Data sources for BGP filtering Job Snijders NLNOG Camp 2018 job@ntt.net¹⁷

What is this about

- Relevance of routing security
- How IRRs work around the world
- Proposed future work
- Time for questions!

BGP Hijacking is lucrative







Amazon Route53 / MyEtherWallet.com hijack

	MYETHERWALLET.COM 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 Tetails 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 (1.2.8 Parameters none Not Valid Before Saturday, 7 April 2018 at 07:05:04 Not Valid After Sunday, 7 April 2019 at 07:05:04 Public Key Info	Eastern European Summer Time

Algorithm RSA Encryption (1.2.840.113549.1.1.1)

It could've been worse!

- The AS 10297 upstreams (NTT, Cogent, Level3) & Equinix route server blocked the hijack attack
- 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
- More info:

https://www.internetsociety.org/blog/2018/04/amazons-route-53-bgp-hijack/

3 reasons to filter

- Creating filters based on public data, forces malicious actors to leave a trial in IRR, WHOIS or other data sources: auditability
- Bugs happen: your router may suddenly ignore parts of your configuration, you'll then rely on your BGP peer's filters
- Everyone makes mistakes a typo is easily made

Filtering recap

- 1. Reject RFC 1918 (private) IP space
- 2. Reject majority of Bogon/Private ASNs
- 3. Reject IXP Nets
- 4. Reject based on "Peerlock" rules
- 5. Allow what is registered in IRR, WHOIS, and RPKI
- 6. Reject the rest

What is the IRR

- "Internet Routing Registry"
- What many operators uses as a source to generate per customer prefix filters
- Publicly available, to help debugging and provide transparency
- By making our source for filter generation publicly available, other parties can inspect what we take into consideration.

What sources are there?

- IRR Sources offered by Regional Internet Registries (RIPE, APNIC, ARIN, etc)
- IRR Sources operated by "third parties" (like RADB, NTT, ALTDB, etc)
- WHOIS sources (ARIN-WHOIS, Registro.BR)
- RPKI

• The sources are NOT equal, some operate by different rules than others

A route object: the atom

\$ whois -h rr.ntt.net 192.147.168.0/24

route: 192.147.168.0/24
descr: Job Snijders
origin: AS15562
notify: job@instituut.net
mnt-by: MAINT-JOB
changed: job@ntt.net 20161003
source: NTTCOM

(only the bold lines are relevant in the process) 10/

Generating a prefix filter

job@vurt ~\$ whois -h rr.ntt.net '!gAS15562'
A212

165.254.255.132/32 165.254.255.26/32 165.254.255.0/25 165.254.255.144/28 165.254.255.133/32 **192.147.168.0/24** 165.254.255.160/28 165.254.255.149/32 209.24.0.0/16 204.42.254.192/26 165.254.255.0/24 67.221.245.0/24 C job@vurt ~\$

Same example, with bgpq3

job@vurt ~\$ bgpq3 -h rr.ntt.net -l AS15562-in AS15562 no ip prefix-list AS15562-in ip prefix-list AS15562-in permit 67.221.245.0/24 ip prefix-list AS15562-in permit 165.254.255.0/24 ip prefix-list AS15562-in permit 165.254.255.0/26 ip prefix-list AS15562-in permit 165.254.255.26/32 ip prefix-list AS15562-in permit 165.254.255.64/26 ip prefix-list AS15562-in permit 165.254.255.132/32 ip prefix-list AS15562-in permit 165.254.255.133/32 ip prefix-list AS15562-in permit 165.254.255.144/28 ip prefix-list AS15562-in permit 165.254.255.149/32 ip prefix-list AS15562-in permit 165.254.255.160/28 ip prefix-list AS15562-in permit 192.147.168.0/24 ip prefix-list AS15562-in permit 204.2.30.0/23 ip prefix-list AS15562-in permit 204.42.254.192/26 ip prefix-list AS15562-in permit 209.24.0.0/16

