Securing Routing Protocols

ITU/APNIC/MICT IPv6 Security Workshop

23rd – 27th May 2016

Bangkok

Last updated 19th May 2016

Dealing with Threats Against Routing & Routing Protocols

- Routing Protocol Security applies equally to IPv4 and IPv6
  - Router Control Plane
  - Routing Protocol Neighbour Authentication
  - BGP Protocol Security
  - Remotely Triggered Black Hole Filtering
  - Route Origin Validation

Router Control Plane

Router Security Considerations

- Ensure limited access to routers & switches across the backbone
  - Addressing for device control plane access comes from dedicated address block
    - Don’t mix customer delegated and backbone infrastructure addressing
  - Filter at network edge and on device to only allow NOC access to control plane
    - Easier with IPv6 than with IPv4 (infrastructure addressing can come out of one /48)

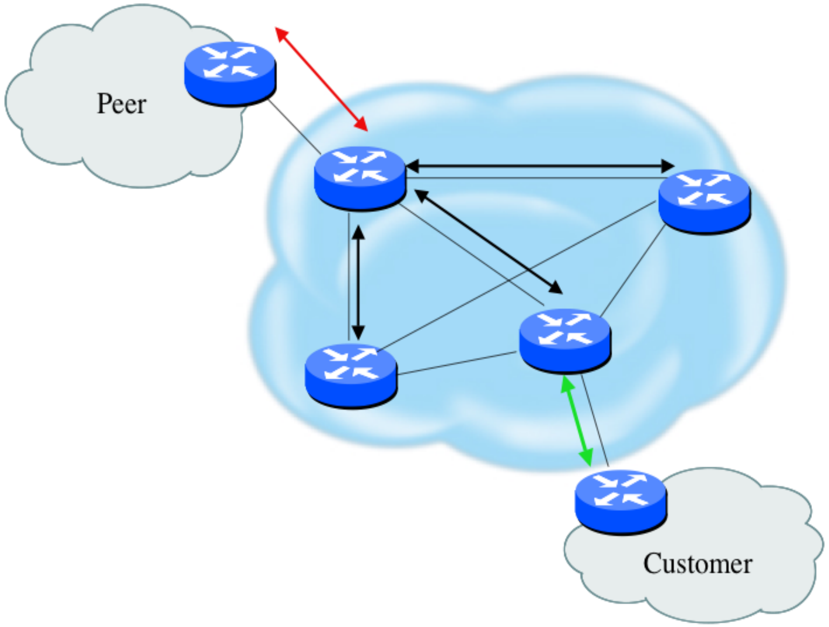
Router Security Considerations

- Segment backbone to simplify route distribution
- Design networks so outages don’t affect entire network but only portions of it
  - Tune IGP parameters for fast reconvergence
  - Use techniques such as Bi-Directional Forwarding Detection

Router Security Considerations

- Control router access
  - Watch for internal attacks on these systems
  - Use different passwords for standard and configuration access to router and monitoring system root access.
  - Never have role accounts
    - One account per user, centrally controlled
- Scanning craze for all kinds of ports – this will be never ending battle
  - Turn off unused features and remove unneeded configuration

Routing Control Plane



- MD-5 authentication
  - Some deploy at customer’s request
- Route filters limit what routes are believed from a valid peer
- Packet filters limit which systems can appear as a valid peer
- Limiting propagation of invalid routing information
  - Prefix filters
  - AS-PATH filters (trend is leaning towards this)
  - Route damping (latest consensus is that it causes more harm than good)
- Not yet possible to validate whether legitimate peer has authority to send routing update

Control Plane (Routing) Filters

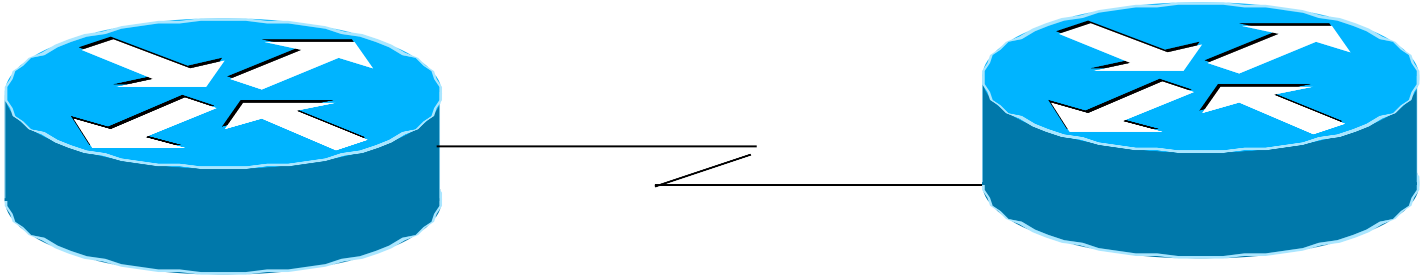
- Filter traffic destined TO your core routers
- Develop list of required protocols that are sourced from outside your AS and access core routers
  - Example: eBGP peering, GRE, IPSec, etc.
  - Use classification filters as required
- Identify core address block(s)
  - This is the protected address space
  - Summarization is critical for simpler and shorter filter lists

Neighbour Authentication

Why Use Neighbour Authentication

- Neighbour Authentication equates to data origin authentication and data integrity
  - Otherwise unauthorised routers can potentially compromise the network!
- In BGP, require TCP resets to be authenticated so malicious person can’t randomly send TCP resets
- In cases where routing information traverses shared networks, someone might be able to alter a packet or send a duplicate packet
- Routing protocols were not initially created with security in mind…..this needs to change….

