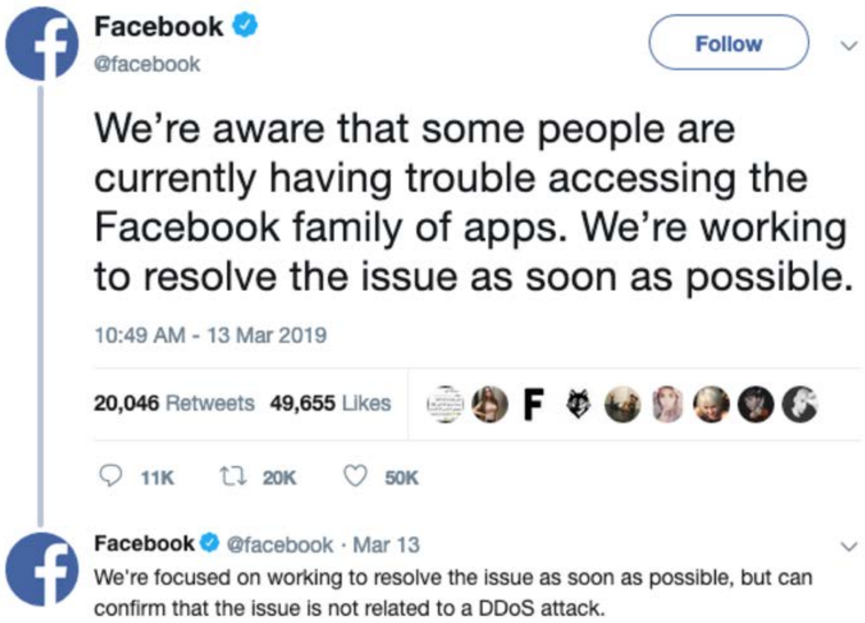
APNIC

RPKI Deployment

Recent - Fat-finger/Hijacks/Leaks

- (Possible) Facebook prefix leaks – **March 2019**
  - FB family of apps not available in Europe (13 March)
    - Potentially a European ISP leaked it to a major transit provider, who propagated it to its peers (and downstream)



Facebook @facebook

Follow

We're aware that some people are currently having trouble accessing the Facebook family of apps. We're working to resolve the issue as soon as possible.

10:49 AM - 13 Mar 2019

20,046 Retweets 49,655 Likes

11K 20K 50K

Facebook @facebook · Mar 13

We're focused on working to resolve the issue as soon as possible, but can confirm that the issue is not related to a DDoS attack.

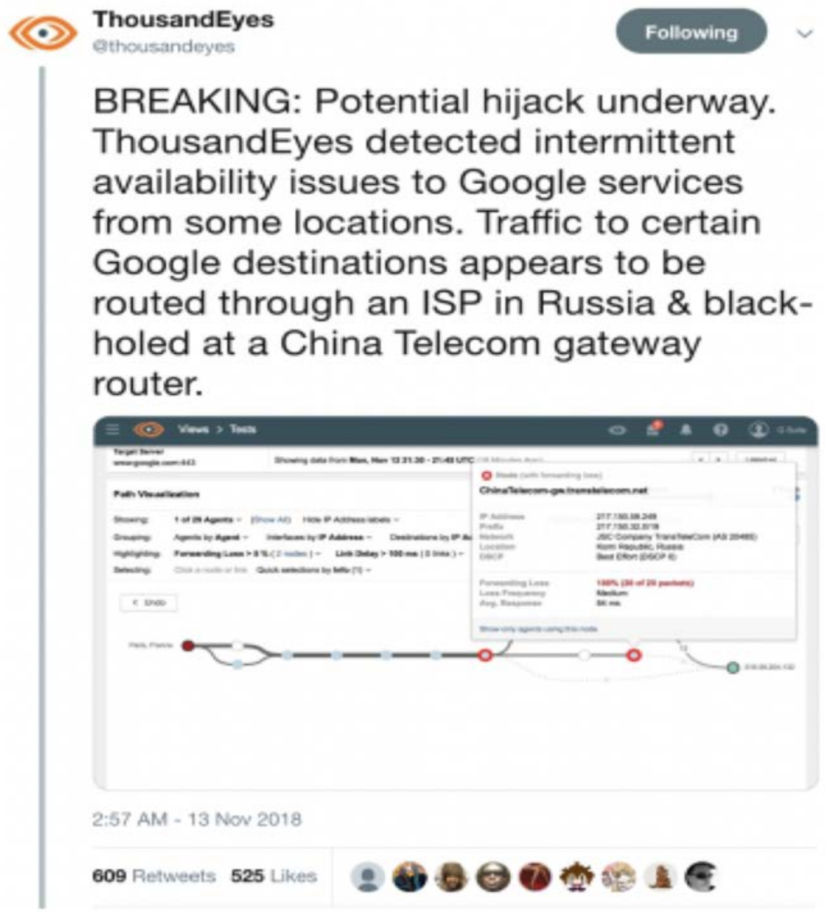
Update: Facebook, Instagram and Messenger were down for many users

Jonathan Shieber @jshieber / 2 weeks ago

Commen

Recent - Fat-finger/Hijacks/Leaks

- 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



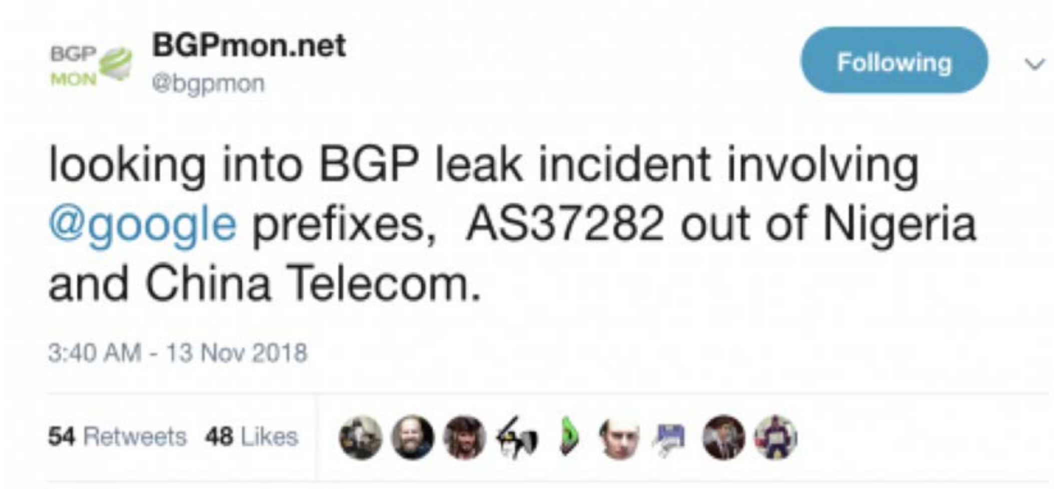
ThousandEyes
@thousandeyes

Following

BREAKING: Potential hijack underway. ThousandEyes detected intermittent availability issues to Google services from some locations. Traffic to certain Google destinations appears to be routed through an ISP in Russia & black-holed at a China Telecom gateway router.

2:57 AM - 13 Nov 2018

609 Retweets 525 Likes



BGPmon.net
@bgpmon

Following

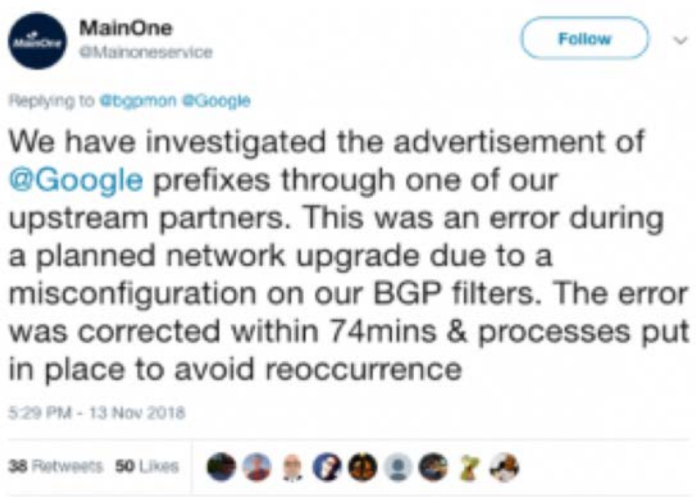
looking into BGP leak incident involving @google prefixes, AS37282 out of Nigeria and China Telecom.

3:40 AM - 13 Nov 2018

54 Retweets 48 Likes

Recent - Fat-finger/Hijacks/Leaks

- Google prefix leaks (contd…)
  - ❑ How did it happen?
    - ▪ AS37282 (MainOne) leaked Google prefixes to AS4809 (CT) at IXPN, who leaked it to other transit providers like AS20485 (TransTelecom)



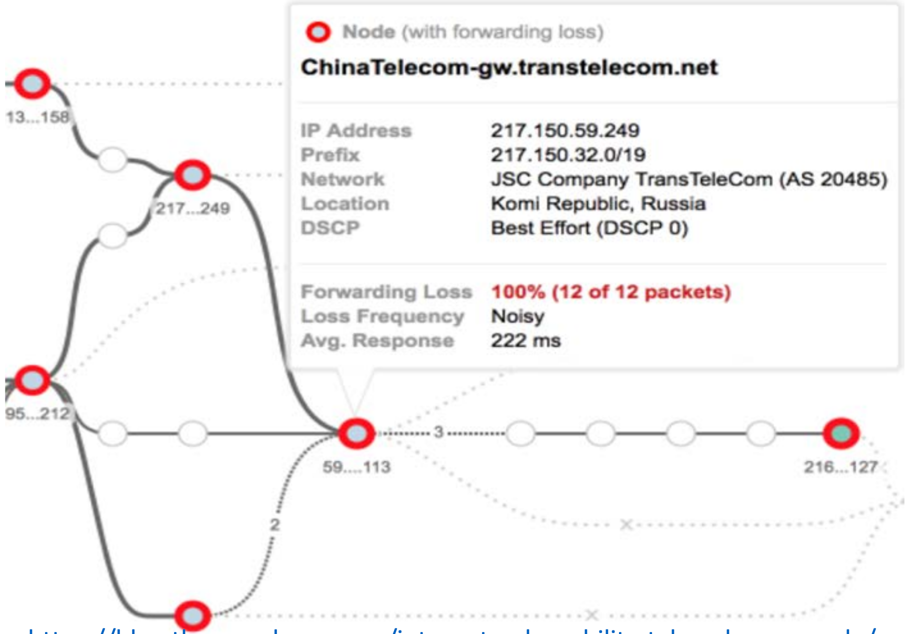
MainOne
@Mainoneservice

Replying to @bgpmon @Google

We have investigated the advertisement of @Google prefixes through one of our upstream partners. This was an error during a planned network upgrade due to a misconfiguration on our BGP filters. The error was corrected within 74mins & processes put in place to avoid reoccurrence

5:29 PM - 13 Nov 2018

38 Retweets 50 Likes



Node (with forwarding loss)

ChinaTelecom-gw.transtelecom.net

IP Address 217.150.59.249
Prefix 217.150.32.0/19
Network JSC Company TransTeleCom (AS 20485)
Location Komi Republic, Russia
DSCP Best Effort (DSCP 0)

Forwarding Loss 100% (12 of 12 packets)
Loss Frequency Noisy
Avg. Response 222 ms

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

Recent - Fat-finger/Hijacks/Leaks

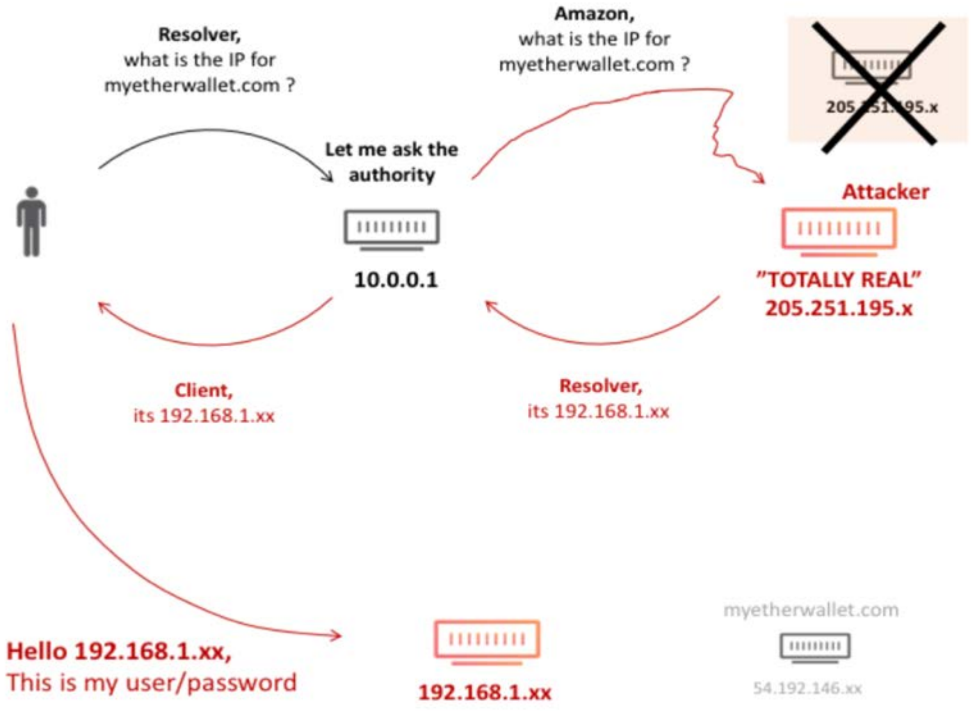
- Amazon (AS16509) Route53 hijack – **April2018**
  - ❑ AS10279 (eNET) originated more specifics (/24s) of Amazon Route53's prefix (205.251.192.0/21)

