ROUTING SECURITY RPKI INTRODUCTION

Melchior Aelmans

maelmans@juniper.net

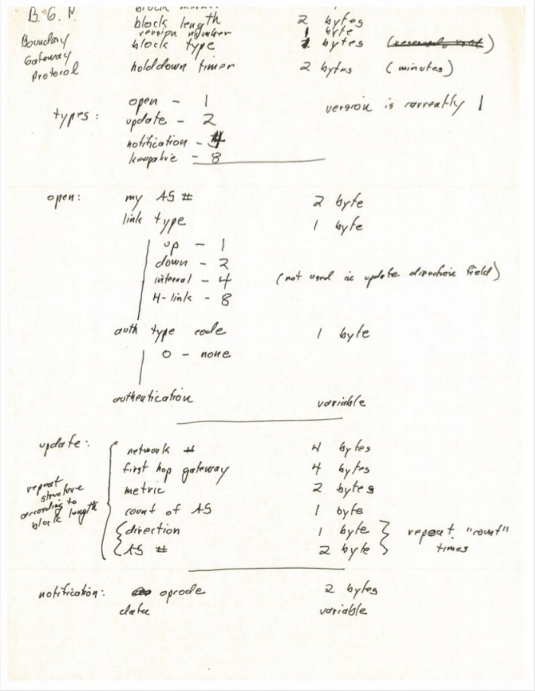
JUNIPER NETWORKS | Engineering Simplicity

AGENDA

- Current state of affairs
- Problem statement
  - Example MyEtherWallet / Twitter
- Resource Public Key Infrastructure (RPKI)
  - Route Origin Authorization (ROA)
  - Difference IRR and RPKI
  - RPKI Origin Validation examples
- RPKI Validators / Cache servers
- Implementation
- Next steps in Routing Security
  - BGPsec, ASPA, AS-Cones
- Statistics and additional resources

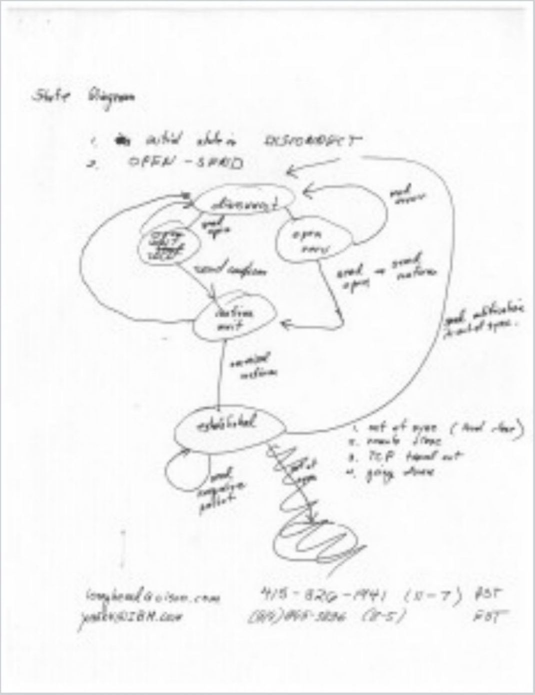
Current situation

HISTORY OF INTERNET ROUTING

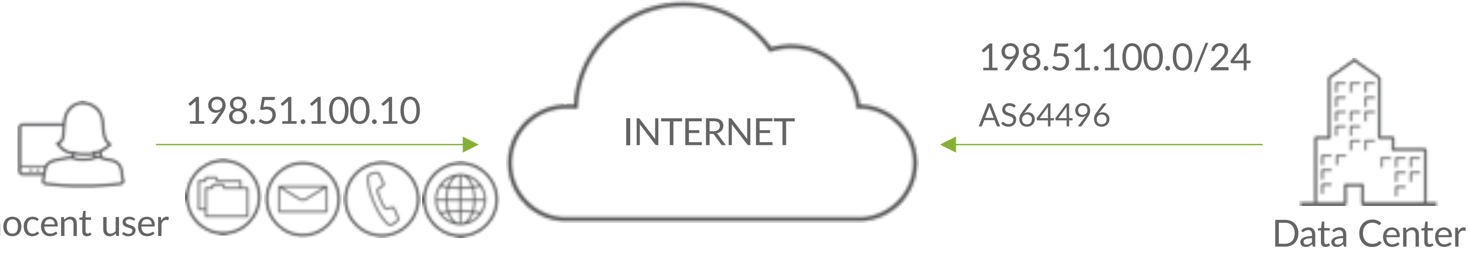


B.G.P.
Boundary
Gateway
Protocol

	block marker		
	block length	2 bytes	
	version number	1 byte	
	block type	~~2~~ 1 bytes	(~~reserved, must~~)
	holddown timer	2 bytes	(minutes)
types:	open - 1		version is currently 1
	update - 2		
	notification - ~~3~~ 4		
	keepalive - 8		
open:	my AS #	2 byte	
	link type	1 byte	
	up - 1		
	down - 2		
	internal - 4		(not used in update direction field)
	H-link - 8		
	auth type code	1 byte	
	0 - none		
	authentication	variable	
update:	network #	4 bytes	
repeat structure according to block length	first hop gateway	4 bytes	
	metric	2 bytes	
	count of AS	1 byte	
	direction	1 byte	repeat "count" times
	AS #	2 byte	
notification:	~~op~~ opcode	2 bytes	
	data	variable	



THE PERFECT WORLD



PREFIX FILTERS, IRR FILTERING, PEER LOCK, ETC. ARE ALL IN PLACE?

- Prefix filters
- Peer lock
- “bignetworks” filter
- Bogon ASN filtering
- Bogon Prefix filtering
- Filter long ASN path
- Filter small prefixes
- IRR filtering

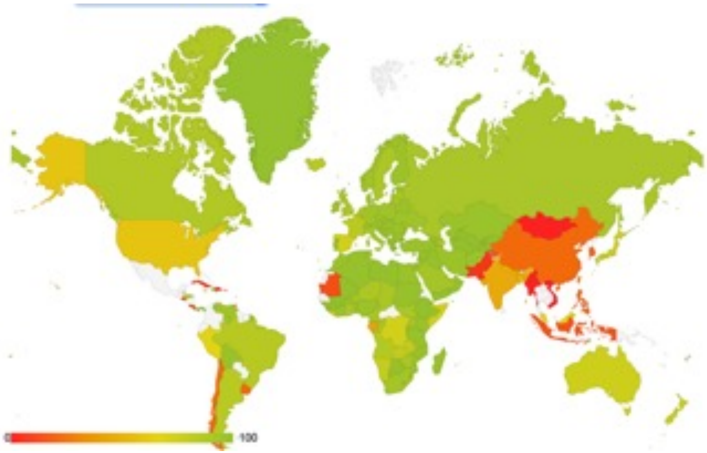
THE PERFECT WORLD…OR NOT (YET)?

I know all my customers and peers (all friends) and have filters and strict IRR applied.

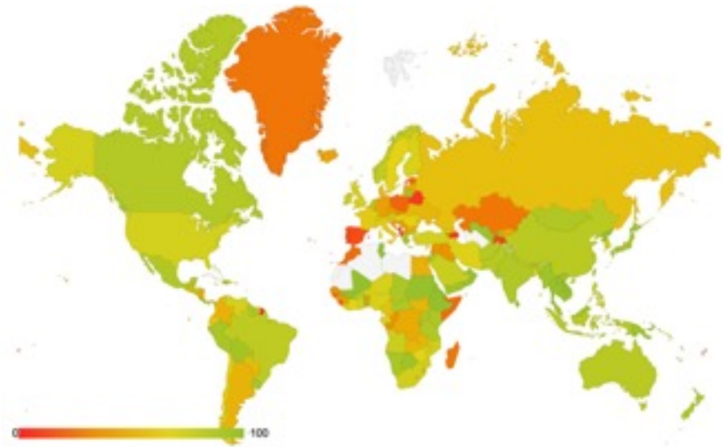
However…

- Prefix filters don’t care about the originating ASN or AS-PATH
- Peer Lock doesn’t cover every network and is arbitrary
- Filtering small prefix outbound is an issue for DDoS mitigation
- Downstream customers might use private ASN
- IRR databases are far from correct, are incomplete or contain outdated data

IRR DATABASE ACCURACY



RIPE IRR



RADB IRR

Problem is that no single IRR database is consistent and can be 100% trusted. Which one doe you trust?

BGP HIJACKS ARE HAPPENING

June 2019 - European telecommunication networks

A Swiss data centre hosting company accidentally leaked over 70,000 routes from its internal routing table to China Telecom. Instead of ignoring the BGP leak, China Telecom re-announced these routes as its own and declared itself as the shortest way to reach the network of the Swiss data centre operator and other nearby European telecommunication companies and ISPs.