Grouping ASNs: AS-SETs

job@vurt ~\$ whois -h rr.ntt.net AS15562:AS-**SNIJDERS** as-set: AS15562:AS-SNIJDERS members: AS15562 # Me members: AS57436 # Samer members: AS-KING # Thomas King members: AS-NETHER # Jared tech-c: **DUMY-RIPE** admin-c: DUMY-RIPE notify: job@instituut.net **ORG-SNIJ1-RIPE** org: mnt-by: SNIJDERS-MNT created: 2018-01-16T17:54:54Z last-modified: 2018-01-16T17:58:36Z RIPE source:

Systematic access to AS-SETS

\$ whois -h rr.ntt.net '!iAS15562:AS-SNIJDERS,1'
A130

AS15562 AS202539 AS205591 AS205593 AS206479 AS206499 AS206551 AS234 AS267 AS31451 AS41731 AS49697 AS51861 AS57436 AS60003 AS61438 C

Wrapping it up:

- An AS-SET is resolved into all its member ASNs
- For each ASN we do a reverse lookup to find all route-objects where the ASN is the "origin:"

How one IRR source is unlike the other..

- Not all IRRs are equal
- They differ in terms of ownership, purpose, policy, validation
- All of IRR is "garbage in, garbage out"
- Some RIRs offer good training materials on how to use the IRR
- Some IRRs have fancy web interfaces, some require interaction via email

- In NTTCOM, any customer can create any route object for any prefix (if it hasn't been covered by another route object in NTTCOM)
- In RADB anyone that pays \$500 per year can create any route object for any prefix (if it hasn't been covered by another route object in RADB)

- In ARIN, any ARIN member can create any route object for any prefix (if it hasn't been covered by another route object in ARIN)
- In ARIN WHOIS, only the owner of the IP block can specify an Origin AS

- In RIPE, only the owner of the IP block can create/designate route objects. Except when it isn't RIPE managed space... then anyone can create any route object for any prefix (if it hasn't been covered by another route object in RIPE)
- In the future RIPE will show the difference between route-objects for which it is authoritative and and which ones it isn't by showing: "source: RIPE" and "source: RIPE-NONAUTH"

- In the RIPE database, when you create route objects, both the IP block owner and the owner of the Origin ASN have to approve it. So double authentication is required.
- This is going to change in 2018, only the IP owner has to approve: aligns with APNIC, others

- In the APNIC and AfriNIC database you can only create route-objects for APNIC/AfriNIC managed space AND with approval from the IP block owner, and no approval from the ASN owner.
- This is the most sane approach, cleanest data

Summary: what the F***



What is this ARIN WHOIS thing?

•Remember from the fifth slide that we **only** care about the CIDR + Origin AS tuple?

<u> </u>	/iew & Manage	e Network
Dashboard	Jerre en la	
Tickets & Messages 🕧 👻		
Your Account	Information	
Settings Profile and security information	NETWORK INFO	
Point of Contact records	Net Range:	198.51.100.0 - 198.51.100.255
View and manage POCs	CIDR:	198.51.100.0/24
Organization Identifiers	Origin AS:	AS19384
View and manage Org IDs	Net Name:	TEST-NET-2
Associations Report Records connected to your	Net Handle:	NET-198-51-100-0
account	Parent:	<u>NET-198-51-100-0</u>
IP Addresses	Public Comments:	THIS NETWORK IS NON-PORTABLE
	Registered Date:	09-26-2008 15:36:10
Search View and manage your networks	Last Modified Date:	09-26-2008 15:36:10

Using ARIN WHOIS

- It is a trustworthy authoritative source of data
- We download a 3.5GB XML dump from ARIN once a day
- We convert the XML into "route:" objects
- We load those IRR objects into rr.ntt.net
- This is to offer customers easier choices
 - Setting up IRR can be cumbersome

Example output

\$ whois -h rr.ntt.net 199.43.0.44 199.43.0.0/24 route: descr: NET-199-43-0-0-1 origin: AS10745 remarks: This route object represents authoritative data retrieved from ARIN's WHOIS service. The original data can be found here: remarks: https://whois.arin.net/rest/net/NET-199-43-0-0 - 1This route object is the result of an remarks: automated WHOIS-to-IRR conversion process. mnt-by: MAINT-JOB changed: job@ntt.net **20150715 ARIN-WHOIS** source:

More WHOIS: Registro.br

job@vurt ~\$ whois -h registro.br 200.160.4.6 | grep -A 1 inetnum

- inetnum: 200.160.0/20
- aut-num: AS22548

job@vurt ~\$ grep AS22548 nicbr-asn-blk-latest.txt AS22548|Núcleo de Inf. e Coord. do Ponto BR - NIC.BR| 05.506.560/0001-36|200.160.0.0/20|2001:12ff::/32

- Computer parseable Registro.br data dump:
 - ftp://ftp.registro.br/pub/numeracao/origin/nicbr-asn-blk-latest.txt
 (Thank you Frederico Neves!)
- 56% of NICBR WHOIS entries exact match with BGP DFZ, quite accurate!

