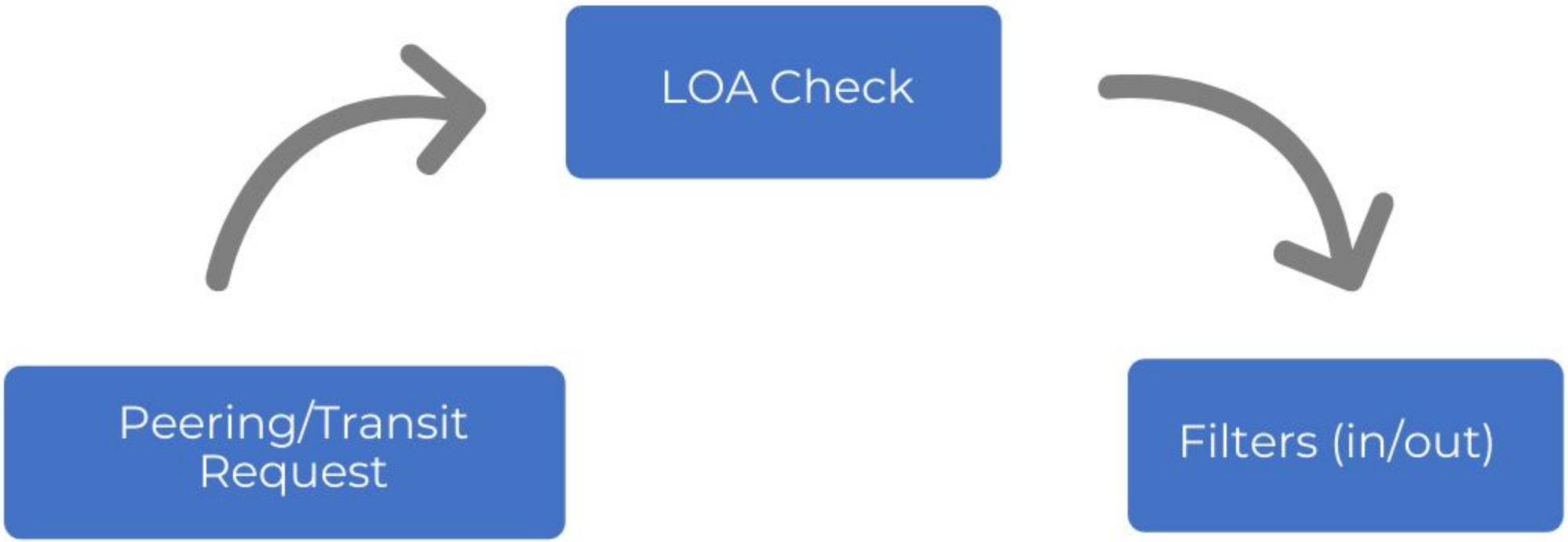
- Because it's just so easy to do bad in routing!

By Source (WP:NFCC#4), Fair use,
https://en.wikipedia.org/w/index.php?curid=42515224

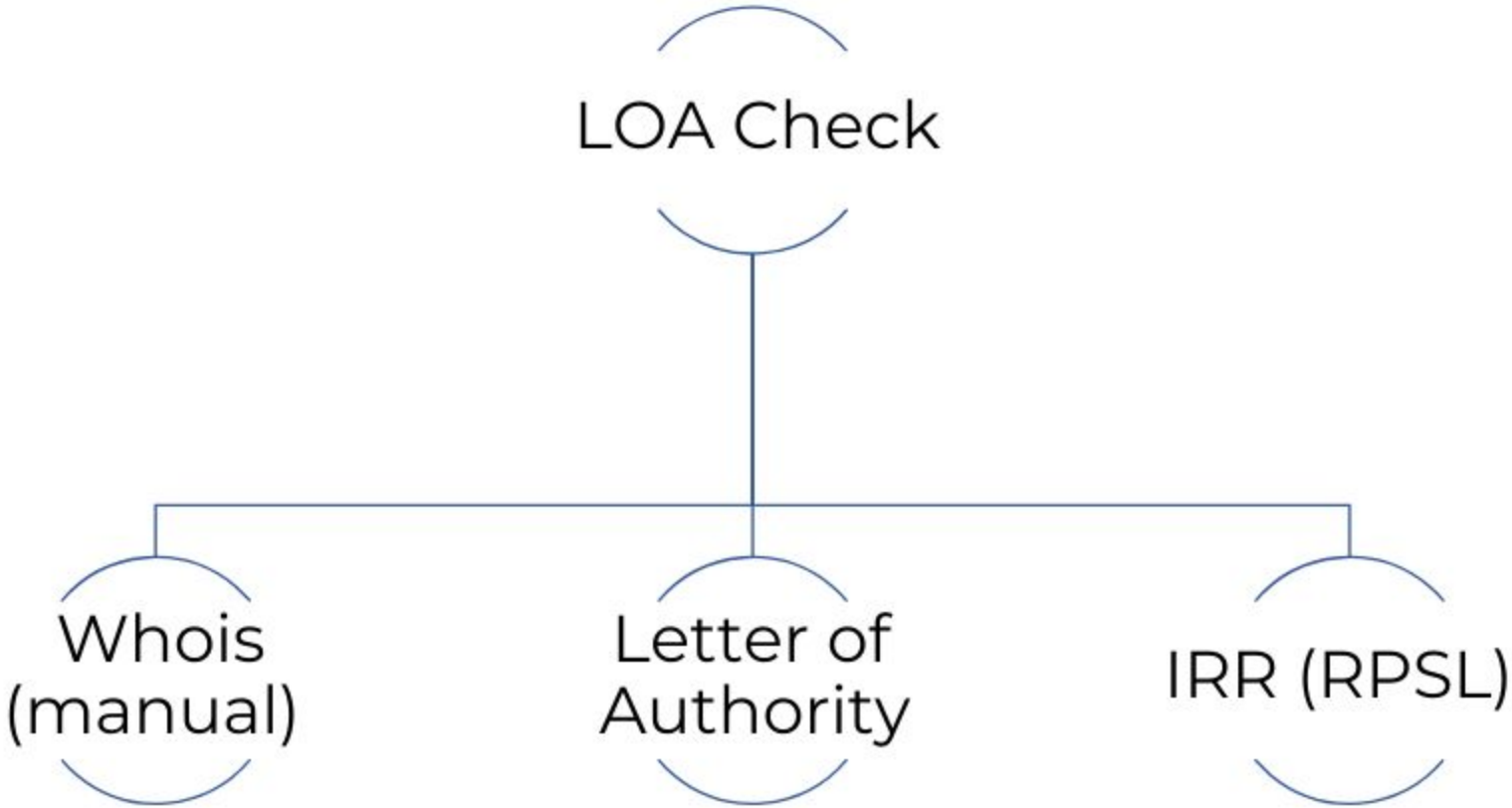
How do we address these?