Some of the most impacted European networks included Swisscom (AS3303) of Switzerland, KPN (AS1130) of Holland, and Bouygues Telecom (AS5410) and Numericable-SFR (AS21502) of France. This particular incident was severe, lasting over two hours. Users of the affected networks suffered slow connections and denial of service to some servers.

April 2020 - Akamai, Amazon and Alibaba

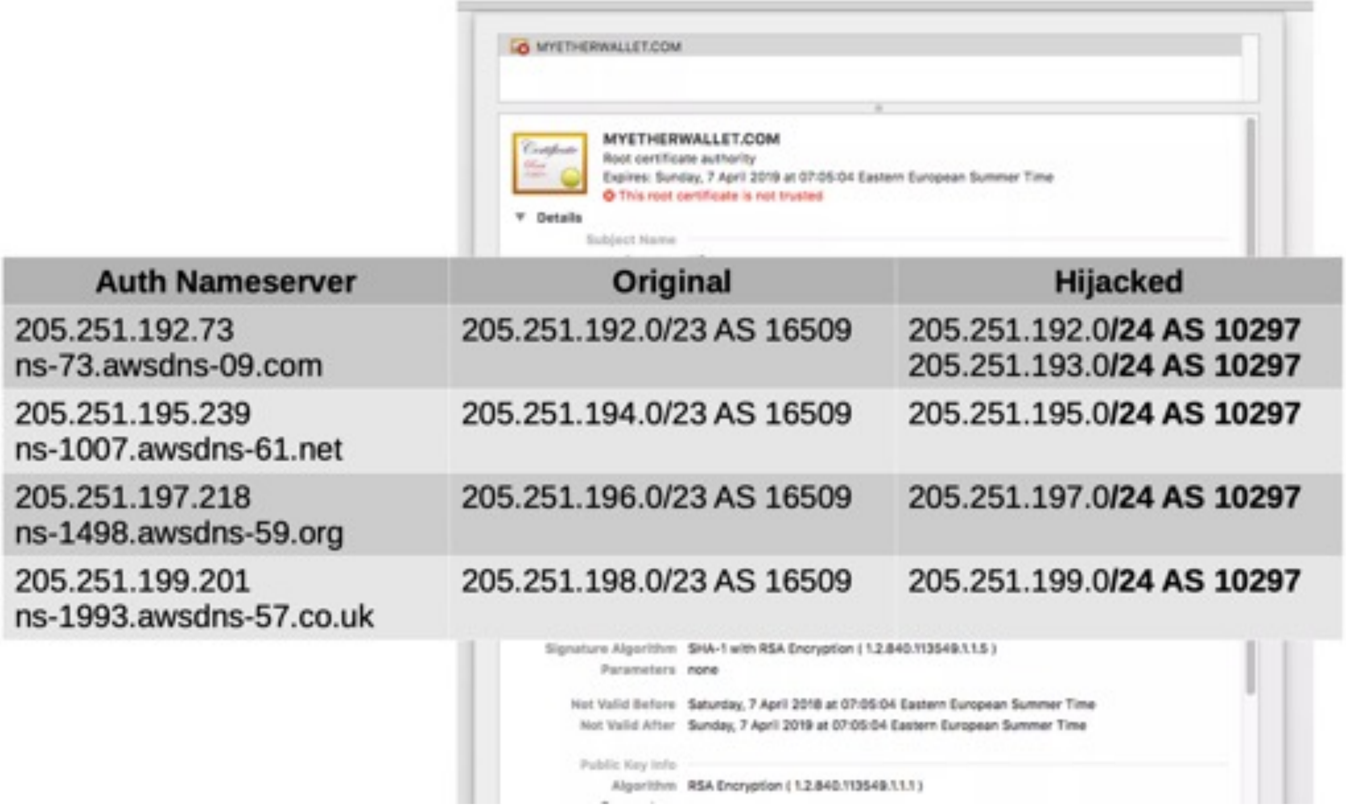
A massive BGP hijack involving over 8,800 prefixes affected companies such as Akamai, Amazon and Alibaba on April 1, 2020. Initiated by a Rostelecom user, the attack caused service disruptions throughout the world. It is currently unknown how much data was leaked or for what purposes, but it generally acknowledged that stricter network filtering by Rostelecom could have prevented the attack.

September 2020 - Telstra

500 prefixes wrongfully advertised as belonging to Telstra caused lengthy data detours via the Australian telecommunications company in September 2020. Telstra later apologised for the unintentional hijacking, stating the incident was caused by post verification testing to address an unrelated software bug. While this incident may have caused widespread connectivity challenges, no data or personal information is suspected to be breached.

Source: https://www.anapaya.net/blog/border-gateway-protocol-hijacking-examples-and-solutions

AMAZON ROUTE53 / MYETHERWALLET.COM HIJACK



MYETHERWALLET.COM

MYETHERWALLET.COM
Root certificate authority
Expires: Sunday, 7 April 2019 at 07:05:04 Eastern European Summer Time
This root certificate is not trusted

▼ Details

Subject Name

Auth Nameserver	Original	Hijacked
205.251.192.73 ns-73.awsdns-09.com	205.251.192.0/23 AS 16509	205.251.192.0/**24 AS 10297** 205.251.193.0/**24 AS 10297**
205.251.195.239 ns-1007.awsdns-61.net	205.251.194.0/23 AS 16509	205.251.195.0/**24 AS 10297**
205.251.197.218 ns-1498.awsdns-59.org	205.251.196.0/23 AS 16509	205.251.197.0/**24 AS 10297**
205.251.199.201 ns-1993.awsdns-57.co.uk	205.251.198.0/23 AS 16509	205.251.199.0/**24 AS 10297**

Signature Algorithm SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
Parameters none

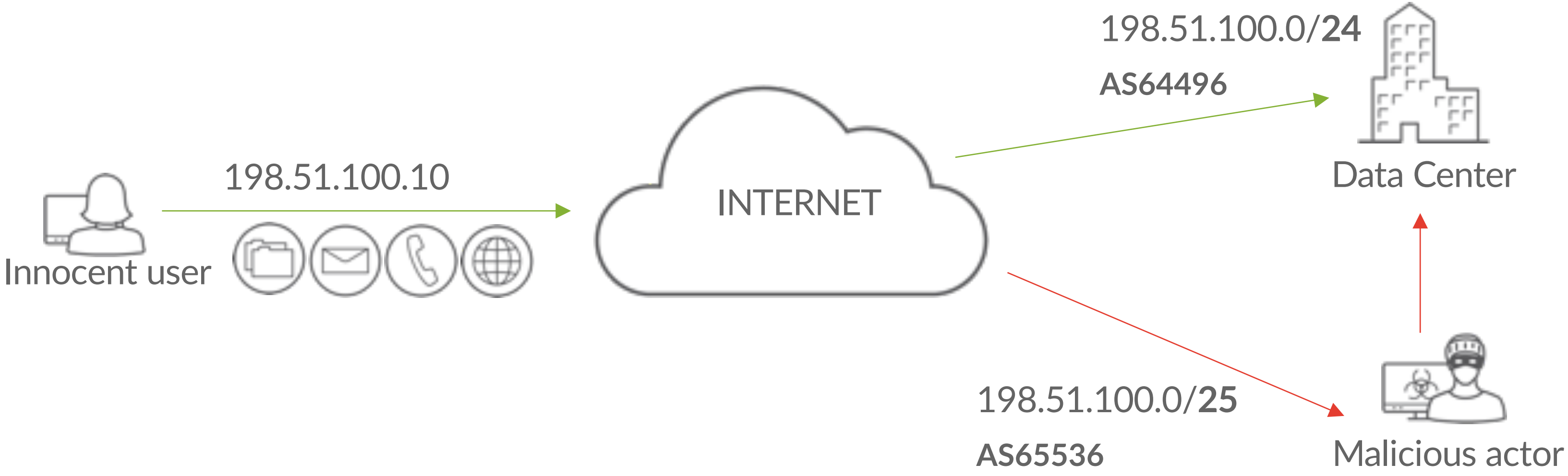
Not Valid Before Saturday, 7 April 2018 at 07:05:04 Eastern European Summer Time
Not Valid After Sunday, 7 April 2019 at 07:05:04 Eastern European Summer Time

Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)

IMPACT: HOW BAD WAS IT?

