

Let's talk about routing security

—

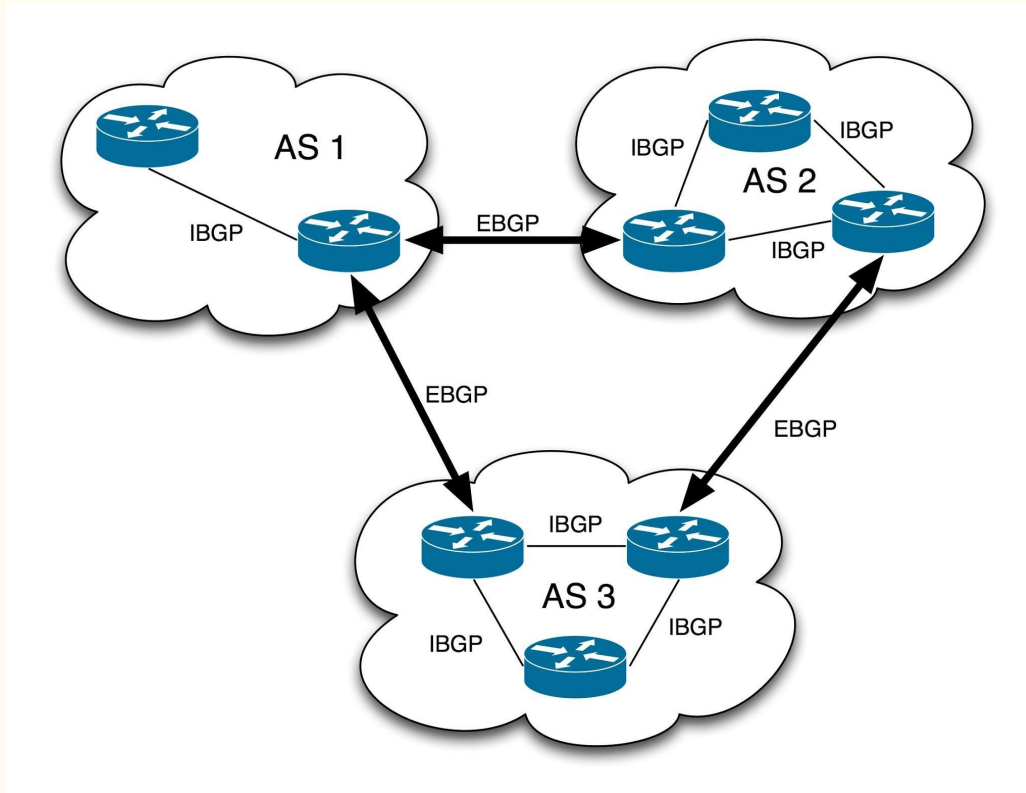
How secure is our routing infrastructure in 2019?

Fundamentals of global routing...

Internet - Network of ASNs...

- Internet is simply network of autonomous networks all connected together and speaking “BGP”.
- There are around 64k autonomous networks (known by their number called ASN) in IPv4 routing and 17k ASNs in IPv6 world.
- A set of around 15 networks stitch these ASNs together by forming a “default free / transit free zone” and essentially all ASNs in the world are direct/indirect customer of either of these ASNs.
- A large part of modern traffic flows from a limited set of ASNs (content networks) to eyeball networks via PNI’s and Internet Exchanges

Internet - Network of ASNs...



Internet - Network of ASNs...(+ DNS!)

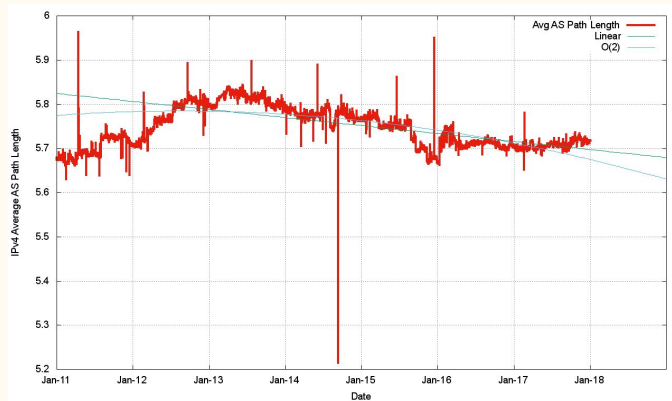
- BGP ensures interconnection of networks and DNS ensures domain to IP mapping.
- DNS relies on set of 13 logical root DNS servers and practically as many as 980 instances across the world via anycast.
- These 13 root DNS addresses are hardcoded in DNS resolver software (like BIND, powerdns etc) and hence security of these 13 IPs is important.
- DNS resolver contacts either of 13 based on reply time and other factors in the resolution algorithm.