- **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

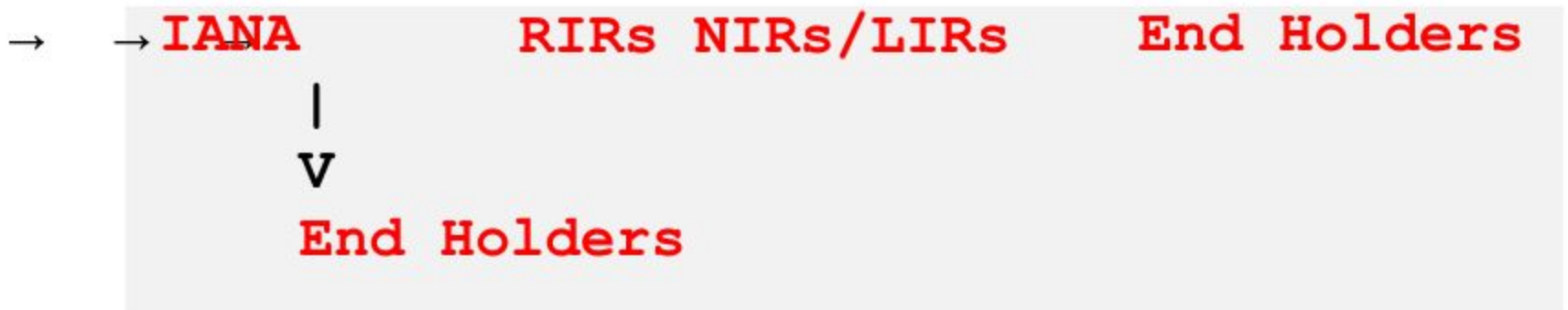
Back to basics –identify GOOD

- 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

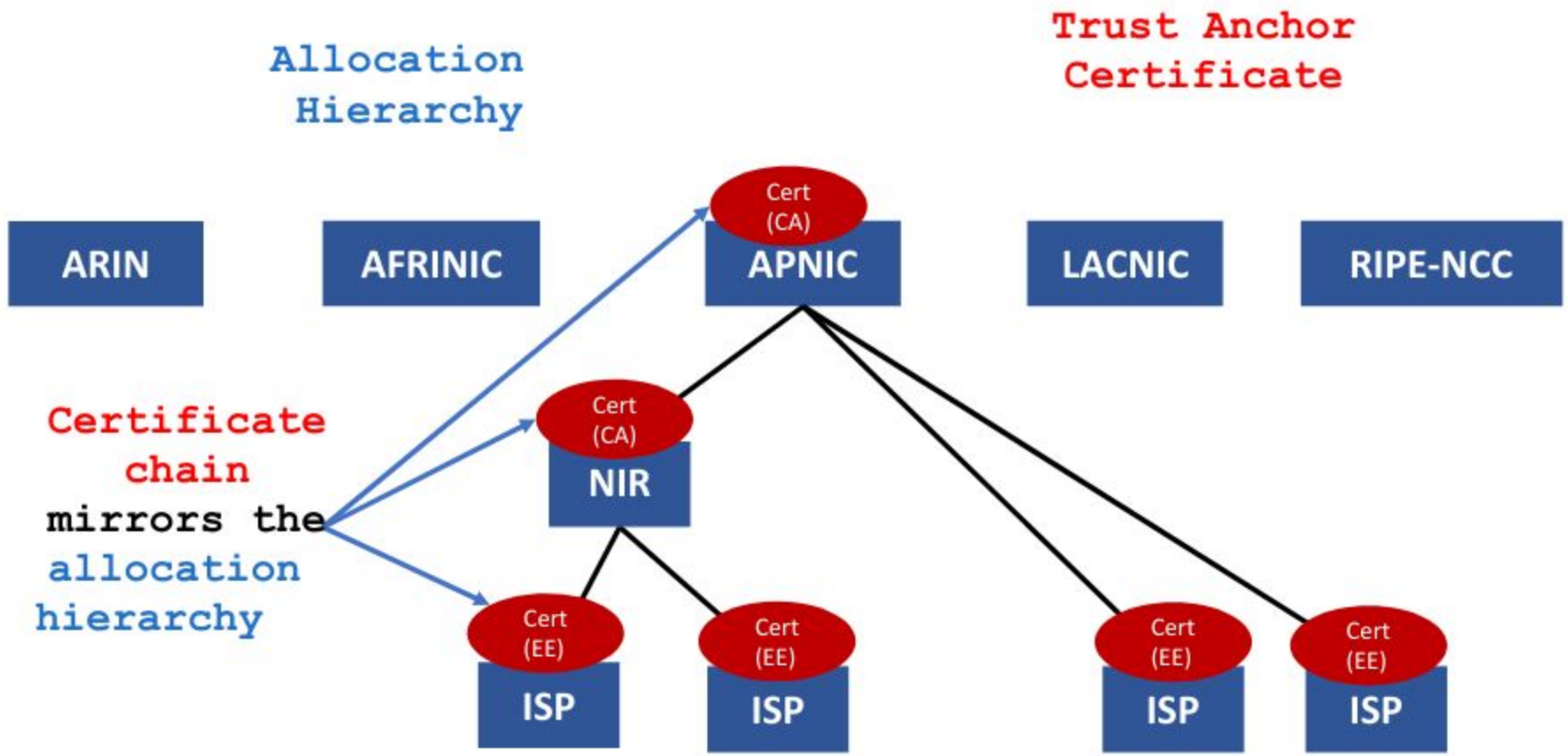
How about trust?

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



    - To describe the address allocation using digital certificates

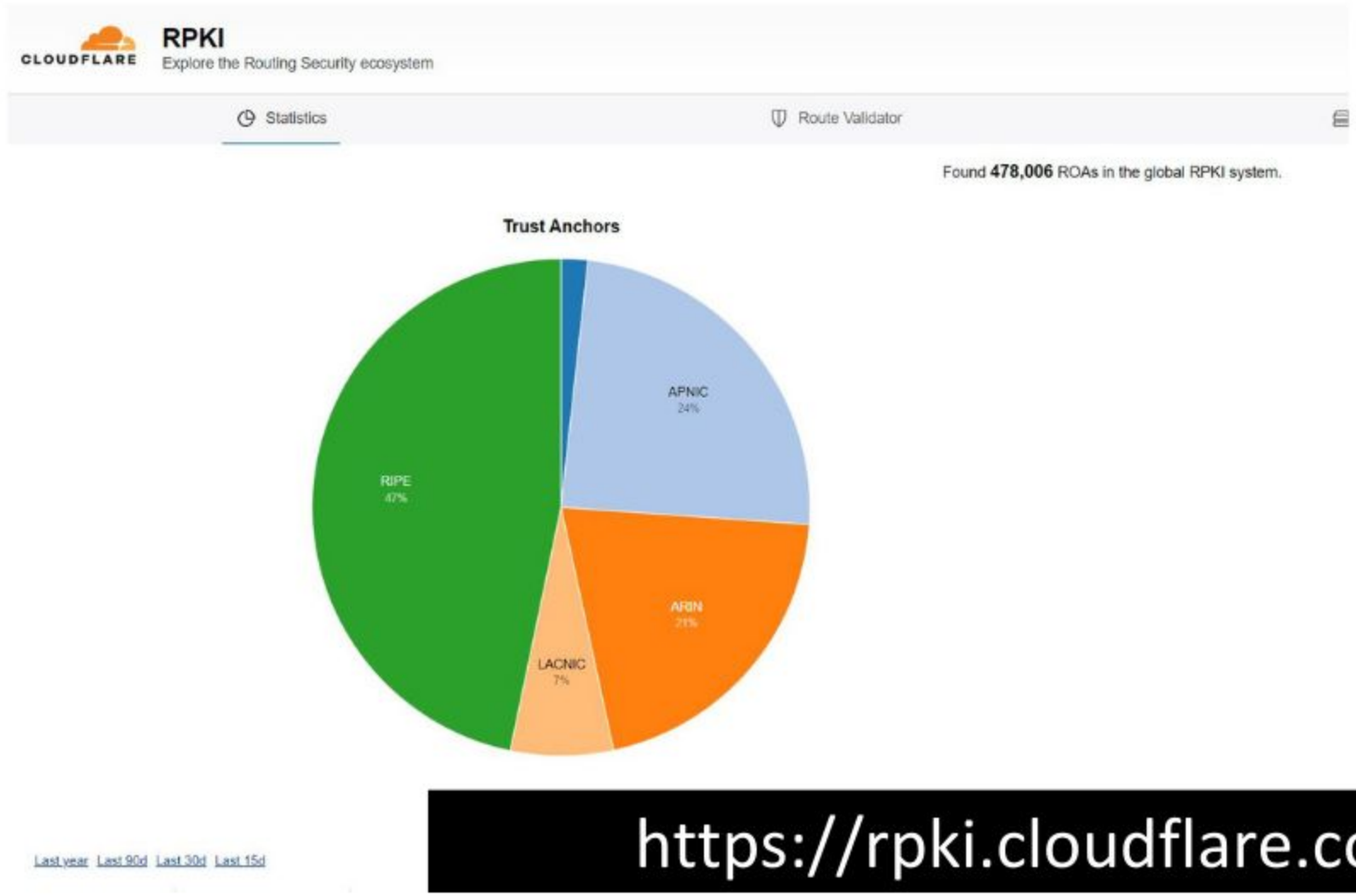
RPKI Chain of Trust



RPKI Chain of Trust

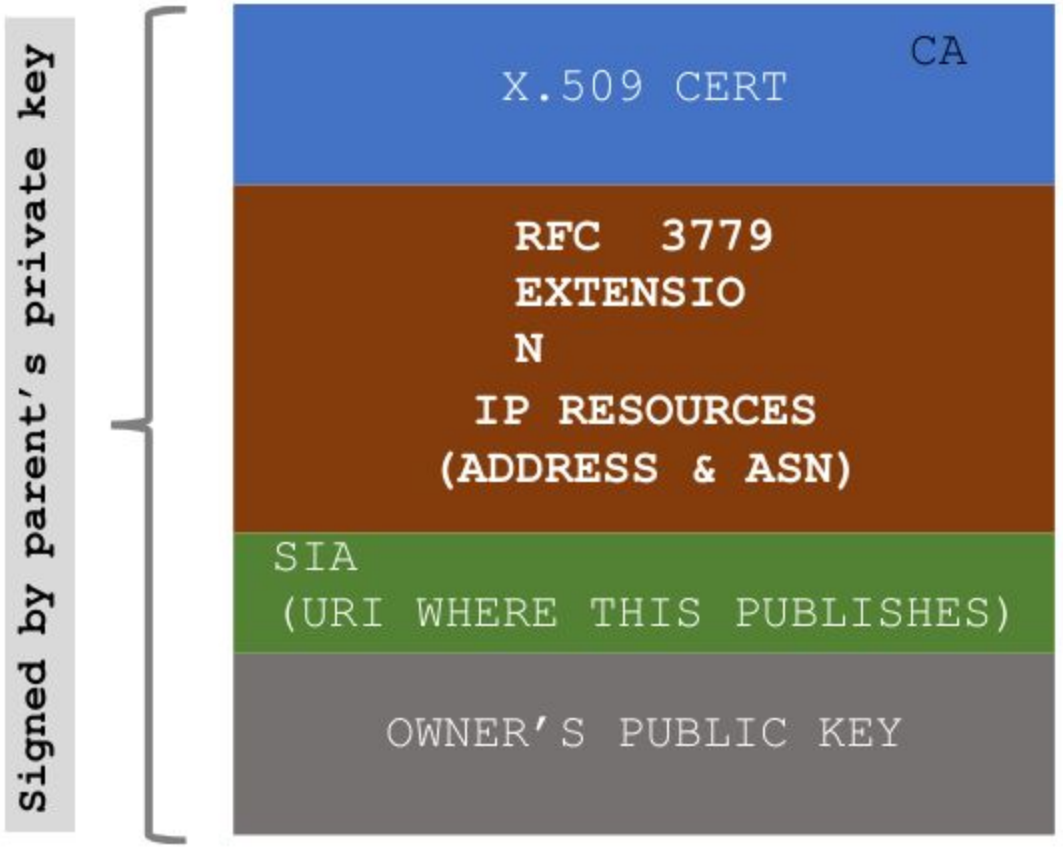
- RIRs hold a self-signed root certificate for all the resources they have in the registry
  - they are the *Trust Anchor* for the system
- The root certificate signs the resource certificates for end-holder allocations
  - binds the resources to the end-holders public key
- Any attestations signed by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

Chain of Trust?



https://rpki.cloudflare.com/

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)
  - *Aissues a resource certificate that binds the allocated address with B'spublic 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	203.176.32.0/1
Max-length	9 /24
Origin ASN	**AS17821**

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

What can RPKI do?