Sample MD-5 Auth Configuration (OSPFv2)



```
interface Loopback0
 ip address 70.70.70.70 255.255.255.255
 ip ospf 10 area 0
!
interface Serial2
 ip address 192.16.64.2 255.255.255.0
 ip ospf 10 area 0
 ip ospf message-digest-key 1 md5 mk6
!
router ospf 10
 area 0 authentication message-digest
```

```
interface Loopback0
 ip address 172.16.10.36 255.255.255.255
 ip ospf 10 area 0
!
interface Serial1/0
 ip address 192.16.64.1 255.255.255.0
 ip ospf 10 area 0
 ip ospf message-digest-key 1 md5 mk6
!
router ospf 10
 area 0 authentication message-digest
```

Sample OSPFv3 IPSec Configuration

```
interface Loopback0
 ipv6 address 2001:DB8::1/128
 ipv6 ospf 100 area 0

interface FastEthernet0/0
 description Area 0 backbone interface
 ipv6 address 2001:DB8:2000::1/64
 ipv6 ospf 100 area 0

interface FastEthernet0/1
 description Area 1 interface
 ipv6 address 2001:DB8:1000::2/64
 ipv6 ospf 100 area 1
 ipv6 ospf authentication ipsec spi 257 sha1 7 091C1E59495546435A5D557879767A6166714054455755
020D0C06015B564D400F0E01050502035C0C

ipv6 router ospf 100
 router-id 10.0.0.1
 log-adjacency-changes detail
 passive-interface Loopback0
 timers spf 0 1
 timers pacing flood 15
 area 0 range 2001:DB8::/64
 area 0 range 2001:DB8:2000::/64
 area 1 range 2001:DB8:1000::/64
 area 0 encryption ipsec spi 256 esp aes-cbc 256 7 075F711C1E59495547435A5D557B7A757961677041
55445153050A0B00075D504B420D0C03070601005E0E53520D02514650520D5D5059771A195E4E5240455C5B sha1
7 00544356540B5B565F701D1F5848544643595E567879767A6166714054455052050D0C07005A574C42
```

Example for IS-IS

- Note that neighbour authentication for IS-IS is IP protocol independent:

```
key-chain isis-as42
 key 1
 key-string as42-pass
!
router isis as42
 authentication mode md5 level-2
 authentication key-chain isis-as42 level-2
!
 address-family ipv6
  multi-topology
!
```

BGP Security Techniques

- BGP prefix filtering
- BGP Community Filtering
- MD5 Keys on the eBGP and iBGP Peers
- Max Prefix Limits
- Max AS Path Length
- Prefer Customer Routes over Peer Routes (RFC 1998)
- GTSM (i.e. TTL Hack)
- Remote Trigger Black Hole (RTBH) Filtering

BGP Prefix Filtering

- Configuring BGP peering without using filters means:
  - All best paths on the local router are passed to the neighbour
  - All routes announced by the neighbour are received by the local router
  - Can have disastrous consequences
- Good practice is to ensure that each eBGP neighbour has inbound and outbound filter applied:

```
router bgp 64511
 neighbor 1.2.3.4 remote-as 64510
 neighbor 1.2.3.4 prefix-list as64510-in in
 neighbor 1.2.3.4 prefix-list as64510-out out
```

BGP Prefix Filtering

- If necessary to receive prefixes from any provider, care is required.
  - Don't accept default (unless you need it)
  - Don't accept your own prefixes
- Special use prefixes for IPv4 and IPv6:
  - http://www.rfc-editor.org/rfc/rfc6890.txt
- For IPv4:
  - Don't accept prefixes longer than /24 (?)
    - /24 was the historical class C
- For IPv6:
  - Don't accept prefixes longer than /48 (?)
    - /48 is the design minimum delegated to a site

BGP Prefix Filtering

- Check Team Cymru's list of "bogons"

  www.team-cymru.org/Services/Bogons/http.html
- For IPv4 also consult:

  www.rfc-editor.org/rfc/rfc6441.txt (BCP171)
- For IPv6 also consult:

  www.space.net/~gert/RIPE/ipv6-filters.html
- Bogon Route Server:

  www.team-cymru.org/Services/Bogons/routeserver.html
  - Supplies a BGP feed (IPv4 and/or IPv6) of address blocks which should not appear in the BGP table

Receiving IPv4 Prefixes

```
router bgp 100
 network 101.10.0.0 mask 255.255.224.0
 neighbor 101.5.7.1 remote-as 101
 neighbor 101.5.7.1 prefix-list in-filter in
!
ip prefix-list in-filter deny 0.0.0.0/0             ! Default
ip prefix-list in-filter deny 0.0.0.0/8 le 32       ! RFC1122 local host
ip prefix-list in-filter deny 10.0.0.0/8 le 32      ! RFC1918
ip prefix-list in-filter deny 100.64.0.0/10 le 32   ! RFC6598 shared addr
ip prefix-list in-filter deny 101.10.0.0/19 le 32   ! Local prefix
ip prefix-list in-filter deny 127.0.0.0/8 le 32     ! Loopback
ip prefix-list in-filter deny 169.254.0.0/16 le 32  ! Auto-config
ip prefix-list in-filter deny 172.16.0.0/12 le 32   ! RFC1918
ip prefix-list in-filter deny 192.0.0.0/24 le 32    ! RFC6598 IETF proto
ip prefix-list in-filter deny 192.0.2.0/24 le 32    ! TEST1
ip prefix-list in-filter deny 192.168.0.0/16 le 32  ! RFC1918
ip prefix-list in-filter deny 198.18.0.0/15 le 32   ! Benchmarking
ip prefix-list in-filter deny 198.51.100.0/24 le 32 ! TEST2
ip prefix-list in-filter deny 203.0.113.0/24 le 32  ! TEST3
ip prefix-list in-filter deny 224.0.0.0/3 le 32     ! Multicast & Expmnt
ip prefix-list in-filter deny 0.0.0.0/0 ge 25       ! Prefixes >/24
ip prefix-list in-filter permit 0.0.0.0/0 le 32
```