So how “trust” in the
BGP works?

Trust in the BGP...

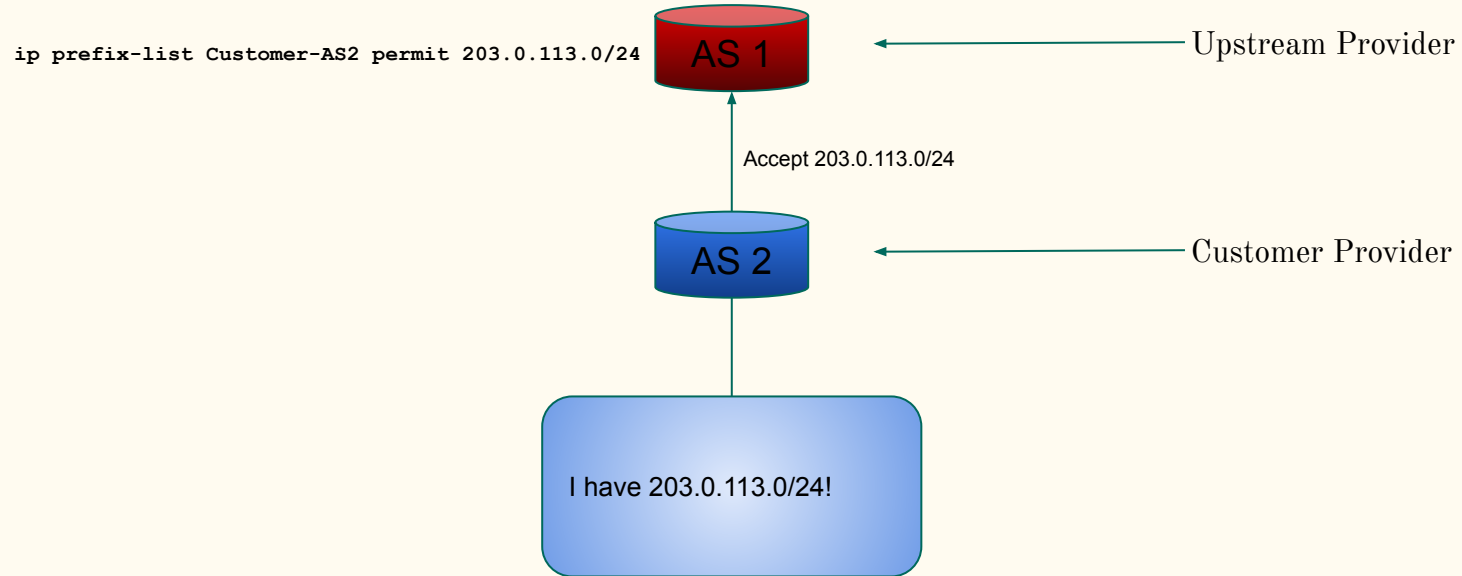
- BGP supports filtering and networks can define in filter what they can accept or reject and the default action (accept/reject).
- Filters can be based on IP prefix, ASN or AS Path or other factors like BGP community.
- It's quite easy to generate filters for networks near the edge but very hard as one goes near the core.
- Edge filtering - Filter the networks which connect to you based on static filter based on prefix and some other basic rules and full stop.
- Filtering beyond the edge - Allow prefixes of your downstream customer + their downstream + further their downstream and so on...

Trust in the BGP...



