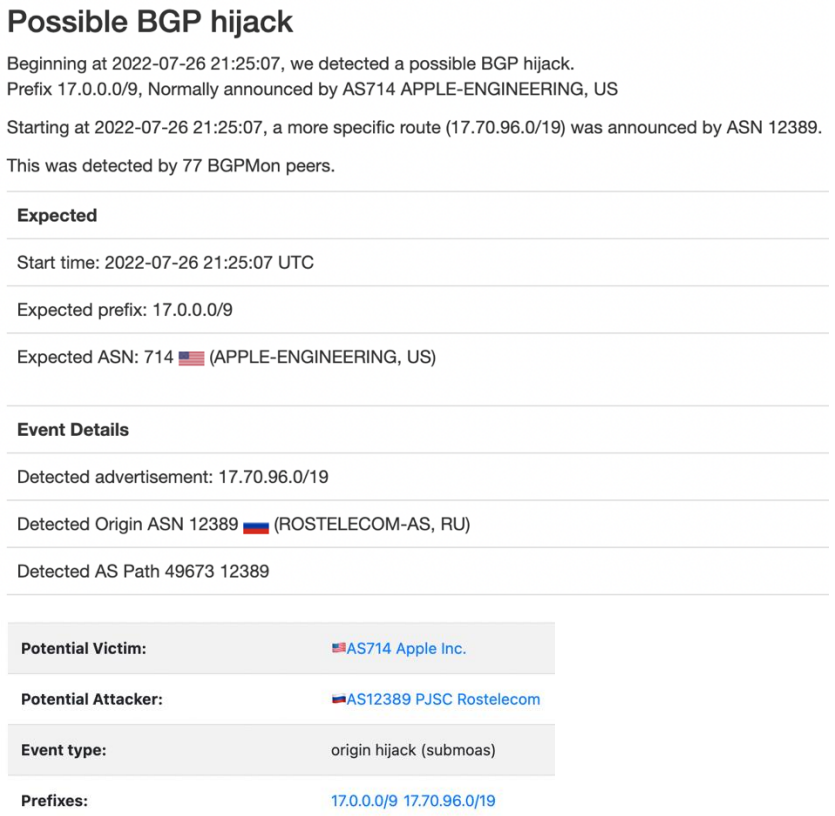
APNIC

Securing Internet Routing with RPKI

Headlines

- AS12389 hijacks one of the Apple's prefix – **26 Jul 2022**
  - Apple's usual announcement 17.0.0.0/9
    - More specific 17.70.96.0/19 was hijacked
  - Main Upstream leakers
    - AS7473 (Singtel)
    - AS1273 (Vodafone UK)
    - AS4826 (Vocus)
  - Apple announced 17.70.96.0/21 to mitigate
    - Affected for more than 5 hours
    - AS12389 withdrew the announcement after 12+ hours



Possible BGP hijack

Beginning at 2022-07-26 21:25:07, we detected a possible BGP hijack.
Prefix 17.0.0.0/9, Normally announced by AS714 APPLE-ENGINEERING, US

Starting at 2022-07-26 21:25:07, a more specific route (17.70.96.0/19) was announced by ASN 12389.

This was detected by 77 BGPMon peers.

Expected

Start time: 2022-07-26 21:25:07 UTC

Expected prefix: 17.0.0.0/9

Expected ASN: 714 (APPLE-ENGINEERING, US)

Event Details

Detected advertisement: 17.70.96.0/19

Detected Origin ASN 12389 (ROSTELECOM-AS, RU)

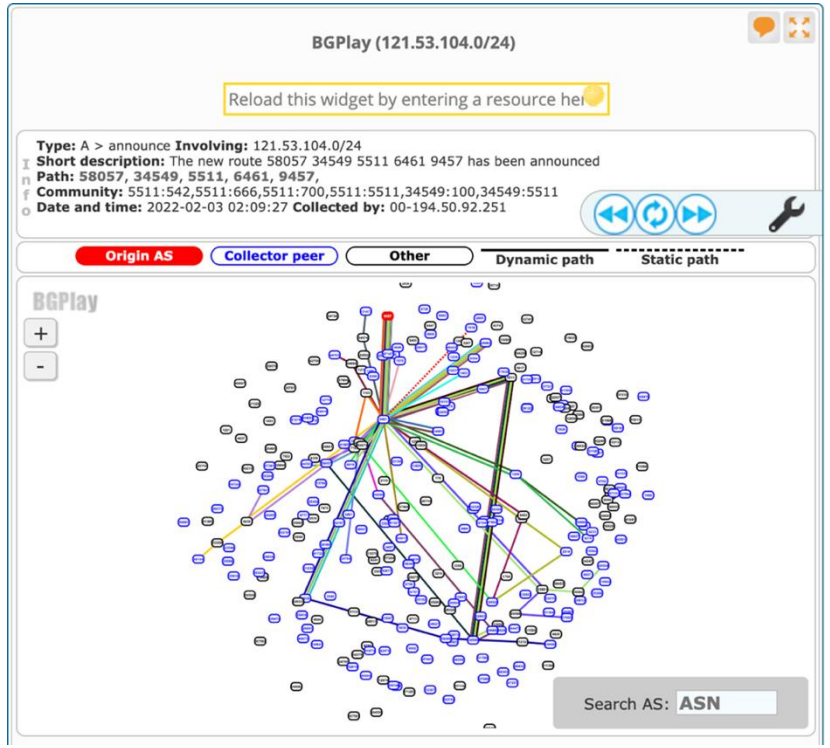
Detected AS Path 49673 12389

Potential Victim:	AS714 Apple Inc.
Potential Attacker:	AS12389 PJSC Rostelecom
Event type:	origin hijack (submoas)
Prefixes:	17.0.0.0/9 17.70.96.0/19