Receiving IPv6 Prefixes

```
router bgp 100
 network 2020:3030::/32
 neighbor 2020:3030::1 remote-as 101
 neighbor 2020:3030::1 prefix-list v6in-filter in
!
ipv6 prefix-list v6in-filter permit 64:ff9b::/96        ! RFC6052 v4v6trans
ipv6 prefix-list v6in-filter permit 2001::/32           ! Teredo
ipv6 prefix-list v6in-filter deny 2001::/23 le 128      ! RFC2928 IETF prot
ipv6 prefix-list v6in-filter deny 2001:2::/48 le 128    ! Benchmarking
ipv6 prefix-list v6in-filter deny 2001:10::/28 le 128   ! ORCHID
ipv6 prefix-list v6in-filter deny 2001:db8::/32 le 128  ! Documentation
ipv6 prefix-list v6in-filter permit 2002::/16           ! 6to4
ipv6 prefix-list v6in-filter deny 2002::/16 le 128      ! 6to4 subnets
ipv6 prefix-list v6in-filter deny 2020:3030::/32 le 128 ! Local Prefix
ipv6 prefix-list v6in-filter deny 3ffe::/16 le 128      ! Old 6bone
ipv6 prefix-list v6in-filter permit 2000::/3 le 48      ! Global Unicast
ipv6 prefix-list v6in-filter deny ::/0 le 128
```

Receiving Prefixes

- Paying attention to prefixes received from customers, peers and transit providers assists with:
  - The integrity of the local network
  - The integrity of the Internet
- Responsibility of all Network Operators to be good Internet citizens

BGP Community Filtering

- Network operators use BGP Communities for:
  - Internal policies
  - Policies for their customers
  - Policies towards their upstream providers
- Policies are aimed at ensuring routing system integrity within networks and between networks
- BGP Community references:
  - Specification (RFC1997) and Example Use (RFC1998)
  - http://bgp4all.com/ftp/isp-workshops/BGP\ Presentations/09-BGP-Communities.pdf

MD5 keys on BGP peerings

- Use passwords on all BGP sessions
  - Not being paranoid, VERY necessary
  - It’s a secret shared between you and your peer
  - If arriving packets don’t have the correct MD5 hash, they are ignored
  - Helps defeat miscreants who wish to attack BGP sessions
- Powerful preventative tool, especially when combined with filters and GTSM

```
router bgp 100
 address-family ipv6
  neighbor 2001:db8::1 remote-as 200
  neighbor 2001:db8::1 description Peering with AS200
  neighbor 2001:db8::1 password 7 030752180500
!
```

BGP Maximum Prefix Tracking

- Allow configuration of the maximum number of prefixes a BGP router will receive from a peer
- Two level control:
  - Warning threshold: log warning message
  - Maximum: tear down the BGP peering, manual intervention required to restart

```
neighbor <x.x.x.x> maximum-prefix <max> [restart N] [<threshold>] [warning-only]
```

- Optional keywords:
  - **`restart`** will restart the BGP session after N minutes
  - <threshold> sets the warning level (default 75%)
  - **`warning-only`** only sends warnings

Limiting AS Path Length

- Some BGP implementations have problems with long AS_PATHS
  - Memory corruption
  - Memory fragmentation
- Even using AS_PATH prepends, it is not normal to see more than 20 ASes in a typical AS_PATH in the Internet today
  - The Internet is around 5 ASes deep on average
  - Largest AS_PATH is usually 16-20 ASNs

```
neighbor x.x.x.x maxas-limit 15
```

Limiting AS Path Length

- Some announcements have ridiculous lengths of AS-paths:

```
*> 3FFE:1600::/24      22 11537 145 12199 10318 10566 13193 1930 2200
   3425 293 5609 5430 13285 6939 14277 1849 33 15589 25336 6830 8002
   2042 7610 i
```

This example is an error in one IPv6 implementation

```
*>i193.105.15.0        2516 3257 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 50404 50404 50404 50404
   50404 50404 50404 50404 50404 50404 50404 i
```

This example shows 100 prepends (for no obvious reason)

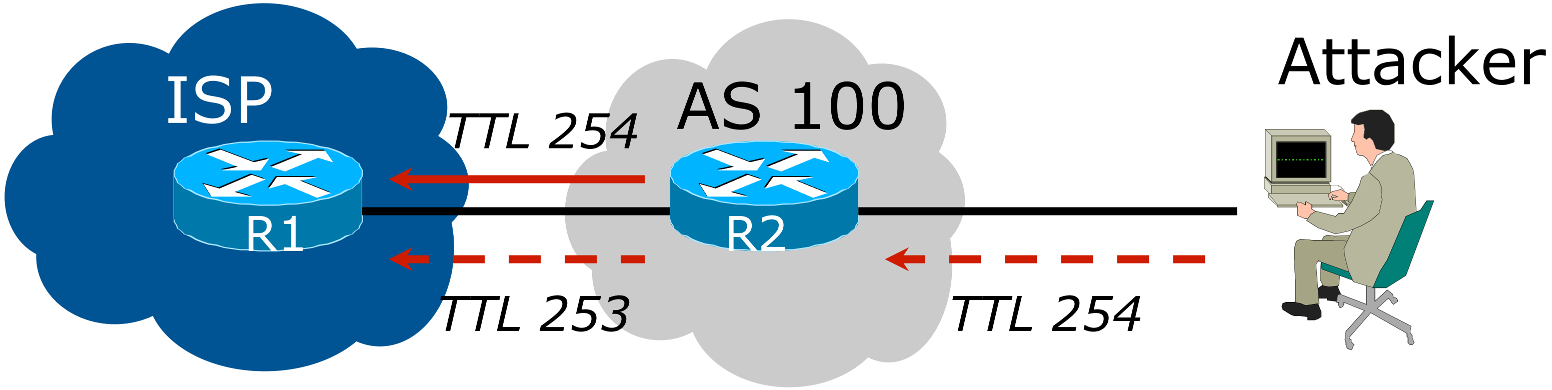
- If your implementation supports it, limit the maximum AS-path length you will accept

Customer routes vs Peer routes

- Common for end organisations to have more than one upstream provider
- Routes heard from the customer have to be preferred over the same routes heard from a peer
  - This is done by increasing BGP local preference for customer routes
  - Provides a degree of protection for its customer routes

GTSM: The BGP TTL "hack"