- AS 10297 upstreams (NTT, Cogent, Level3) & Equinix route server blocked the hijack
- Hijack was unnoticed for 2 hours
- Some peers of AS 10297 (Google, Hurricane Electric, BBOI, others) accepted the hijack
- Hijack impact was limited thanks to filters, but still an absolute disaster for all involved
- Was also using a rogue HTTPS certificate so users clicked through certificate errors
- More info: https://www.internetsociety.org/blog/2018/04/amazons-route-53-bgp-hijack/
- **The hackers were able to steal approximately $400,000 in cryptocurrency. Some 'news sites' even reported $17 million was stolen!**

WHAT HAPPENED TO OUR INNOCENT USER

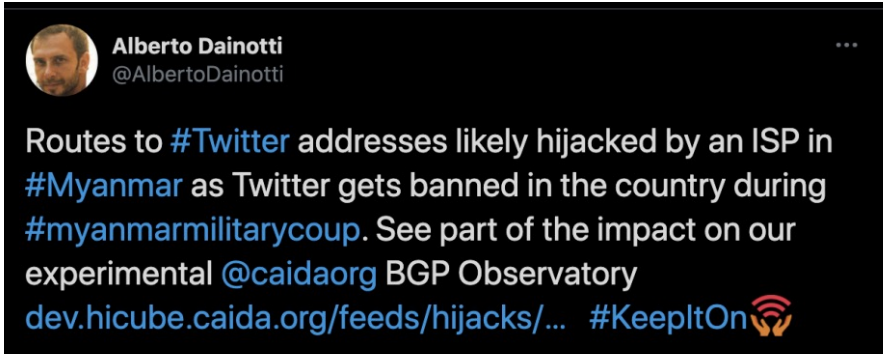


MOST RECENT BGP HIJACK

On February 1, 2021, Myanmar's army took power in a coup against the elected government and detained the civilian leadership. Restrictions to the Internet were reported as people woke up to the news. Internet Society's Insight portal covered the Internet shutdown in detail.

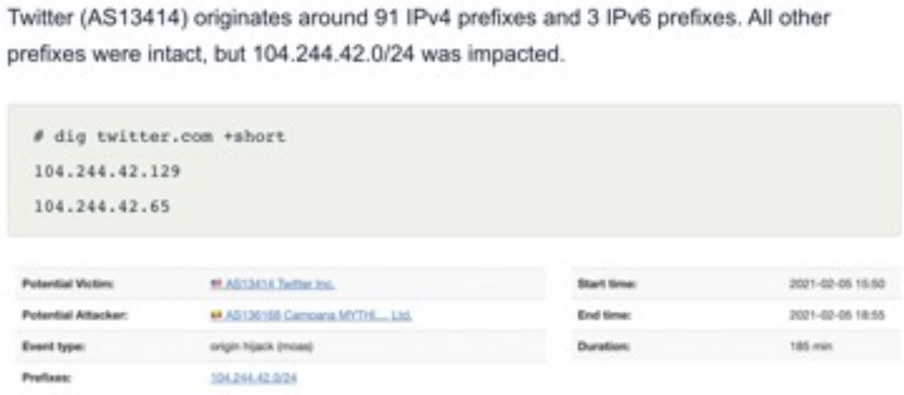
On Friday, 5 February, the Myanmar Ministry of Transport and Communications issued a notification to mobile networks and internet service providers (ISPs) in the country to block Twitter and Instagram. ISPs and Telcos operating in the country followed the order and blocked said services.

After a few hours, Dr Alberto Dainotti (Research Scientist and Principal Investigator at CAIDA) shared the following tweet suggesting a hijack attempt originating from one of the ISPs in Myanmar.



Alberto Dainotti
@AlbertoDainotti

Routes to #Twitter addresses likely hijacked by an ISP in #Myanmar as Twitter gets banned in the country during #myanmarmilitarycoup. See part of the impact on our experimental @caidaorg BGP Observatory dev.hicube.caida.org/feeds/hijacks/… #KeepItOn



Twitter (AS13414) originates around 91 IPv4 prefixes and 3 IPv6 prefixes. All other prefixes were intact, but 104.244.42.0/24 was impacted.

```
# dig twitter.com +short
104.244.42.129
104.244.42.65
```

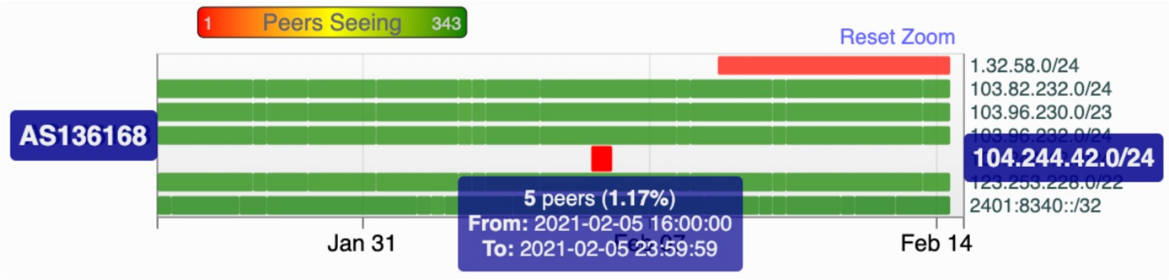
Potential Victim:	AS13414 Twitter Inc.	Start time:	2021-02-05 15:50
Potential Attacker:	AS136168 Campana MYTHI… Ltd.	End time:	2021-02-05 18:55
Event type:	origin hijack (moas)	Duration:	185 min
Prefixes:	104.244.42.0/24		

https://www.manrs.org/2021/02/did-someone-try-to-hijack-twitter-yes/

CAIDA BGP Observatory

MOST RECENT BGP HIJACK

Even though the propagation wasn't widespread, it must have impacted the end users of those ASNs who accepted this bogus announcement. It could have been much worse.

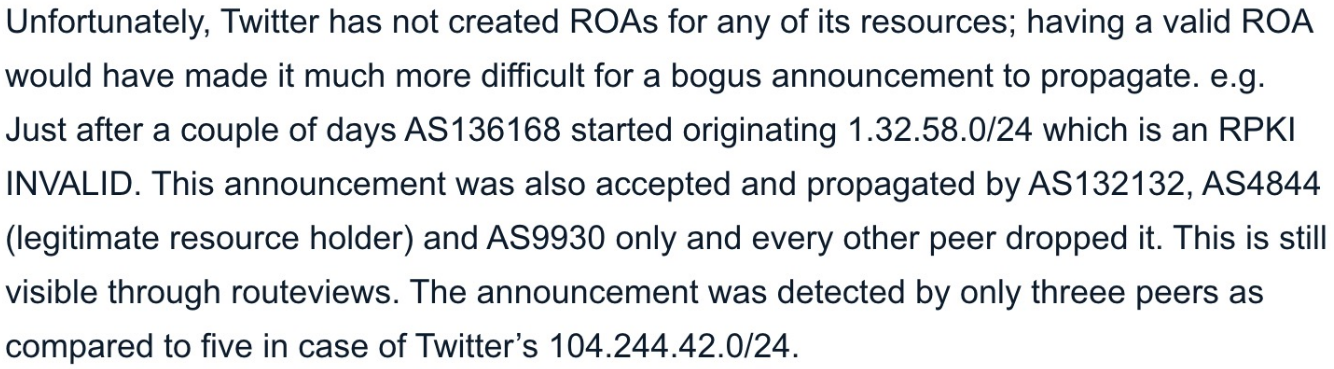


RIPE Stats

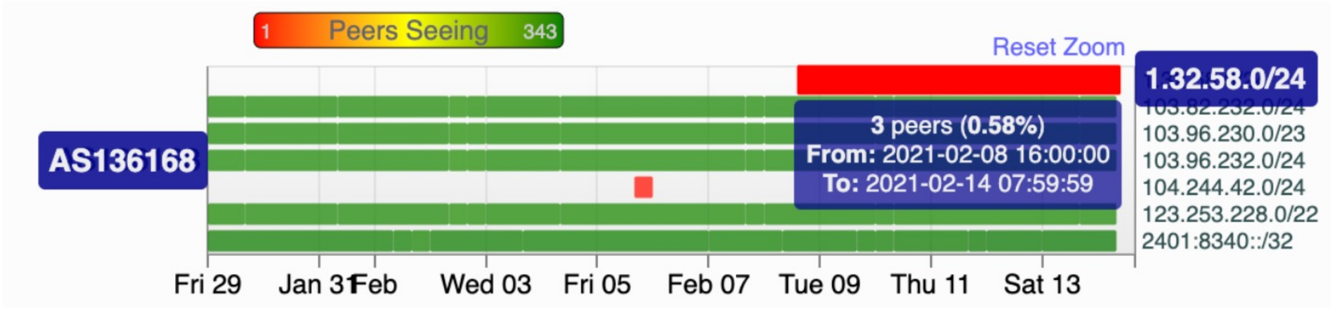
It is very encouraging to see that many service providers didn't accept the bogus announcement by AS136168. As mentioned in RIPE Stats, only five peers were able to detect this and it did not create any widespread issue on the Internet. Except the handful networks mentioned above, all other service providers dropped this announcement on the basis of route object.

https://www.manrs.org/2021/02/did-someone-try-to-hijack-twitter-yes/

MOST RECENT BGP HIJACK – RPKI WOULD HAVE LIMITED THE IMPACT



Unfortunately, Twitter has not created ROAs for any of its resources; having a valid ROA would have made it much more difficult for a bogus announcement to propagate. e.g. Just after a couple of days AS136168 started originating 1.32.58.0/24 which is an RPKI INVALID. This announcement was also accepted and propagated by AS132132, AS4844 (legitimate resource holder) and AS9930 only and every other peer dropped it. This is still visible through routeviews. The announcement was detected by only threee peers as compared to five in case of Twitter’s 104.244.42.0/24.



https://www.manrs.org/2021/02/did-someone-try-to-hijack-twitter-yes/

BGP Origin Validation

Resource Public Key Infrastructure (RPKI)

SO NOW WHAT? WHAT IS RPKI?