    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

Recent - Fat-finger/Hijacks/Leaks

- 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

Recent - Fat-finger/Hijacks/Leaks

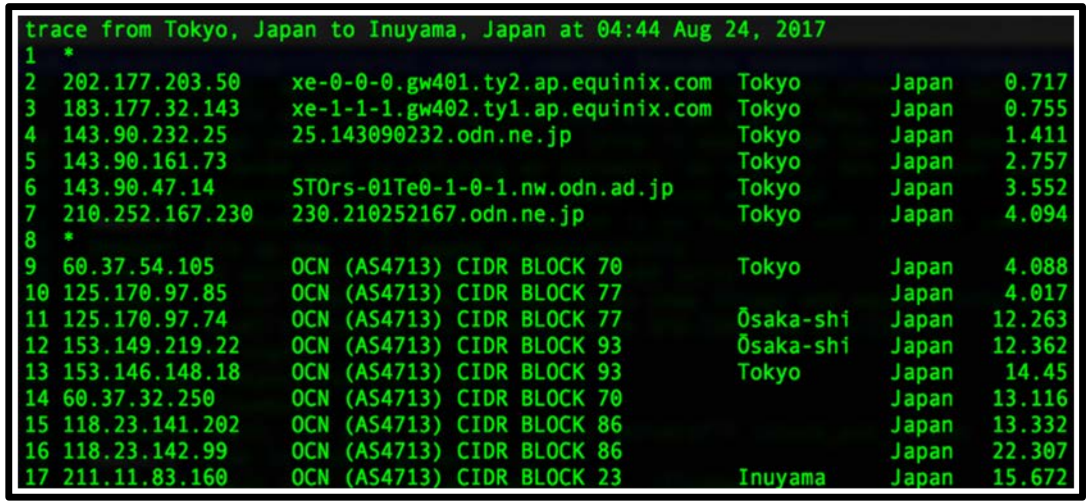
- ~~Bharti (AS9498) originates 103.0.0.0/10 - **Dec 2017**~~
  - ~~~ 2 days~~
  - ~~No damage done – more than 8K specific routes!~~

- 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!

Recent - Fat-finger/Hijacks/Leaks

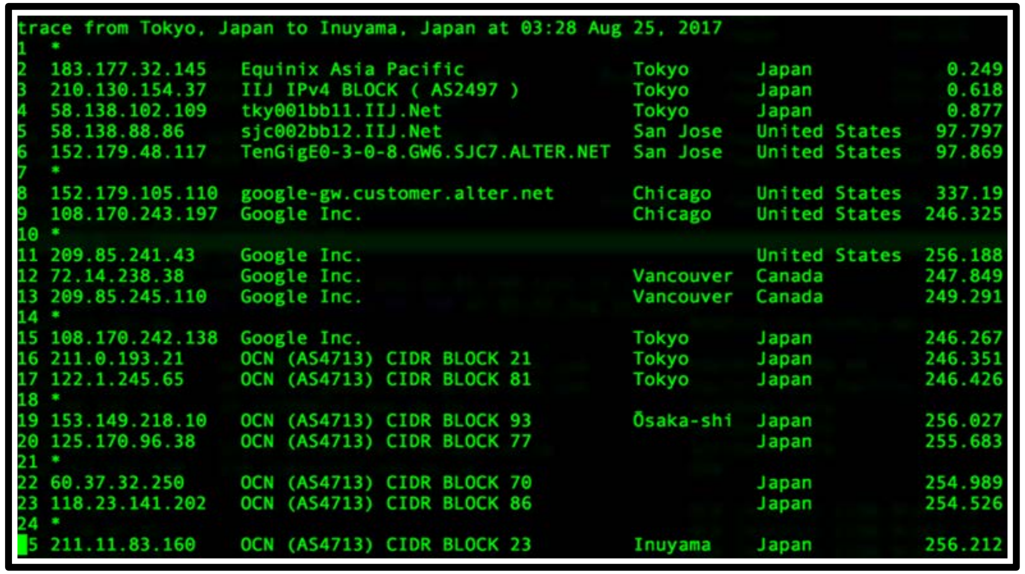
- Google leak (contd…)



```
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
```

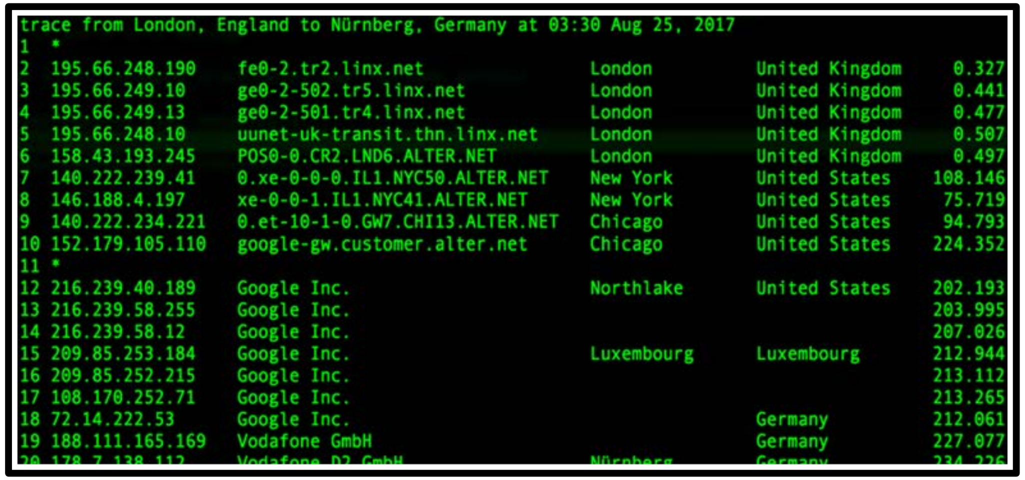
Before leak (JP->JP)

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/

Fat-finger/Hijacks/Leaks

- YouTube (AS36561) Incident - **Feb 2008**
  - ~ 2 hours
  - AS17557 (PT) 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
    - Decentralised distributed environment
  - Meaning no reference point for what’s right in routing
  - Which means, no clear way of knowing what is wrong

Why do we keep seeing these?

- Routing works by RUMOUR
  - Self learning routing protocols that does topology discovery
    - Tell what you know to your neighbors, and
    - Learn what your neighbors know
  - Assume everyone is correct (and *honest*)
  - Makes it difficult to determine if a rumour is incorrect
    - 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
  - ❑ Which means, no reference outcome to compare the local view ☹
- It is NOT deterministic
  - ❑ Does not always generate the same outcomes for the same inputs

Why do we keep seeing these?

- Routing in reality is a NEGOTIATION
  - Does two things:
    - Topology discovery
    - Policy negotiation *(~traffic engineering*)
  - Policy is a negotiation
    - I have import preferences
    - You have export preferences

Why do we keep seeing these?

- Routing works in REVERSE
  - Outbound advertisement affects inbound traffic
    - I can announce your prefix, get traffic that used to go to you to come to me
  - Inbound (*Accepted*) advertisement influence outbound traffic
    - You could announce someone’s prefix to me, and make me send, traffic that used to go to them, to you

Why do we keep seeing these?

- And as always, there is no E-bit (evil!)
  - A bad routing update does not identify itself as BAD
  - All we can do is identify GOOD updates
    - So, if we identify what’s good, rest is bad? ;-)

  - But how do we identify what is GOOD???

Why should we worry?

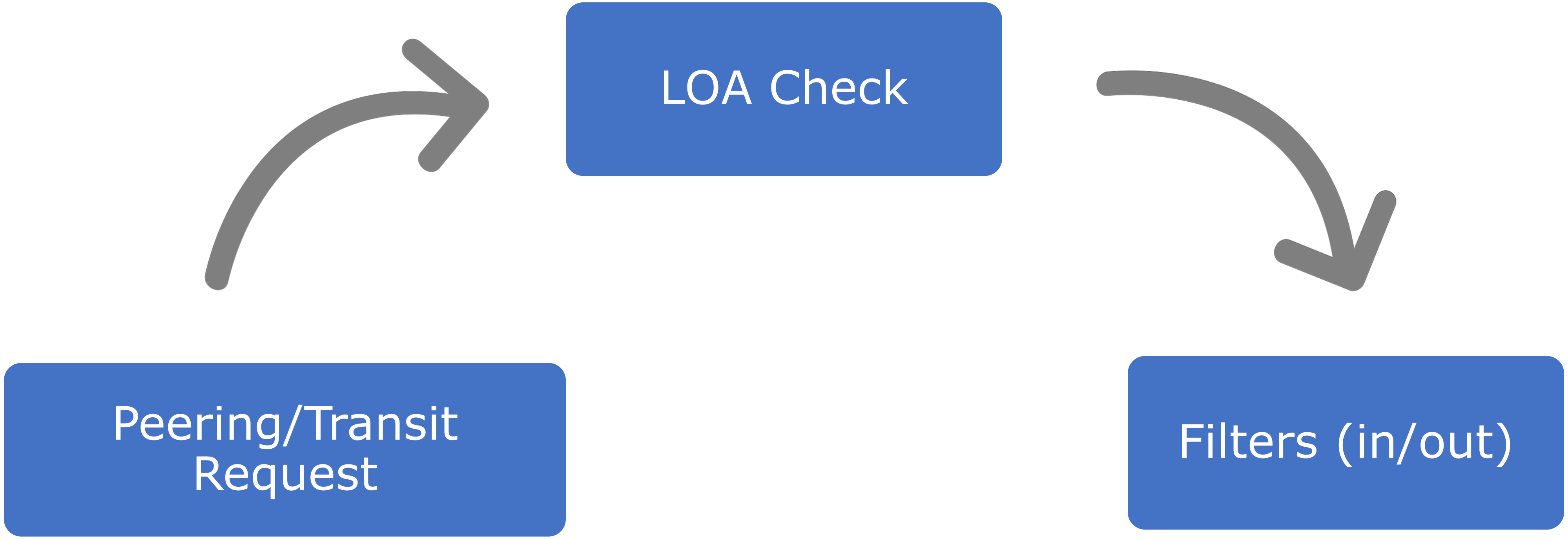
- Because it’s just so easy to do bad in routing!

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

How do we address these?

- **Filtering!**
  - ❑ Filters with your peers, upstream(s) and customers
    - AS Path filters
    - Prefix filters
    - Maximum Prefix limit

Current practice

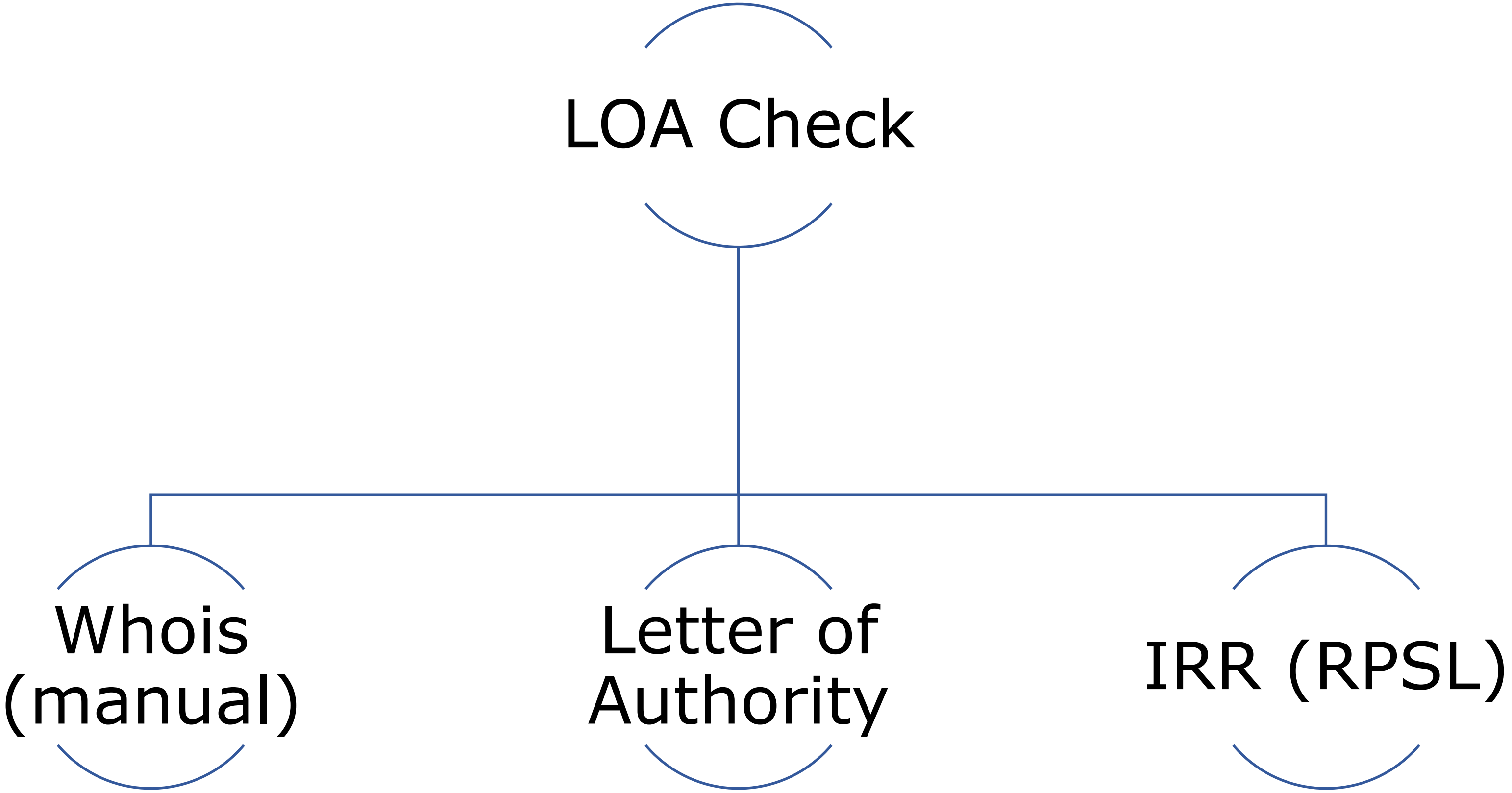


Peering/Transit Request

LOA Check

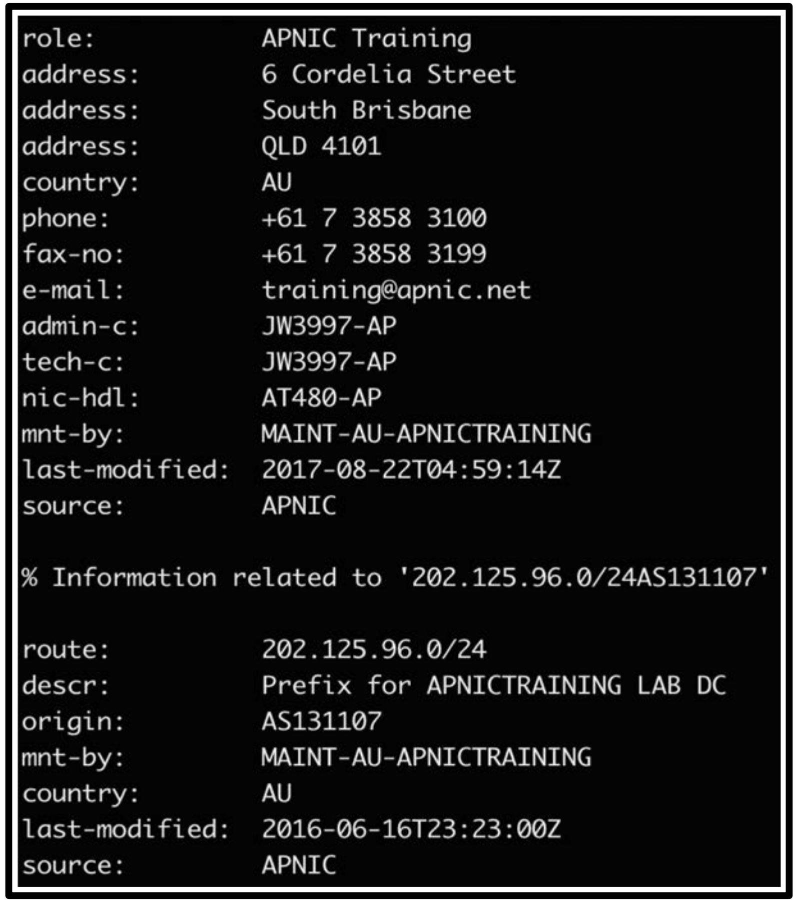
Filters (in/out)

Tools & Techniques



Tools & Techniques

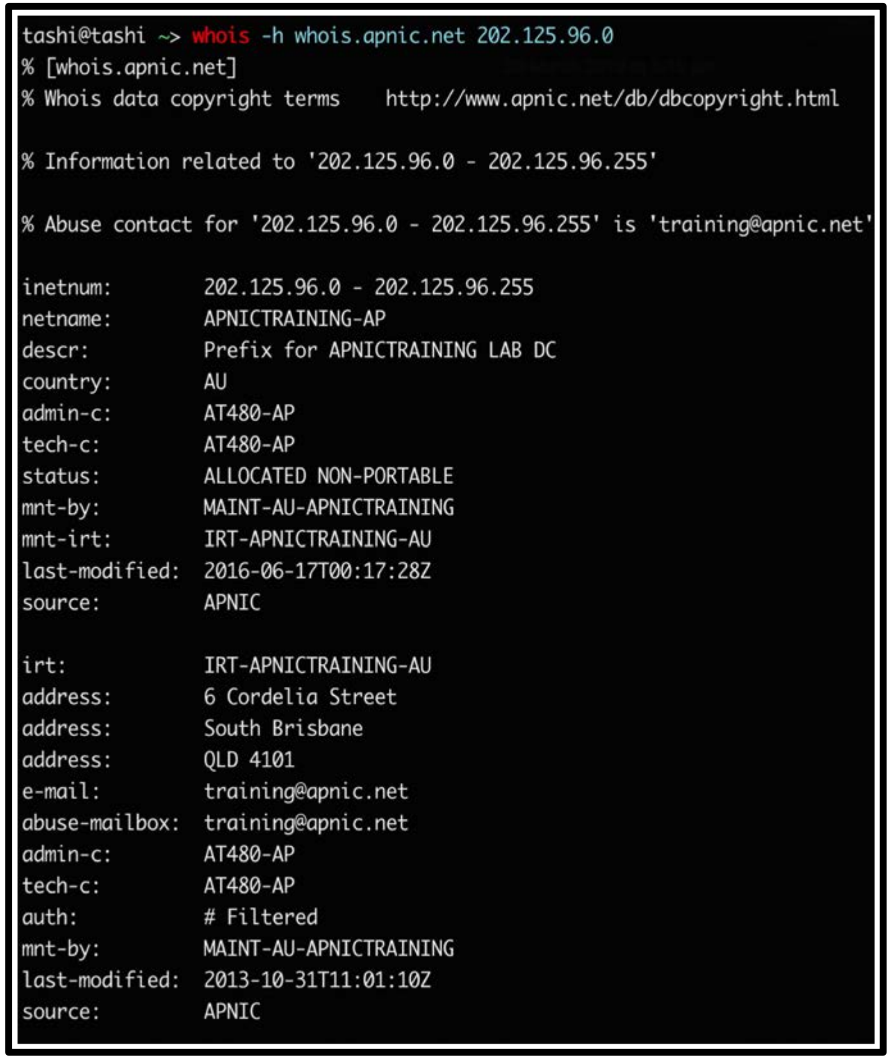
- Look up **whois**
  - verify holder of a resource