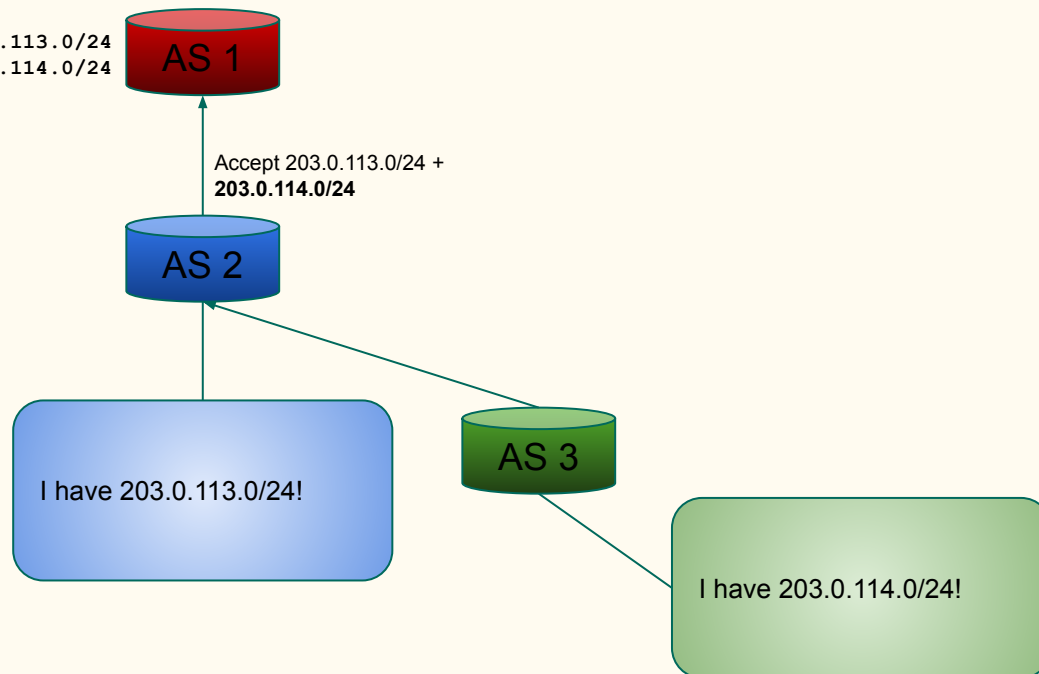
As per research data by Mr Geoff Huston (Scientist at APNIC) average AS path length in IPv4 world is around 5.7 and hence for a case like $AS\ 1 < AS2 < AS3 < AS4 < AS5$ it's very hard for AS1 to what to allow for AS4 (learnt via AS2).

Filtering chain...



Filtering chain...

```
ip prefix-list Customer-AS2 permit 203.0.113.0/24  
ip prefix-list Customer-AS2 permit 203.0.114.0/24
```



How does filtering works at the “Internet scale” ?

IRR - Internet Routing Registries

- IRRs are the public “registers” where one can log what they want to do and then just do it.
- IRRs use RPSL (Routing Policy Specific Language) to define “route object” where one defines prefix, origin AS, description etc and upstream can generate filters based on that.
- IRRs use “AS SETs” which define ASNs in a set (for instance a set of customer ASNs) and that is used to define customers ASNs.
- AS SETs can further have AS Sets of customer and that helps to generate downstream’s downstream’s downstream filter.

IRR - Route Object Example

```
whois -h whois.radb.net 216.218.128.0/17
```

```
route:      216.218.128.0/17
descr:      Hurricane Electric
             55 South Market St
             San Jose, CA
origin:      AS6939
notify:      noc@he.net
changed:     noc@he.net 20170407
mnt-by:      HE-NOC
source:      RADB
```

IRR - Route Object Example

```
whois -h whois.radb.net 216.218.128.0/17
```

```
route:      216.218.128.0/17    <- Prefix
descr:     Hurricane Electric
           55 South Market St
           San Jose, CA
origin:    AS6939      <- Origin AS
notify:    noc@he.net
changed:   noc@he.net 20170407
mnt-by:    HE-NOC
source:    RADB
```

IRR - AS SET Example

```
whois -h whois.radb.net AS-Google
```

```
as-set:      AS-GOOGLE
descr:      Google
members:    AS11344
members:    AS13949
members:    AS15169
members:    AS15276
members:    AS19425
members:    AS22577
members:    AS26910
members:    AS36040
members:    AS36384
members:    AS36492
members:    AS36561
members:    AS394725
members:    AS40873
members:    AS41264
members:    AS43515
members:    AS55023
members:    AS6432
members:    AS19527
members:    AS26684
members:    AS395973
members:    AS36039
members:    AS24424
members:    AS-GOOGLE-IT
members:    AS-MEEBO
members:    AS-METAWEB-2
mnt-by:     MAINT-AS15169
changed:    raybennett@google.com 20180614 #14:42:00Z
source:     RADB
```

