



# YA~~t~~aBf

*test*

Yet Another ~~talk~~ about BGP filtering

Markus “FvD” Weber  
AS286, KPN Eurorings (Germany B.V.)

# Help



CAT	PATH	RDM	POINTS	COUNT	FILE
T	13	A1	5		
		A2	5		
		A3	5		
		A4	5		
		A5	5		
14	15	A6	5		
		A7	5		
		A8	5		
		A9	5		
		A10	5		
17	18	A11	5		
		A12	5		
		A13	5		
		A14	5		
		A15	5		
19	20	A16	5		
		A17	5		
		A18	5		
		A19	5		
		A20	5		
21	22	A21	5		
		A22	5		
		A23	5		
		A24	5		
		A25	5		
24	25	A26	5		
		A27	5		
		A28	5		
		A29	5		
		A30	5		
27	28	A31	5		
		A32	5		
		A33	5		
		A34	5		
		A35	5		

CAT	PATH	RDM	POINTS	COUNT	FILE
A	20	A1	5		
		A2	5		
		A3	5		
		A4	5		
		A5	5		
70	26	A6	5		XXXXXXXX
		A7	5		XXXXXXXX
		A8	5		
		A9	5		
		A10	5		
52	33	A11	5		
		A12	5		
		A13	5		
		A14	5		
		A15	5		
99	36	A16	5		
		A17	5		
		A18	5		
		A19	5		
		A20	5		
37	38	A21	5		
		A22	5		
		A23	5		
		A24	5		
		A25	5		

What?

No news is ...

... too bad there had been/are/will be -bad- news  
e.g. leaks

\* -> 200020 -> 3356 -> \*

\* -> 21217 -> 4134 -> \*

(\* ->) 33154 -> 396531 -> 701 -> \*

\* -> 6939 -> 9498 -> 6453 -> \*

# A simple “suspected T1-nT1-T1 leak live monitor”

- connected to the great RIS Live service
- used a dumb script(s) checking for paths matching

```
".* (Tier1-ASes)+ (nonTier1-ASes)+ (Tier1-ASes)+ .*"
```

- adjusted Tier1 list
- changed to 10min+ intervals reporting
- and waited ...

<https://puck.nether.net/bgp/leakinfo.cgi>

<https://bgpstream.com/>

# A simple “suspected leak live monitor”

The screenshot shows an email client window with a list of emails on the left and the content of a selected email on the right.