```
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
```



```
tashi@tashi ~> 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'

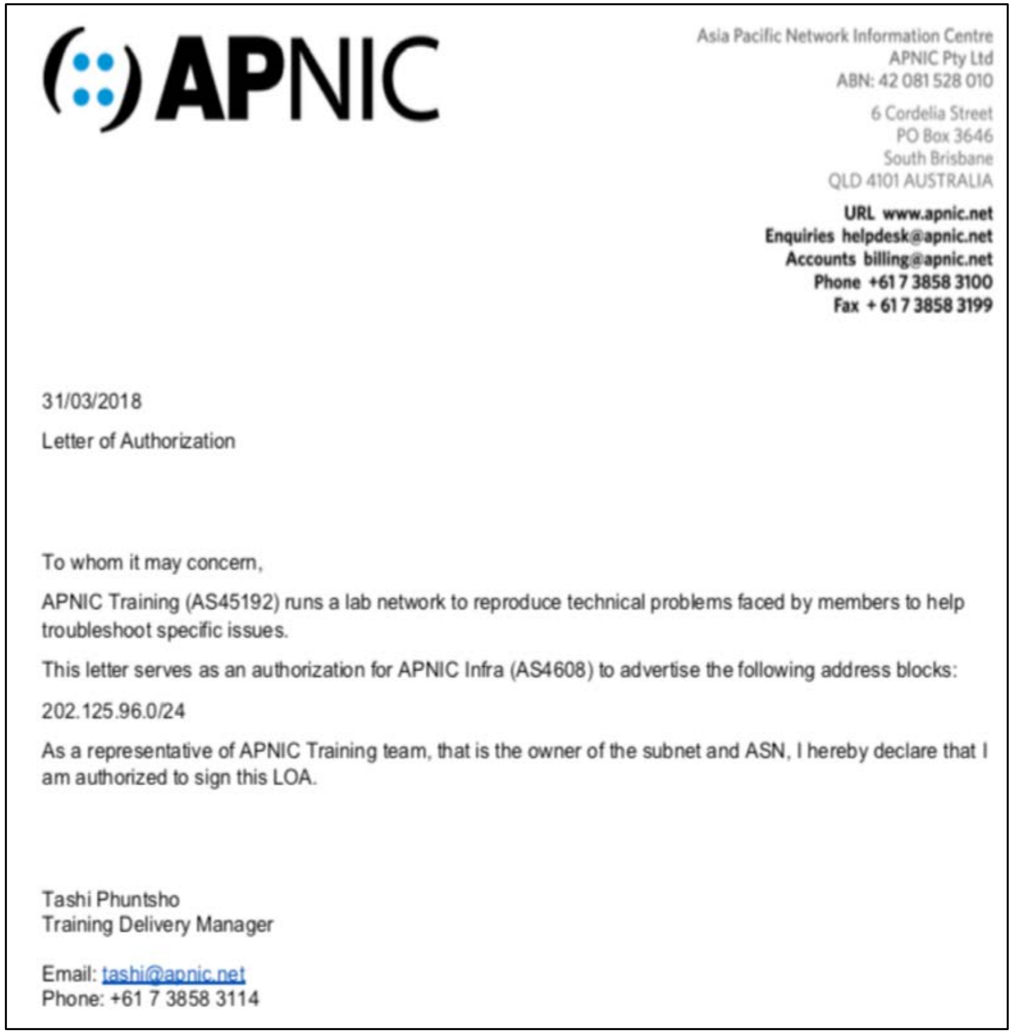
% Abuse contact for '202.125.96.0 - 202.125.96.255' is 'training@apnic.net'

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
```

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

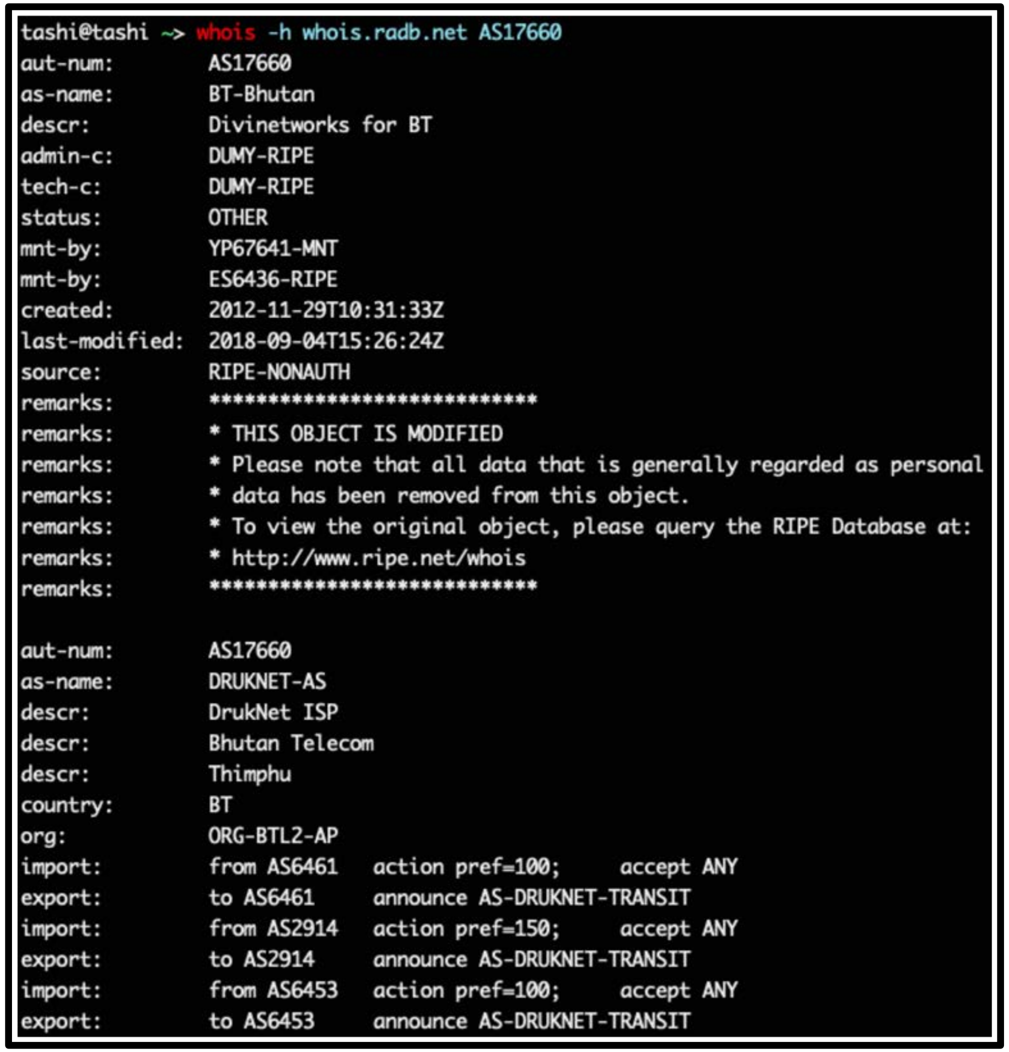
As a representative of APNIC Training team, that is the owner of the subnet and ASN, I hereby declare that I am authorized to sign this LOA.

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

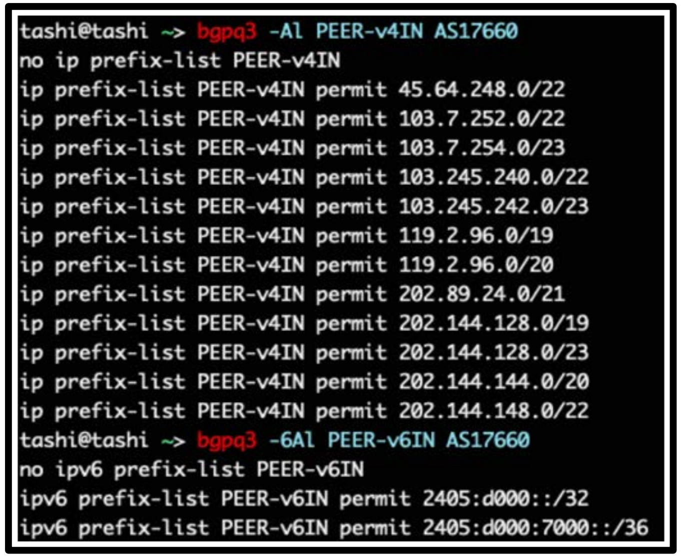


```
tashi@tashi ~> whois -h whois.radb.net AS17660
aut-num:        AS17660
as-name:        BT-Bhutan
descr:          Divinetworks for BT
admin-c:        DUMY-RIPE
tech-c:         DUMY-RIPE
status:         OTHER
mnt-by:         YP67641-MNT
mnt-by:         ES6436-RIPE
created:        2012-11-29T10:31:33Z
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:        ****************************

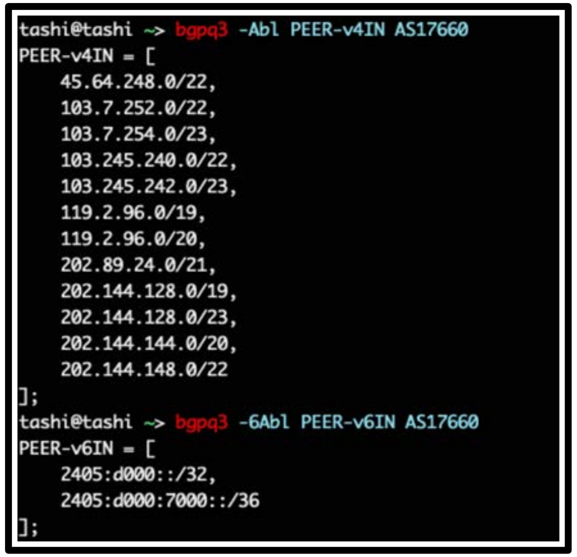
aut-num:        AS17660
as-name:        DRUKNET-AS
descr:          DrukNet ISP
descr:          Bhutan Telecom
descr:          Thimphu
country:        BT
org:            ORG-BTL2-AP
import:         from AS6461   action pref=100;     accept ANY
export:         to AS6461     announce AS-DRUKNET-TRANSIT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