IRR - AS SET Example

```
whois -h whois.radb.net AS-Google
```

```
as-set:      AS-GOOGLE    <- Name of AS Set
descr:      Google
members:    AS11344 <- Member ASN in the AS SET
members:    AS13949
members:    AS15169
members:    AS15276
members:    AS19425
members:    AS22577
members:    AS26910
members:    AS36040
members:    AS36384
members:    AS36492
members:    AS36561
members:    AS394725
members:    AS40873
members:    AS41264
members:    AS43515
members:    AS55023
members:    AS6432
members:    AS19527
members:    AS26684
members:    AS395973
members:    AS36039
members:    AS24424
members:    AS-GOOGLE-IT <- Member AS SET in the AS SET
members:    AS-MEEBO
members:    AS-METAWEB-2
mnt-by:     MAINT-AS15169
changed:    raybennett@google.com 20180614 #14:42:00Z
source:     RADB
```


More on Internet Routing Registries

- There are as many as 25 IRRs and were created for different reasons historically.
- RIR (Regional Internet Registry) based RIRs like APNIC, ARIN are common across their member users.
- Non-for profit RADB used mostly by larger organisations, free option ALTDB (for general Internet).
- One can define which IRR one is using at the peeringdb e.g RADB::AS-HURRICANE for Hurricane Electric or APNIC::AS9498:AS-BHARTI-IN for Airtel.
- RADB mirrors all major IRRs and thus a query to RADB includes it's own database as well as data of other mirrors IRRs.
- Most of filtering tools by default use RADB for generating filters.

Query to RADB...

```
whois -h whois.radb.net 163.47.158.0/24
```

```
route:          163.47.158.0/24
origin:         AS10075
descr:         Fiber @ Home Limited
                House - 7/B, Road - 13, Gulshan - 1
mnt-by:        MAINT-FIBERATHOME-BD
last-modified: 2019-04-03T13:58:31Z
source:        APNIC
```

Query to RADB...

```
whois -h whois.radb.net 163.47.158.0/24
```

```
route:          163.47.158.0/24
origin:         AS10075
descr:          Fiber @ Home Limited
                House - 7/B, Road - 13, Gulshan - 1
mnt-by:         MAINT-FIBERATHOME-BD
last-modified: 2019-04-03T13:58:31Z
source:         APNIC <- Shows the source database
```

bgpq3 - Tool for generating filters

- Open source tool bgpq3 can be used for generating filters based on IRR.
- It supports syntax of Cisco, JunOS out of the box.
- It also supports generating filter list based on custom syntax of any given hardware.
- Supports JSON based output format.
- Includes supports for AS Path based filters as well as IPv6.
- Supports only generation of filters and one needs to have a mechanism to push these filters to the routers.

bgpq3 - in action

```
bgpq3 -l Anurag AS58901 -6
no ipv6 prefix-list Anurag
ipv6 prefix-list Anurag permit 2402:b580::/32
ipv6 prefix-list Anurag permit 2402:b580:1::/48
ipv6 prefix-list Anurag permit 2402:b580:2::/48
ipv6 prefix-list Anurag permit 2402:b580:3::/48
```

```
bgpq3 -J -l Anurag AS58901 -6
policy-options {
  replace:
    prefix-list Anurag {
      2402:b580::/32;
      2402:b580:1::/48;
      2402:b580:2::/48;
      2402:b580:3::/48;
    }
}
```

bgpq3 - in action

```
bgpq3 -l Anurag AS58901 -6
no ipv6 prefix-list Anurag
ipv6 prefix-list Anurag permit 2402:b580::/32
ipv6 prefix-list Anurag permit 2402:b580:1::/48
ipv6 prefix-list Anurag permit 2402:b580:2::/48    <- Cisco iOS style syntax
ipv6 prefix-list Anurag permit 2402:b580:3::/48
```

```
bgpq3 -J -l Anurag AS58901 -6
policy-options {
  replace:
    prefix-list Anurag {
      2402:b580::/32;
      2402:b580:1::/48;    <- JunOS syntax
      2402:b580:2::/48;
      2402:b580:3::/48;
    }
}
```

It's querying RADB and formatting

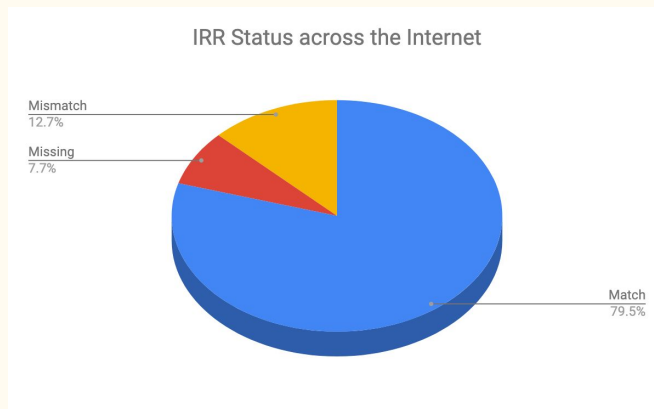
```
whois -h whois.radb.net '!6as58901'  
A66  
2402:b580:1::/48 2402:b580:3::/48 2402:b580:2::/48 2402:b580::/32  
C
```

More on this on RADB here: <https://www.radb.net/query/help>

So how well IRR based
filtering works?