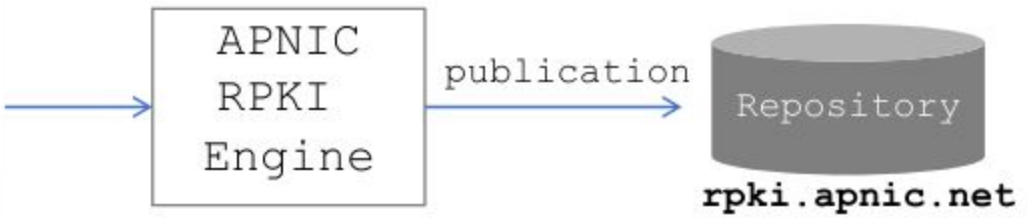
- Authoritatively proof:
  - Who is the legitimate owner of an address, and
  - Identify which ASNs have the permission from the holder to originate the address

- 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

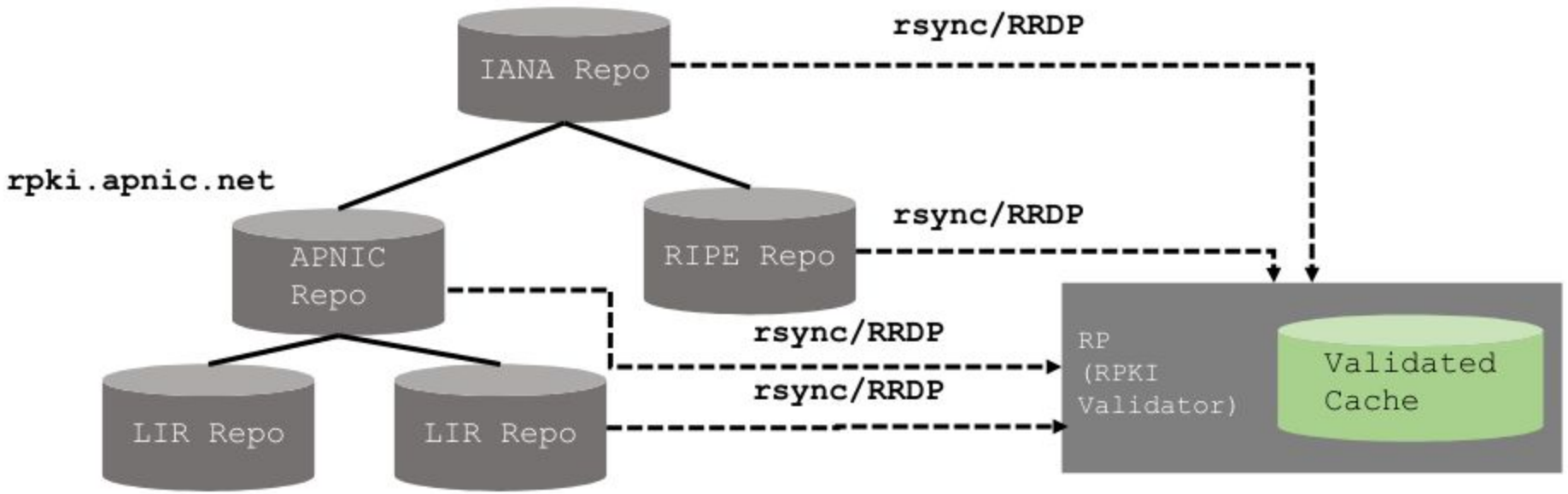
MyAPNIC GUI



rpki.apnic.net

RPKI Components

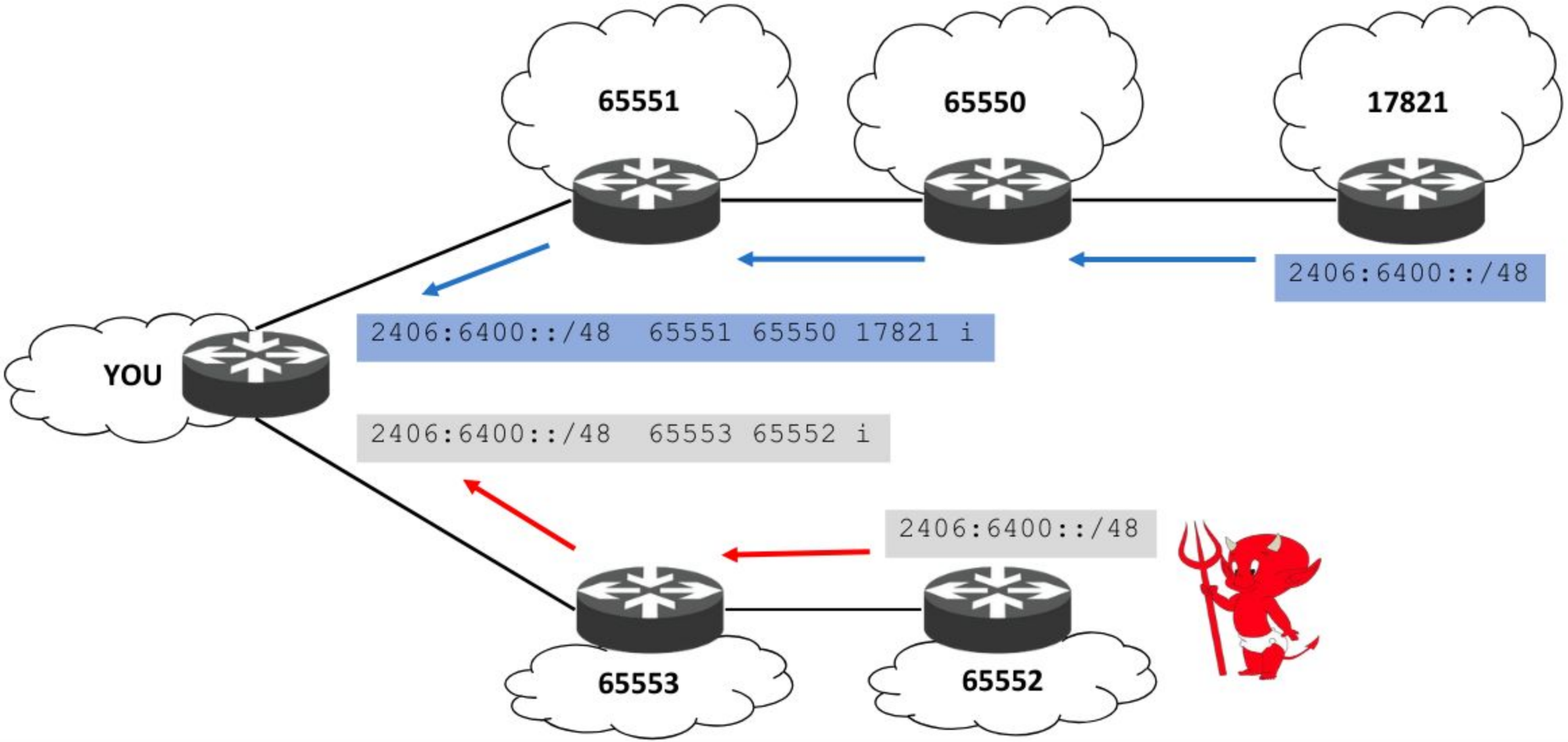
- **Relying Party (RP)**
  - RPKI Validator that gathers data (ROA) from the distributed RPKI repositories
  - Validates each entry's signature against the TA to build a "*Validated cache*"