**Email List (Left Panel):**

FROM	SUBJECT	RECEIVED	SIZE	MENT...
KPN INS Gra...	LEAK-mmm (44)[14]101	20190824 - 06:58 UTC	295 KB	
KPN INS Gra...	LEAK-mmm (43)[19]306	20190824 - 06:48 UTC	647 KB	
KPN INS Gra...	LEAK-mmm (130)[27]223	20190824 - 06:38 UTC	490 KB	
KPN INS Gra...	LEAK-III (66)[14]117	20190824 - 06:28 UTC	361 KB	
KPN INS Gra...	LEAK-mmm (34)[9]80	20190824 - 06:18 UTC	281 KB	
KPN INS Gra...	LEAK-mmm (47)[10]324	20190824 - 06:08 UTC	1,004...	
KPN INS Gra...	LEAK-mmm (43)[23]458	20190824 - 05:58 UTC	953 KB	
KPN INS Gra...	LEAK-III (145)[20]44	20190824 - 05:48 UTC	221 KB	
KPN INS Gra...	LEAK-III (38)[8]30	20190824 - 05:38 UTC	215 KB	
KPN INS Gra...	LEAK-mmm (51)[27]348	20190824 - 05:28 UTC	677 KB	
KPN INS Gra...	LEAK-mmm (26)[4]108	20190824 - 05:18 UTC	219 KB	
KPN INS Gra...	LEAK-mmm (30)[6]356	20190824 - 05:07 UTC	793 KB	
KPN INS Gra...	LEAK-mmm (110)[14]263	20190824 - 04:57 UTC	725 KB	
KPN INS Gra...	LEAK-mmm (41)[15]224	20190824 - 04:47 UTC	594 KB	
KPN INS Gra...	LEAK-mmm (30)[13]115	20190824 - 04:37 UTC	306 KB	
KPN INS Gra...	LEAK-mmm (37)[18]144	20190824 - 04:27 UTC	353 KB	
KPN INS Gra...	LEAK-III (35)[10]88	20190824 - 04:17 UTC	246 KB	
KPN INS Gra...	LEAK-III (109)[11]46	20190824 - 04:07 UTC	209 KB	
KPN INS Gra...	LEAK-mmm (39)[12]148	20190824 - 03:57 UTC	350 KB	
KPN INS Gra...	LEAK-III (43)[22]444	20190824 - 03:47 UTC	1 MB	
KPN INS Gra...	LEAK-mmm (35)[19]114	20190824 - 03:37 UTC	339 KB	
KPN INS Gra...	LEAK-mmm (45)[16]87	20190824 - 03:27 UTC	270 KB	
KPN INS Gra...	LEAK-mmm (113)[17]397	20190824 - 03:17 UTC	1 MB	
KPN INS Gra...	LEAK-mmm (35)[20]169	20190824 - 03:07 UTC	398 KB	
KPN INS Gra...	LEAK-mmm (36)[16]525	20190824 - 02:57 UTC	1,012...	
KPN INS Gra...	LEAK-III (47)[9]34	20190824 - 02:47 UTC	203 KB	
KPN INS Gra...	LEAK-mmm (30)[8]234	20190824 - 02:37 UTC	478 KB	
KPN INS Gra...	LEAK-mmm (107)[18]372	20190824 - 02:27 UTC	754 KB	
KPN INS Gra...	LEAK-mmm (36)[6]329	20190824 - 02:17 UTC	962 KB	
KPN INS Gra...	LEAK-mmm (22)[6]295	20190824 - 02:07 UTC	618 KB	
KPN INS Gra...	LEAK-III (39)[9]20	20190824 - 01:57 UTC	178 KB	
KPN INS Gra...	LEAK-mmm (31)[8]32	20190824 - 01:47 UTC	186 KB	
KPN INS Gra...	LEAK-mmm (24)[9]111	20190824 - 01:37 UTC	383 KB	
KPN INS Gra...	LEAK-mmm (112)[26]256	20190824 - 01:27 UTC	219 KB	
KPN INS Gra...	LEAK-mmm (30)[6]2	20190824 - 01:17 UTC	644 KB	
KPN INS Gra...	LEAK-mmm (34)[8]107	20190824 - 01:07 UTC	282 KB	
KPN INS Gra...	LEAK-mmm (39)[26]160	20190824 - 00:57 UTC	358 KB	
KPN INS Gra...	LEAK-mmm (41)[11]30	20190824 - 00:47 UTC	196 KB	
KPN INS Gra...	LEAK-mmm (106)[10]153	20190824 - 00:37 UTC	452 KB	
KPN INS Gra...	LEAK-III (37)[13]32	20190824 - 00:27 UTC	237 KB	
KPN INS Gra...	LEAK-mmm (83)[22]253	20190824 - 00:17 UTC	637 KB	
KPN INS Gra...	LEAK-mmm (32)[8]240	20190824 - 00:07 UTC	556 KB	
KPN INS Gra...	LEAK-III (38)[6]16	20190823 - 23:57 UTC	107 KB	
KPN INS Gra...	LEAK-III (107)[11]26	20190823 - 23:47 UTC	237 KB	
KPN INS Gra...	LEAK-mmm (36)[11]35	20190823 - 23:37 UTC	202 KB	

**Email Content (Right Panel):**

Reply Reply All Forward IM  
Sat 24/08/2019 05:48

KI  
LEAK-III (43)[22]444 20190824 - 03:47 UTC

To: Weber, Markus

L-pfx: 43 / L-erp: 22 / L-path: 444  
=====

69.88.246.0/23 -- 27541 (1)  
6939==>36222==>6461 <nT1bT1> (1)  
27541->6939=>36222=>6461=>42708=>57381 (1)