- Implement RFC5082 on BGP peerings
  - (Generalised TTL Security Mechanism)
  - Neighbour sets TTL to 255
  - Local router expects TTL of incoming BGP packets to be 254
  - No one apart from directly attached devices can send BGP packets which arrive with TTL of 254, so any possible attack by a remote miscreant is dropped due to TTL mismatch



BGP TTL "hack"

- TTL Hack:
  - Both neighbours must agree to use the feature
  - TTL check is much easier to perform than MD5
  - (Called BTSH – BGP TTL Security Hack)
- Provides "security" for BGP sessions
  - In addition to packet filters of course
  - MD5 should still be used for messages which slip through the TTL hack
  - See https://www.nanog.org/meetings/nanog27/presentations/meyer.pdf for more details

Remotely Triggered Black Hole Filtering

- A simple technique whereby the Network Operator can use their entire backbone to block mischievous traffic to a specific address within their network or their customers' network
- Chris Morrow's presentation at NANOG 30 in 2004 describing the technique:
  - https://www.nanog.org/meetings/nanog30/presentations/morrow.pdf
- Deployed and supported by many of the world's largest network operators

RTBH – How it works

- Network Operator deploys:
  - RTBH support across their entire backbone
    - Simply a null route for a specific next-hop address
    - (Router Null interfaces simply discard packets sent to them – negligible overhead in modern hardware)
  - A trigger router (usually in the NOC)
    - Talks iBGP with the rest of the backbone (typically as a client to route-reflectors in the core)
    - Used to trigger a blackhole route activity for any address under attack, as requested by a customer

RTBH – Backbone Configuration

- Network Operator sets up a null route for the 100::1 address on all the backbone routers which participate in BGP

```
ipv6 route 100::1/128 null 0  254
```

- 100::1 is part of 100::/64, the Discard Prefix, one of the reserved IPv6 address blocks listed in the IANA registry
  - http://www.iana.org/assignments/iana-ipv6-special-registry
  - It is not used or routed on the public Internet

RTBH – Trigger Router (1)

- Create a route-map to catch routes which need to be blackholed
  - Static routes can be tagged in Cisco IOS – we will tag routes to be blackholed with the value of 66
  - Set origin to be iBGP
  - Set local-preference to be 150
    - higher than any other local-preference set in the backbone
  - Set community to be *no-export* and internal marker community (ASN:666)
    - Don’t want prefix to leak outside the AS
  - Set next-hop to 192.0.2.1 (IPv4) or 100::1 (IPv6)

RTBH – Trigger Router (2)

- The whole route-map:

```
route-map v6blackhole-trigger permit 10
 description Look for Route 66
 match tag 66
 set local-preference 200
 set origin igp
 set community no-export 100:666
 set ip next-hop 100::1
!
route-map v6blackhole-trigger deny 20
 description Nothing else gets through
```

RTBH – Trigger Router (3)

- Then introduce the route-map into the BGP configuration
  - NB: the iBGP on the trigger router cannot use “next-hop-self” – Cisco IOS over writes the route-map originated next-hop with “next-hop-self”

```
router bgp 100
 address-family ipv6
  redistribute static route-map v6blackhole-trigger
  neighbor 2001:dbd::2 remote-as 100
  neighbor 2001:dbd::2 description iBGP with RR1
  neighbor 2001:dbd::2 update-source Loopback 0
  neighbor 2001:dbd::2 send-community
  neighbor 2001:dbd::3 remote-as 100
  neighbor 2001:dbd::3 description iBGP with RR2
  neighbor 2001:dbd::3 update-source Loopback 0
  neighbor 2001:dbd::3 send-community
!
```

RTBH – Trigger Router (4)

- To implement the trigger, simply null route whatever address or address block needs to be blackholed
  - With Tag 66

```
ipv6 route 2001:db8:f::e0/128 null0 tag 66
```

  - And this ensures that (for example) 2001:db8:f::e0/128 is announced to the entire backbone with next-hop 100::1 set

RTBH – End Result

- Prefixes which need to be null routed will come from the trigger router and look like this in the BGP table:

```
 *>i 2001:DB8:F::E0/128   100::1         0    200        0 i
```

- Routing entry for 2001:db8:f::e0 is this:

```
cr1>sh ipv6 route 2001:db8:f::e0
Routing entry for 2001:DB8:F::E0/128
  Known via "bgp 100", distance 200, metric 0, type
internal
  Route count is 1/1, share count 0
  Routing paths:
    100::1
      MPLS label: nolabel
      Last updated 00:00:03 ago
```

RTBH – End Result

- Routing entry for 100::1 is this:

```
cr1>sh ipv6 route 100::1
Routing entry for 100::1/128
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    directly connected via Null0
      Last updated 00:05:21 ago
```

- Traffic to 2001:db8:f::e0 is sent to null interface

Audit and Validate Your Routing Infrastructures

- Are appropriate paths used?
  - Check routing tables
  - Verify configurations
- Is router compromised?
  - Check access logs

Routing Security Conclusions

- Current routing protocols do not have adequate security controls
- Mitigate risks by using a combination of techniques to limit access and authenticate data
- Be vigilant in auditing and monitoring your network infrastructure
- Consider MD5 authentication
- Always filter routing updates….especially be careful of redistribution

But Wait…There’s More…

- RPKI – Resource Public Key Infrastructure, the Certificate Infrastructure to Support the other Pieces
    - We need to be able to authoritatively prove who owns an IP prefix and what AS(s) may announce it
    - Prefix ownership follows the allocation hierarchy (IANA, RIRs, ISPs, etc)
    - Origin Validation
        - Using the RPKI to detect and prevent mis-originations of someone else’s prefixes (early 2012)
    - AS-Path Validation AKA BGPsec
        - Prevent Attacks on BGP (future work)

BGP – Why Origin Validation?

- Prevent YouTube accident & Far Worse
- Prevents most accidental announcements
- Does not prevent malicious path attacks
- That requires ‘Path Validation’ and locking the data plane to the control plane, the third step, BGPsec

What is RPKI?