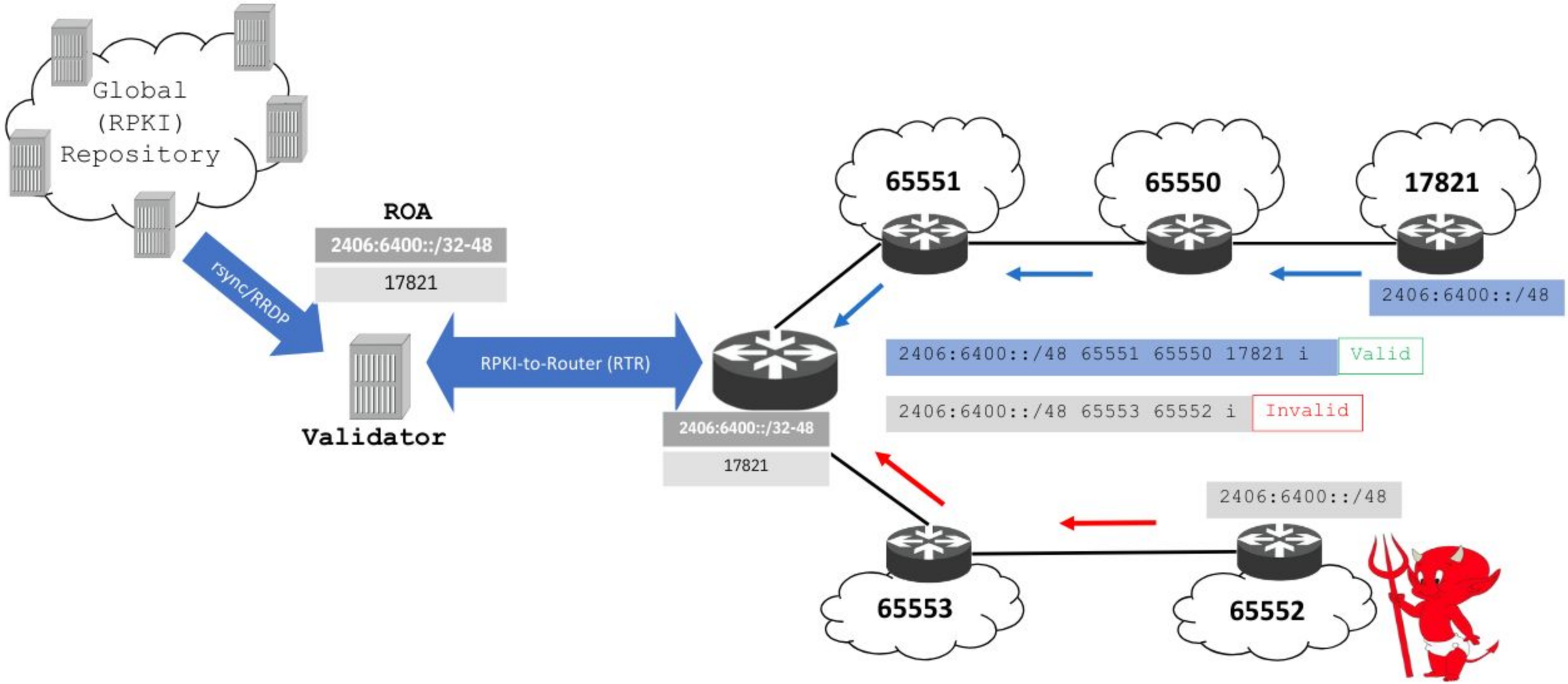
RPKI Service Models

- 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]
```

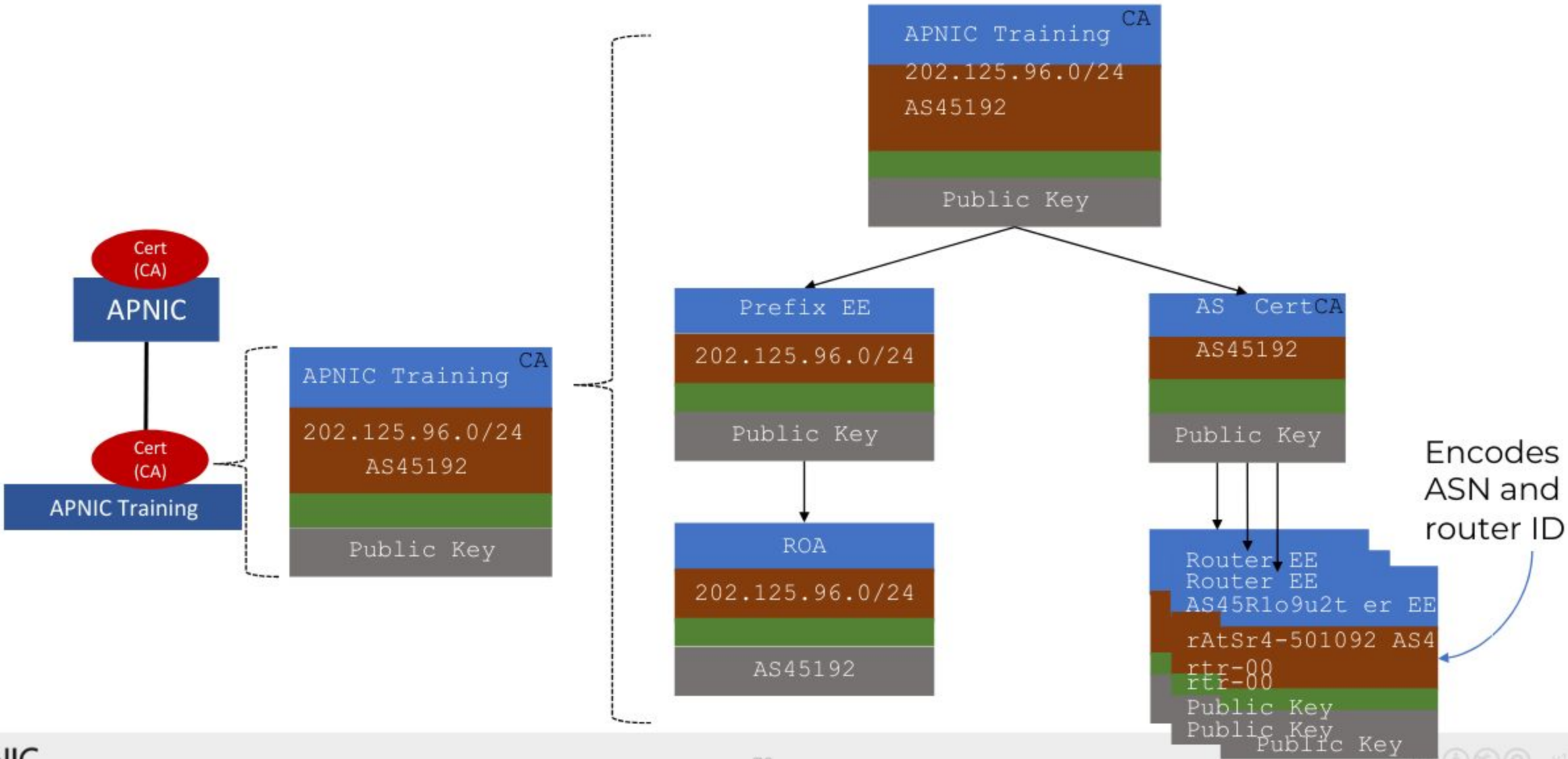
- Drop Invalids

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

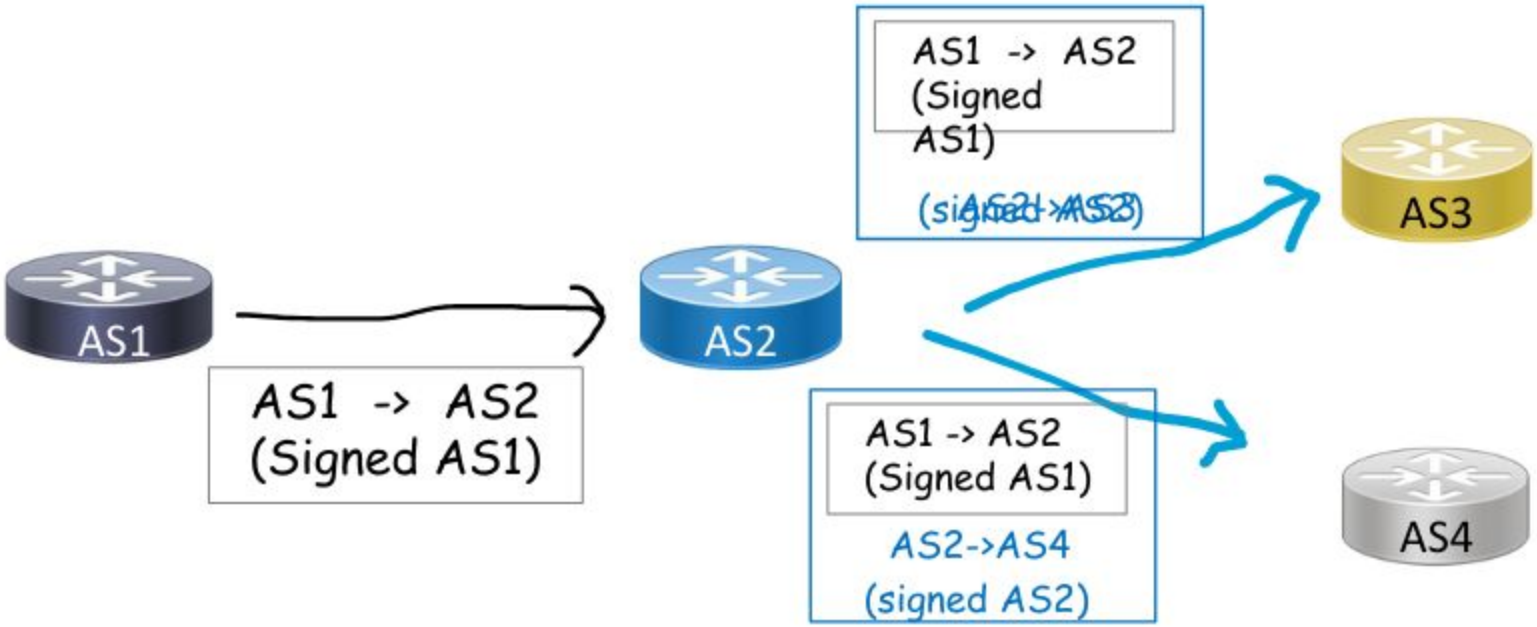
Are ROAs enough?

- 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 BGPsecspeaker 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 BGPsecwould 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 (IfAS(0)) UNKNOWN
  - 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, VALID

RPKI Further Reading