69.88.252.0/24 -- 27541 (1)  
6939==>36222==>6461 <nT1bT1> (1)  
27541->6939=>36222=>6461=>42708=>57381 (1)

103.4.201.0/24 -- 54994 (1)  
3491==>7473==>6461 <nT1bT1> (1)  
54994->3491->7473=>6461=>37271=>328112=>328474=>328474=>328474=>328474 (1)

103.54.43.0/24 -- 134116 (8)  
2914==>58601==>17494==>7473==>6762 <nT1bT1> (11)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762 (2)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762->20800 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762->28917 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>50877 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>41497 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>52863 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>6762=>31133=>41722 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>31463=>56738 (2)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>55720 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>29075=>198385 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6762=>29075=>198385=>202194 (2)  
2914==>58601==>17494==>7473==>6453 <nT1bT1> (6)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453 (2)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453=>1403 (5)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453=>29686 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453=>6453=>6453=>47692 (3)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453=>28716=>59919 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>6453=>29075=>58308 (2)  
2914==>58601==>17494==>7473==>3491 <nT1bT1> (9)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>23520=>265721 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>57866=>8283 (2)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>63956 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>38001 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>64050 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>17639 (2)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>57866=>15703=>8283 (1)  
134116==>58889=>58717=>174=>2914=>58601=>17494->7473=>3491=>132825 (1)  
3491==>9498=>58601==>17494==>7473==>6762 <nT1bT1> (23)  
134116==>58889=>58717=>174=>3491=>9498=>58601=>17494->7473=>6762 (2)  
134116==>58889=>58717=>174=>3491=>9498=>58601=>17494->7473=>6762=>41497 (2)  
134116==>58889=>58717=>174=>3491=>9498=>58601=>17494->7473=>6762=>20800 (1)

# The test/game!

(on basics)



# The test/game - rules



- collect the points
  - *weight is a bit random ...*
  - *some guides might include or overlap with others*
  - only if you do it 1:1 (or stricter and with equal results)
  - only if you do it consistent everywhere (with special exceptions)
  - if you don't understand - 0 points (blame me for bad/no explanation)
- two categories
  - downstreams (you providing transit) and peers (no transit relation)
  - not everything is always for both applicable or makes equal sense
- if equal points, higher AS number wins, 286 can not win
- **not scoring highest doesn't mean your network is insecure nor a higher scored network is more secure**

# Some basics / definitions

And time to speed up!

# Sources

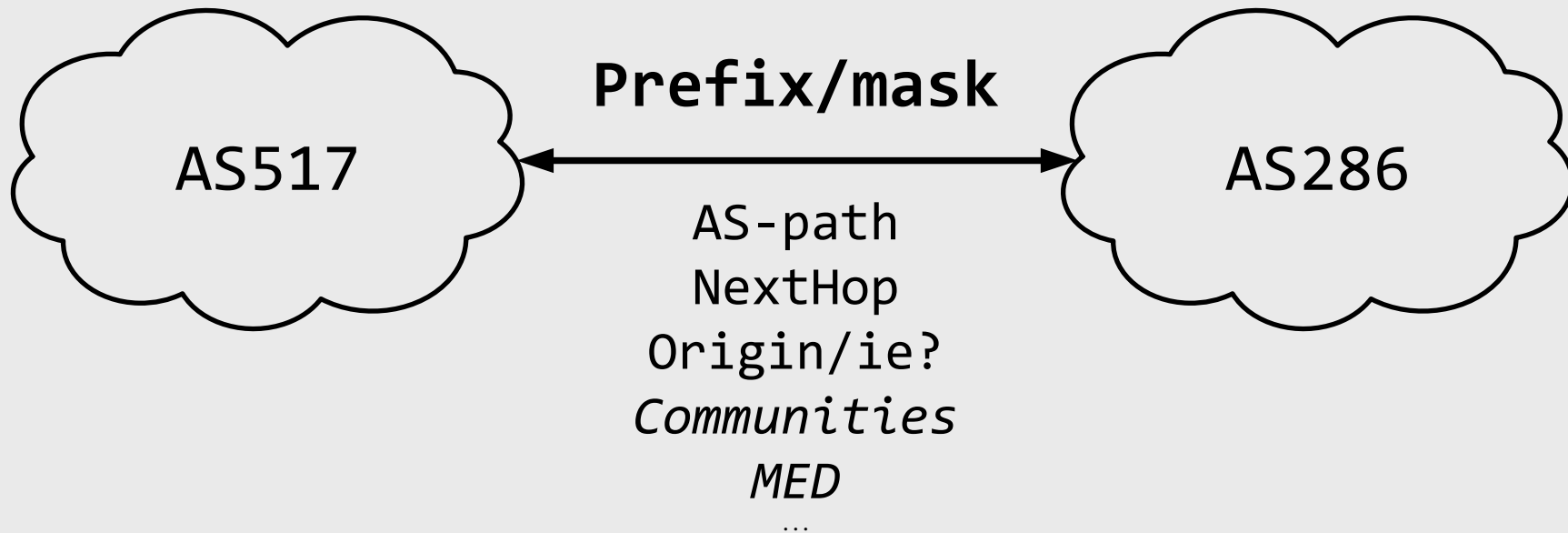
- BGP neighbor / peer information
  - v4/v6, peer's ASN and address
  - AS-set, route-set or just ASN
  - eventually static list of expected prefixes
- RFCs, IP/ASN allocation, other sources
  - mostly for bogon filters
  - Tier-1 ASes
- IRR, Whois or similar data
  - for AS-path and prefix filter
  - as-set, route/route6 (and aut-num) objects
- RPKI