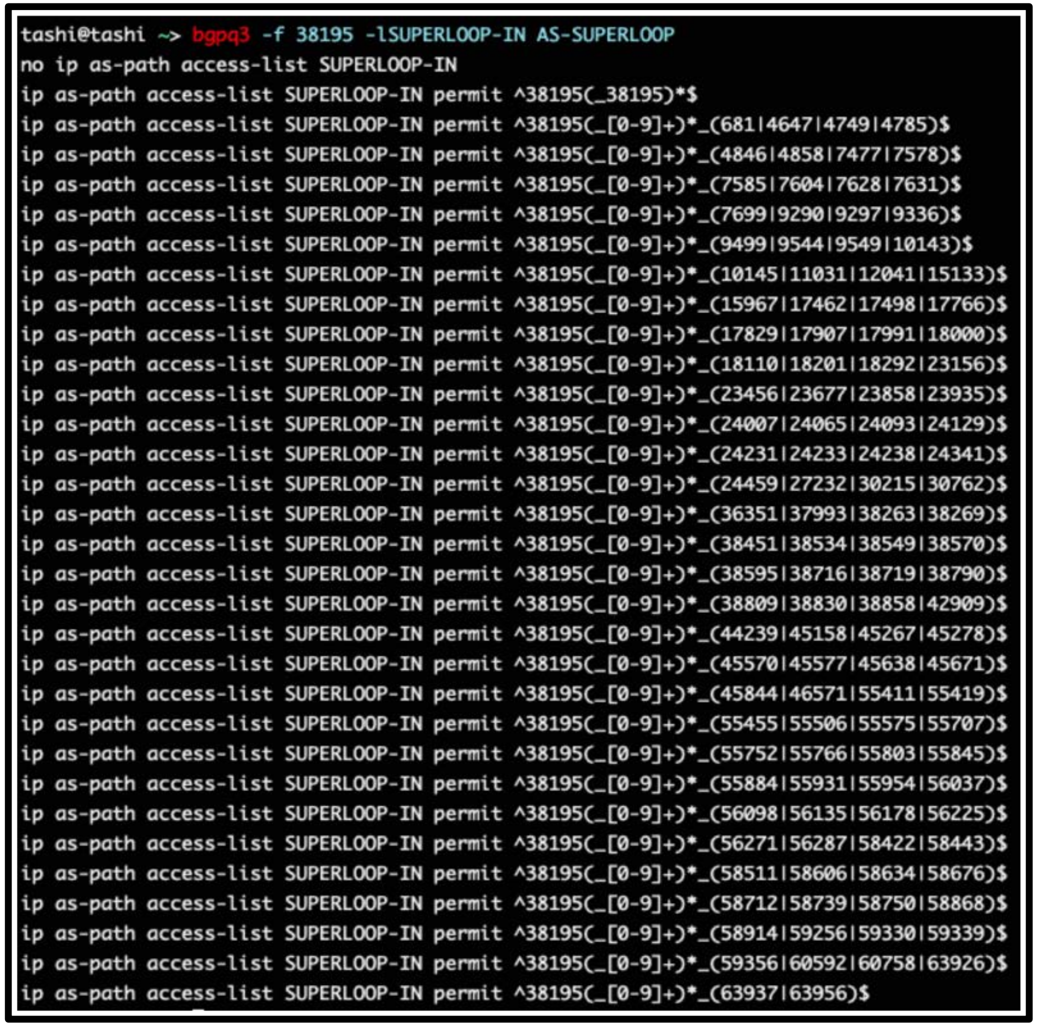
- IRR
  - *Helps auto generate network (prefix/as-path) filters using RPSL tools*
    - Filter out route advertisements not described in the registry



```
tashi@tashi ~> bgpq3 -Al PEER-v4IN AS17660
no ip prefix-list PEER-v4IN
ip prefix-list PEER-v4IN permit 45.64.248.0/22
ip prefix-list PEER-v4IN permit 103.7.252.0/22
ip prefix-list PEER-v4IN permit 103.7.254.0/23
ip prefix-list PEER-v4IN permit 103.245.240.0/22
ip prefix-list PEER-v4IN permit 103.245.242.0/23
ip prefix-list PEER-v4IN permit 119.2.96.0/19
ip prefix-list PEER-v4IN permit 119.2.96.0/20
ip prefix-list PEER-v4IN permit 202.89.24.0/21
ip prefix-list PEER-v4IN permit 202.144.128.0/19
ip prefix-list PEER-v4IN permit 202.144.128.0/23
ip prefix-list PEER-v4IN permit 202.144.144.0/20
ip prefix-list PEER-v4IN permit 202.144.148.0/22
tashi@tashi ~> bgpq3 -6Al PEER-v6IN AS17660
no ipv6 prefix-list PEER-v6IN
ipv6 prefix-list PEER-v6IN permit 2405:d000::/32
ipv6 prefix-list PEER-v6IN permit 2405:d000:7000::/36
```



```
tashi@tashi ~> bgpq3 -Abl PEER-v4IN AS17660
PEER-v4IN = [
    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
];
tashi@tashi ~> bgpq3 -6Abl PEER-v6IN AS17660
PEER-v6IN = [
    2405:d000::/32,
    2405:d000:7000::/36
];
```



```
tashi@tashi ~> bgpq3 -f 38195 -lSUPERLOOP-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]+)*_(4846|4858|7477|7578)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7585|7604|7628|7631)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7699|9290|9297|9336)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9499|9544|9549|10143)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(10145|11031|12041|15133)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(15967|17462|17498|17766)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17829|17907|17991|18000)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(18110|18201|18292|23156)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23456|23677|23858|23935)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24007|24065|24093|24129)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24231|24233|24238|24341)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24459|27232|30215|30762)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(36351|37993|38263|38269)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38451|38534|38549|38570)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38595|38716|38719|38790)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38809|38830|38858|42909)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(44239|45158|45267|45278)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(45570|45577|45638|45671)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(45844|46571|55411|55419)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(55455|55506|55575|55707)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(55752|55766|55803|55845)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(55884|55931|55954|56037)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(56098|56135|56178|56225)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(56271|56287|58422|58443)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(58511|58606|58634|58676)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(58712|58739|58750|58868)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(58914|59256|59330|59339)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(59356|60592|60758|63926)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(63937|63956)$
```

Tools & Techniques

- Problem(s) with IRR
  - No single authority model
    - How do I know if a RR entry is genuine and correct?
    - Is it maintained by the authoritative owner of the resource?
    - 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
  - ❑ Small mistakes could have big impacts
    - check your outputs before committing

Back to basics – identify GOOD

- Could we use a digital signature to convey the "*authority to use*"?
  - Using a private key to *sign* the *authority*, and
  - the public key to *validate* the *authority*

- The idea being:
  - If the holder of the resource has the private key, it can sign/authorize the use of the resource

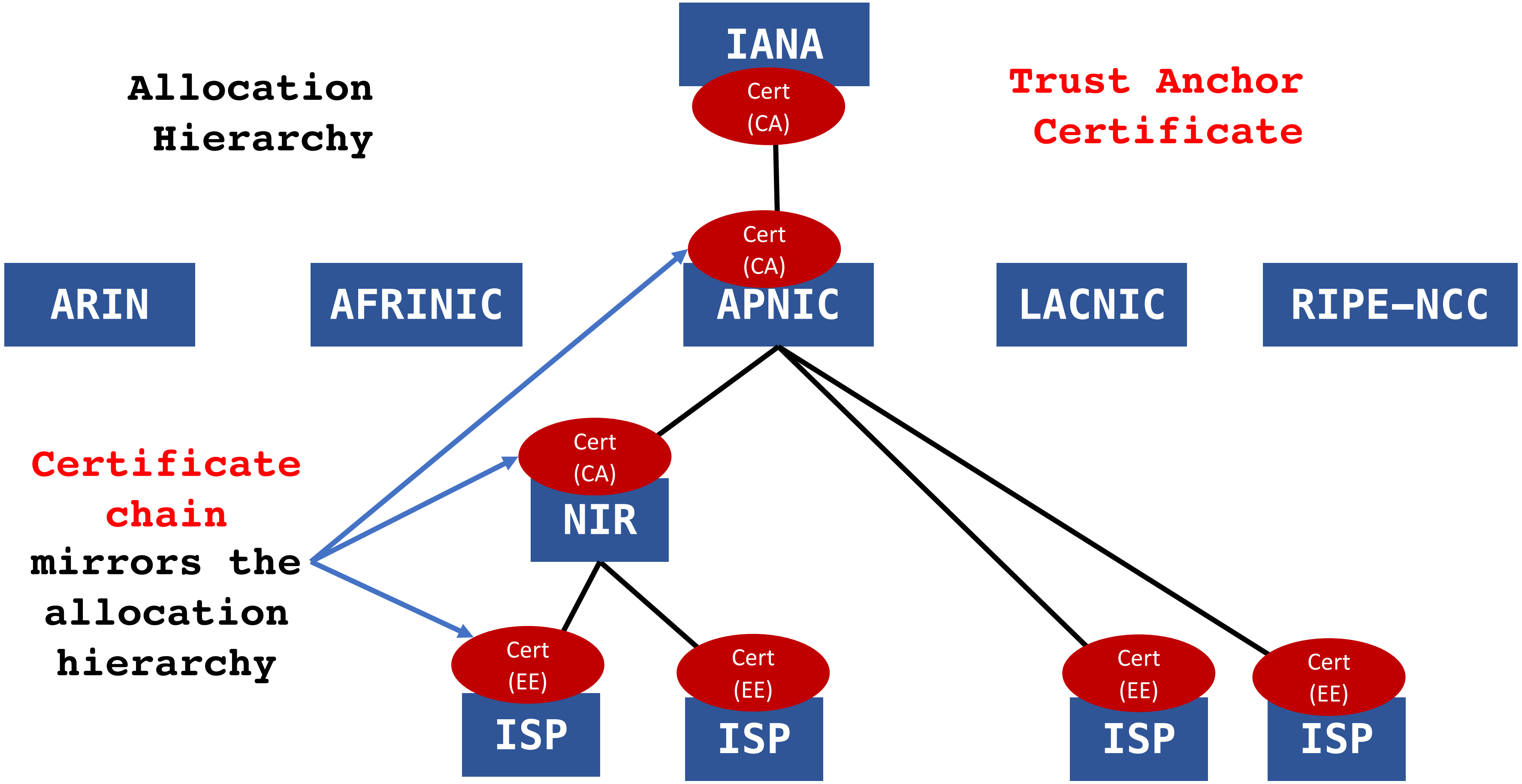
How about trust?

- 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



Allocation Hierarchy

Trust Anchor Certificate

Certificate chain mirrors the allocation hierarchy

Image 4

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

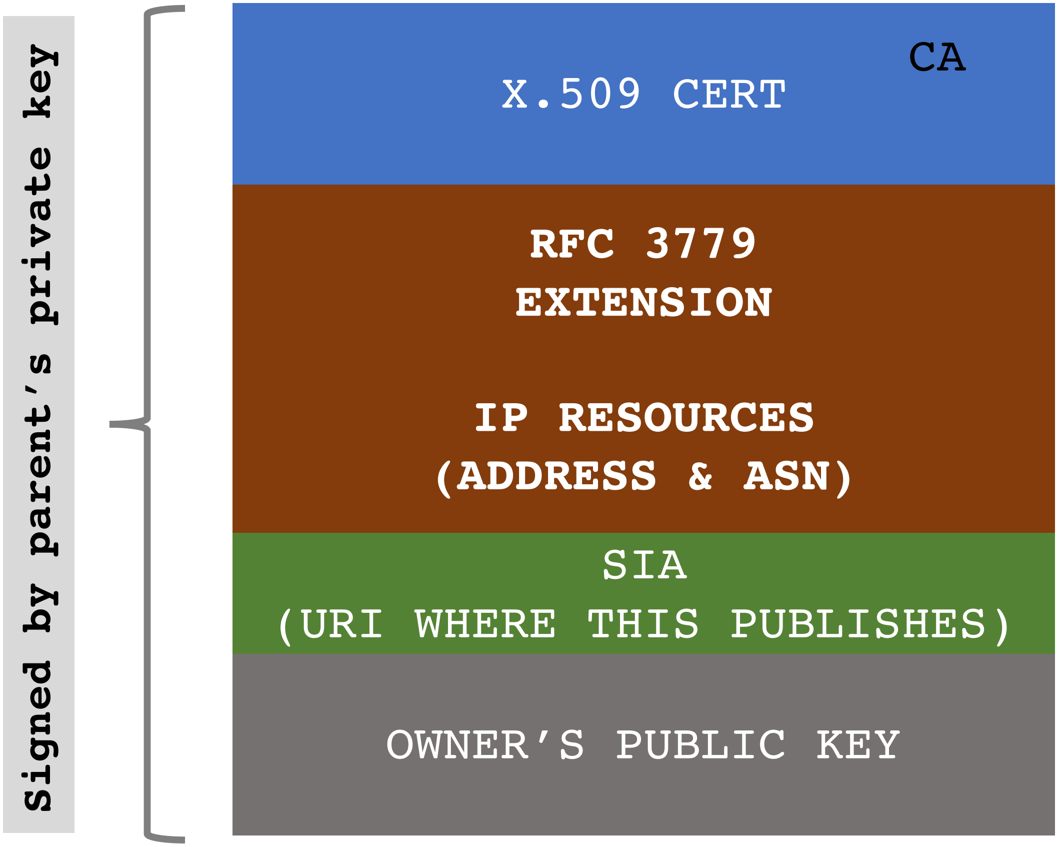
- Any attestations signed by the end-holder’s private key, can now be validated up the chain of trust

X.509 Certificates recap (RFC5280)

- Associates a public key with an individual or an organization

VERSION	Version of X.509
SERIAL NUMBER	Uniquely identifies the certificate
SIGNATURE ALGORITHM	Algorithms used by the CA to sign the cert
ISSUER NAME	Id of the CA (that issued the cert)
VALIDITY PERIOD	Cert validity
SUBJECT NAME	Entity associated with the public key
SUBJECT PUBLIC KEY	Owner’s public key
EXTENSIONS (ISSUER KEY ID)	Identify the pub key of issuer of the cert
EXTENSIONS (SUBJECT KEY ID)	Extra info (owner of the cert)
EXTENSIONS (CRL)	Extensions (CRL)
CA DIGITAL SIGNATURE	Certifies the binding between the pub key & subject of the cert

RPKI profile ~ Resource Certificates


Signed by parent's private key
X.509 CERT
CA
RFC 3779 EXTENSION
IP RESOURCES (ADDRESS & ASN)
SIA (URI WHERE THIS PUBLISHES)
OWNER'S PUBLIC KEY

- 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 public key certificate (resource certificate) that binds the allocated address with **B's** public key, all signed by **A's** (certification authority) 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)

- The resource holder (B) can now sign *attestations (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 into the BGP routing system, and **sign** this permission with its private key (ROA)

Route Origin Authorization (ROA)

- Digitally signed object
  - 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*

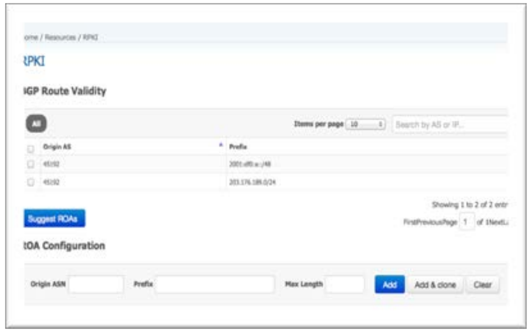
What can RPKI do?