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

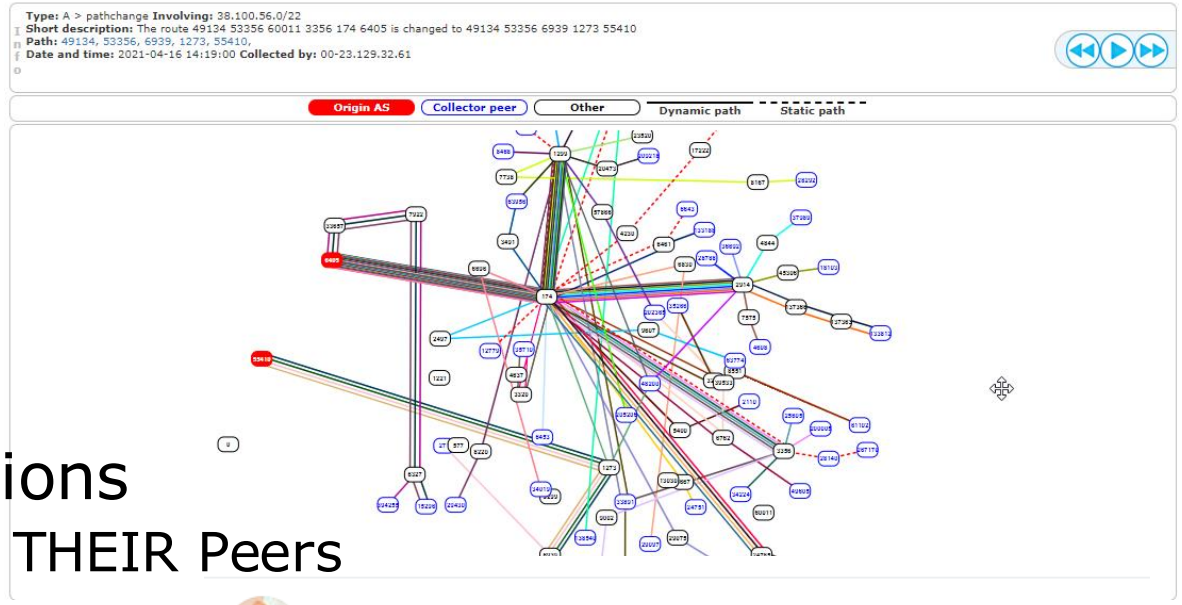
Headlines

- Hackers steal 1.9m worth of crypto currency – **03 Feb 2022**
  - AS38099 (Kakao Corp) hosts KLAYswap on 121.53.104.157
    - AS9457 (Dreamline Co) delegated 121.53.104.0/23 to Kakao Corp
      - It's announced to KINX only
      - **No ROA coverage**
    - Attacker announced **121.53.104.0/24** in global routing table with **AS9457**
      - **AS_PATH:** 40630 6939 **6461 9457**
      - Managed to announce through Zayo
      - Traffic rerouted to hackers' network
      - More detailed analysis:
        https://www.manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets/



Headlines

- AS55410 Leaks ~30k Prefixes – **16 Apr 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.

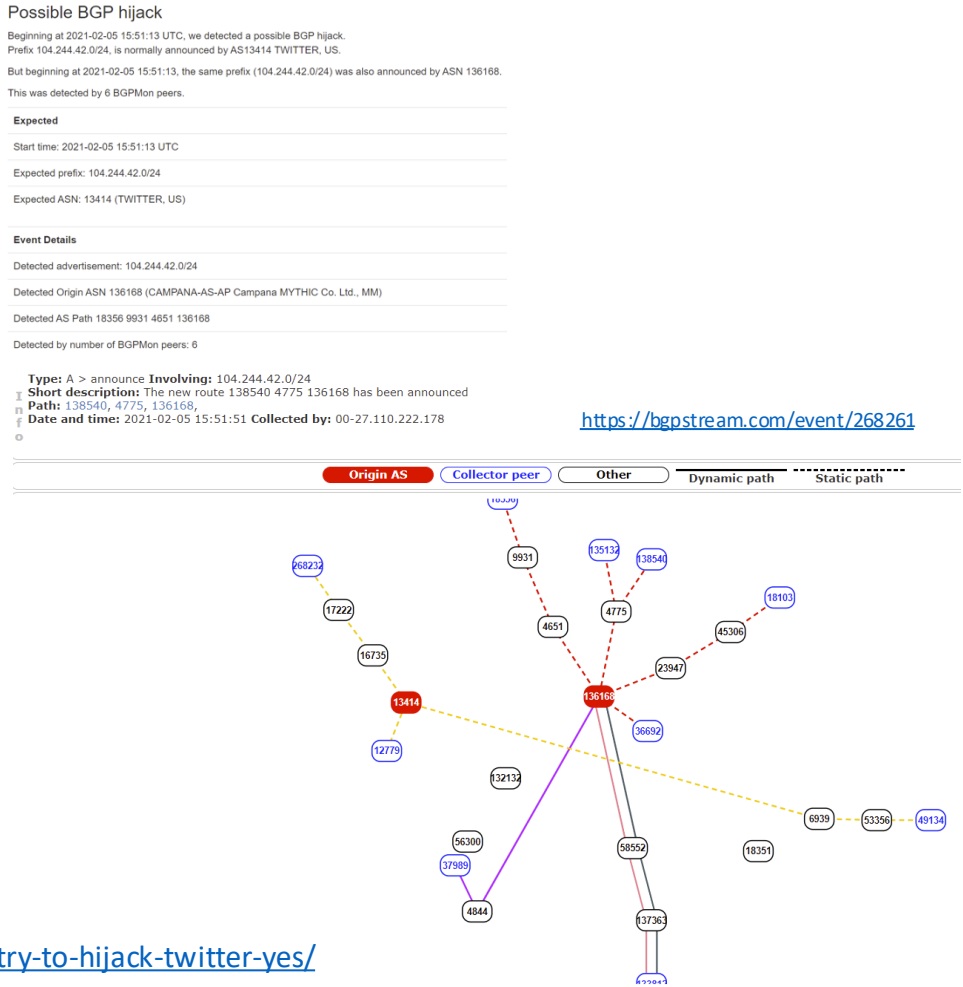
https://bgpstream.com/event/271479 https://bgpstream.com/event/271478