# IRR - Internet Routing Registries

- Getting “useful” data from (meta) IRRs for an AS-set
  - **e-AS-set-ASes**
    - “explode” the AS-set = ask the (meta) IRR to return a list of all ASes for a given AS-set
    - recursively “explode” all AS-sets listed as members
  - **e-AS-set-prefixes**
    - for each <AS> in e-AS-set-ASes ask the (meta) IRR to return all <prefixes> for which a route object with “origin: <AS>” exists (reverse lookup)
    - either merge all prefixes into a single list or keep <prefix-originAS> tuples / prefix list per AS
- Not all IRRs are trustworthy ...

# Prefix lists

- Exact (as published/documented) or “covered” address space
  - Example:  
1.1.0.0/23, 1.1.0.0/24, 1.1.1.0/24, 1.1.2.0/23
  - **1:1 match (exact)**  
1.1.0.0/23, 1.1.0.0/24, 1.1.1.0/24, 1.1.2.0/23  
or 1.1.0.0/23-24, 1.1.2.0/23
  - **“covered” (aggregate, accept all more-specifics)**  
1.1.0.0/23, 1.1.2.0/23  
=> accepts as well 1.1.2.0/24 and 1.1.3.0/24  
or 1.1.0.0/22  
=> accepts as well 1.1.0.0/22, 1.1.2.0/24 and 1.1.3.0/24
- Example: AS-KPN results in (v4/v6)
  - 236.707 / 35.940 unique prefixes
  - 72.909 / 7.714 aggregated prefixes
  - 26.337 / 6.728 aggregated (aggressive) prefixes
  - 61.432 / 14.905 route-filter/p-l-r prefixes for exact match



Get ready!

Round I

# Miscellaneous things related to filtering

- Maintain your AS-set
  - consider using prefixed AS<ASnumber>:AS-set or your brand name
  - keep members up-to-date
  - ensure others easily can find it
- Maintain your route-objects
  - delete no-longer in use
  - add route-objects before announcing new prefixes
- Maintain your aut-num object
  - keep (mp-)import/export up-to-date
- Maintain your ROAs

m1

1

m2

1

m3

1

m4

2



# Miscellaneous things related to filtering

- Document and share your filtering policies
  - at least internal, nice if public, last resort configuration
- **Check scalability and monitor the “growth” of filters**
  - test extensively upfront scaling of filters (large/many small)
  - monitor growth of filters / config sizes to prevent surprises
  - filtering impacts (e)BGP convergence speed / CPU load
  - *you unlikely can do everything for everyone*
- Use a framework or central admin/maintained configs
  - does your setup allow easy update and introduction of new filters?
    - e.g. centrally generated configuration of sessions
    - e.g. common shared policies
  - does your setup allow easy reading why something was rejected?