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

- Hence, can help:
  - ❑ prevent **route hijacks**
    - ▪ A prefix originated by an AS without authorization
  - ❑ prevent **mis-origination**
    - ▪ A prefix that is mistakenly originated by an AS which does not own it

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



ome / Resources / RPKI

RPKI

IGP Route Validity

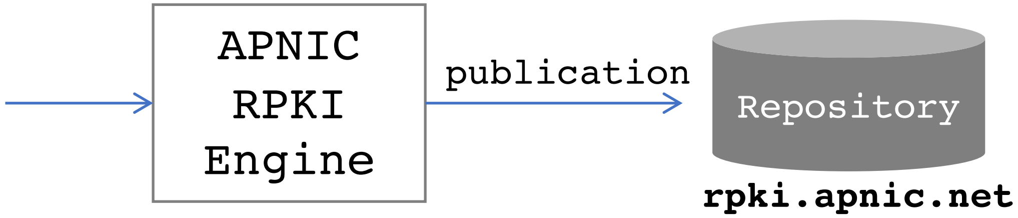
□	Origin AS	Prefix
□	45192	2001:df0:a::/48
□	45192	203.176.189.0/24

Showing 1 to 2 of 2 entr

OA Configuration

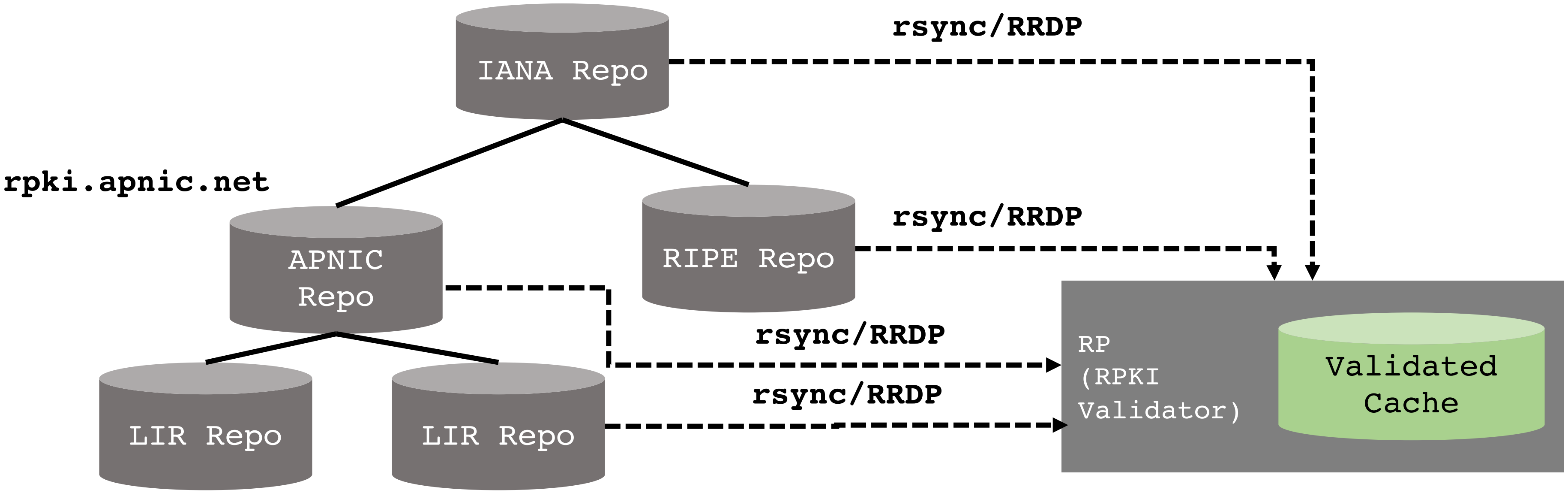
Origin ASN □ Prefix □ Max Length □

MyAPNIC GUI



RPKI Components

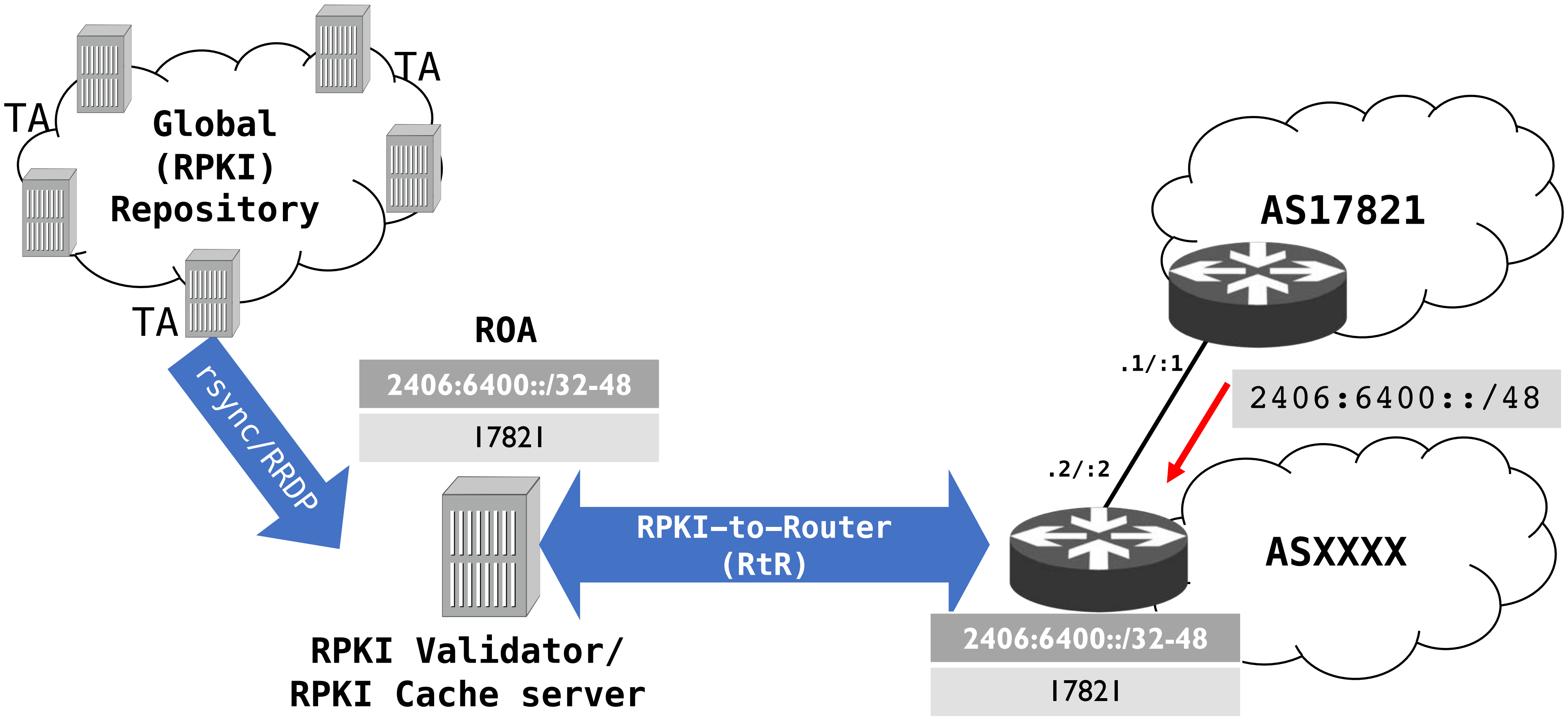
- **Relying Party (RP)**
  - ❑ RPKI Validator tool that gathers data (ROA) from the distributed RPKI repositories
  - ❑ Validates each entry's signature against the TA to build a "*Validated cache*"



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

- 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 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

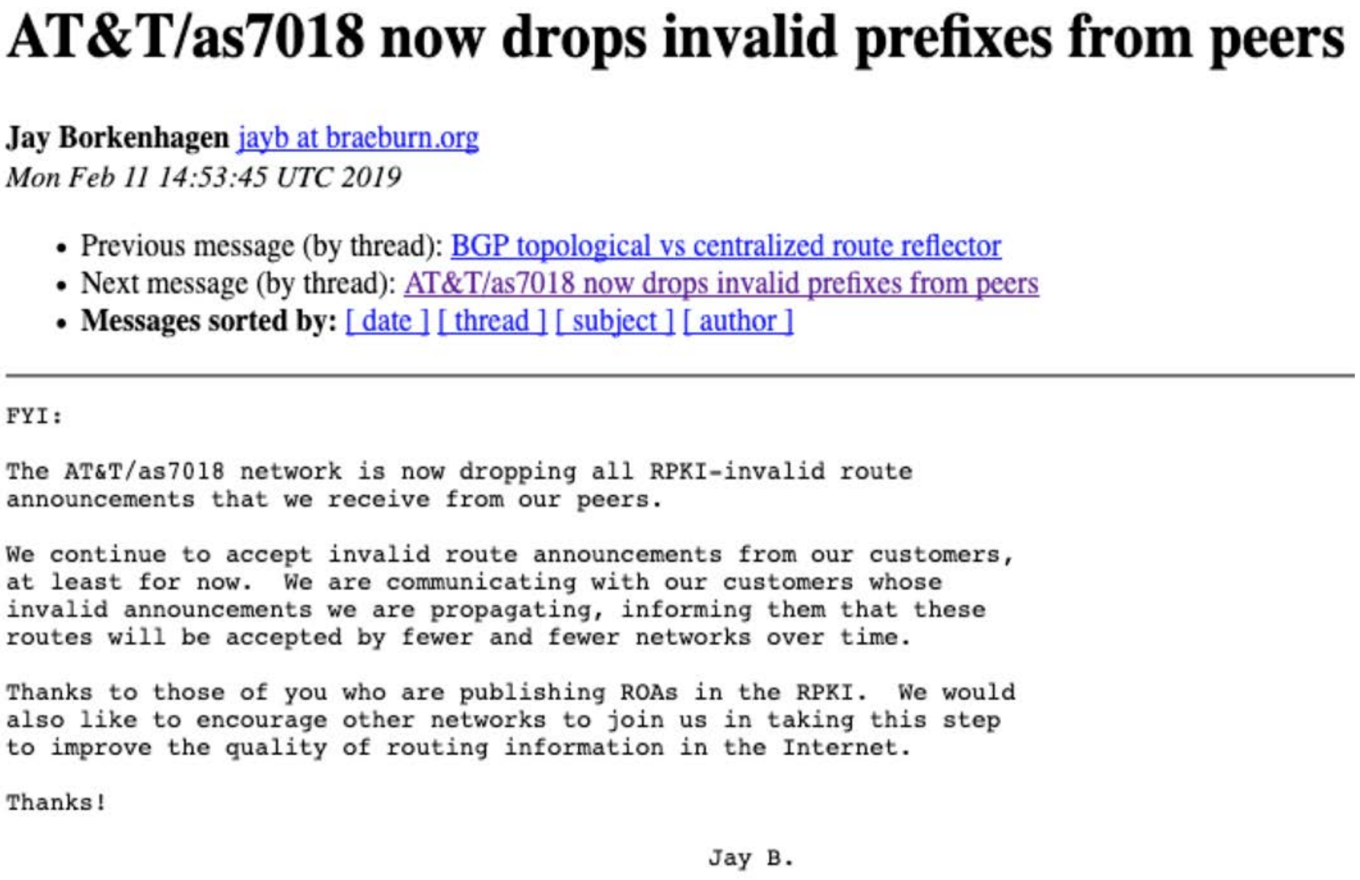
ASN	Prefix	RPKI State
65420	10.0.0.0/16	VALID
65420	10.0.128.0/17	VALID
65421	10.0.0.0/16	INVALID
65420	10.0.10.0/24	INVALID
65430	10.0.0.0/8	NOT FOUND

Possible actions - RPKI states

- **Do Nothing** (observe & learn)
- **Tag with BGP communities**
  - If you have downstream customers or run a route server (IXP)
    - Let them decide
  - Ex:
    - `Valid (ASN:65XX1)`
    - `Not Found (ASN:65XX2)`
    - `Invalid (ASN:65XX3)`
- **Modify preference values**
  - *RFC7115 (High, Low, Lowest)*
- **Drop Invalids**
  - ~6K IPv4 routes (might want to check your top flows)
    https://rpki-monitor.antd.nist.gov/index.php?p=3&s=0