X.509 PKI Certificates

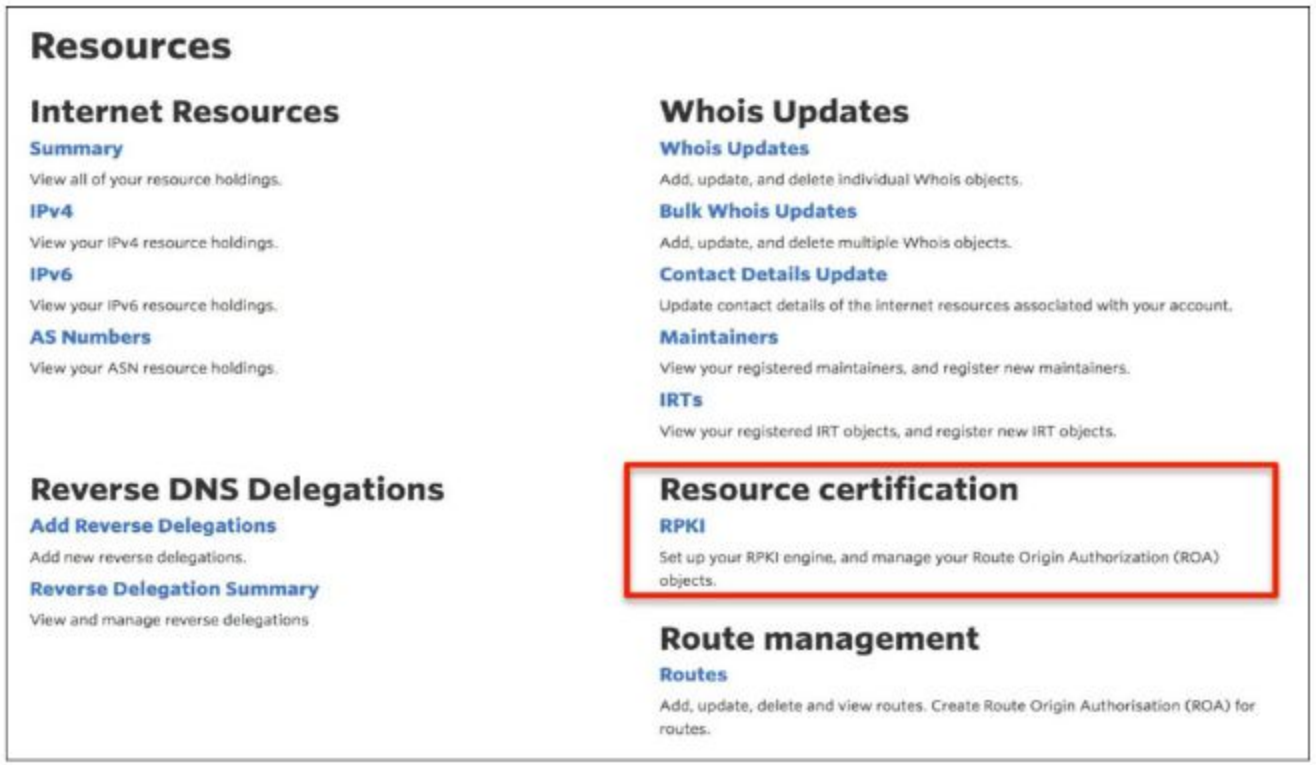
Extensions for IP Addresses and ASNs

Resource Public Key Infrastructure

Implementation

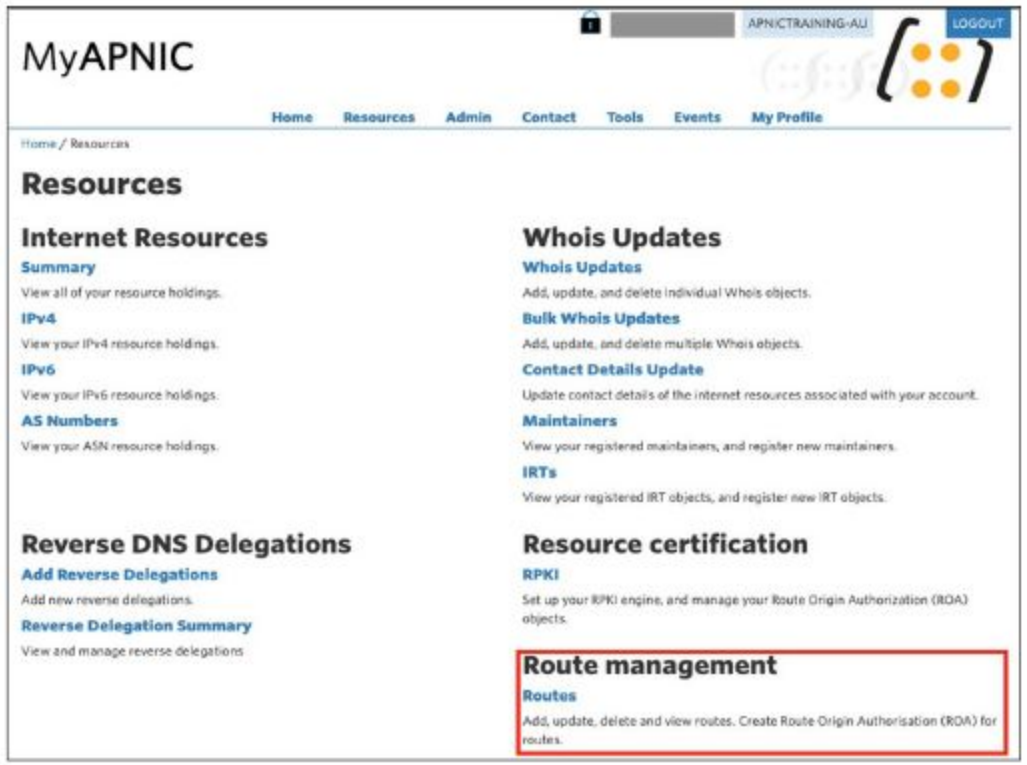
1.sign & publish your ROA

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



Create & publish your ROA

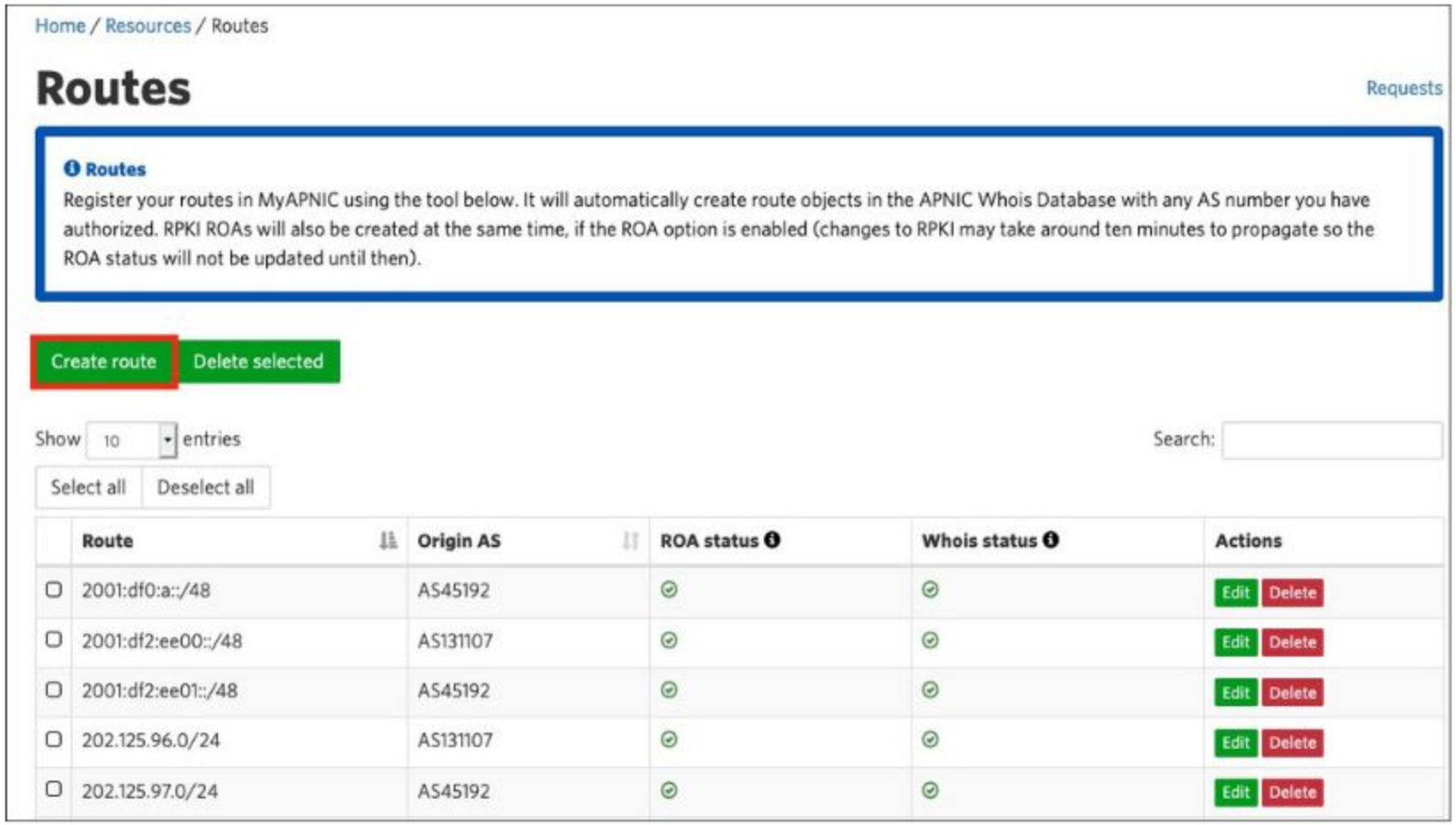
- Then go to the Routespage
  - → Go to **Resources** **Route Management** → **Routes** (see image below)



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

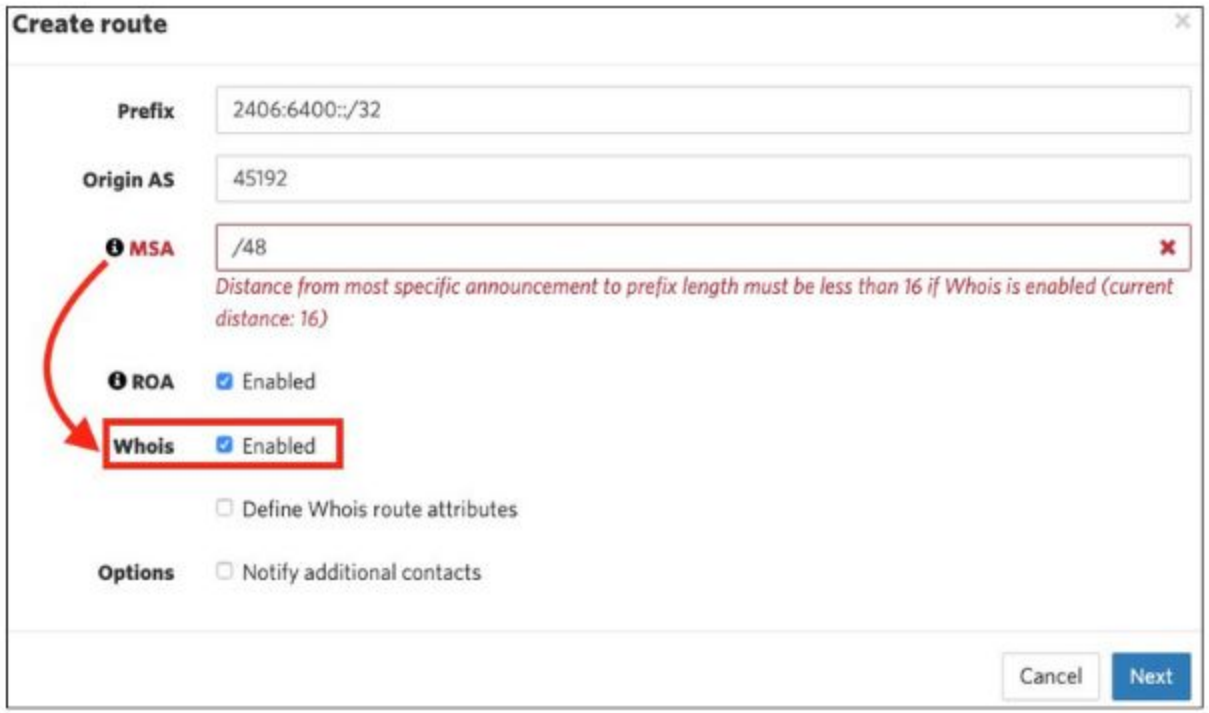
Create (publish) your ROA