m5

1

m6

3

m7

1

# Miscellaneous things related to filtering

- Review impact of filtering on/by other “services”
  - e.g.
    - BlackHole /32 | /128 announcements vs exact route-object match;
    - prefix-originAS filtering vs. direct downstream is announcing BH route on behalf of one of its own downstreams
    - private-AS sessions
    - exact filtering vs. “customer attempting to mitigate prefix hijack”
- Know your BGP implementation
  - implied filters and default action
  - what they do and when (+order) they are performed
  - at least you are aware that there are some
- Make config updates of filters/lists/configs/... atomic
  - full or no update
  - partial configuration during updates are NOT used or cause harm

m8

1

m9

1

m10

1

# Miscellaneous things related to filtering

- Be transparent on your rejects

- what you reject and why to avoid surprises for e.g. your customers
- <https://routing.he.net>
- <https://as286.net/data/hidden-pfx-downstream.txt>

m11  
1

- Be transparent on your rejects (+)

- “bother” your customers with opt-out emails
- peers? IXes should ...

m12  
1

# Miscellaneous things related to filtering

- Review once in a while static lists
  - regularly review and update “static” (RFC, other info based) filters like bogons, Tier-1 lists, ...
- Update your IRR/\* generated filters
  - regularly / (semi-)automated
- Update your IRR/\* generated filters (+)
  - on-request (via NOC) immediately
- Update your IRR/\* generated filters (++)
  - self-service for customers (peers)

m13  
1

m14  
1

m15  
1

m16  
2

Get ready!

Round II

# BGP - miscellaneous

- **Accept only agreed address family**
  - disable other
- **Check Next-Hop received**
  - is on-link and is not local/own interface
  - and is peer's address - unless route-server or "IX/LAN specials"
- **Think about / Normalize MEDs**
  - you have a clear policy how to handle MEDs and know the risks
- **Set Prefix-limits everywhere**
  - so simple, so important !!!
  - received vs. accepted, auto-recovery
  - (semi-)automated adjustment of limits

b1

1

b2

1

b3

1

b4

7

# BGP - miscellaneous

- Remove communities - internal
  - you remove on ingress all communities with internal (for you) meaning
- Remove communities - limit access to actions
  - you limit access to communities with an associated action in your network to those who are allowed to use it - downstream / peer
- Remove communities - meaning to others
  - you have thought of removing (or keeping) communities received from your customers/peers or generated internally having eventually a meaning for your peers/upstreams (BH/prepends/...)
- Limit number of communities
  - you considered (or opted against) dropping announcements with too many communities (or remove communities)

b5

3

b6

2

b7

2

b8

1

Get ready!

Round III



# Filtering on prefix

- Reject “standard” bogon prefixes
- Reject “full” bogon prefixes
  - you reject the full bogon list (CYMRU), includes standard (get both)
- Reject the default route
- Reject too specifics (more-specific than /24|/48)
  - eventually accept from downstreams (/24|/48 single-AS-multihomed)
- Reject too un-specifics (less-specific than /8|/16?)
  - 2 \* /1 is again a default route
  - unavailable NH might become suddenly available

p1

4

p2

3

p3

2

p4

2

p5

2

# Filtering on prefix

- Reject own address space (incl. more-specifics)
- Reject IX space you are connected to (incl. more-specifics)
- Accept only prefixes within e-AS-set-prefixes - or longer
  - “covered” address space, ignoring origin ASor
- Accept only prefixes within e-AS-set-prefixes - exact
  - only as “documented” (matching route-object), ignoring origin AS

p6

4

p7

5

p8

6

p9

8

Get ready!