ROV – Industry trends

- **AT&T** (AS7018) drops Invalids!
  - ❑ 11 Feb 2019



AT&T/as7018 now drops invalid prefixes from peers

Jay Borkenhagen jayb at braeburn.org
Mon Feb 11 14:53:45 UTC 2019

- Previous message (by thread): BGP topological vs centralized route reflector
- Next message (by thread): AT&T/as7018 now drops invalid prefixes from peers
- **Messages sorted by:** [date] [thread] [subject] [author]

```
FYI:

The AT&T/as7018 network is now dropping all RPKI-invalid route
announcements that we receive from our peers.

We continue to accept invalid route announcements from our customers,
at least for now.  We are communicating with our customers whose
invalid announcements we are propagating, informing them that these
routes will be accepted by fewer and fewer networks over time.

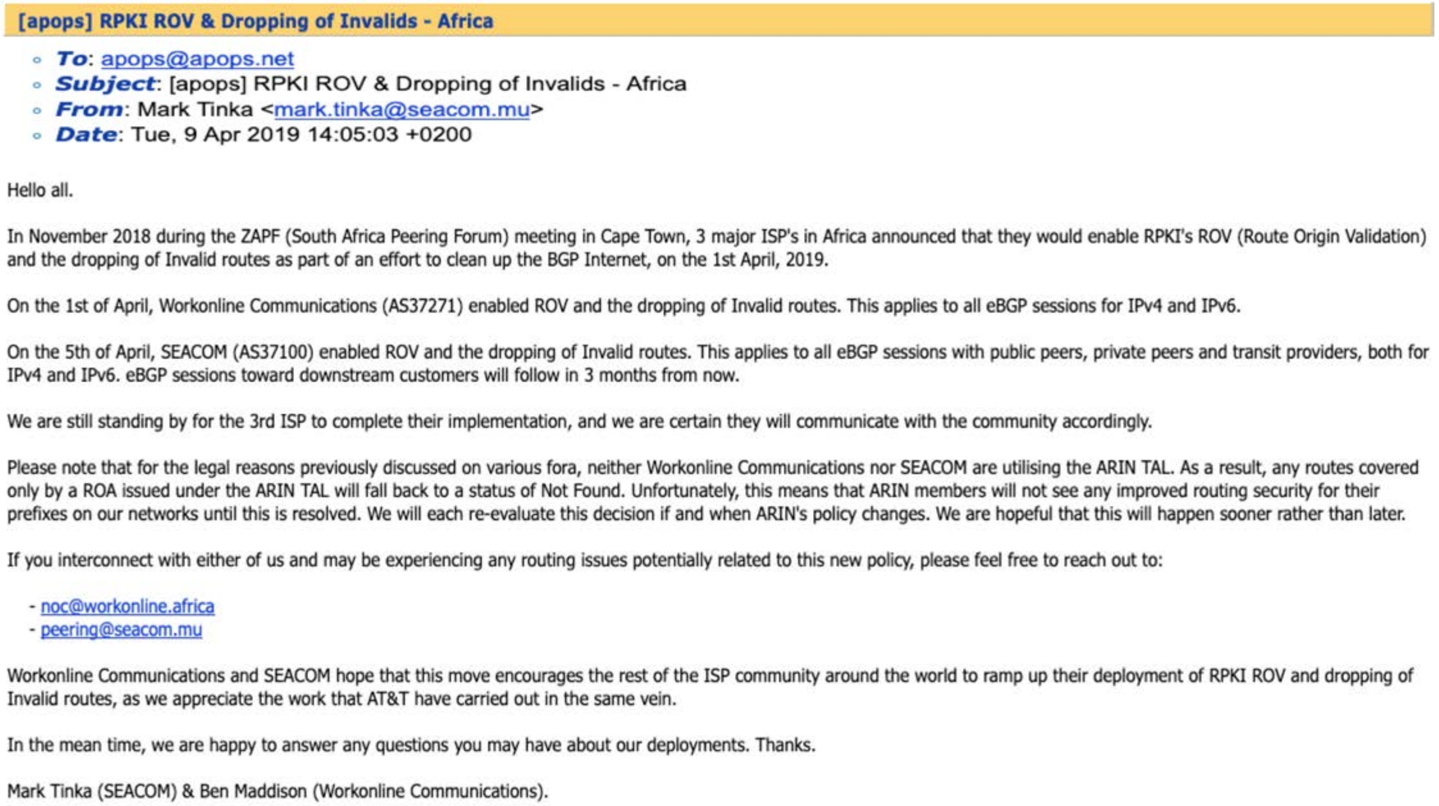
Thanks to those of you who are publishing ROAs in the RPKI.  We would
also like to encourage other networks to join us in taking this step
to improve the quality of routing information in the Internet.

Thanks!

                                        Jay B.
```

ROV – Industry trends

- **Workonline Comms** (AS37271) & **SEACOM** (AS37100) drops Invalids!
  - 1 and 5 April 2019 (does not use ARIN's TAL)



[apops] RPKI ROV & Dropping of Invalids - Africa

- ***To***: apops@apops.net
- ***Subject***: [apops] RPKI ROV & Dropping of Invalids - Africa
- ***From***: Mark Tinka <mark.tinka@seacom.mu>
- ***Date***: Tue, 9 Apr 2019 14:05:03 +0200

Hello all.

In November 2018 during the ZAPF (South Africa Peering Forum) meeting in Cape Town, 3 major ISP's in Africa announced that they would enable RPKI's ROV (Route Origin Validation) and the dropping of Invalid routes as part of an effort to clean up the BGP Internet, on the 1st April, 2019.

On the 1st of April, Workonline Communications (AS37271) enabled ROV and the dropping of Invalid routes. This applies to all eBGP sessions for IPv4 and IPv6.

On the 5th of April, SEACOM (AS37100) enabled ROV and the dropping of Invalid routes. This applies to all eBGP sessions with public peers, private peers and transit providers, both for IPv4 and IPv6. eBGP sessions toward downstream customers will follow in 3 months from now.

We are still standing by for the 3rd ISP to complete their implementation, and we are certain they will communicate with the community accordingly.

Please note that for the legal reasons previously discussed on various fora, neither Workonline Communications nor SEACOM are utilising the ARIN TAL. As a result, any routes covered only by a ROA issued under the ARIN TAL will fall back to a status of Not Found. Unfortunately, this means that ARIN members will not see any improved routing security for their prefixes on our networks until this is resolved. We will each re-evaluate this decision if and when ARIN's policy changes. We are hopeful that this will happen sooner rather than later.

If you interconnect with either of us and may be experiencing any routing issues potentially related to this new policy, please feel free to reach out to:

- noc@workonline.africa
- peering@seacom.mu

Workonline Communications and SEACOM hope that this move encourages the rest of the ISP community around the world to ramp up their deployment of RPKI ROV and dropping of Invalid routes, as we appreciate the work that AT&T have carried out in the same vein.

In the mean time, we are happy to answer any questions you may have about our deployments. Thanks.

Mark Tinka (SEACOM) & Ben Maddison (Workonline Communications).

RPKI Further Reading

RFC 5280

X.509 PKI Certificates

RFC 3779

Extensions for IP Addresses and ASNs

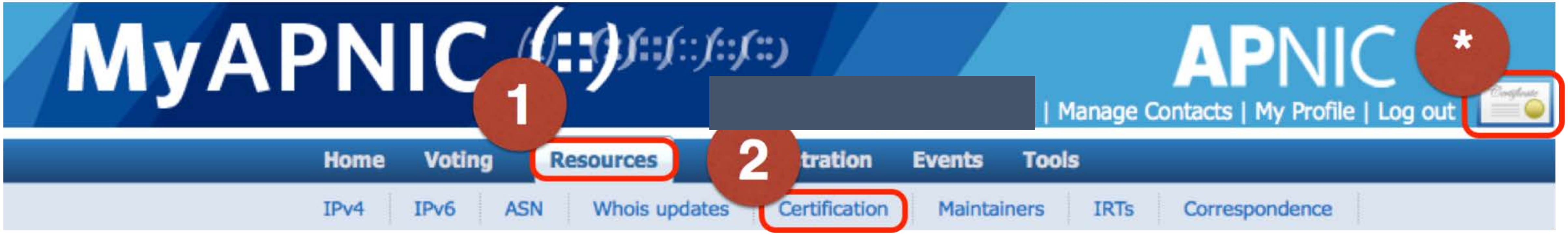
RFC 6481-6493

Resource Public Key Infrastructure

Implementation

Create & publish your ROA

- MyAPNIC portal
  - ❑ **`Resources > RPKI`**

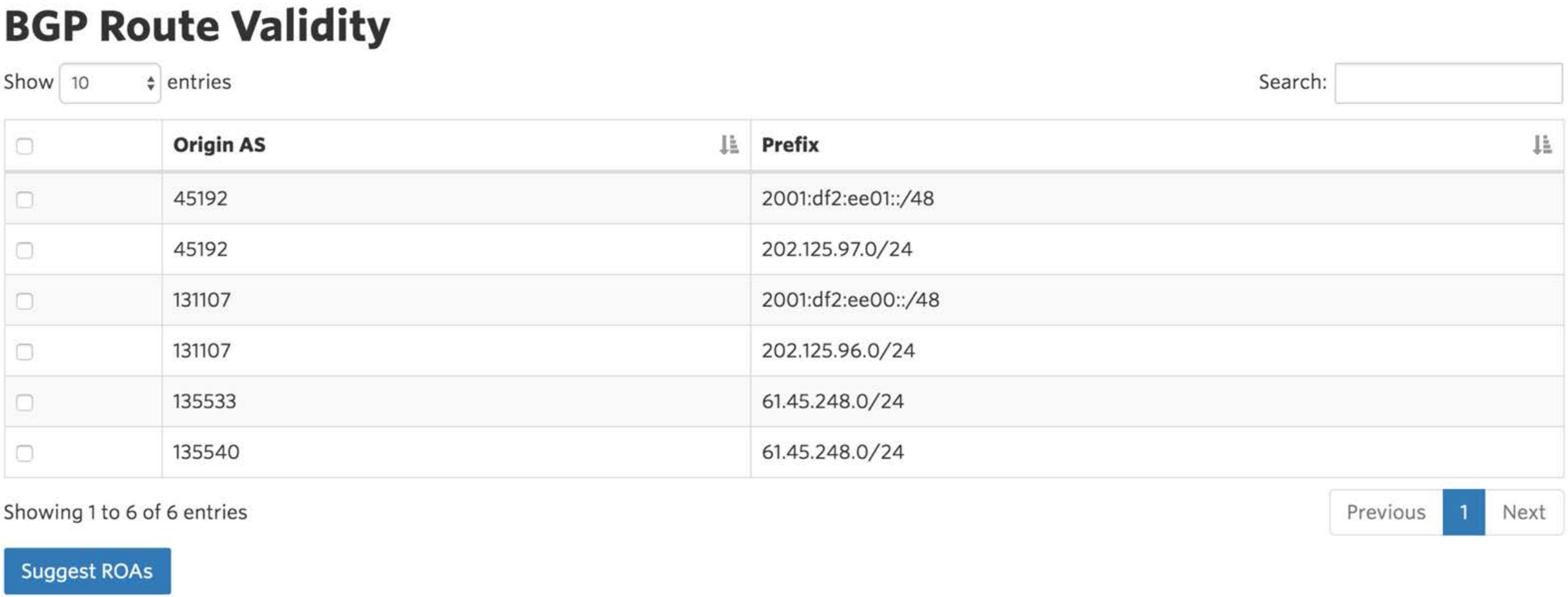

[redacted]

Here is a detailed guide:

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

Create (publish) your ROA

- Available prefixes for which you can create ROA



BGP Route Validity

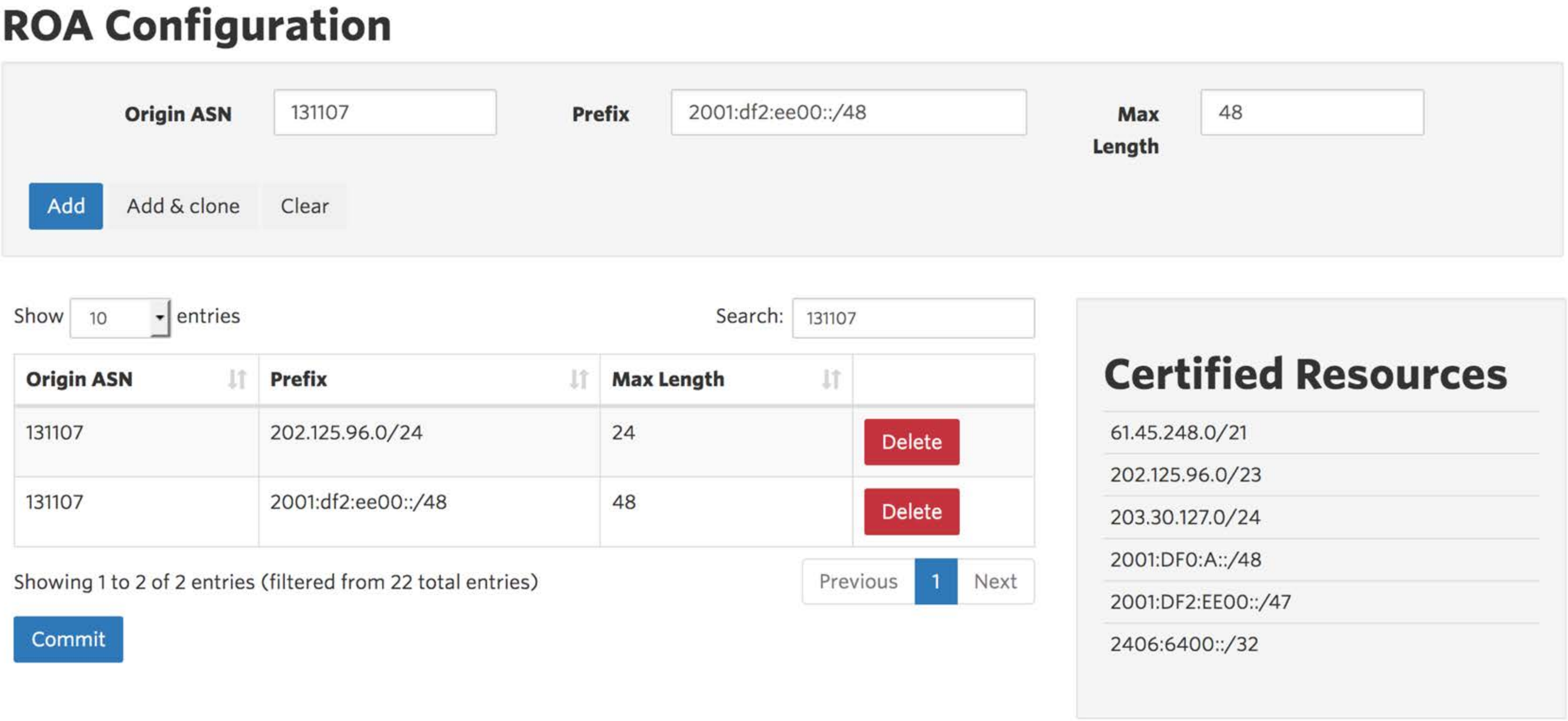
Show 10 entries

Search:

□	Origin AS	Prefix
□	45192	2001:df2:ee01::/48
□	45192	202.125.97.0/24
□	131107	2001:df2:ee00::/48
□	131107	202.125.96.0/24
□	135533	61.45.248.0/24
□	135540	61.45.248.0/24

Showing 1 to 6 of 6 entries

Create (publish) your ROA



ROA Configuration

Origin ASN: 131107

Prefix: 2001:df2:ee00::/48

Max Length: 48

Show 10 entries

Search: 131107

Origin ASN	Prefix	Max Length	
131107	202.125.96.0/24	24	Delete
131107	2001:df2:ee00::/48	48	Delete

Showing 1 to 2 of 2 entries (filtered from 22 total entries)

Certified Resources

- 61.45.248.0/21
- 202.125.96.0/23
- 203.30.127.0/24
- 2001:DF0:A::/48
- 2001:DF2:EE00::/47
- 2406:6400::/32

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

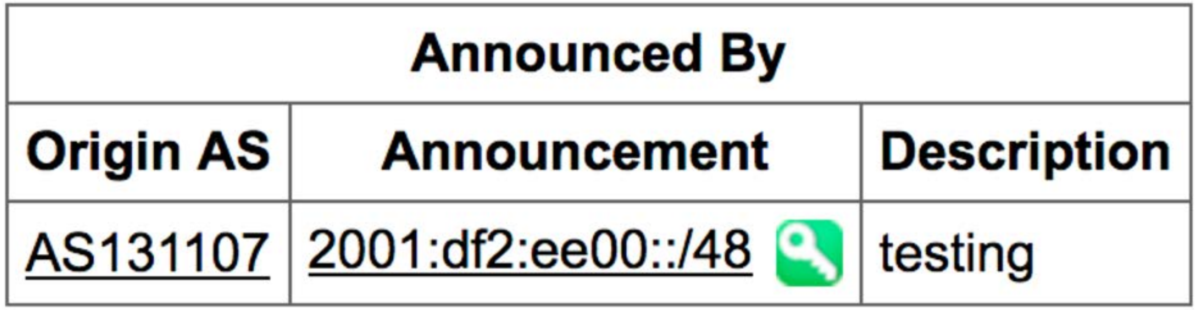
Prefix:                     2001:df2:ee00::/48
Prefix description:         APNICTRAINING-DC
Country code:               AU
Origin AS:                  131107
Origin AS Name:             APNICTRAINING LAB DC
RPKI status:                ROA validation successful
First seen:                 2016-06-30
Last 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)
```

Check your ROA

```
https://bgp.he.net/
```



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

Deploy RPKI Validator

- Many options:
  - ❑ RIPE RPKI Validator

    ```
    https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources
    ```

  - ❑ Dragon Research Labs RPKI Toolkit

    ```
    https://github.com/dragonresearch/rpki.net
    ```

  - ❑ Routinator

    ```
    https://github.com/NLnetLabs/routinator
    ```

  - ❑ OctoRPKI & GoRTR (Cloudflare's RPKI toolkit)

    ```
    https://github.com/cloudflare/cfrpki
    ```

  - ❑ RTRlib* (bird, FRR, Quagga…)

    ```
    https://rtrlib.realmv6.org/
    ```

RIPE Validator

- Download RPKI Validator

```
# wget https://lirportal.ripe.net/certification/content/static/validator/rpki-validator-app-2.25-dist.tar.gz
```

- Installation

```
tar -zxvf rpki-validator-app-2.25-dist.tar.gz
cd rpki-validator-app-2.25
./rpki-validator.sh start
```

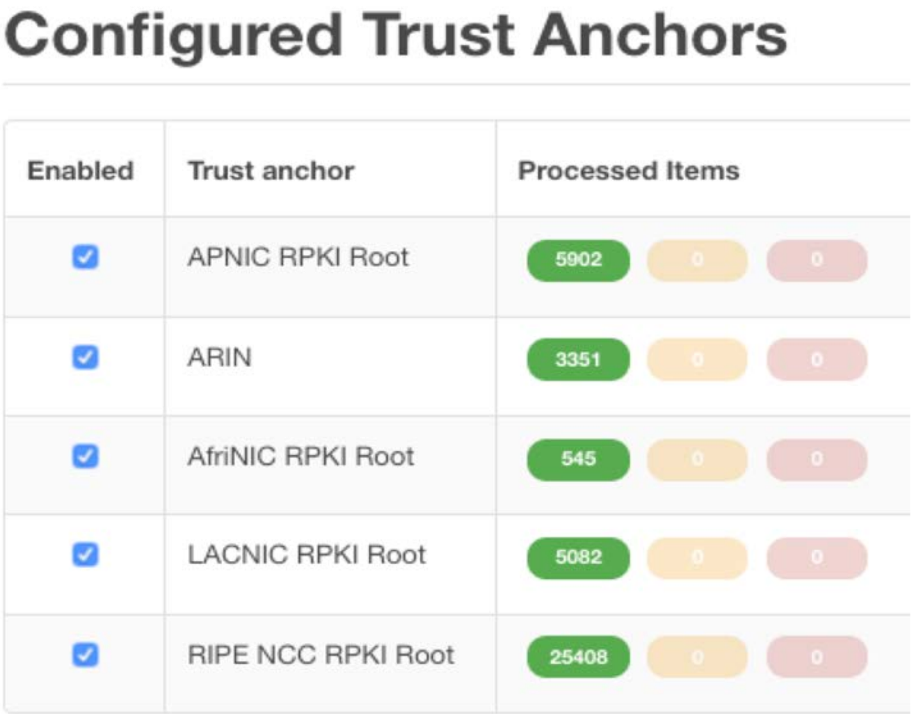
  - ❑ Need to download ARIN’s TAL separately

```
wget https://www.arin.net/resources/rpki/arin-ripevalidator.tal
```

    - Move it to **“<base-folder>/conf/tal”** and restart

RIPE Validator

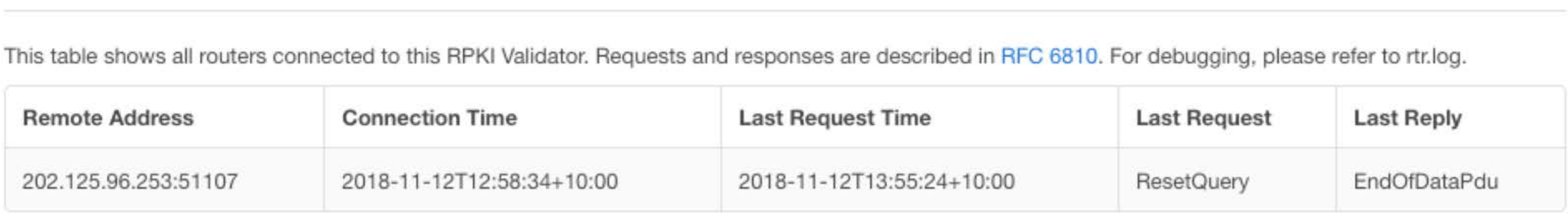
```
http://rpki-validator.apnictraining.net:8080/
```



Configured Trust Anchors

Enabled	Trust anchor	Processed Items
☑	APNIC RPKI Root	5902 0 0
☑	ARIN	3351 0 0
☑	AfriNIC RPKI Root	545 0 0
☑	LACNIC RPKI Root	5082 0 0
☑	RIPE NCC RPKI Root	25408 0 0

Router Sessions



This table shows all routers connected to this RPKI Validator. Requests and responses are described in RFC 6810. For debugging, please refer to rtr.log.

Remote Address	Connection Time	Last Request Time	Last Request	Last Reply
202.125.96.253:51107	2018-11-12T12:58:34+10:00	2018-11-12T13:55:24+10:00	ResetQuery	EndOfDataPdu

Dragon Research - Validator

- Installation on Ubuntu 16.04 Xenial

```
https://github.com/dragonresearch/rpki.net/blob/master/doc/quickstart/xenial-rp.md
```

- Installation
  - ❑ Add the GPG public key

```
# wget -q -O /etc/apt/trusted.gpg.d/rpki.gpg https://download.rpki.net/APTng/apt-gpg-key.gpg
```

  - ❑ Add the repo to the APT source list

```
# wget -q -O /etc/apt/sources.list.d/rpki.list https://download.rpki.net/APTng/rpki.xenial.list

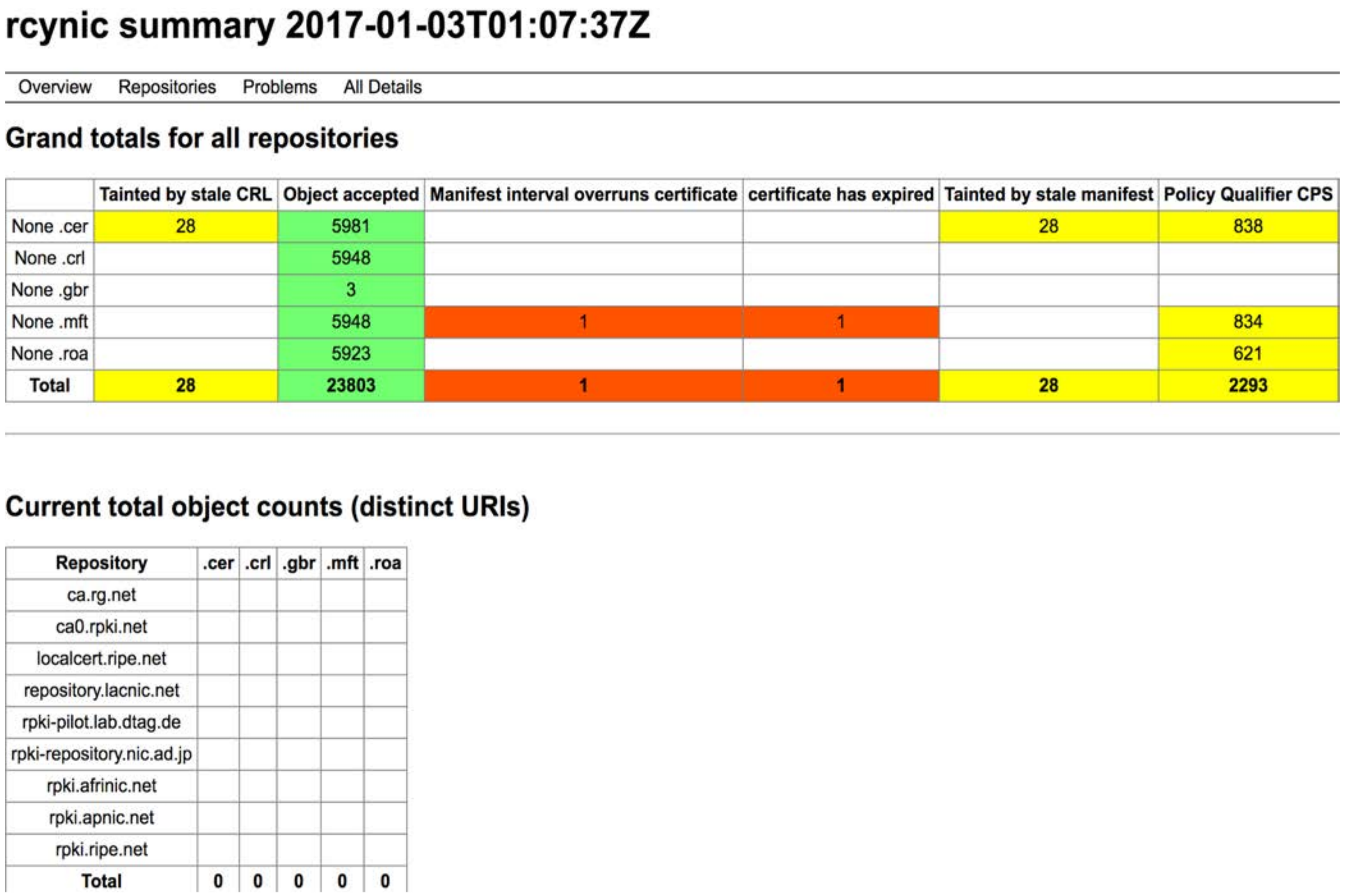
-q: quite (wget output)
-O: output to <file>
```

```
# apt update
```

```
# apt install rpki-rp
```

Dragon Research - Validator

http://rpki-dragonresearch.apnictraining.net/rcynic/



rcynic summary 2017-01-03T01:07:37Z

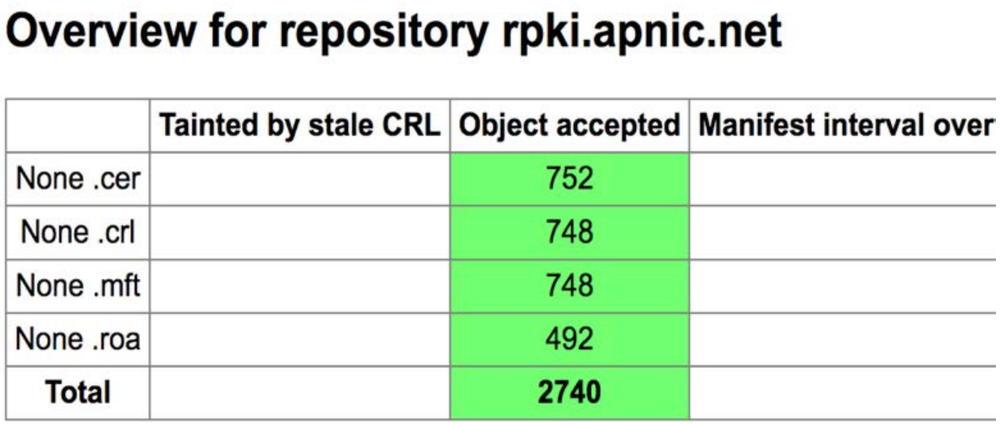
Overview Repositories Problems All Details

Grand totals for all repositories

	Tainted by stale CRL	Object accepted	Manifest interval overruns certificate	certificate has expired	Tainted by stale manifest	Policy Qualifier CPS
None .cer	28	5981			28	838
None .crl		5948				
None .gbr		3				
None .mft		5948	1	1		834
None .roa		5923				621
Total	**28**	**23803**	**1**	**1**	**28**	**2293**

Current total object counts (distinct URIs)

Repository	.cer	.crl	.gbr	.mft	.roa
ca.rg.net					
ca0.rpki.net					
localcert.ripe.net					
repository.lacnic.net					
rpki-pilot.lab.dtag.de					
rpki-repository.nic.ad.jp					
rpki.afrinic.net					
rpki.apnic.net					
rpki.ripe.net					
Total	**0**	**0**	**0**	**0**	**0**



Overview for repository rpki.apnic.net

	Tainted by stale CRL	Object accepted	Manifest interval over
None .cer		752	
None .crl		748	
None .mft		748	
None .roa		492	
Total		**2740**	

Routinator - Validator

- Installation on Ubuntu 16.04 Xenial
  - Will use rustup to install and manage rust