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/

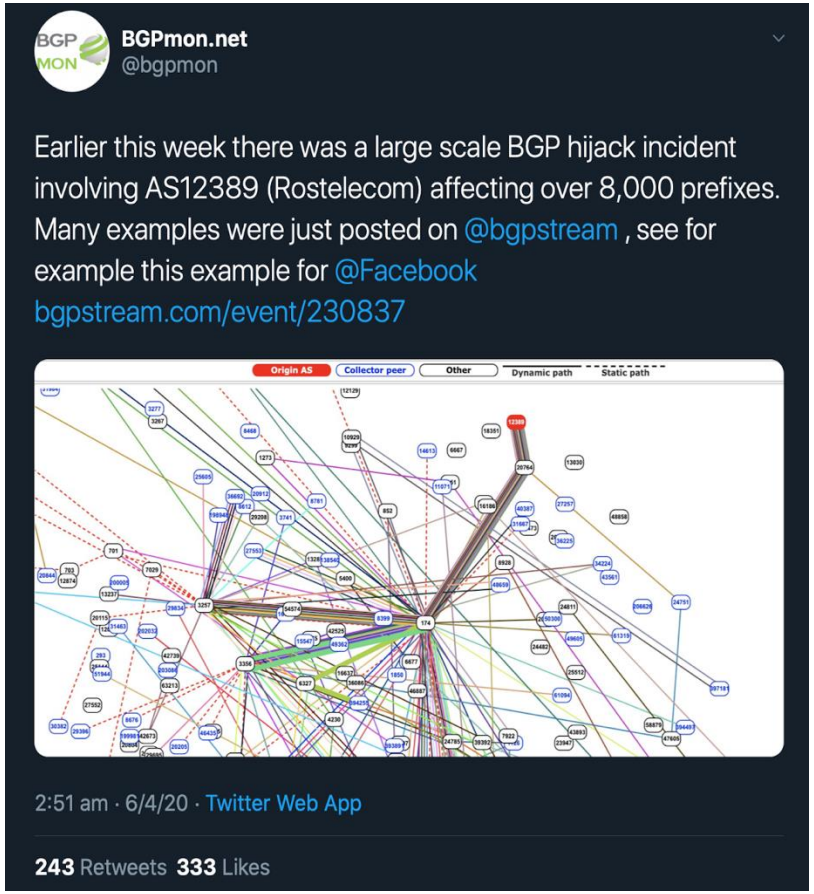


Headlines

- Not so funny ☹ – **1 Apr 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)

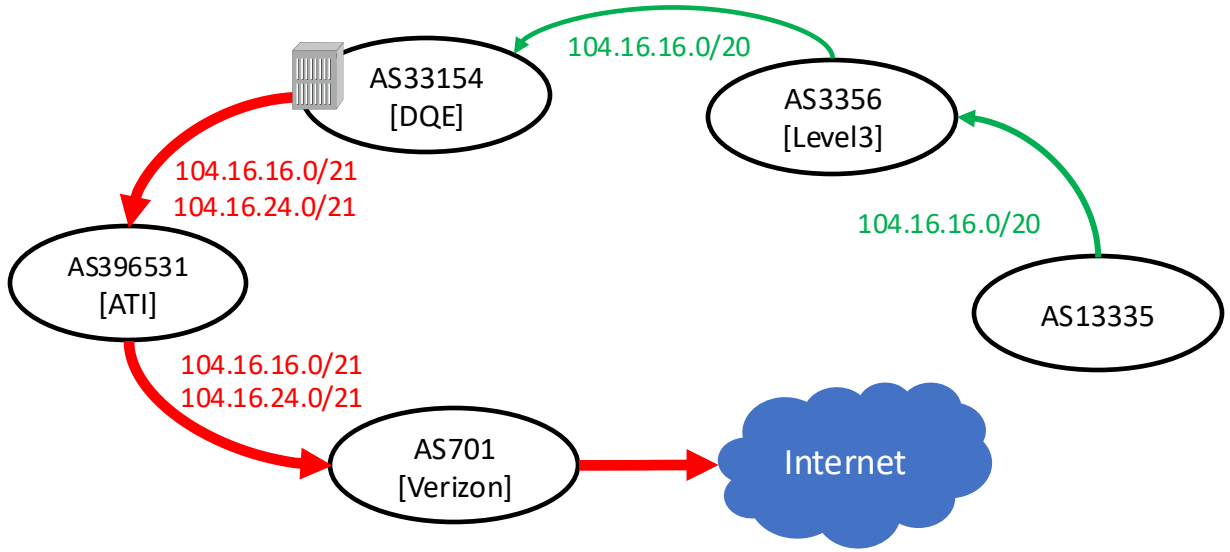
Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 104.18.216.0/21	AS13335 - CLOUDFLARENET - [US]: 265 - 104.18.208.0/20 from 2020-04-01 19:33 to 2020-04-01 20:04 [high] 265 - 104.16.0.0/12 from 2020-04-01 19:33 to 2020-04-01 20:04 [high]	2020-04-01 19:33	0:31:00
Created Hijack	AS12389 - ROSTELECOM-AS - [RU] 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

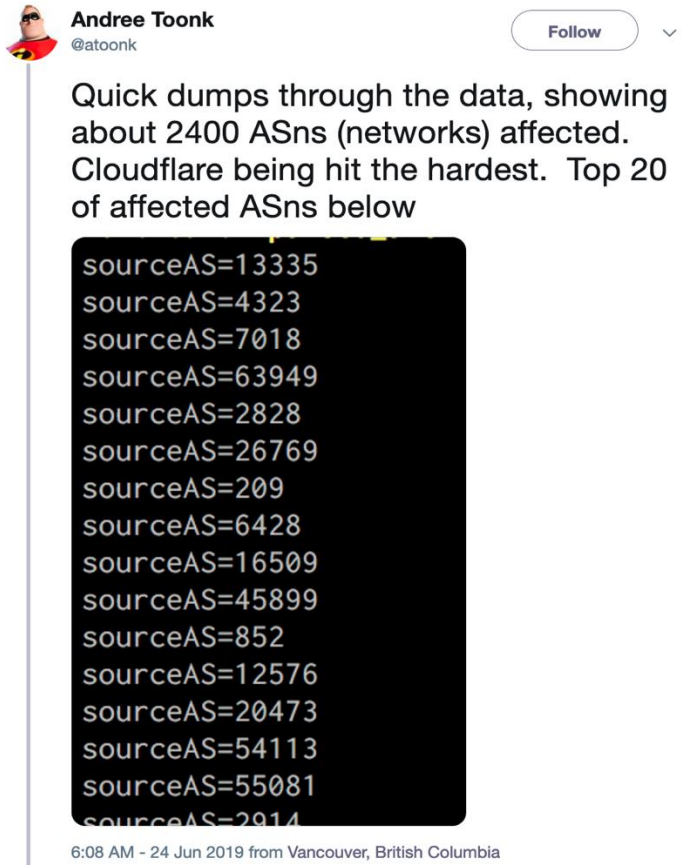
https://blog.qrator.net/en/how-you-deal-route-leaks_69/



Headlines

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





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

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

- 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?