Round IV

# Filtering on AS-path

- Reject long as-paths
  - average length v4 ~4.3, v6 ~3.5
- Drop announcements with bogon ASes in AS-path
- Make sure left most (last added) AS in AS-path == peer AS
  - by filter / knobs; don't do on transparent route-server sessions
- Make sure right most AS in AS-path is in e-AS-set-ASes
  - origin AS
- Make sure every AS in AS-path is in e-AS-set-ASes
  - includes origin AS filter (get both points)

a1

1

a2

3

a3

3

a4

4

a5

7

# Filtering on AS-path

- Do not accept “Tier1” ASes in AS-paths from downstream
  - T1 in AS-paths from downstream is very, very likely a leak
- Do not accept "<Tier1>\* <nonTier1>+ <Tier1>+ .\*" from peers
  - (T1)-nonT1-T1 sequences in AS-paths from peers indicate very, very likely leaks - unless nonT1 is a non-transparent route-server (RISK!)
- Implement Job S./NTT’s “peer-lock” (some)
  - [http://instituut.net/~job/peerlock\\_manual.pdf](http://instituut.net/~job/peerlock_manual.pdf)
  - idea: if a network A interconnects with network B, why should A receive and accept announcements from B via network C or D?
  - sometimes there are good reasons to accept, e.g. because C is upstream of B and D is peer of C and A-B is down or TE prefixes sent by B via C or because B uses it’s AS split behind different ASes or ...
  - human information exchange, changes cause large config updates
  - *(you do it at least for >10%/>5 ASes of your larger BGP peers)*

a6

5

a7

5

a8

9

# Filtering on AS-path – But ...

- The fun with **AS\_SET (BGP)** in AS-paths
  - as a result of non-ATOMIC aggregation ... the thing with the { }
- What is the right-most AS in "**24785 16003 {8455 27970}**"?
  - J (as-number matching): it's "27970" (not 16003 nor (8455 or 27970))
  - C IOS (string regexp): it's "27970}"
- the following matches "t1\*-nt1+~t1"
  - 94.131.240.0/20 AS path: 174 44600 {6939 29491 35297 49720} I
- ASes in AS\_SET ({...}) might not be listed in IRR AS-set
- RFC6472, BCP 172, 2011:
  - "Recommendation for Not Using AS\_SET and AS\_CONFED\_SET in BGP"
- draft-kumari-**deprecate**-as-set-confed-set-14 (v0 2012)

# RPKI route validation and AS\_SET (BGP) ...

- RFC6483/2. [...]:  
*If the AS\_PATH contains a path segment of type AS\_SET, indicating that the route is an aggregate, then the origin AS cannot be determined.*
- Is it unknown or invalid (or valid)?
- RFC6907 (brief 7.1.8-7.1.12) – recommendation (ref RFC6472)
  - AS\_SET in path and covering ROAs => invalid (ignoring AS (mis-)match)
  - AS\_SET in path and no covering ROA => unknown
- What is your implementation doing?
  - JunOS 16.1R7/17.3R3/18.2R3 doesn't follow RFC6907 and uses unknown
  - happily bypass RPKI/RTR/RV without spoofing source AS and works even for more-specifics (and works with drafted ASPA; “unverifiable”)

... draft-kumari-deprecate-as-set-confed-set-14

Get ready!

Round V



# Filtering on prefix–origin-AS / RPKI

- Accept only prefixes from documented (IRR) origin-AS
  - prefix announcement must originate from AS mentioned in the route object (<prefix,origin-AS> tuples)
  - almost like RPKI/ROA - just on IRR route objects
  - again - decide for exact match or “same or more-specific”
  - configuration size & CPU killer (but “nicely” on bird)
- Reject RPKI “invalid” announcements
  - e.g. using RTR
  - invalid = >0 (validated correct) ROAs covering the network range and not a single of them match <pfx/mask, oAS>

ap1  
9

ap2  
8

# Filtering on prefix-origin-AS / RPKI

- ROA only works
  - what do you do if there's only a ROA, but no IRR route object?
  - hint: use rr.ntt.net rather than whois.radb.net ...
- Use RPKI/ROA data to improve your IRR filters
  - e.g. you convert ROAs to route-objects and hide all IRR route-objects underneath (if origin AS is within e-AS-set-Ases)

ap3  
4

ap4  
6

Get ready!