- Resource Public Key Infrastructure (RPKI)
- A robust security framework for verifying the association between resource holder and their Internet resources
- Created to address the issues in RFC 4593 “Generic Threats to Routing Protocols”
- Helps to secure Internet routing by validating routes
  - Proof that prefix announcements are coming from the legitimate holder of the resource

RFC 6480 – An Infrastructure to Support Secure Internet Routing (Feb 2012)

Benefits of RPKI - Routing

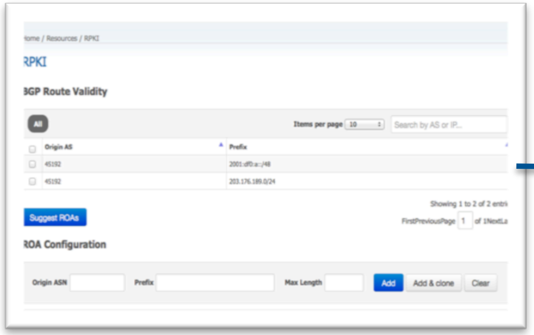
- Prevents route hijacking
  - A prefix originated by an AS without authorization
  - Reason: malicious intent
- Prevents mis-origination
  - A prefix that is mistakenly originated by an AS which does not own it
  - Also route leakage
  - Reason: configuration mistake / fat finger

BGP Security (BGPsec)

- Extension to BGP that provides improved security for BGP routing
- Being worked on by the SIDR Working Group at IETF
- Implemented via a new optional non-transitive BGP attribute that contains a digital signature
- Two components:
  - BGP Prefix Origin Validation (using RPKI)
  - BGP Path Validation

Issuing Party

- Internet Registries (RIR, NIR, Large LIRs)
- Acts as a Certificate Authority and issues certificates for customers
- Provides a web interface to issue ROAs for customer prefixes
- Publishes the ROA records