Photo by Markus Spiske on Unsplash

BGP Origin Validation using RPKI

Resource Public Key Infrastructure (RPKI) is a method of cryptographic signing records that associate a prefix with an originating AS number.

All the five RIRs (AFRINIC, APNIC, ARIN, LACNIC & RIPE) provide a method for members to take a prefix/ASN pair and sign those with a ROA (Route Origin Authorization) record.

The ROA can then be used by operators to validate route advertisements. They are sure a route advertisement is intended by the legitimate owner.

Summary: “RPKI is a database holding statements of the rightful resource owner which can be cryptographically verified”

THE MAJOR DIFFERENCES BETWEEN IRR AND RPKI

With IRR, for most IRR databases operators don’t know if the owner of a prefix actually was the same entity that created the IRR route objects.

→ *With RPKI on the other hand, they have can trust the owner (and only the owner) created the ROAs!*

IRR does not offer “proof of termination”, there are many IRR databases and the same set of databases is not used by everyone. NTT uses a different list of mirrors than RADB, than YYCIX, than AMS-IX, than Level3, etc etc.

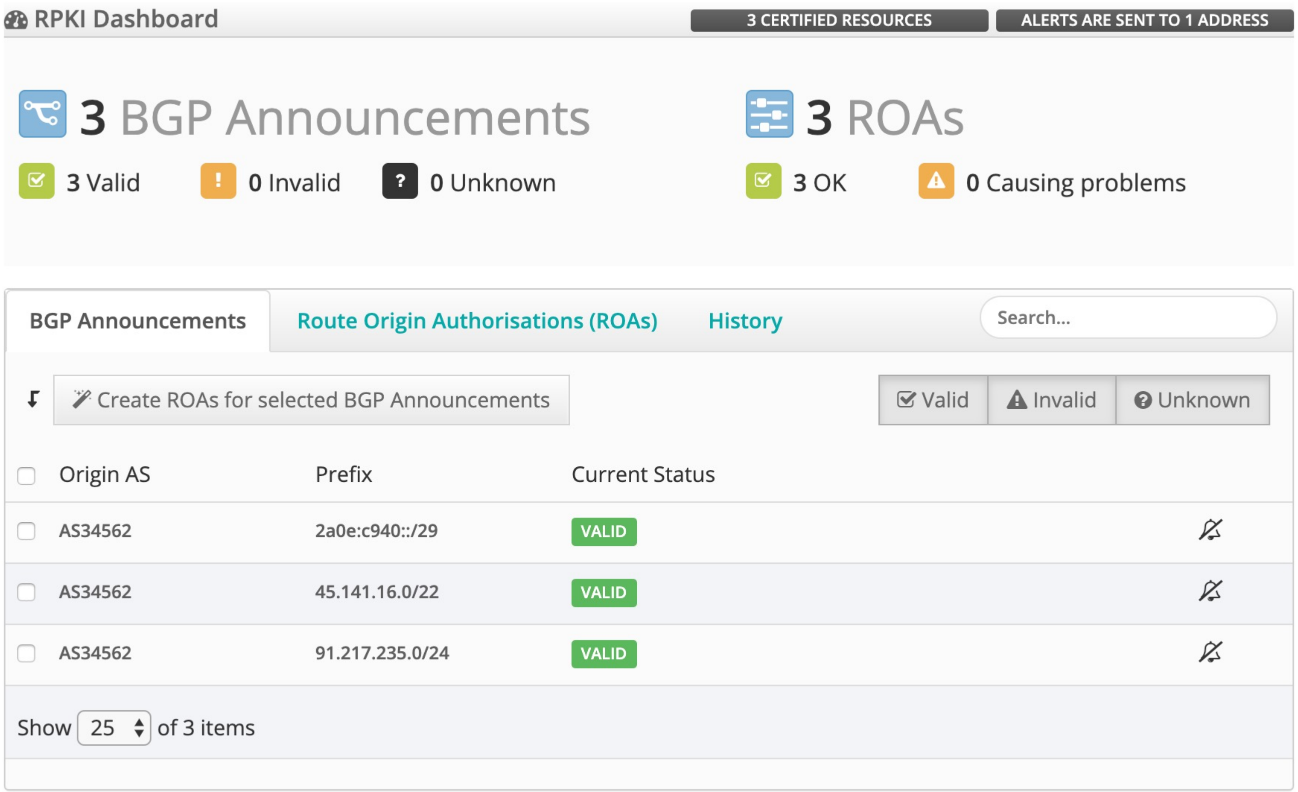
→ *With RPKI – the internet community agreed to use 5 “roots” (ARIN, RIPE, LACNIC, APNIC, AFRINIC)*

WHAT DO THESE TWO DIFFERENCES MEAN?

Operators can’t apply IRR based filters on their large (tier 1) peering partners, since they don’t have a full list of every IRRs that exist, nor do they know the order in which to parse them.

With RPKI they know for sure that they are honouring the legitimate prefix owner’s wishes when doing so!

HOW TO CREATE ROAS



RPKI Dashboard

3 CERTIFIED RESOURCES | ALERTS ARE SENT TO 1 ADDRESS

3 BGP Announcements

✓ 3 Valid | ! 0 Invalid | ? 0 Unknown

3 ROAs

✓ 3 OK | ⚠ 0 Causing problems

BGP Announcements | Route Origin Authorisations (ROAs) | History | Search...

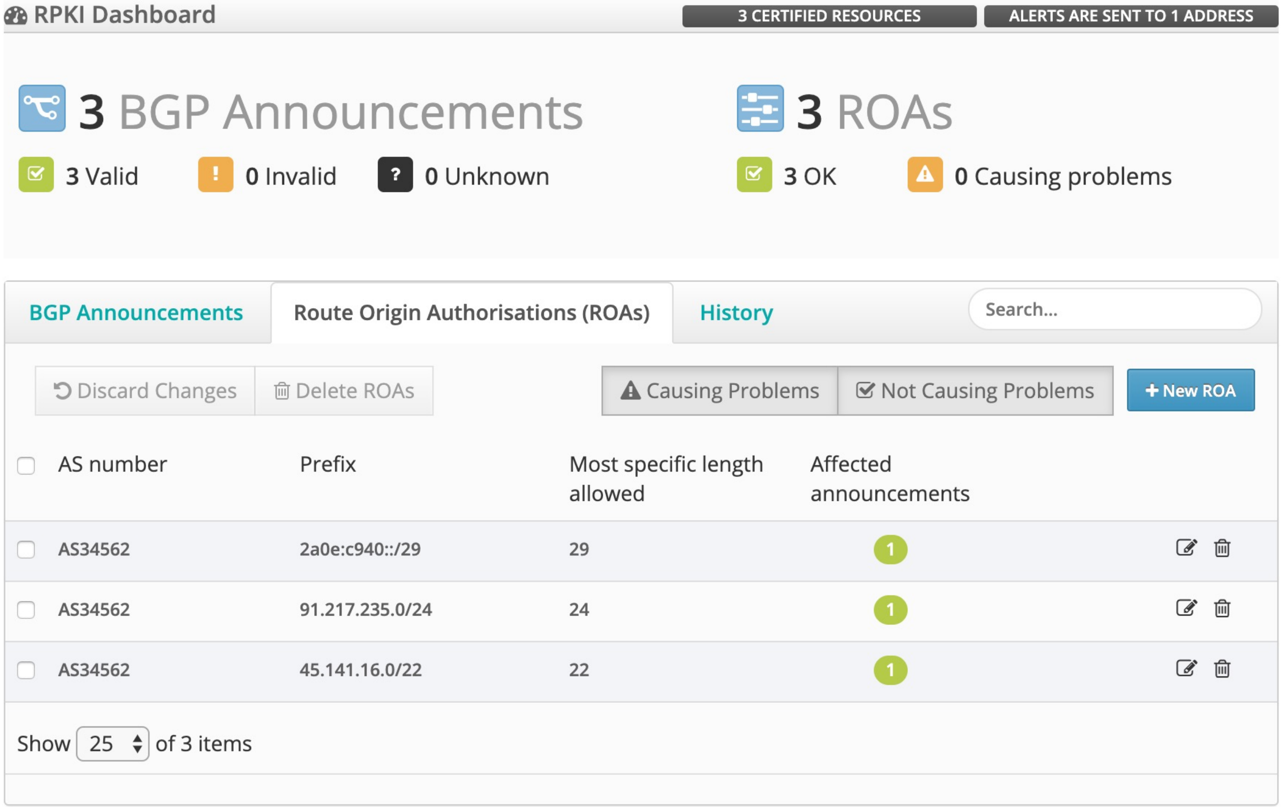
Create ROAs for selected BGP Announcements