Round VI - final!

# Outbound (+upstream)

- Normalize MEDs
  - whatever you consider as normalized ...
- Set Next-Hop self
- Remove private ASes from AS-path
- Remove communities - meaning to others
  - internal / not-on-purpose ones which have meaning in other networks
- Execute action communities
  - e.g. suppress announcement (filter ;-)

o1

1

o2

1

o3

2

o4

3

o5

2

# Outbound (+upstream)

- Filter out bogon prefixes
- Do not announce default route
  - except if your customer wants it
- Do not announce too un-specifics ( $/<8|/<16$ )
- Remove too specifics ( $/>24|/>48$ )
  - you might have different polices on your downstreams
- Do not announce IX space you are connected to

o6

1

o7

1

o8

1

o9

1

o10

2

# Outbound (+upstream)

- Announce/Originate only your aggregated address space
  - and customer PIs
  - no more-specifics unless needed (TE, mitigation, ...)
- Re-announce only “correct” prefixes learned from eBGP
  - customer (+own) routes to customers + peers + upstreams
  - peer and transit routes to customers
- **Do not use prefix-lists as only filter for re-announcement of downstream routes**
  - if you have more than one non-downstream eBGP session
  - **YOU GET MINUS POINTS**

o11  
2

o12  
5

o13  
-9

# Outbound

- Prevent redistribution (BGP) of wrongly typed addresses
  - only announce to others your (+customer PI) space
  - fat fingers cause no harm!

o14  
8

Well done!



# More ...

- If you thought that's all ...
  - aut-num
    - if these would be reasonable (content / for use) maintained
    - filter on (pre-computed) possible AS-paths (graph)
  - more reasonable IRR filter builds
    - just as a starting point <https://http://routing.he.net/algorithm.html>
  - have some feeds to route collecting services (RIS, RouteViews, ...)
    - helpful on monitoring and post mortem analysis
  - NOC procedures for “you leak / you announce bad things” calls
    - to get directly the attention of the right people
  - procedures for mitigation
    - and documentation of special filters + removal
  - BGP monitoring / anomaly detection
    - and making use of the findings
  - using AS0 ROAs as filter
  - doing outbound RV (again) and filtering
  - ROA maxlen ...
  - ... ..

While you sum your points ...

# An experiment - RTR for IRR route-origin-validation

- IRR route-origin-validation often doesn't scale well
  - and how does RTR / RPKI RV scale?
- converted IRR AFRINIC, APNIC, ARIN, RIPE (AUTH), JPIRR, NTTCOM and RADB route-objects into 1:1 ROA data (~2.1mIn)
- fed data with Cloudflares gortr to a router (~14mIn DFZ routes via iBGP, but no validation / no complex policies, RE-1800x4-32G, 16.1R7-S?/18.2R3-S?)
- restarted RPD / reload box and waited ...

# An experiment - RTR for IRR route-origin-validation

- BGP converged after a while - RTR took 45min - 1h++ (+1GB)
  - every update triggers re-evaluation of affected prefixes in RIB
- Filtering options
  - option A (invalid drop, valid/unknown accept):
    - unprotected / only partially protected during this time
  - option B (invalid/unknown drop, valid accept; “all space covered”):
    - slow time for BGP convergence, maybe ok for eBGP peers, but customers?
    - risk when servers offline, RV cache expired; NEVER do this (for now)
- So what?
  - if all others do ROV, propagation of wrong announcements is limited
  - it is a - kind of scalable - way to be more strict; as add-on (IRR)
  - do you really want to be / can you really be that strict?
    - “IRR exact” invalids: ~255K, unknown: ~100K (\*), invalids without alt: ?!?
    - “IRR covered” invalids: ~75K, unknown: ~100K (\*), invalids without alt: ?!?
  - too bad, there’s just a single validation cache (on JunOS)

# And the winner is?

>160? Use a calculator ;-)

# Questions?

Markus Weber fvd-nid2019@uucp.de

Slides will show up on <https://as286.net>