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

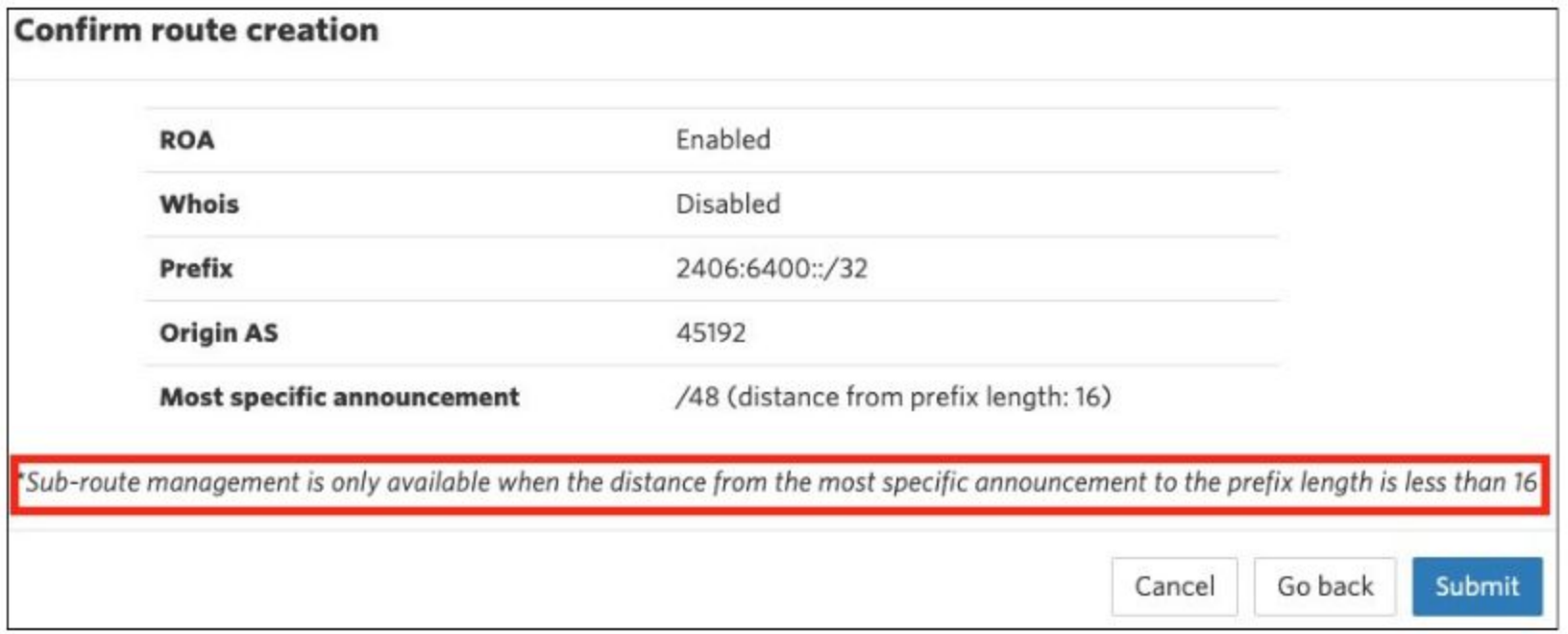


Create (publish) your ROA

- Example for **IPv6** below

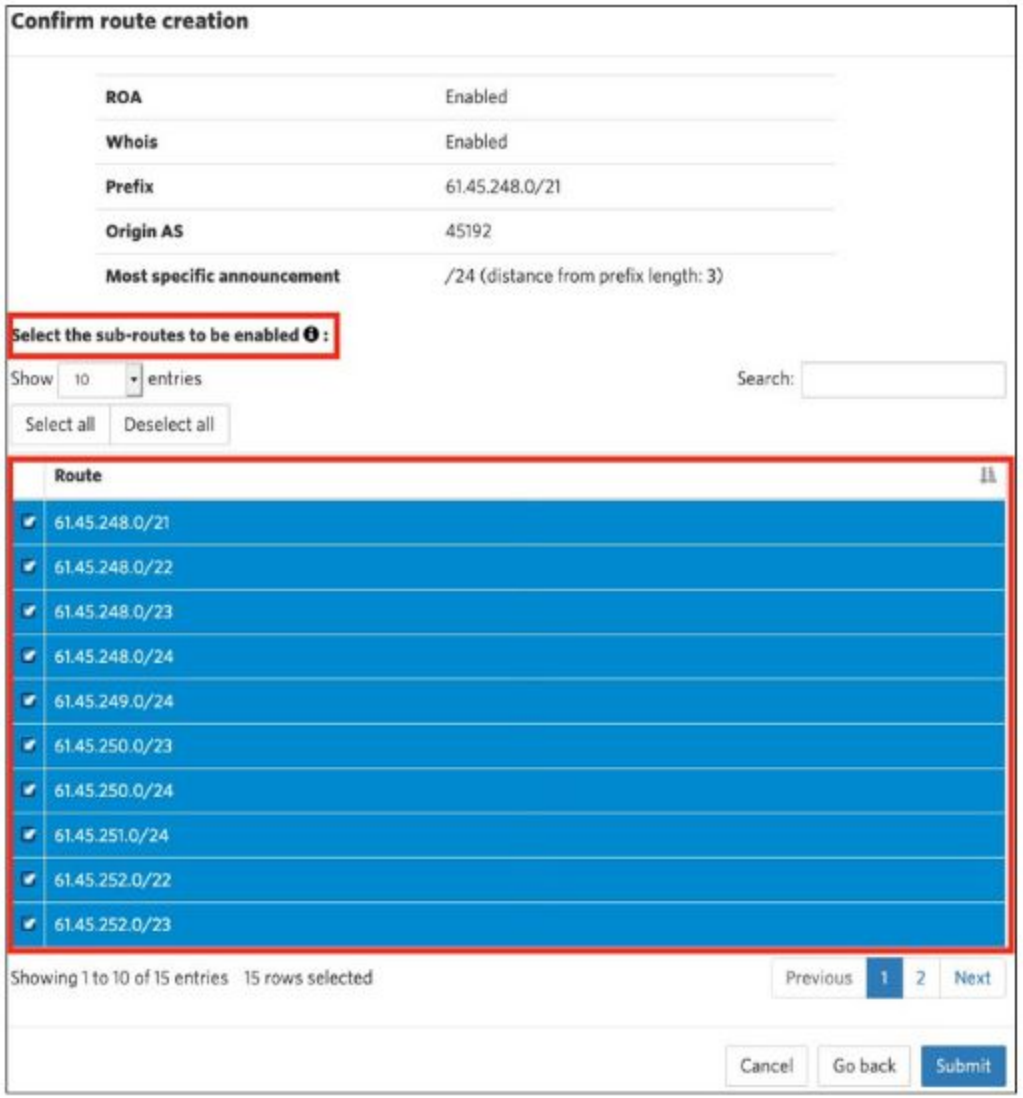


Create (publish) your ROA



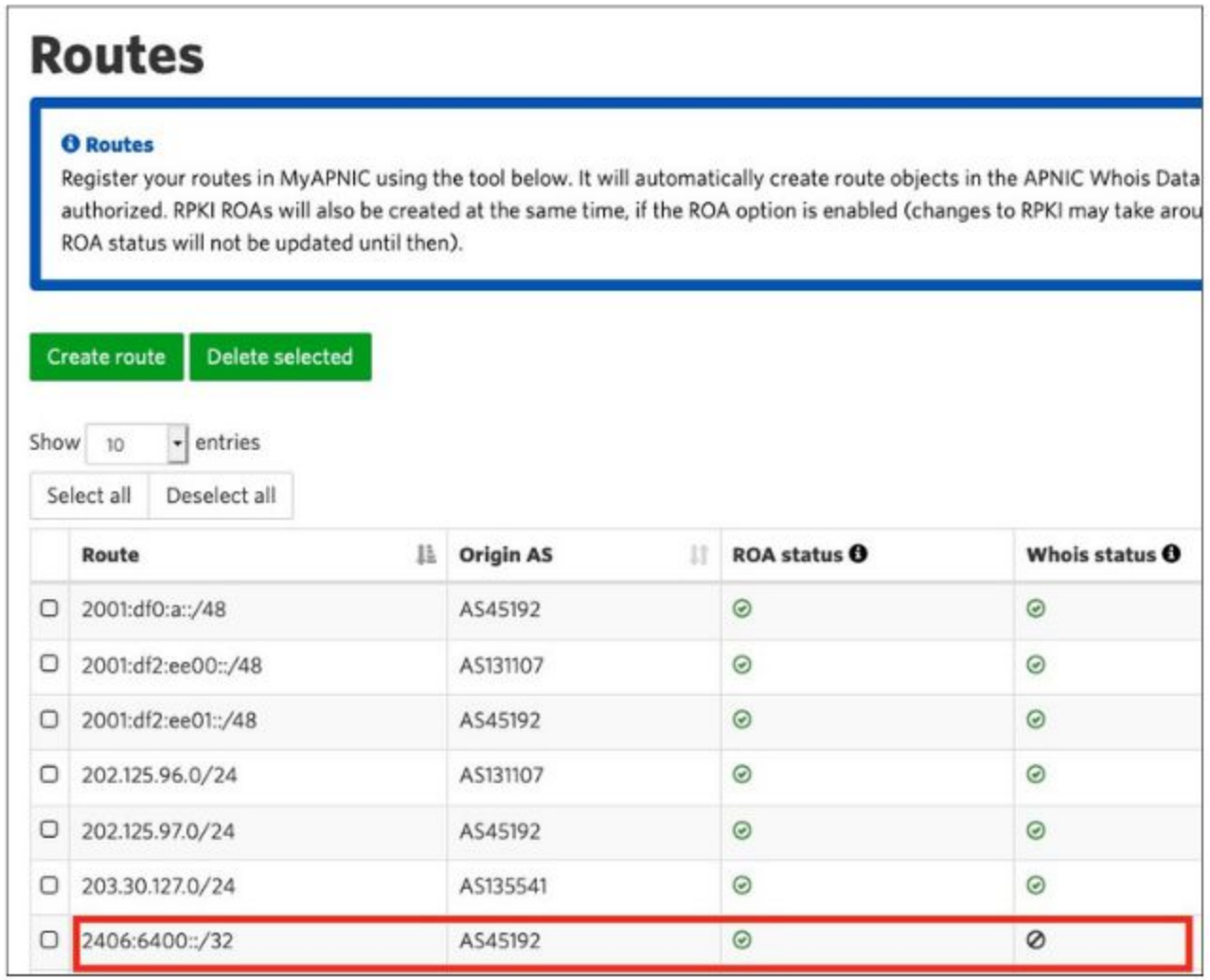
Create (publish) your ROA

- Example for **IPv4**



Create (publish) your ROA

- Your ROAs are ready!



Check your ROA

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

Validated ROAs

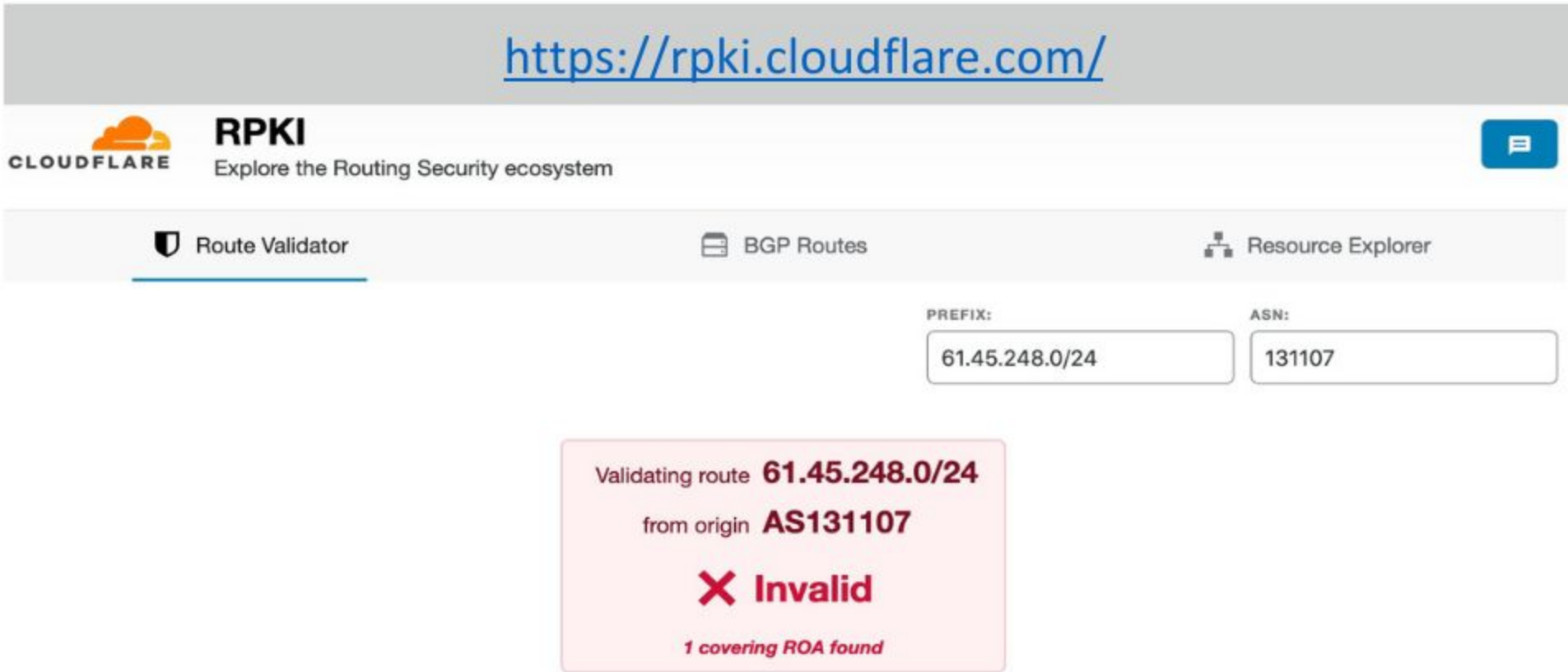
Show 10 entries

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

Demo: Using whois