✓ Valid | ⚠ Invalid | ? Unknown

□	Origin AS	Prefix	Current Status
□	AS34562	2a0e:c940::/29	VALID
□	AS34562	45.141.16.0/22	VALID
□	AS34562	91.217.235.0/24	VALID

Show 25 of 3 items

HOW TO CREATE ROAS



3 BGP Announcements

✓ 3 Valid | ! 0 Invalid | ? 0 Unknown

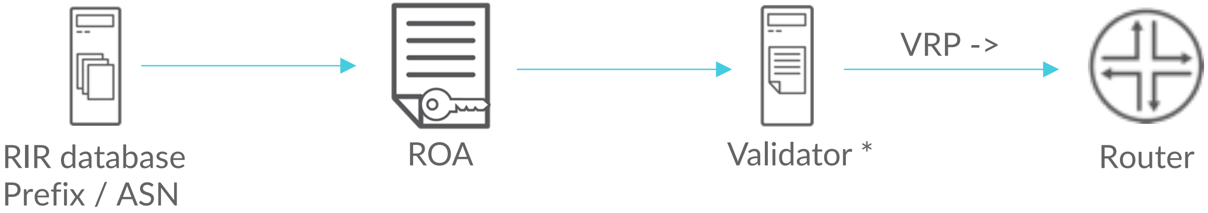
3 ROAs

✓ 3 OK | ⚠ 0 Causing problems

□	AS number	Prefix	Most specific length allowed	Affected announcements
□	AS34562	2a0e:c940::/29	29	1
□	AS34562	91.217.235.0/24	24	1
□	AS34562	45.141.16.0/22	22	1

Show 25 of 3 items

ORIGIN VALIDATION USING RPKI



* Cryptographic validation of ROAs happen at the RPKI Validator, to avoid burdening internet routers with the process. Only VRPs (Validated ROA Payload) are sent to the router.

THE ORIGIN VALIDATION PROCEDURE: ROAS & VRPS

A RPKI ROA is converted into a “VRP” (Validated ROA Payload), a VRP is a plain-text, crypto verified, representation of a ROA:

```
{
        "roas" : [ {
                "asn" : "AS34562",
                "prefix" : "195.114.12.0/24",
                "maxLength" : 24,
                "ta" : ”RIPE”
        },
}
```

HOW TO VALIDATE A BGP UPDATE WITH A VRP?

First, find which VRPs cover the BGP update. If no VRPs cover the prefix in the BGP UPDATE, the route is marked “Not-Found”.

If one or more VRPs cover the BGP announcement, iterate through each VRP and try to match them as following:

1. Does the Origin ASN as seen in the BGP UPDATE match with any of the RPKI ROAs?
2. Is the Prefix Length as seen in the BGP UPDATE aligned with what the ROA from (1) states?

If any one of the ROAs with the above two checks are positive matches, the route is “RPKI valid”.

If the BGP UPDATE fails to positive match: “RPKI Invalid”.

EXAMPLE #1

List of VRPs:

Prefix 123.0.0.0/16	Origin AS 444	MaxLength 16
Prefix 2001:67c:208c::/48	Origin AS 15562	MaxLength 48

BGP Update:

Prefix 172.48.0.0/24	Origin AS 555

Result…. ?

NOT FOUND!

EXAMPLE #2

List of VRPs:

Prefix 123.0.0.0/16	Origin AS 444	MaxLength 16
Prefix 2001:67c:208c::/48	Origin AS 15562	MaxLength 48

BGP Update:

Prefix 123.0.0.0/16	Origin AS 444

Result…. ?

VALID!

EXAMPLE #3

List of VRPs:

Prefix 123.0.0.0/16	Origin AS 666	MaxLength 16
Prefix 2001:67c:208c::/48	Origin AS 15562	MaxLength 48

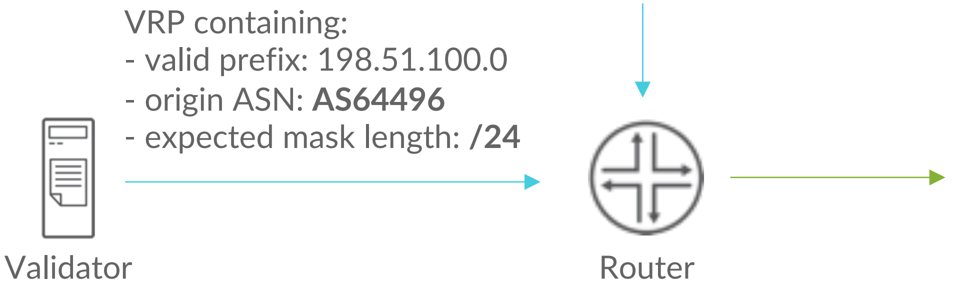
BGP Update:

Prefix 123.0.0.0/16	Origin AS 123

Result…. ?

INVALID!

PERFECT WORLD ROUTING



Route advertisement:

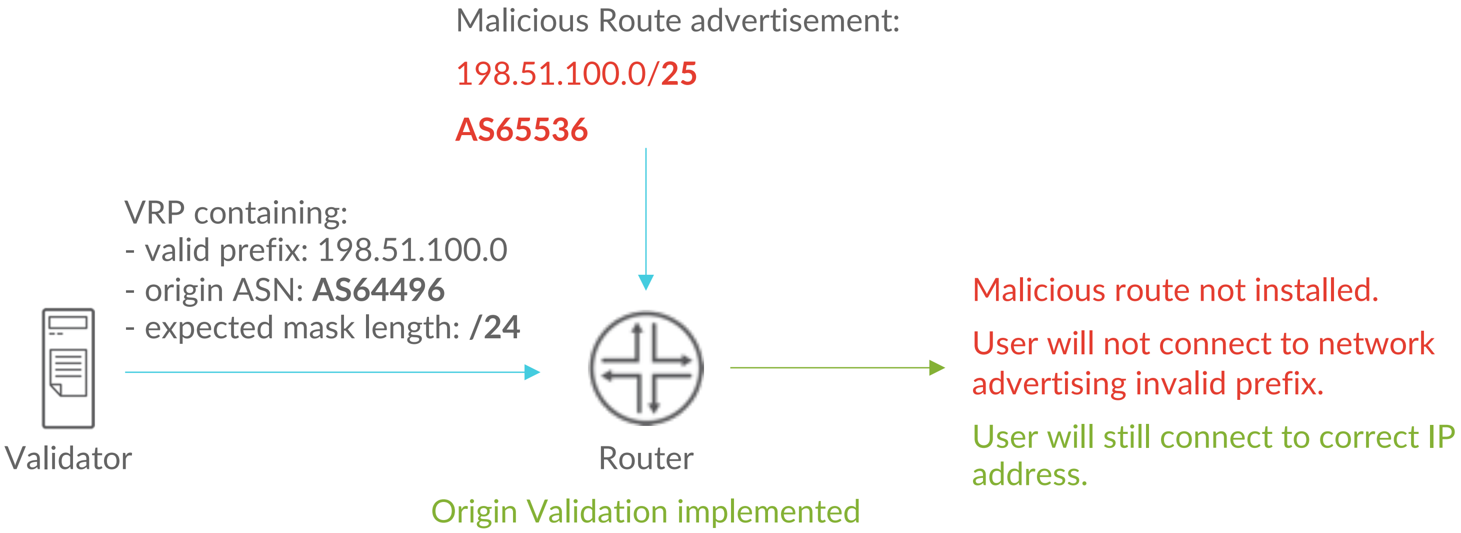
198.51.100.0/**24**

AS64496

VRP containing:
- valid prefix: 198.51.100.0
- origin ASN: **AS64496**
- expected mask length: **/24**

Happy user being able to connect to right IP address

ORIGIN VALIDATION IMPLEMENTED



Malicious Route advertisement:

198.51.100.0/**25**

AS65536

VRP containing:
- valid prefix: 198.51.100.0
- origin ASN: **AS64496**
- expected mask length: **/24**

Validator

Router

Origin Validation implemented

Malicious route not installed.

User will not connect to network advertising invalid prefix.

User will still connect to correct IP address.

SUMMARY:

“RPKI IS USED TO LET THE LEGITIMATE HOLDER OF A BLOCK OF IP ADDRESSES MAKE AN AUTHORITATIVE STATEMENT ABOUT WHICH AS IS AUTHORISED TO ORIGINATE THEIR PREFIX IN THE BGP.