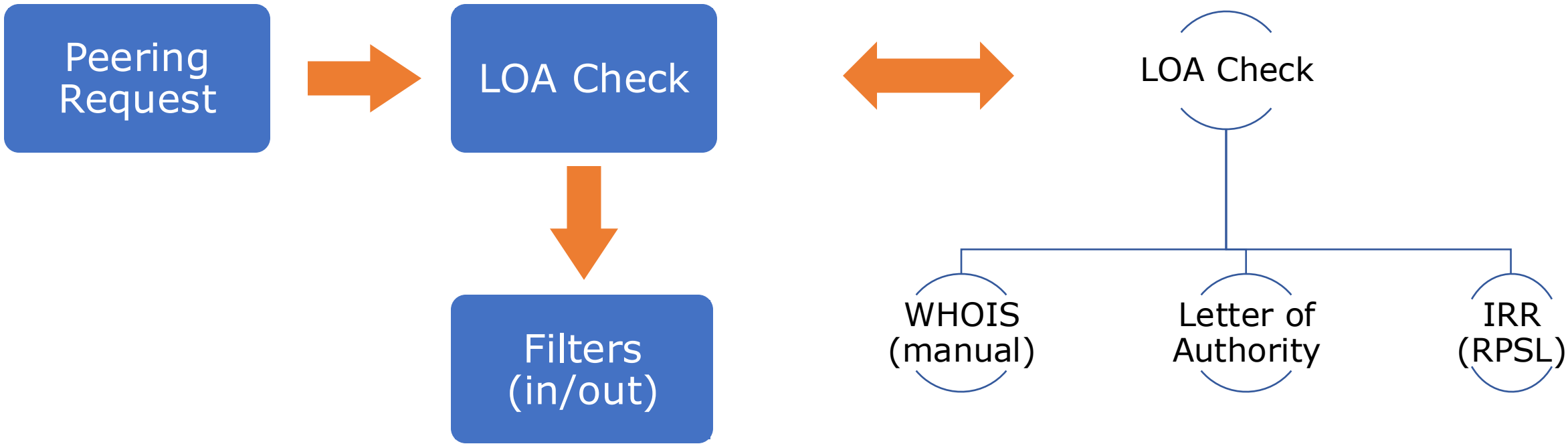
- Routing works in REVERSE
  - Outbound advertisement affects inbound traffic
  - Inbound (*Accepted*) advertisement influence outbound traffic

- 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 ☹

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

- 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

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

```
~
❯ bgpq4 -Al PREF-V4-IN AS24016
no ip prefix-list PREF-V4-IN
ip prefix-list PREF-V4-IN permit 103.197.164.0/22 le 24
ip prefix-list PREF-V4-IN permit 115.84.128.0/19 le 24
ip prefix-list PREF-V4-IN permit 202.21.176.0/20 le 24
ip prefix-list PREF-V4-IN permit 220.158.220.0/22 le 24

~
❯ bgpq4 -6Al PREF-V6-IN AS24016
no ipv6 prefix-list PREF-V6-IN
ipv6 prefix-list PREF-V6-IN permit 2401:8300::/32 le 40
ipv6 prefix-list PREF-V6-IN permit 2401:8300:f000::/47 ge 48 le 48
ipv6 prefix-list PREF-V6-IN permit 2401:8300:f002::/48
```

```
~
❯ bgpq4 -Al PREF-V4-IN AS24016:AS-ALL
no ip prefix-list PREF-V4-IN
ip prefix-list PREF-V4-IN permit 36.255.104.0/23 le 24
ip prefix-list PREF-V4-IN permit 103.71.57.0/24
ip prefix-list PREF-V4-IN permit 103.76.2.0/24
ip prefix-list PREF-V4-IN permit 103.84.134.0/24
ip prefix-list PREF-V4-IN permit 103.103.66.0/24
ip prefix-list PREF-V4-IN permit 103.110.109.0/24
ip prefix-list PREF-V4-IN permit 103.110.110.0/23 le 24
ip prefix-list PREF-V4-IN permit 103.119.75.0/24
ip prefix-list PREF-V4-IN permit 103.143.252.0/24
ip prefix-list PREF-V4-IN permit 103.191.77.0/24
ip prefix-list PREF-V4-IN permit 103.197.164.0/22 le 24
ip prefix-list PREF-V4-IN permit 115.84.128.0/19 le 24
ip prefix-list PREF-V4-IN permit 202.21.176.0/20 le 24
ip prefix-list PREF-V4-IN permit 220.158.220.0/22 le 24
```

```
~
❯ bgpq4 -3f 24016 -l ROL-IN AS24016:AS-ALL
no ip as-path access-list ROL-IN
ip as-path access-list ROL-IN permit ^24016(_24016)*$
ip as-path access-list ROL-IN permit ^24016(_[0-9]+)*_(132218|133742|136238|137056)$
ip as-path access-list ROL-IN permit ^24016(_[0-9]+)*_(137981|150125)$
```