ome / Resources / RPKI

RPKI

3GP Route Validity

Items per page 10

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

Showing 1 to 2 of 2 entri

FirstPreviousPage 1 of 1NextLa

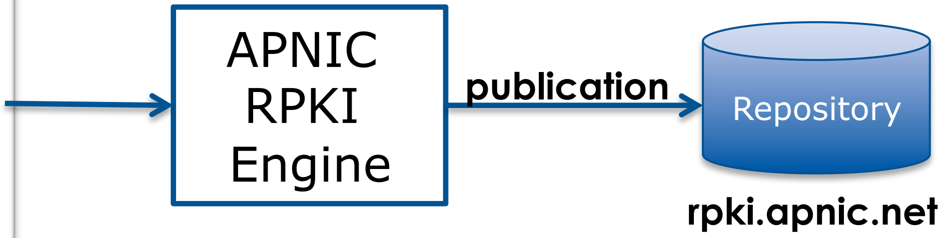
ROA Configuration

Origin ASN

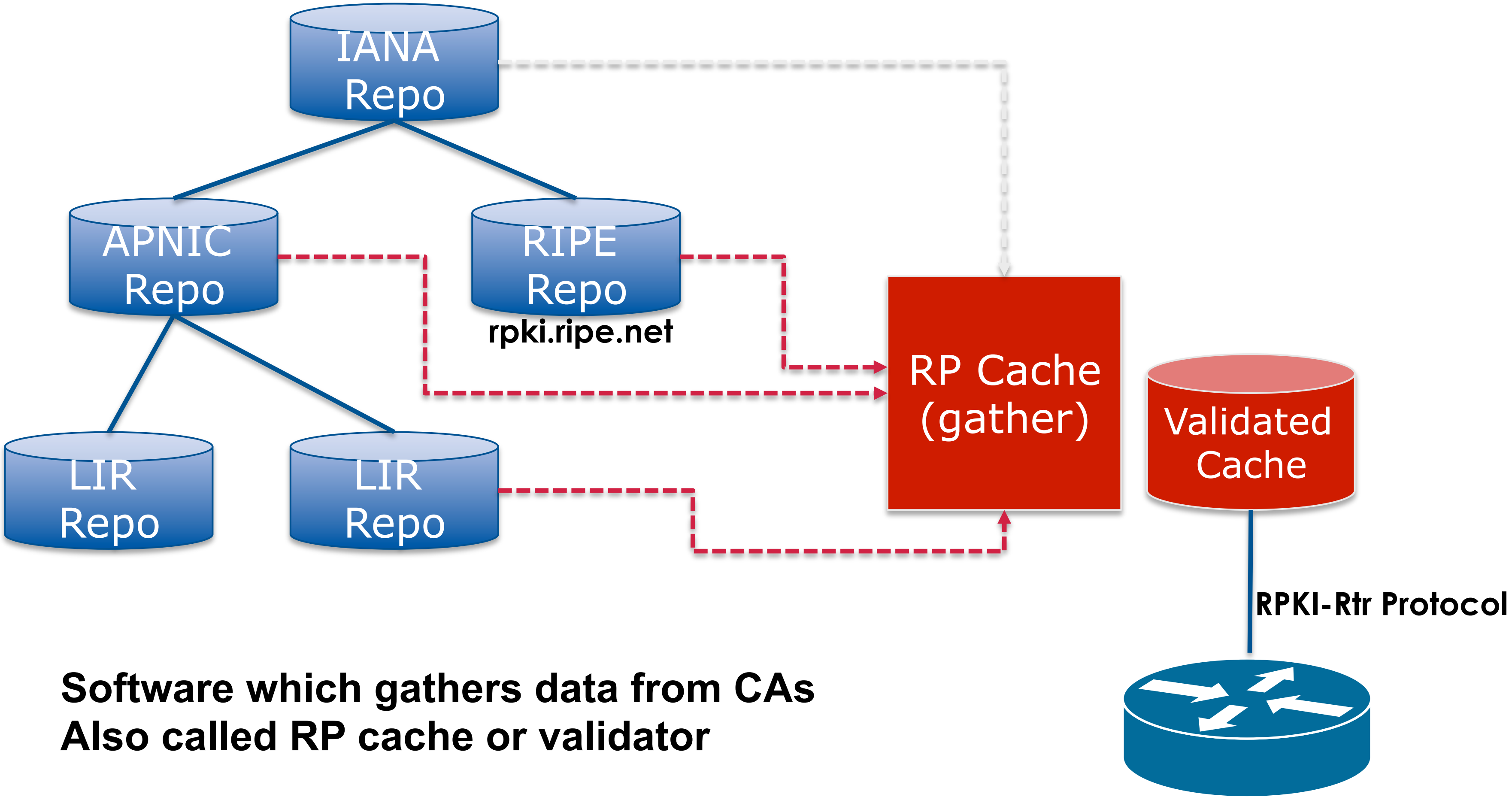
Prefix

Max Length

MyAPNIC GUI

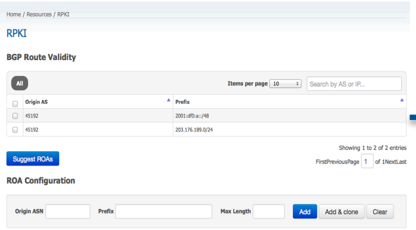


Relying Party (RP)



Software which gathers data from CAs
Also called RP cache or validator

RPKI Components



RPKI

BGP Route Validity

Items per page 10

Search by AS or IP...

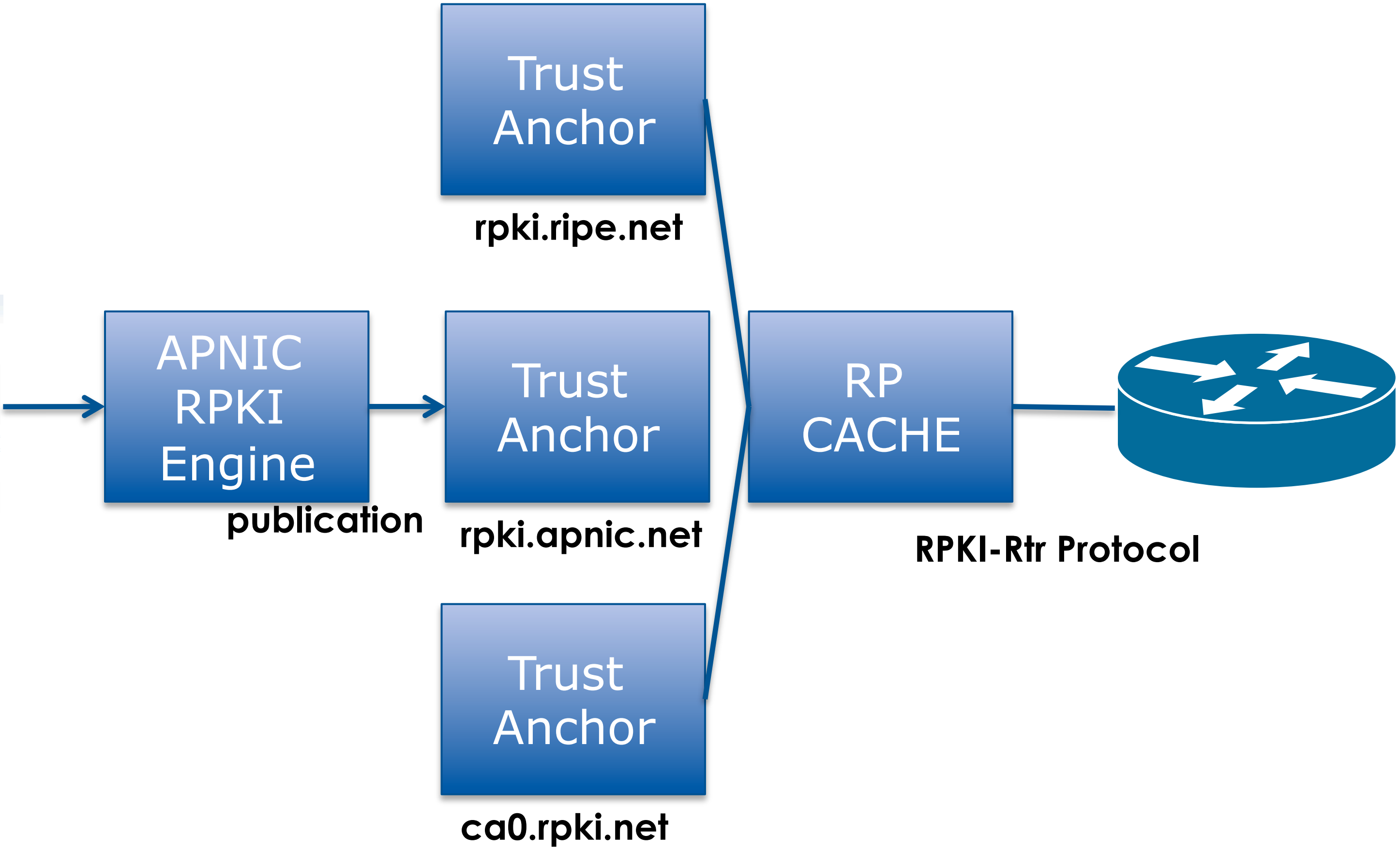
□	Origin AS	Prefix
□	45192	2001:df0:a::/48
□	45192	203.176.189.0/24

Showing 1 to 2 of 2 entries

ROA Configuration

Origin ASN ____ Prefix ____ Max Length ____

MyAPNIC GUI



Route Origin Authorization (ROA)