```
curl https://sh.rustup.rs -sSf | sh
```

  - downloads and runs a script to install rustup and rust

- Installation
  - Using “cargo” (the rust pkg manager) to install Routinator

```
cargo install routinator
```

- run Routinator

```
routinator vrps
```

  - The command prints the list of valid ROAs (valid roa payload - vrp)
  - If this is the first time running Routinator, it creates `$HOME/.rpki-cache` (example `/home/tashi/.rpki-cache/tals/`) to place the TALs of the five RIRs (will complain ARIN’s TAL is missing)

Routinator – Validator (contd..)

- Download ARIN’s TAL and move it to the base folder
  - Make sure it is the RFC7330 format

```
wget https://www.arin.net/resources/rpki/arin-rfc7730.tal
```

```
mv arin-rfc7730.tal /home/tashi/.rpki-cache/tals/
```

- Rerun the command

```
routinator vrps
```

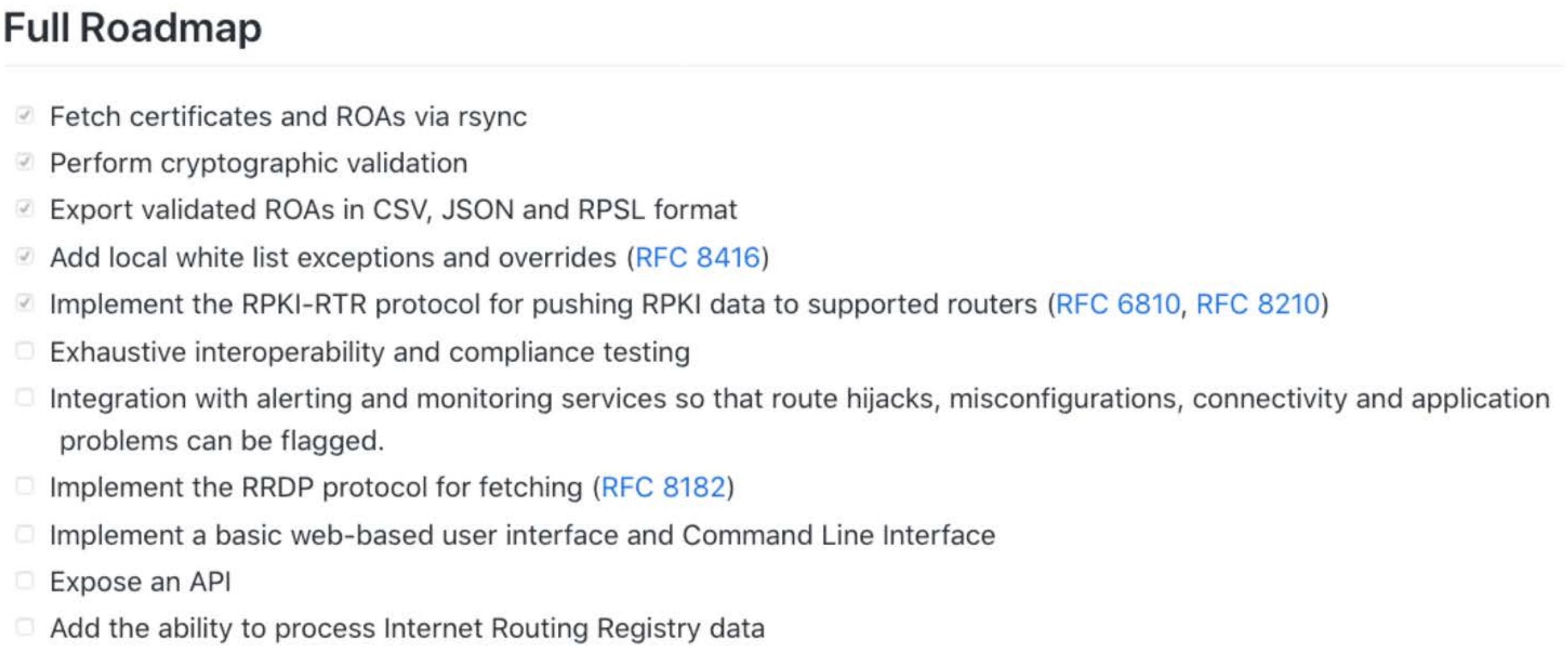
  - It will rsync the whole rpki repo to the local machine and produce a list of valid ROAs

- Feeding routers with RTR
  - In order to run it as a RTR server (port 3323) on both IPv4/v6, use the rtrd subcommand

```
routinator rtrd -l 202.125.96.48:3323 -l [2001:df2:ee00:ee00::48]:3323 --refresh=900
```

Routinator – Validator (contd..)

- Routinator does not yet have a web interface/GUI



Full Roadmap

- ☑ Fetch certificates and ROAs via rsync
- ☑ Perform cryptographic validation
- ☑ Export validated ROAs in CSV, JSON and RPSL format
- ☑ Add local white list exceptions and overrides (RFC 8416)
- ☑ Implement the RPKI-RTR protocol for pushing RPKI data to supported routers (RFC 6810, RFC 8210)
- □ Exhaustive interoperability and compliance testing
- □ Integration with alerting and monitoring services so that route hijacks, misconfigurations, connectivity and application problems can be flagged.
- □ Implement the RRDP protocol for fetching (RFC 8182)
- □ Implement a basic web-based user interface and Command Line Interface
- □ Expose an API
- □ Add the ability to process Internet Routing Registry data

`https://github.com/NLnetLabs/routinator/blob/master/README.md`

Configuration (IOS)

- Establishing session with the validator

```
router bgp 131107
 bgp rpki server tcp <validator-IP> port <323/8282/3323> refresh 120
```

- Note:
  - Cisco IOS by default does not include invalid routes for best path selection!
  - If you don’t want to drop invalids, we need explicitly tell BGP (under respective address families)

```
bgp bestpath prefix-validate allow-invalid
```

Configuration (IOS)

- Policies based on validation:

```
route-map ROUTE-VALIDATION permit 10
 match rpki valid
 set local-preference 110
!
route-map ROUTE-VALIDATION permit 20
 match rpki not-found
 set local-preference 100
!
route-map ROUTE-VALIDATION permit 10
 match rpki invalid
 set local-preference 90
!
```

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
!
```

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 110;
               validation-state valid;
               accept;
           }
       }
       term invalid {
           from {
               protocol bgp;
               validation-database invalid;
           }
           then {
               local-preference 90;
               validation-state invalid;
               accept;
           }
       }
```

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

Router Configuration (JunOS)

- Apply the policy to inbound updates

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

IOS should have something similar!

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

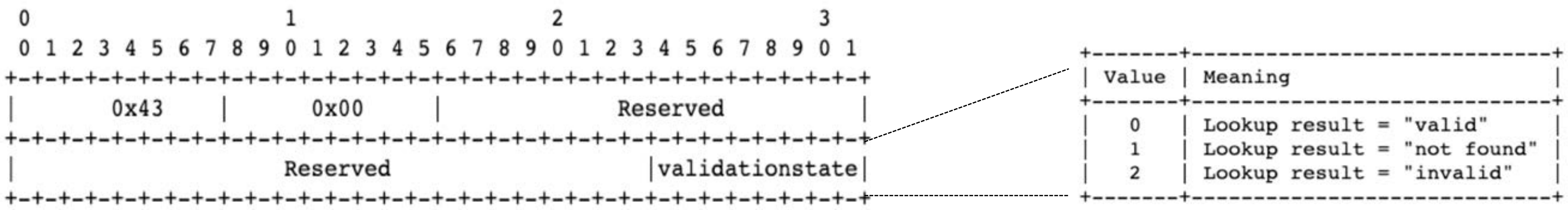
2001:201::/32     *[BGP/170] 21:18:14, MED 0, localpref 100, from
2001:df2:ee00::1
                     AS path: 65332 I, validation-state: unknown
                   >to fe80::dab1:90ff:fedc:fd07 via ge-1/1/0.0
```

Propagating RPKI states to iBGP peers

- To avoid every BGP speaker having an RTR session, and

- All BGP speakers have consistent information

  - Relies on extended BGP communities (RFC8097)



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0x43    |      0x00     |             Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Reserved                   |validationstate|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Value	Meaning
0	Lookup result = "valid"
1	Lookup result = "not found"
2	Lookup result = "invalid"

- Sender (one that has 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 (one with RTR session)

```
policy-statement ROUTE-VALIDATION {
    term valid {
        from {
            protocol bgp;
            validation-database valid;
        }
        then {
            local-preference 110;
            validation-state valid;
            community add origin-validation-state-valid;
            accept;
        }
    }
    term invalid {
        from {
            protocol bgp;
            validation-database invalid;
        }
        then {
            local-preference 90;
            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;
   }

}
```

Propagating RPKI states – potential issues

- IOS as BR, propagating states to JunOS iBGP peers

```
unknown iana 4300
```

  - Hack:
    - Either act on the states at the border, or
    - Match and tag them with custom communities before propagating

Configuration - Reference Link

- **Cisco**
  - ❑ https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-3s/irg-xe-3s-book/irg-origin-as.pdf

- **Juniper**
  - ❑ https://www.juniper.net/documentation/en_US/junos/topics/topic-map/bgp-origin-as-validation.html

- **RIPE:**
  - ❑ https://www.ripe.net/manage-ips-and-asns/resource-management/certification/router-configuration

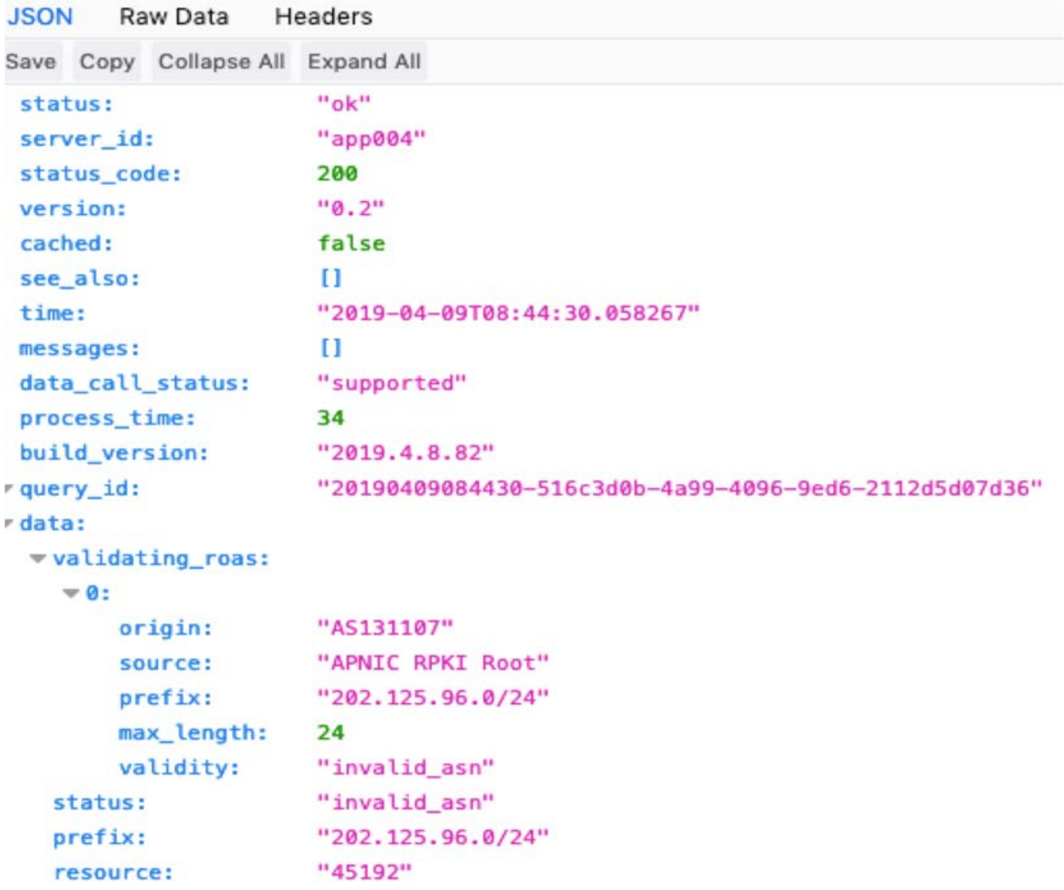
Operational Caveats

- When RTR session goes down, the validation state changes to **Not Found** for all routes after a while
  - ***Invalid*** **=>** ***Not Found***
  - **we need at least two RTR sessions** and/or need careful filtering policies

- During a router reload, do we receive ROAs first or BGP updates first?
  - If BGP update is faster than ROA, will propagate even invalid routes to its iBGP peers

Useful tools

- RIPEstat – prefix/ASN
  - https://stat.ripe.net/data/rpki-validation/data.json?resource=45192&prefix=202.125.96.0/24



```
status:                   "ok"
server_id:                "app004"
status_code:              200
version:                  "0.2"
cached:                   false
see_also:                 []
time:                     "2019-04-09T08:44:30.058267"
messages:                 []
data_call_status:         "supported"
process_time:             34
build_version:            "2019.4.8.82"
query_id:                 "20190409084430-516c3d0b-4a99-4096-9ed6-2112d5d07d36"
data:
  ▼ validating_roas:
    ▼ 0:
         origin:          "AS131107"
         source:          "APNIC RPKI Root"
         prefix:          "202.125.96.0/24"
         max_length:      24
         validity:        "invalid_asn"
  status:                 "invalid_asn"
  prefix:                 "202.125.96.0/24"
  resource:               "45192"
```

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

Thank You!