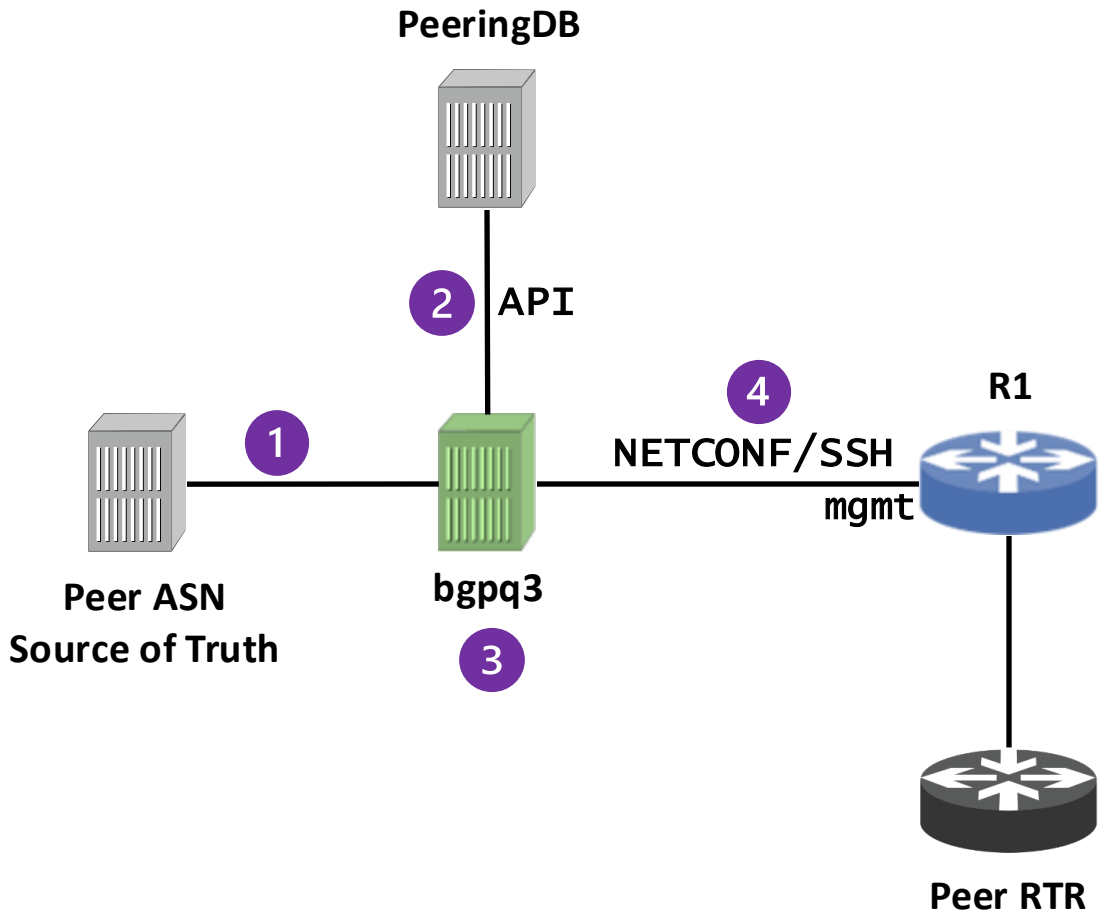
bgpq4 Demo

Aside: bgpq4/bgpq3

- bgpq4 has some advantages over bgpq3
  - Faster response time
  - Included Arista, MikroTik
  - More flags and syntaxes – see man page or help option
- Installation
  - Ubuntu/Debian: `sudo apt install [bgpq4/bgpq3]`
  - MacOS: `brew install [bgpq4/bgpq3]`
- More info:
  - https://github.com/bgp/bgpq4
  - https://github.com/snar/bgpq3

Automatic Filtering with bgpq4

1. Collect list of Peer ASN (API, text file or other means)
2. API Call: Get the AS-SET of Peer ASN from PeeringDB
3. bgpq3:
   - Generate `prefix/asn` filters
   - Compare with the current filter
   - If changed, override the saved filter with the new one
4. Push the new filter to the router with NETCONF/SSH



Limitation of Prefix-list and AS-PATH filtering

- Prefix-list and AS-PATH filters are suitable to filter
  - downstream customers
  - Peers
- Not ideal to filter routes in the global BGP table
  - Wrong prefixes can be injected anytime
    - Due to mistakes (fat finger)
    - Intentionally (Hijack)
- To preventing invalid routes from internet, RPKI will be able to help

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)?

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

What is RPKI?

- A cryptographic framework that
  - Allows internet resource (IPv4, IPv6, ASNs) holders to create ROA
  - Cryptographically validate the prefix and its origin ASN
- ROA – Route Origin Authorization
  - Digital object generated cryptographically by the resource holder
  - Published in the RPKI repository
- ROV – Route Origin Validation
  - Which ASN(s) have the authority to originate the prefix?

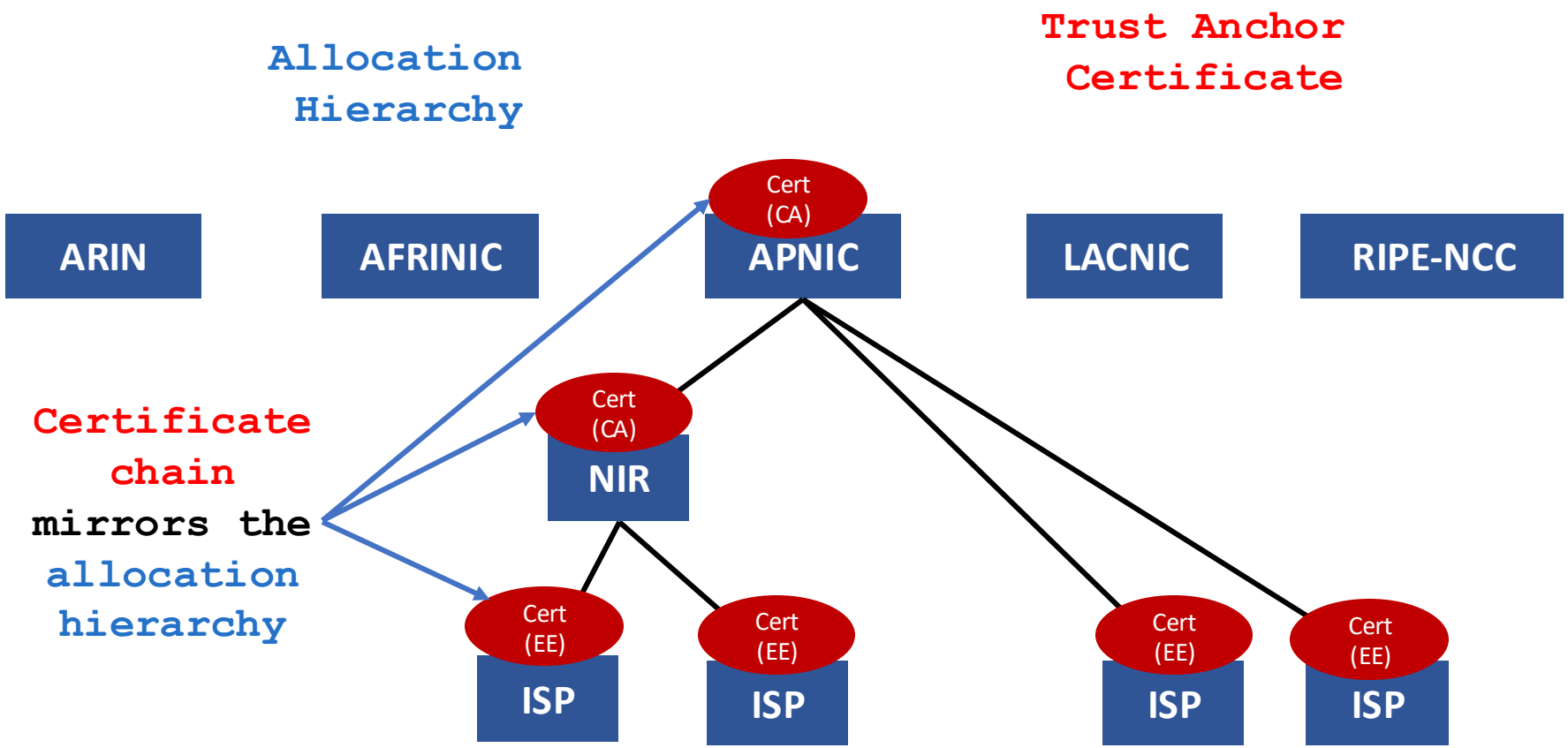
How about trust?