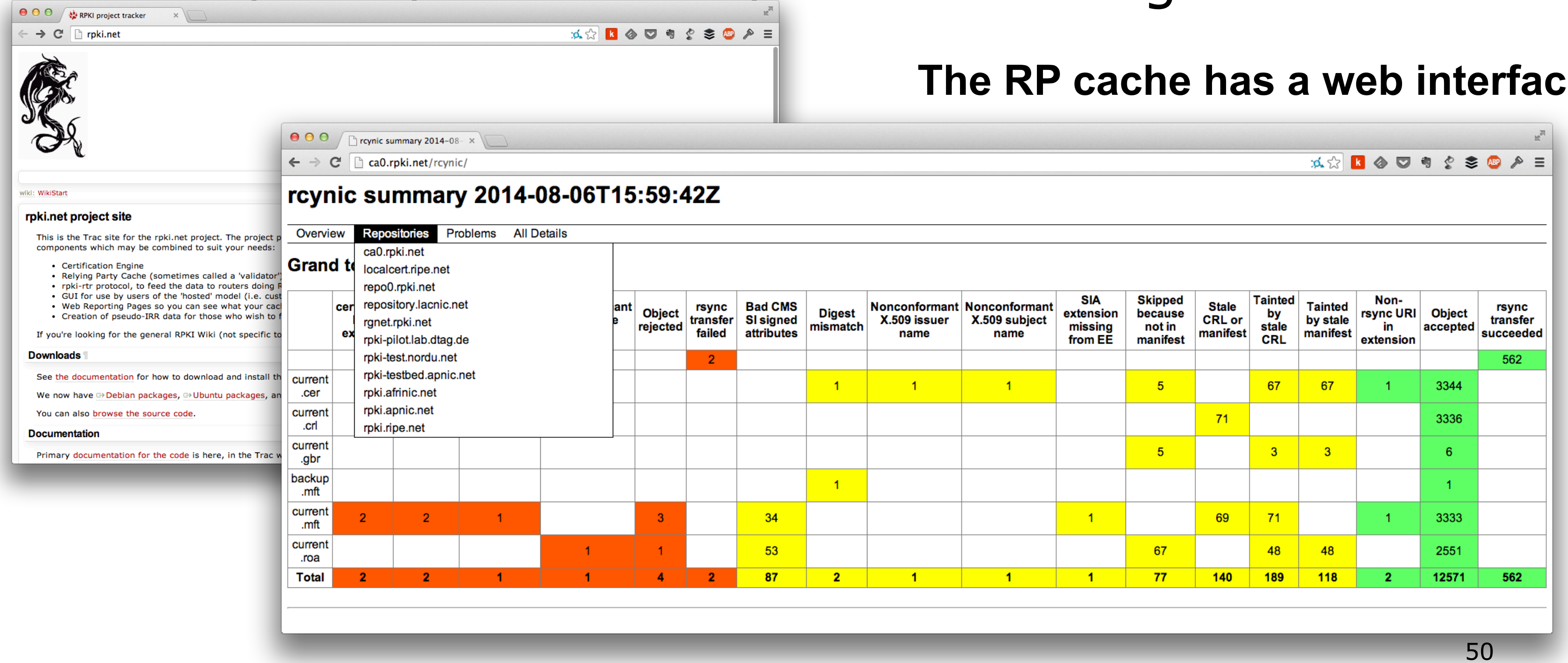
- A digital object that contains a list of address prefixes and one AS number
- It is an authority created by a prefix holder to authorize an AS Number to originate one or more specific route advertisements
- Publish a ROA using MyAPNIC

Router Origin Validation

- Router must support RPKI
- Checks an RP cache / validator
- Validation returns 3 states:
  - Valid = when authorization is found for prefix X
  - Invalid = when authorization is found for prefix X but not from ASN Y
  - Unknown = when no authorization data is found
- Vendor support:
  - Cisco IOS – available in release 15.2
  - Cisco IOS/XR – available in release 4.3.2
  - Juniper – available in release 12.2
  - Nokia (Alcatel Lucent) – close to availability

Build an RP Cache

- Download and install from rpki.net
  - Instructions here: https://trac.rpki.net/wiki/doc/RPKI/Installation/UbuntuPackages



The RP cache has a web interface

wiki: WikiStart

rpki.net project site

This is the Trac site for the rpki.net project. The project p
components which may be combined to suit your needs:

- Certification Engine
- Relying Party Cache (sometimes called a 'validator'
- rpki-rtr protocol, to feed the data to routers doing R
- GUI for use by users of the 'hosted' model (i.e. cust
- Web Reporting Pages so you can see what your cac
- Creation of pseudo-IRR data for those who wish to f

If you're looking for the general RPKI Wiki (not specific to

Downloads

See the documentation for how to download and install th

We now have Debian packages, Ubuntu packages, an

You can also browse the source code.

Documentation

Primary documentation for the code is here, in the Trac w

rcynic summary 2014-08-06T15:59:42Z

ca0.rpki.net
localcert.ripe.net
repo0.rpki.net
repository.lacnic.net
rgnet.rpki.net
rpki-pilot.lab.dtag.de
rpki-test.nordu.net
rpki-testbed.apnic.net
rpki.afrinic.net
rpki.apnic.net
rpki.ripe.net

Grand t

	cer ex			ant e	Object rejected	rsync transfer failed	Bad CMS SI signed attributes	Digest mismatch	Nonconformant X.509 issuer name	Nonconformant X.509 subject name	SIA extension missing from EE	Skipped because not in manifest	Stale CRL or manifest	Tainted by stale CRL	Tainted by stale manifest	Non-rsync URI in extension	Object accepted	rsync transfer succeeded
						2												562
current .cer								1	1	1		5		67	67	1	3344	
current .crl													71				3336	
current .gbr												5		3	3		6	
backup .mft								1									1	
current .mft	2	2	1		3		34				1		69	71		1	3333	
current .roa				1	1		53					67		48	48		2551	
Total	**2**	**2**	**1**	**1**	**4**	**2**	**87**	**2**	**1**	**1**	**1**	**77**	**140**	**189**	**118**	**2**	**12571**	**562**

Configure Router to Use Cache

- Point router to the local RPKI cache
  - Server listens on port 43779
  - Cisco IOS example:

```
router bgp 64512
 bgp rpki server tcp 10.0.0.3 port 43779 refresh 60
```

Some commands

- **`show ip bgp rpki servers`**
  - Provide connection status to the RPKI server
- **`show ip bgp rpki table`**
  - Shows the VRPs (validated ROA payloads)
- **`show ip bgp`**
  - Shows the BGP table with status indication next to the prefix