```
apnic@group30:~$ whois -h whois.bgpmon.net " --roa 135533 61.45.248.0/24"
0 - Valid
------------------------
ROA Details
------------------------
Origin ASN:       AS135533
Not valid Before: 2020-12-04 01:50:09
Not valid After:  2025-07-30 00:00:00  Expires in 4y121d18h21m22.4000000059605s
Trust Anchor:     rpki.apnic.net
Prefixes:         61.45.248.0/24 (max length /24)
apnic@group30:~$ whois -h whois.bgpmon.net " --roa 135533 [redacted]:/48"
2 - Not Valid: Invalid Origin ASN, expected 131107
apnic@group30:~$ whois -h whois.bgpmon.net " --roa 131107 [redacted]3"
0 - Valid
------------------------
ROA Details
------------------------
Origin ASN:       AS131107
Not valid Before: 2021-02-23 03:30:07
Not valid After:  2025-07-30 00:00:00  Expires in 4y121d18h19m51.4000000059605s
Trust Anchor:     rpki.apnic.net
Prefixes:         [redacted] (max length /48)
                  202.125.96.0/24 (max length /24)
                  2406:6400::/32 (max length /36)
```

Using whois

- Check details about prefix or ASN
  - whois-h whois.apnic.net as135533
  - whois-h rr.ntt.net as135533
  - whois-h rr.ntt.net 61.45.248.0/24
- RPKI Check
  - VALID
    - whois -h whois.bgpmon.net " --roa 135533 61.45.248.0/24"
    - whois -h whois.bgpmon.net " --roa 131107 2406:6400::/32"
  - NOT VALID
    - whois -h whois.bgpmon.net " --roa 135533 61.45.248.0/22"
    - whois -h whois.bgpmon.net " --roa 135533 2406:6400::/48"

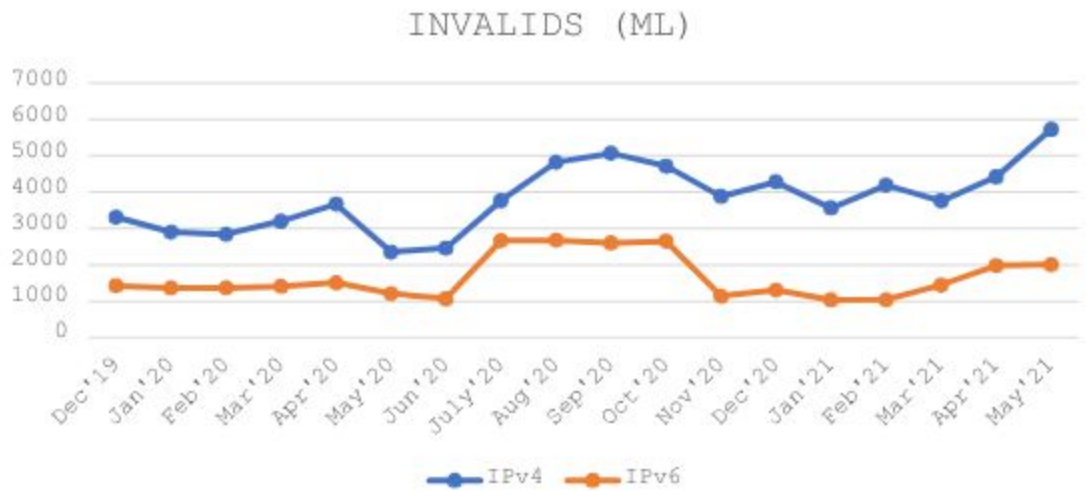
Check your ROA