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

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

    - To describe the address allocation using digital certificates

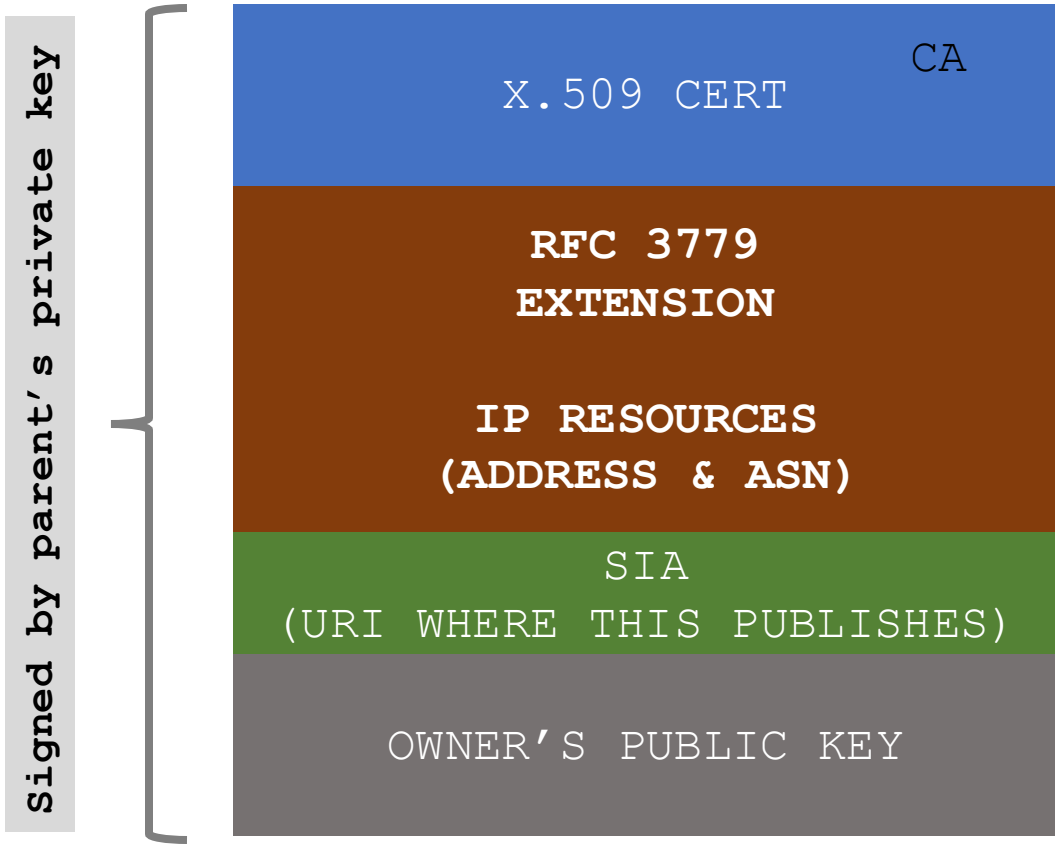
RPKI Chain of Trust



RPKI Chain of Trust

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

RPKI profile ~ Resource Certificates



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

Resource Certificates

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

Route Origin Authorization (ROA)

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

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

Route Origin Authorization (ROA)

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

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

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

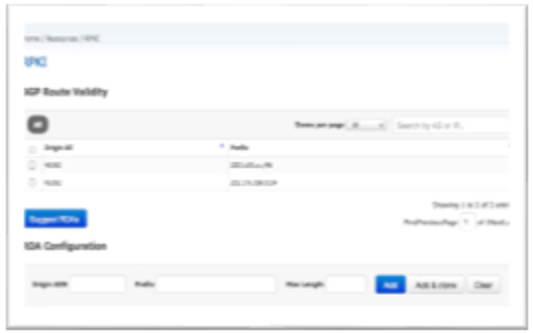
What can RPKI do?