Check Server

```
br2# show ip bgp rpki servers
BGP SOVC neighbor is 10.0.0.3/43779 connected to port 43779
Flags 64, Refresh time is 60, Serial number is 1463446452
InQ has 0 messages, OutQ has 0 messages, formatted msg 1
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 25243
  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: 103.245.240.6, Local port: 39475
Foreign host: 202.144.129.117, Foreign port: 43779
Connection tableid (VRF): 0
```

RPKI Table

```
3115 BGP sovc network entries using 348880 bytes of memory
3249 BGP sovc record entries using 64980 bytes of memory

Network                Maxlen  Origin-AS  Source  Neighbor
2001:240::/32          32      2497       0       2C0F:FEB0:B:1::2/43779
2001:348::/32          64      7679       0       2C0F:FEB0:B:1::2/43779
2001:500:4::/48        48      10745      0       2C0F:FEB0:B:1::2/43779
2001:500:13::/48       48      393225     0       2C0F:FEB0:B:1::2/43779
2001:500:30::/48       48      10745      0       2C0F:FEB0:B:1::2/43779
2001:500:31::/48       48      393220     0       2C0F:FEB0:B:1::2/43779
2001:500:F0::/48       48      53535      0       2C0F:FEB0:B:1::2/43779
2001:504:32::/48       48      21654      0       2C0F:FEB0:B:1::2/43779
2001:608::/32          32      5539       0       2C0F:FEB0:B:1::2/43779
2001:610::/32          48      1103       0       2C0F:FEB0:B:1::2/43779
2001:610:240::/42      42      3333       0       2C0F:FEB0:B:1::2/43779
2001:620::/32          32      559        0       2C0F:FEB0:B:1::2/43779
2001:620::/29          29      559        0       2C0F:FEB0:B:1::2/43779
2001:630::/32          48      786        0       2C0F:FEB0:B:1::2/43779
...
```

Courtesy of SEACOM: http://as37100.net

BGP Table

```
br2# show ip bgp

RPKI validation codes: V valid, I invalid, N Not found

Network                 Metric LocPrf Weight Path
N*>i 1.0.4.0/24              0    250      0 58587 6939 4637 1221 38803 56203 i
N*>i 1.0.5.0/24              0    250      0 58587 6939 4637 1221 38803 56203 i
...
V*>i 1.9.0.0/16           9037    210      0 2914 4788 i
N*>i 1.10.8.0/24             0    250      0 58587 10026 18046 17408 58730 i
N*>i 1.10.64.0/24            0    210      0 2914 3491 133741 i
N*>i 1.10.66.0/24            0    210      0 2914 3491 133741 i
...
V*>i 1.37.0.0/16             0    220      0 4635 9304 4775 i
N*>i 1.38.0.0/23             0    210      0 2914 1273 55410 38266 ?
N*>i 1.38.0.0/17             0    210      0 2914 1273 55410 38266 {38266} ?
N*>i 1.38.4.0/24             0    210      0 2914 1273 55410 3826
...
```

RPKI BGP State: Valid

```
BGP routing table entry for 2001:240::/32, version 109576927
Paths: (2 available, best #2, table default)
  Not advertised to any peer
  Refresh Epoch 1
  37100 2497
    2C0F:FEB0:11:2::1 (FE80::2A8A:1C00:1560:5BC0) from
2C0F:FEB0:11:2::1 (105.16.0.131)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 37100:2 37100:22000 37100:22004 37100:22060
      path 0828B828 RPKI State valid
      rx pathid: 0, tx pathid: 0x0
```

Courtesy of SEACOM: http://as37100.net

RPKI BGP State: Invalid

```
BGP routing table entry for 5.8.240.0/23, version 420181499
Paths: (2 available, no best path)
 Not advertised to any peer
 Refresh Epoch 1
 37100 44217 3178
  105.22.32.1 from 105.22.32.1 (105.16.0.163)
   Origin IGP, metric 0, localpref 100, valid, external
   Community: 37100:2 37100:22000 37100:22004 37100:22025
   path 2B49B17C RPKI State invalid
   rx pathid: 0, tx pathid: 0
```

Courtesy of SEACOM: http://as37100.net

RPKI BGP State: Not Found

```
BGP routing table entry for 2001:200::/32, version 124240929
Paths: (2 available, best #2, table default)
  Not advertised to any peer
  Refresh Epoch 1
  37100 2914 2500
    2C0F:FEB0:11:2::1 (FE80::2A8A:1C00:1560:5BC0) from
2C0F:FEB0:11:2::1 (105.16.0.131)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 37100:1 37100:13
      path 19D90E68 RPKI State not found
      rx pathid: 0, tx pathid: 0x0
```

Courtesy of SEACOM: http://as37100.net

Using RPKI

- Network operators can make decisions based on RPKI state:
  - Invalid – discard the prefix
  - Not found – let it through (maybe low local preference)
  - Valid – let it through (high local preference)
- Some operators even considering making "not found" a discard event
  - But then Internet IPv4 BGP table would shrink to about 20k prefixes and the IPv6 BGP table would shrink to about 3k prefixes!

RPKI Summary

- All AS operators should consider deploying
- An important step to securing the routing system
  - Origin validation
- Doesn’t secure the path, but that’s the next hurdle to cross
- With origin validation, the opportunities for malicious or accidental mis-origination disappear

Routing Security

- Implement the recommendations in https://www.routingmanifesto.org/manrs
  1. Prevent propagation of incorrect routing information
     - Filter BGP peers, in & out!
  2. Prevent traffic with spoofed source addresses
     - BCP38 – Unicast Reverse Path Forwarding
  3. Facilitate communication between network operators
     - NOC to NOC Communication
  4. Facilitate validation of routing information
     - Route Origin Authorisation using RPKI

Summary

- Secure routing protocols
  - OSPF, IS-IS, BGP
- Secure access to the control plane
- Deploy RPKI
- Filtering helps everyone
  - PLEASE deploy anti-spoofing filters
  - PLEASE filter all BGP neighbours

Securing Routing Protocols

ITU/APNIC/MICT IPv6 Security Workshop

23rd – 27th May 2016

Bangkok