New use case for RPKI data?

\leftrightarrow \rightarrow C (i) Not secure lo	calcert.ripe.net:80	88/roas				☆ 🔽 I
RPKI Validator ноте	Trust Anchors	ROAs	Ignore Filters	Whitelist	BGP Preview	Export and

Validated ROAs

Validated ROAs from APNIC from AFRINIC RPKI Root, APNIC from ARIN RPKI Root, APNIC from IANA RP LACNIC RPKI Root, APNIC from RIPE RPKI Root, ARIN RPKI Root, AfriNIC RPKI Root, LACNIC RPKI Roo NCC Pilot (RRDP prefetch), RIPE NCC RPKI Root, RIPE NCC RPKI Root (RRDP prefetch), RIPE NCC prej prepdev (RRDP prefetch), altca, apnic-testbed.

!	Show 10 •	entries		Search: 2
	ASN 🔺	Prefix 🔶	Maximum Length	Trust Anchor
	15562	2001:67c:208c::/48	48	RIPE NCC RPKI Root

New use case for RPKI data?

- RPKI can be used for Origin Validation but it has more uses!
- A RPKI ROA kind of looks like a route object:
 - A ROA has a "prefix" and an "origin"
 - Use can use RPKI ROAs for provisioning

RPKI is trustworthy data (better than IRR), we know for sure that the owner of the IP space created the ROA

Simple example to convert RPKI ROA to prefix-list

job@vurt ~\$ ftp -VM -o - ∖

http://localcert.ripe.net:8088/export.json | \

jq '.roas[] | select(.asn | contains("AS3333"))? | .prefix' \

| sort | uniq | grep -v ::

"193.0.0/21"

"193.0.10.0/23"

"193.0.12.0/23"

"193.0.18.0/23"

"193.0.20.0/23"

"193.0.22.0/23"

"193.0.24.0/22"

"193.0.24.0/24"

Use RPKI to "drown out" IRR?

- Perhaps... given that we now know some IRR sources are less than stellar...
 - ... We could leverage RPKI data, to ignore rogue/stale/historic/wrong IRR route-objects?
- Maybe when a ROA exists... the ROA should suppress the output on IRRD query output?
- If there is an IRR route-object that conflicts with a RPKI ROA, maybe IRRd only return the ROA?

Using RPKI data: rtrsub

\leftarrow \rightarrow C $$ GitHub, Inc. [US] https://github.co	om/job/rtrsub	९ 🖈 📭 🛯 📀 📭 🛕 :
This repository Search	Pull requests	▲ Issues Marketplace Explore
⊑ job / rtrsub 🗵 ▾		⊙ Unwatch ▼
<> Code () Issues 1 () Pull	requests 0 Boards	🔺 Reports 🔲 Projects 🗿 🗉
RTR Substitution		
🕝 63 commits	1 branch 🛇 22	releases 🎎 1 contributo
Branch: master New pull request		Create new file Upload files
job bump version		
🖬 rtrsub	bump version	
template-examples	make better use of the na	tive bird rpki functions
template-examples .gitignore	-	tive bird rpki functions

Todo list for the community

- Deploy/Fix outstanding IRR issues: RIPE, ARIN
- Increase RPKI's reliability:
 - Certificate Transparency
 - Gain more operational experience!
 - Use RPKI data to "drown out" stale/rogue not-validated IRR data
- A new IRRd daemon from scratch: IRRdv4
 - https://github.com/irrdnet/irrd4
 - The future of IRRd is to be a routing data source policy engine (easy acces to WHOIS sources, and make RPKI a first class)
 - More news at RIPE77!
- Create an RPKI equivalent of AS-SETs
 - (come to the routing-wg for more news!)