IN TURN, OTHER NETWORK OPERATORS CAN DOWNLOAD AND VALIDATE THESE STATEMENTS AND MAKE ROUTING DECISIONS BASED ON THEM. THIS PROCESS IS REFERRED TO AS ROUTE ORIGIN VALIDATION (ROV) “.

Source: https://rpki.readthedocs.io/en/latest/about/introduction.html

RPKI Validators

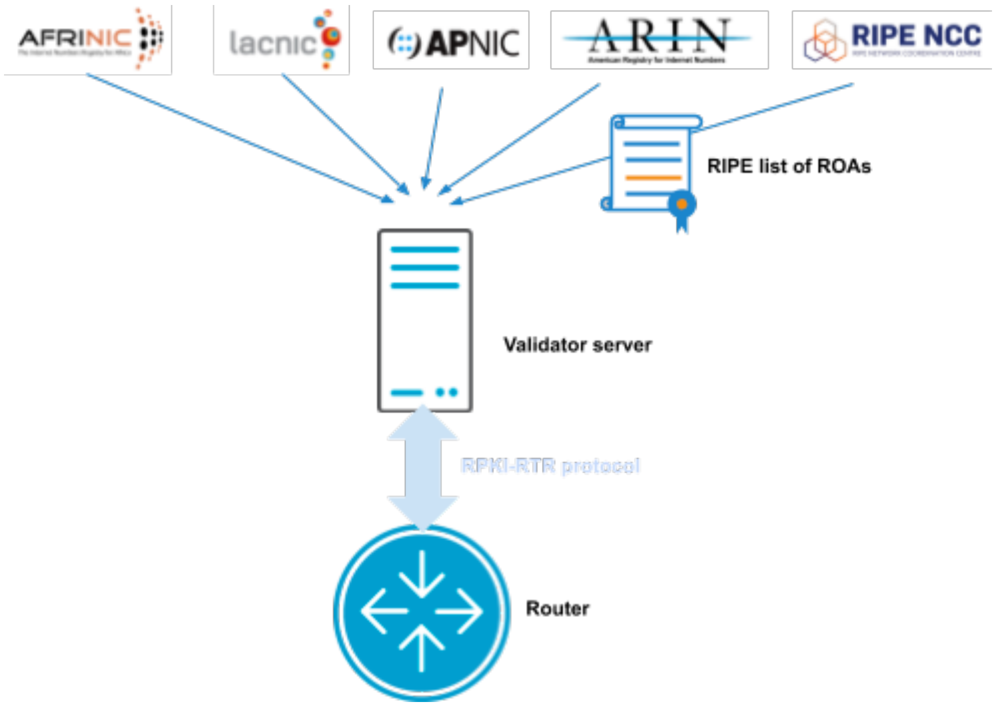
WHAT IS A VALIDATOR AND WHAT DOES IT DO?

Routers do not perform any cryptographic operations to perform Route Origin Validation.

ROAs are checked by external software, called Relying Party software or RPKI Validator, which feeds the processed data “VRP” (Validated ROA Payload) to the router over a light-weight protocol (RPKI-RTR).

This architecture causes minimal overhead for routers.

VALIDATOR EXPLAINED



WHICH VALIDATORS ARE AVAILABLE

Validators are open source or free available pieces of software.

Available validators (maintained):

-	Routinator	https://github.com/NLnetLabs/routinator
-	OctoRPKI	https://github.com/cloudflare/cfrpki
~~-~~	~~RIPE RPKI Validator~~	~~https://github.com/RIPE-NCC/rpki-validator-3~~

And a few others (not maintained, still in BETA, not widely used, etc.):

-	FORT	https://fortproject.net/
-	RPSTIR	http://www.rpstir.net/
-	rpki-client	https://github.com/kristapsdz/rpki-client

EXAMPLE VALIDATOR INSTALLATION: ROUTINATOR

Assuming you have a newly installed Debian or Ubuntu machine, you will need to install rsync, the C toolchain and Rust.
You can then install Routinator and start it up as an RTR server listening on 127.0.0.1 port 3323 and HTTP on port 9556:

```
apt install rsync build-essential
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
source ~/.cargo/env
cargo install routinator
routinator init
# Start routinator
routinator server --rtr 192.0.2.13:3323 --http 192.0.2.13:9556
```

Implementation

STRATEGY TO IMPLEMENT RPKI

1. Install at least 2 RPKI validators.

2. Configure sessions on all border routers with the validators.
   This allows the routers to populate a validation database that contains prefix, prefix lengt, max lengths and origin ASN combinations.
   Nothing is dropped yet and routing still continues to function as it was.

3. Implement BGP filters on all your external BGP sessions. On all inbound BGP sessions with peers, transit and BGP customers, you need to add a policy to **reject** any received advertisement containing RPKI invalid prefixes.

   Think about the order of filtering, RPKI validation should be one of the first.

   Possible order of enabling filtering: start with transit, then peers and last BGP downstream customers.

TIPS AND TRICKS

- Implement RPKI validation on *all* upstream BGP sessions (peering and transit) as invalid routes will still be available in your network if they are learned from one upstream and propagated via iBGP. You *need* to implement it everywhere, or be sure that all upstreams do it for you.

- The *only* correct way to deal with RPKI invalids is to reject them. There are (mostly older) examples mentioning assigning a lower local preference, but that does not help at all, since a more specific route (with a lower local preference) will always win from a less specific route (with a higher local preference).

- There is still quite a number of prefixes which are RPKI invalid present in the DFZ. Mostly, these are misconfigurations. This is not a problem to start rejecting, however in specific cases it can be problematic. Validators and routers offer options to whitelist or ignore specific prefixes.

- Make sure you reject RPKI invalid routes on your outbound sessions as well (not all vendors have implemented this feature). You don't want to be propagating any invalid routes.

- Train customers support staff and provide them with tooling to check RPKI validation status for IP’s. Support staff needs to understand that complaints about destinations being unreachable can be related to RPKI, they need to know how to identify those and how to deal with them.

JUNIPER RPKI IMPLEMENTATION

- RPKI Origin Validation exists in Junos since Junos OS 12.2R1

- Keep in mind when running a Junos OS version it contains bug fixes for the following PRs:
  https://prsearch.juniper.net/InfoCenter/index?page=prcontent&id=PR1483097
  https://prsearch.juniper.net/InfoCenter/index?page=prcontent&id=PR1461602
  https://prsearch.juniper.net/InfoCenter/index?page=prcontent&id=PR1309944

- No need to tweak timers for validator sessions. Just a couple of lines of config needed to get started:

```
routing-options {
    validation {
            group rpki-validator {
                    session 2001:db8::f00:baa {
                            port 8323;
                            local-address 2001:db8::1;
                    }
            }
    }
}
```

JUNIPER RPKI FILTERING POLICY 1/2

```
set policy-options policy-statement RPKI-CHECK term valid from protocol bgp
set policy-options policy-statement RPKI-CHECK term valid from validation-database valid
set policy-options policy-statement RPKI-CHECK term valid then validation-state valid
set policy-options policy-statement RPKI-CHECK term valid then community add origin-validationstate-valid

set policy-options policy-statement RPKI-CHECK term unknown from protocol bgp
set policy-options policy-statement RPKI-CHECK term unknown from validation-database unknown
set policy-options policy-statement RPKI-CHECK term unknown then validation-state unknown
set policy-options policy-statement RPKI-CHECK term unknown then community add origin-validationstate-unknown

set policy-options policy-statement RPKI-CHECK term invalid from protocol bgp
set policy-options policy-statement RPKI-CHECK term invalid from validation-database invalid
set policy-options policy-statement RPKI-CHECK term invalid then validation-state invalid
set policy-options policy-statement RPKI-CHECK term invalid then community add origin-validationstate-invalid

set policy-options community origin-validation-state-invalid members 0x4300:0.0.0.0:2
set policy-options community origin-validation-state-unknown members 0x4300:0.0.0.0:1
set policy-options community origin-validation-state-valid members 0x4300:0.0.0.0:0
```

This is totally non-intrusive

JUNIPER RPKI FILTERING POLICY 2/2

```
set policy-options policy-statement RPKI-CHECK term invalid then reject
```

So, are we safe now?

SO, ARE WE SAFE NOW?

Unfortunately not…we still need another parachute.

Or in other words, we can now perform Origin Validation for prefixes but spoofing the originating ASN is still possible.