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

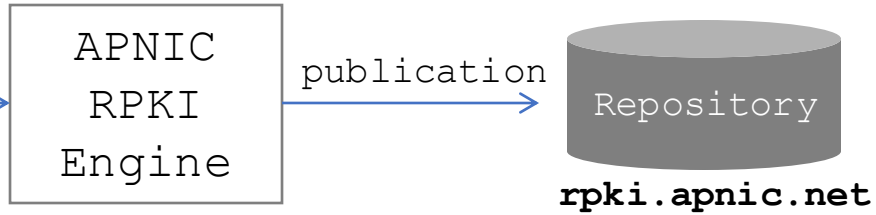
- Can help:
  - prevent **route hijacks/mis-origination/misconfiguration**

RPKI Components

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

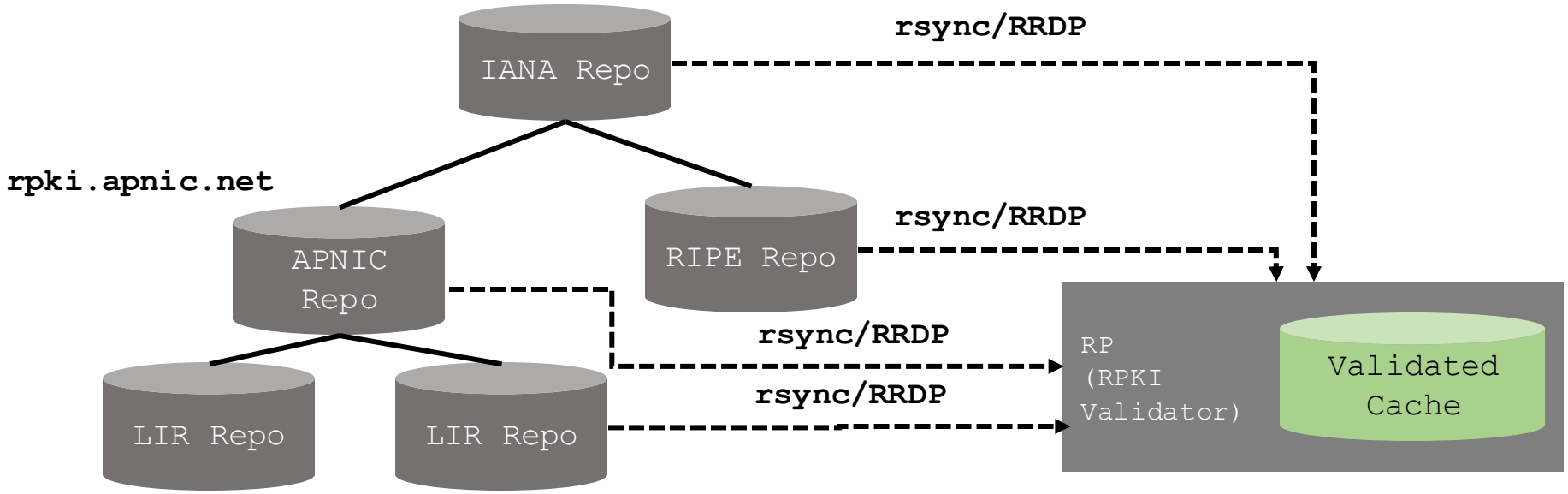


MyAPNIC GUI



RPKI Components

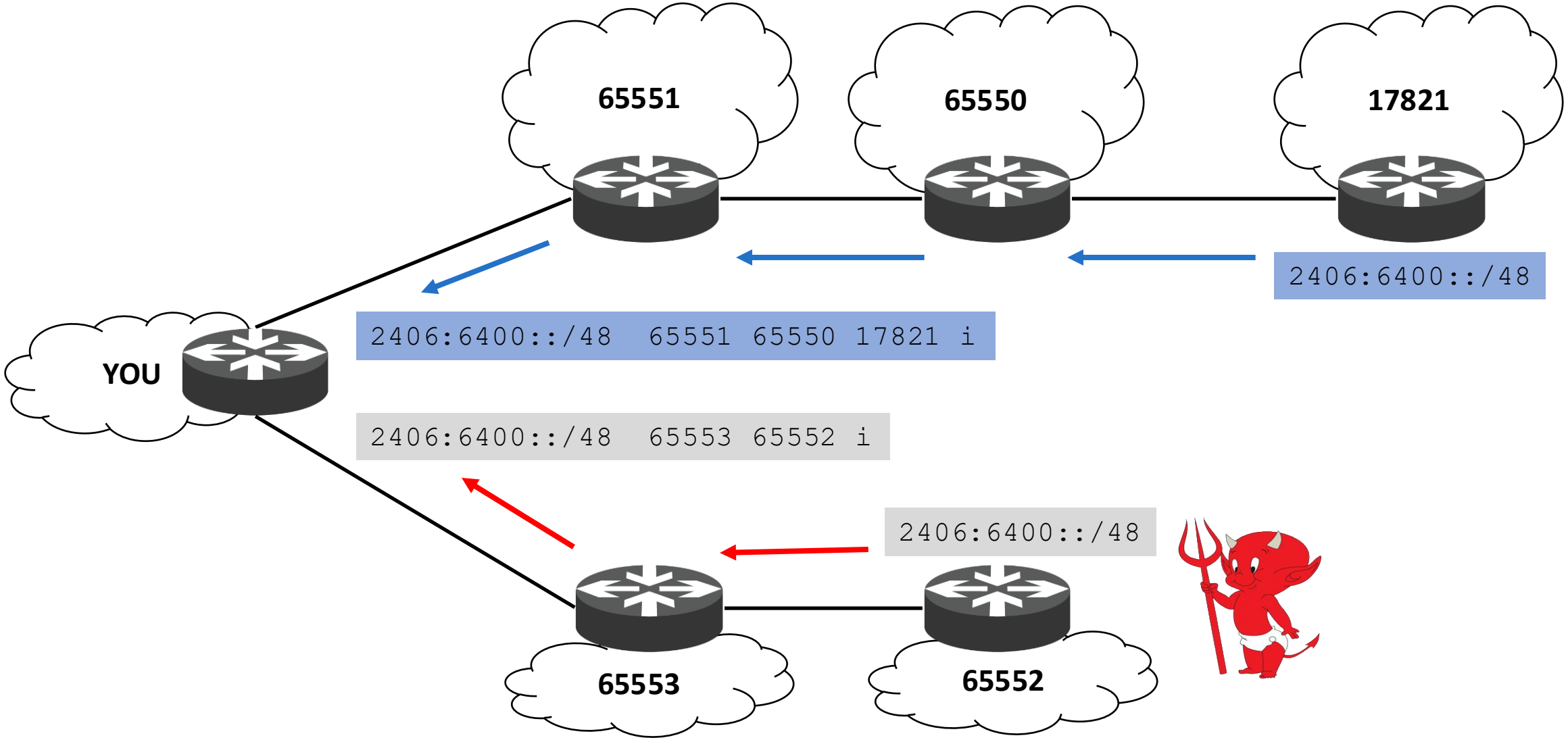
- **Relying Party (RP)**
  - RPKI Validator that gathers data (ROA) from the distributed RPKI repositories
  - Validates each entry's signature against the TA to build a "*Validated cache*"