```
# whois-h rr.ntt.net2001: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: maxLength48
origin: AS131107
mnt-by: MAINT-JOB
changed: job@ntt.net20180802
source: RPKI # Trust Anchor: APNIC RPKI Root

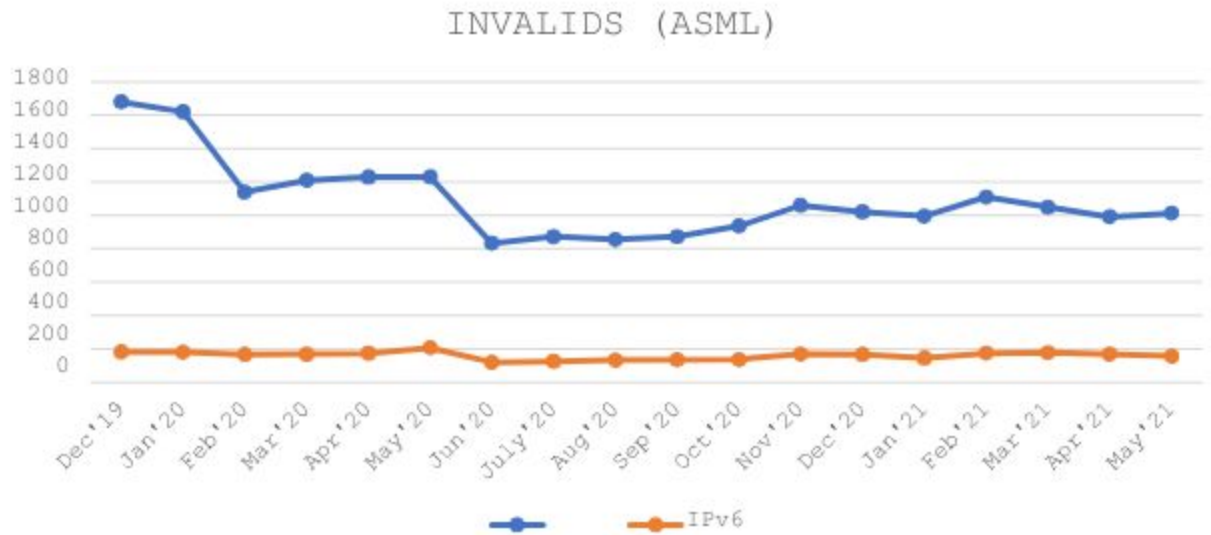
ROA Considerations

- 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



ROA Considerations

- Know your network
  - Do you have multiple ASes?
    - Are they independent ASes?
    - Transit AS + access/stub ASes?



https://blog.apnic.net/2020/04/10/rise-of-the-invalids/

2. Run (your own) RPKI Validator

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

2. Run (your own) RPKI Validator

- Options ~ we haven't tested:
  - rpki-client https://github.com/rpki-client
  - rpki-prover https://github.com/lolepezy/rpki-prover
  - rpstir2 -https://github.com/bgpsecurity/rpstir2

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)

Validator considerations

- VRP output (in)consistency

```
apnic@group01:~$ wc -l rout_sorted.csv
236787 rout_sorted.csv
apnic@group01:~$ wc -l fort_sorted.csv
236787 fort_sorted.csv
```

```
--- fort_sorted.csv     2021-04-13 05:15:09.313592360 +0000
+++ rout_sorted.csv     2021-04-13 05:01:16.434269134 +0000
@@ -144598,7 +144598,6 @@
 AS41937,109.198.0.0/19,24
 AS41937,109.72.48.0/20,24
 AS41937,109.94.228.0/22,24
-AS41937,178.237.216.0/21,21
 AS41937,178.250.136.0/21,24
 AS41937,185.10.140.0/22,24
 AS41937,185.10.142.0/23,24
@@ -206496,7 +206495,8 @@
 AS65530,2001:e68:2001:1::/64,64
 AS65530,202.188.100.0/24,24
 AS65530,202.188.95.0/24,24
-AS65530,2605:9cc0:384::/48,48
+AS65530,2605:9cc0:357::/48,48
+AS65530,2605:9cc0:382::/48,48
 AS65533,2001:718:ffff::/48,48
 AS65535,185.3.28.0/23,23
 AS65535,2001:67c:2b5c::/48,48
@@ -236784,4 +236784,4 @@
 AS9989,202.79.196.0/24,24
 AS999999,2001:678:d44:100::/64,64
 AS999999,2001:678:d44:2::/64,64
```

3. Router Configuration (IOS)

- Enable RTR on your routers
  - eBGPspeakers (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 bgp131107
    bgprpkiserver tcp<validatorIP>port < 323/8282/3323> refresh <secs>
```

Validation State

- Acting on the validation states
  - Tag & do nothing: You have downstream/route server @IXPs

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

  - RFC7115 –preference

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

  - Drop Invalids

```
IPv4    ~
7K   IPv6
~ 2K
```

Configuration (IOS)

- Policies based on validation:

```
route-map ROUTE-VALIDATION permit 10
match rpkivalid
 set local200eference
!
route-map ROUTE-VALIDATION permit 20
match rpkinot-found
 set local100eference
!
route-map ROUTE-VALIDATION permit 30
match rpkiinvalid
 set local50reference
!
```

OR

```
route-map ROUTE-VALIDATION deny 30
 match rpkinvalid
```

Configuration (IOS)

- Apply the route-map to inbound updates

```
router bgp131107
!---output omitted-----!
address-family ipv4
    bgpbestpathprefix-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
    bgpbestpathprefix-validate allow-invalid
neighbor X6:X6:X6:X6::151 activate
neighbor X6:X6:X6:X6::151 route-map   ROUTE -VALI DATION in
exit-address-family
!
```

3. 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

```
#shbgpipv6 unicast rpki?
  serversDisplay RPKI cache server information
  tableDisplay RPKI table entries


#shbgpipv4 unicast rpki?
  serversDisplay RPKI cache server information
  tableDisplay 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 MaxlenOrigin-ASSourceNeighbor
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 MaxlenOrigin-ASSourceNeighbor
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 StateFlaps Uptime #IPv4/IPv6 records X.X.X.46 Up 7509:20:59 40894/6747



>show validation session 202.125.96.46
Session StateFlaps Uptime #IPv4/IPv6 records X.X.X.46 Up 7509:21:18 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
```

RPKI Verification (JunOS)

- Can filter per origin ASN

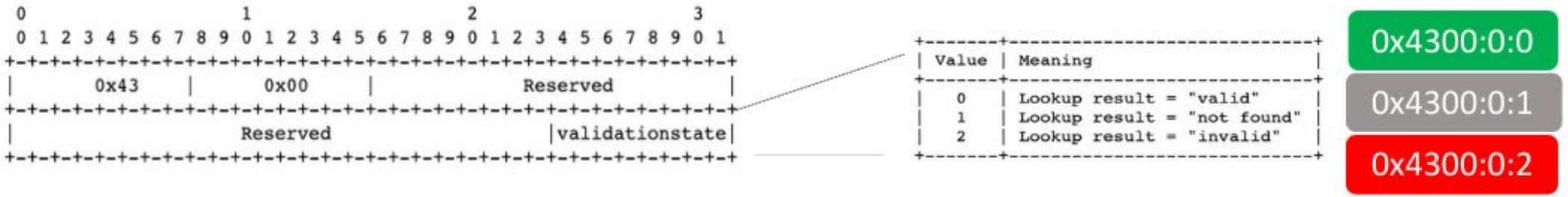
```
>show validation database origin-autonomous-system 45192
RV database for instance master


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
```

Propagating RPKI states to iBGPpeers

- 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 bgp131107
     bgprpkiserver 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 rpkistate
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 rpkistate
exit-address-family
!
```

Propagating RPKI states (IOS)

- Receiver (iBGPpeer)

```
router bgp131107
!---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 rpkistate
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 rpkistate
exit-address-family
!
```

  - If announce rpkistate is not configured for the neighbor, all prefixes received from the iBGPneighbor will be marked VALID!

Propagating RPKI states (JunOS)

- Sender (router with an RTR session)

```
ROUTE-VALIDATION
policy-statement {
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 (iBGPpeer)

```
 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

- VRFs?
  - Know your platform
    - RPKI (RTR) supported on VRF instances? or
    - just the global table?

Operational Considerations

- iBGP state propagation ~ vendor interop?
  - Ex: IOS propagating states to JunOSpeers

```
unknown iana4300
```

  - *Options (hack ~ 17.4R3, 18.2R3, 18.4R2):*
    - *Act on the states at the border, OR*
    - *Tag/match with custom (standard) communities*

Other developments

- ROA with `AS0` origin `(RFC6483/RFC7607)`
  - Negative attestation
    - No valid ASN has been granted authority
    - Not to be routed (Eg: IXP Peering LAN prefixes)
  - Overridden by another ROA (*with an origin AS other than AS0*)
  - *APNIC's RPKI backend supported this since Nov 2018*

Other developments

- Prop-132based AS0ROA
  - APNIC is directed to publish an AS0 ROA for undelegated and unassigned APNIC space
    - ~ comparable to *RFC6491* for special use/reserved/unallocated IANA space
  - *APNIC implemented on 2 Sept 2020*
    - *Separate TAL ~ opt-in* (the main RPKI TAL is included in all RPs)

    https://blog.apnic.net/2020/09/02/policy-prop-132-as0-for-unallocated-space-deployed-in-service/
  - Process:
    - *"fast to remove"* (within 5mins of delegation)
    - *"slow to add"* (undelegated/reclaimed resources added in a cron-job)

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

Any questions?