More work is to be done…

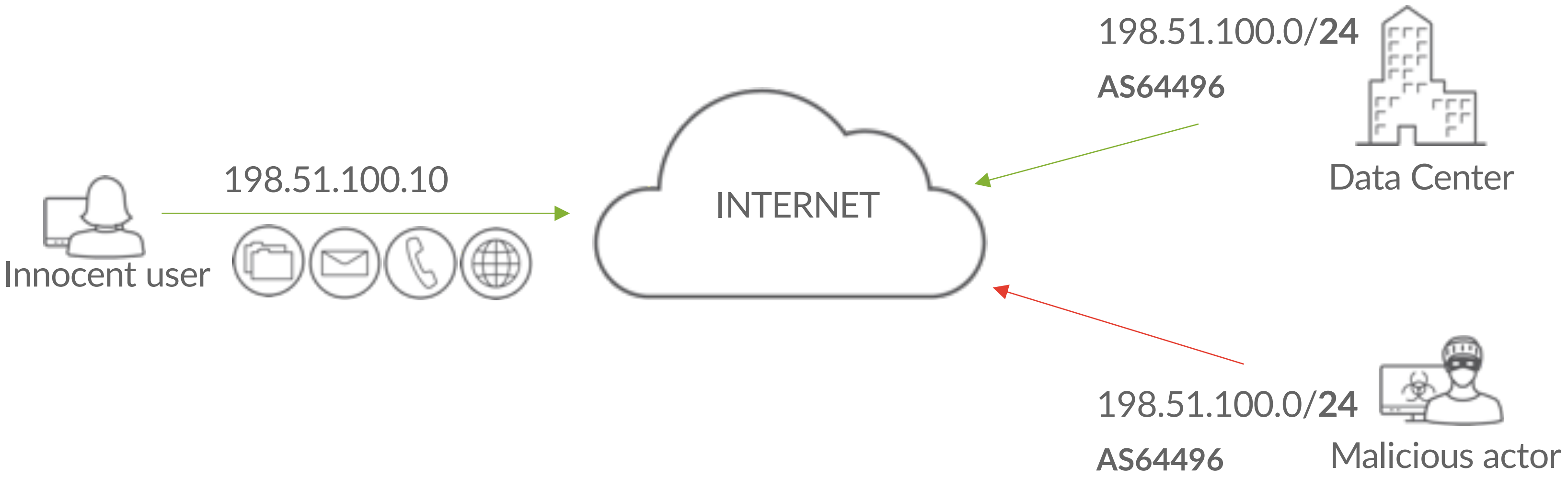
There is work in IETF addressing this problem:

https://datatracker.ietf.org/doc/draft-azimov-sidrops-aspa-profile/

And

https://datatracker.ietf.org/doc/draft-ietf-grow-rpki-as-cones/

WHAT HAPPENED TO OUR INNOCENT USER WHEN THE ORIGINATING ASN IS SPOOFED



If the malicious actor is the shortest AS_PATH the route will be accepted and installed, even with Origin Validation implemented as the advertisement is valid according to the ROA!

THERE IS WORK IN IETF ADDRESSING THIS PROBLEM

https://datatracker.ietf.org/doc/draft-azimov-sidrops-aspa-profile/

And

https://datatracker.ietf.org/doc/draft-ietf-grow-rpki-as-cones/

VERIFICATION OF AS_PATH USING THE RPKI INFRASTRUCTURE AND AUTONOMOUS SYSTEM PROVIDER AUTHORIZATION (DRAFT-IETF-SIDROPS-ASPA-VERIFICATION)

- uses a shared signed database of customer-to-provider relationships using a new RPKI object - Autonomous System Provider Authorization (ASPA).

- ASPAs are digitally signed objects that bind, for a selected AFI, a Set of Provider AS numbers to a Customer AS number (in terms of BGP announcements not business), and are signed by the holder of the Customer AS.

- An ASPA attests that a Customer AS holder (CAS) has authorized Set of Provider ASes (SPAS) to propagate the Customer's IPv4/IPv6 announcements onward, e.g. to the Provider's upstream providers or peers.

- The procedure for validation is comparable to how Origin Validation works; it checks that a pair of ASNs (AS1, AS2) is included in the set of signed ASPAs. The procedure takes (AS1, AS2, ROUTE_AFI) as input parameters and returns one of three results: "valid", "invalid" and "unknown".

- A relying party (RP) must have access to a local cache of the complete set of cryptographically valid ASPAs when performing customer-provider verification procedure.
    - 1. Retrieve all cryptographically valid ASPAs in a selected AFI with a customer value of AS1. The union of SPAS forms the set of "Candidate Providers."
    - 2. If the set of Candidate Providers is empty, then the procedure exits with an outcome of "unknown."
    - 3. If AS2 is included in the set of Candidate Providers, then the procedure exits with an outcome of "valid."
    - 4. Otherwise, the procedure exits with an outcome of "invalid."

RPKI AUTONOMOUS SYSTEMS CONES: A PROFILE TO DEFINE SETS OF AUTONOMOUS SYSTEMS NUMBERS TO FACILITATE BGP FILTERING (DRAFT-IETF-GROW-RPKI-AS-CONES)

- AS-Cones are a way to define groups of Autonomous System numbers in RPKI [RFC6480]. We call them AS-Cones. AS-Cones provide a mechanism to be used by operators for filtering BGP-4 [RFC4271] announcements.

- An AS-Cone is a digitally signed object with the goal to enable operators to define a set of customers that can be found as "right adjacencies", or transit customer networks, facilitating the construction of prefix filters for a given ASN, thus making routing more secure.

- AS-Cones are composed of two types of distinct objects: Policy definitions and the AS-Cones themselves.

- Objects are stored in ASN.1 format and are digitally signed according to the same rules and conventions applied for RPKI ROA Objects.

- A policy definition contains a list the upstream and peering relationships for a given Autonomous System that need an AS-Cone to be used for filtering. For each relationship, an AS-Cone is referenced to indicate which BGP networks will be announced to the other end of the relationship.
  The default behaviour for a neighbour, if the relationship is not explicitly described in the policy, is to only accept the networks originated by the ASN. This means that a stub ASN neither has to set up any AS-Cone, description, nor policy.
  Only one AS-Cone can be supplied for a given relationship. If more than one AS-Cone needs to be announced in the relationship, then it is mandatory to create a third AS-Cone that includes those two.

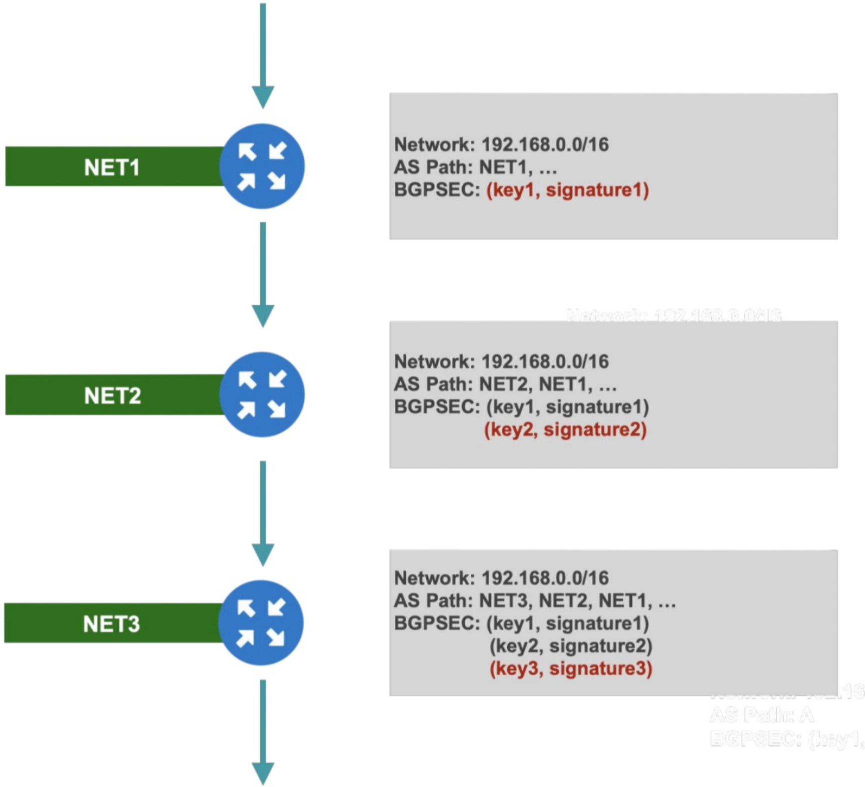
- AS-Cones are very similar to AS-Set RPSL Objects, so they could also be published in IRR Databases as AS-Set objects. Every ASN contained in an AS-Cone, and all the AS-Cones referenced should be considered as member: attributes. The naming convention for AS-Cones (ASX:AS- Cone) should be maintained, in order to keep consistency between the two databases.

WAIT....BUT WHAT ABOUT BGPSEC?

Why this didn't take of (yet)?

- BGPSEC is resource-intensive and therefor hard to deploy in the real world.
  If large number of networks would deploy and need to verify BGPSec signed updates, routers would need much more powerful control plane.

- It potentially exposes new security issues such as exposing not only which AS signed the update, but the actual router that signed the update.