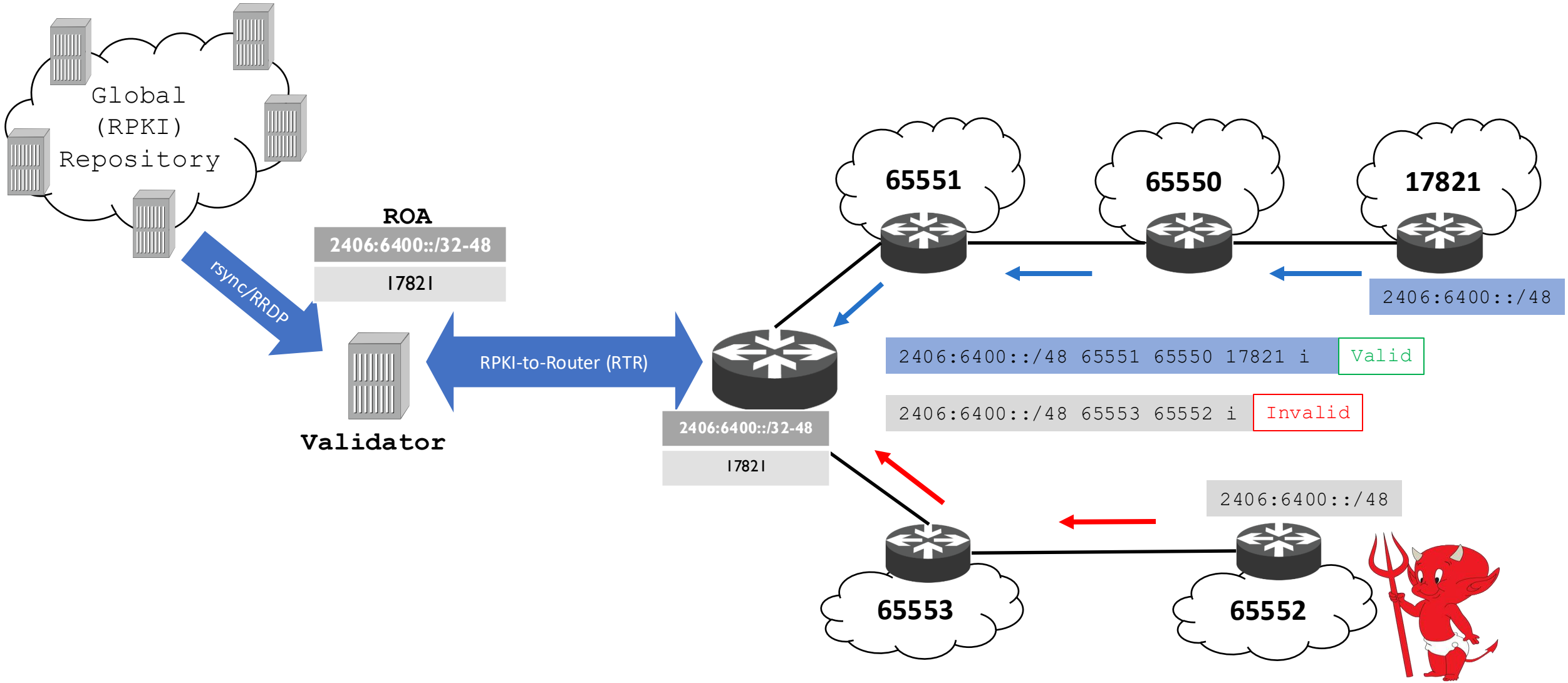
RPKI Service Models

- 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
```

RPKI ROV Configuration

Router Configuration (IOS)

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

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

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

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

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

OR

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

Configuration (IOS)

- Apply the route-map to inbound updates

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

Router Configuration (JunOS)

- Establishing session with the validator

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

Configuration (JunOS)

- Define policies based on the validation states

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

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


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

Router Configuration (JunOS)

- Apply the policy to inbound updates

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

RPKI Verification (IOS)

- IOS has only

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


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

RPKI Verification (IOS)

- Check the RTR session

```
#show 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

```
#show 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
```

```
#show 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)

```
#show 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
```

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

RPKI Verification (JunOS)

- Check the RPKI cache

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


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

RPKI Verification (JunOS)

- Check the RPKI cache

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

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

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

Would have been nice if per AF!

RPKI Verification (JunOS)

- Can filter per origin ASN

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


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


 IPv4 records: 2
 IPv6 records: 1
```

Check routes (JunOS)

```
>show route protocol bgp 202.144.128.0

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

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

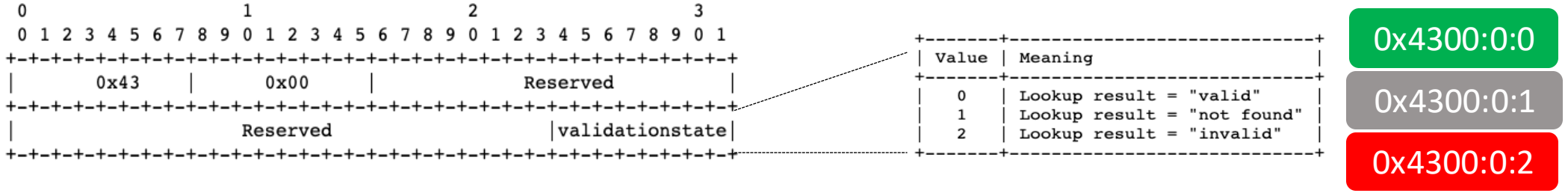
```
>show route protocol bgp 2001:201::/32

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

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

Propagating RPKI states to iBGP peers

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



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

Propagating RPKI states (IOS)

- Sender (one with RTR session)

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

Propagating RPKI states (IOS)

- Receiver (iBGP peer)

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

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

Propagating RPKI states (JunOS)

- Sender (router with an RTR session)

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

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

Propagating RPKI states (JunOS)

- Receiver (iBGP peer)

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

}
```

Any questions?