Network: 192.168.0.0/16
AS Path: NET1, ...
BGPSEC: (key1, signature1)

Network: 192.168.0.0/16
AS Path: NET2, NET1, ...
BGPSEC: (key1, signature1)
(key2, signature2)

Network: 192.168.0.0/16
AS Path: NET3, NET2, NET1, ...
BGPSEC: (key1, signature1)
(key2, signature2)
(key3, signature3)

PRO-TIP: PEER DIRECTLY WITH AS MANY NETWORKS POSSIBLE

- When an attacker spoofs the origin ASN & same prefix length their AS_PATH is probably longer

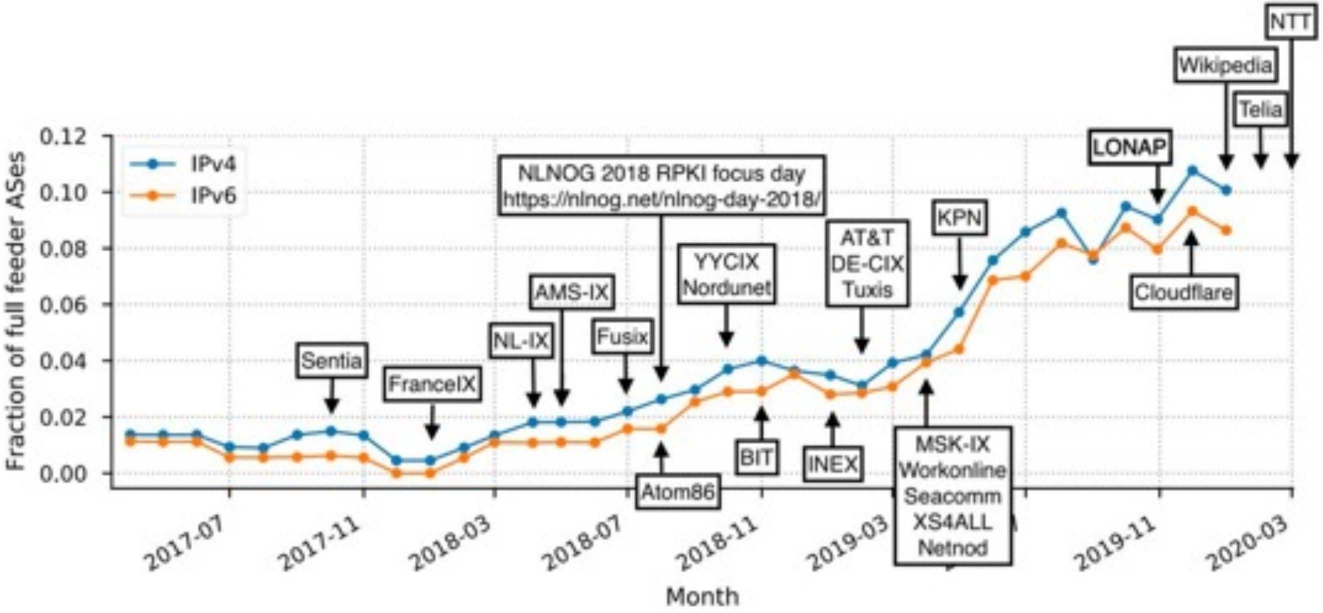
- When an attacker originates more-specifics you reject the route based on ROA

- Direct peering is sort of "AS_PATH Validation ", for just 1 AS Hop

Some statistics

INTERESTING GRAPH

RPKI enforcement is starting to gain traction

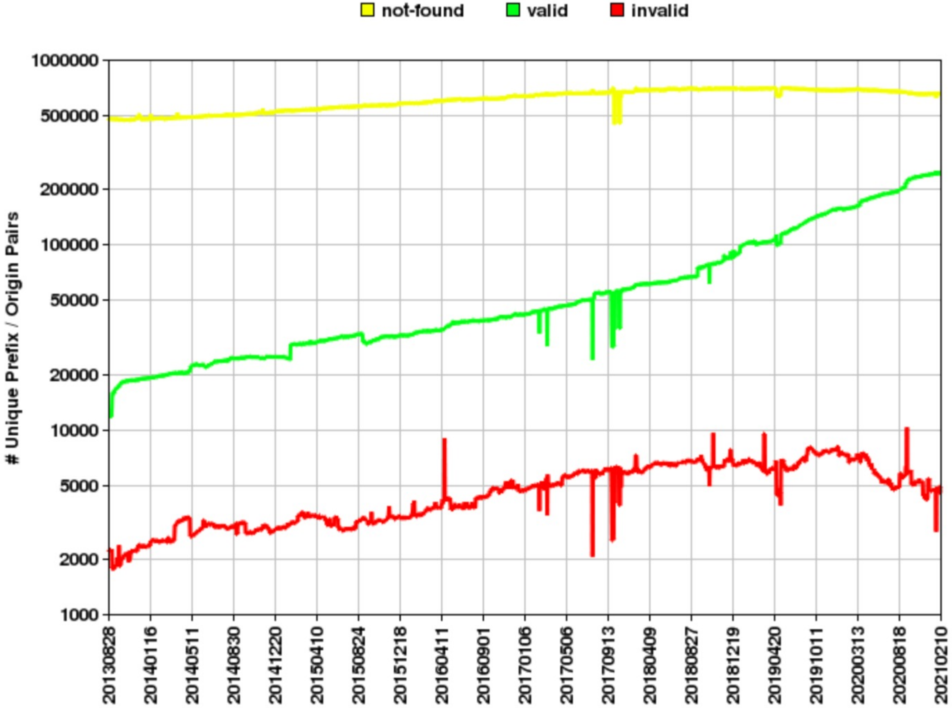


April 30, 2020
Source: Job Snijders via Twitter

LITTLE OVER 27% OF BGP ROUTES ARE COVERED BY VALID ROAS

Global: Validation History of Unique P/O pairs

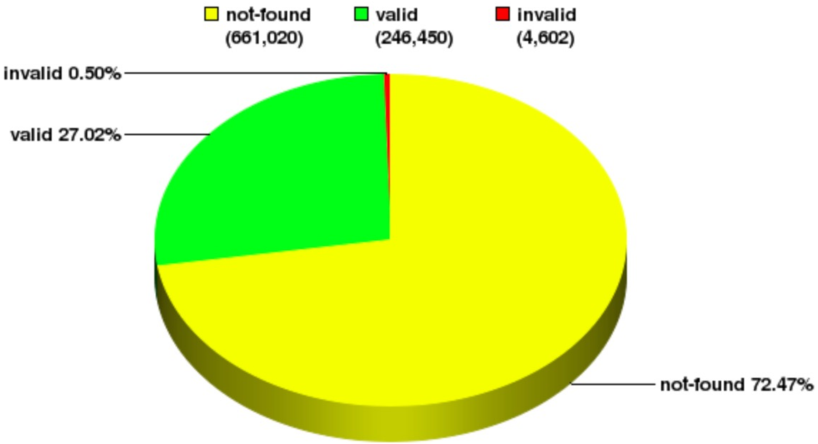
Only IPv4 Prefixes



NIST RPKI Monitor 2021-02-11

Global: Validation Snapshot of Unique P/O pairs

912,072 Unique IPv4 Prefix/Origin Pairs



NIST RPKI Monitor 2021-02-11

Additional resources

RESOURCES

General Routing Security

- NLNOG BGP Filter Guide: http://bgpfilterguide.nlnog.net/
- https://www.nlnetlabs.nl/projects/rpki/faq/
- https://rpki.readthedocs.io/

Deploying RPKI

- Deploying BGP Routing Security: https://www.juniper.net/documentation/en_US/day-one-books/DO_BGP_SecureRouting2.0.pdf
- Deployment Guide: https://www.juniper.net/documentation/en_US/junos/topics/concept/bgp-origin-as-validation.html
- BGP RPKI: Instructions for use: https://labs.ripe.net/Members/flavio_luciani_1/bgp-rpki-instructions-for-use

RIPE

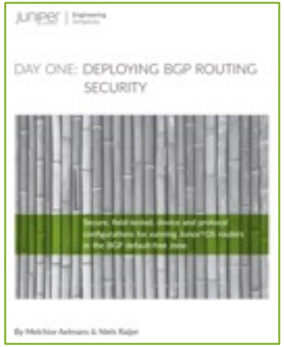
- https://www.ripe.net/rpki/

Measurement data

- https://rpki-monitor.antd.nist.gov

RPKI Validators

- NLnet Labs Routinator: https://www.nlnetlabs.nl/projects/rpki/routinator/
- Cloudflare OctoRPKI: https://github.com/cloudflare/cfrpki
- ~~RIPE Validator: https://github.com/RIPE-NCC/rpki-validator-3~~



DAY ONE: DEPLOYING BGP ROUTING SECURITY

Secure, field-tested, device and protocol configurations for running Junos® OS routers in the BGP default-free zone.

By Melchior Aelmans & Niels Raijer

Day One: Deploying BGP Routing Security

by Melchior Aelmans and Niels Raijer

Questions? Comments? Discussion.

maelmans@juniper.net