So how well IRR based
filtering works? < - *Not so well!*

Filtering Statistics across the Internet



- There are as many as 758313 prefixes visible in global routing table (IPv4 + IPv6)
- Out of total routes: 603185 (79.54%) have valid route objects, 58587 (7.73%) have no valid route objects and 96514 (12.73%) have mismatching route object.
- Thus IRR based filtering can filter/blackhole 155101 routes or 20.45 of the total routes in the global table.

Challenges with IRR based filtering

- IRR is old and not very easy to integrate with the routers.
- There no one-solution-fits-all projects which can generate filters and push to all possible hardwares. (Things are much better off in route-servers run at the internet exchanges)
- IRRs by design are log books and whatever goes in there, usually stays in there. In other words they are full of old outdated route objects.
- The “software speaking to the routers & pushing config” isn’t very common across smaller networks.

Some of developments in routing security...

- Some larger networks including Hurricane Electric are now filtering over 98% peers (~25k BGP sessions) based on IRR.
- Google has announced to start filtering based on IRR by Sept 2019 (at NANOG 75).
- Internet Exchanges like BharatIX (in Mumbai), DECIX (in Frankfurt), INEX (in Dublin), Equinix IX Singapore etc are actively filtering prefixes based on IRR.
- RPKI is being pushed for to use cryptography to validate prefix origin and is supported in latest version of various vendors. For supported hardware, it's very to implement in a route-map / routing-policy.
- RPKI is being integrated in next version of IRR (IRR 4) to ensure route objects cannot be created where ROA mismatch happens.
- AT&T has announced that it now drops prefixes where RPKI validation fails.

RPKI in action...

```
router bgp 58901

  address-family ipv4 unicast
  neighbor 1.2.3.4 route-map Customer-IN in
  bgp bestpath prefix-validate allow-invalid
  !
  route-map Customer-IN permit 10
  match rpki invalid
  set local-preference 50
  !
  route-map Customer-IN permit 20
  match rpki not-found
  set local-preference 100
  !
  route-map Customer-IN permit 30
  match rpki valid
  set local-preference 200
  !
  route-map Customer-IN permit 40
```






RPKI in action...

```
router bgp 58901

  address-family ipv4 unicast
  neighbor 1.2.3.4 route-map Customer-IN in
  bgp bestpath prefix-validate allow-invalid
  !
  route-map Customer-IN permit 10
  match rpki invalid
  set local-preference 50  <- Low localpref on route if RPKI check is invalid (Remember: High localpref wins)
  !
  route-map Customer-IN permit 20
  match rpki not-found
  set local-preference 100 <- Mid level localpref on route is no ROA is present
  !
  route-map Customer-IN permit 30
  match rpki valid
  set local-preference 200 <- High localpref when RPKI check is valid and route is preferred
  !
  route-map Customer-IN permit 40
```

Easy way to check IRR as
well as RPKI for prefixes...







Check for IRR / RPKI ROA validation

- Hurricane Electric's BGP toolkit (free web tool!) supports both IRR as well as RPKI checks.
- Simply go to bgp.he.net and search with network name or AS number or prefix and you will see the status of prefixes.
-  Reflects when correct matching route object exists.
-  Reflects when parent route object exists (say for /17 or /21 etc when announcement is for /22).
-  Reflects when there is a mismatch of route objects.
-  Reflects when RPKI check is valid.
-  Reflects when RPKI check is invalid.

Check for IRR / RPKI ROA validation

<u>104.20.128.0/20</u>	 	Cloudflare, Inc.	
<u>104.20.144.0/20</u>	 	Cloudflare, Inc.	
<u>104.20.160.0/20</u>	 	Cloudflare, Inc.	
<u>104.20.176.0/20</u>	 	Cloudflare, Inc.	

https://bgp.he.net/AS13335#_prefixes

<u>103.229.82.0/23</u>	 	Fiber @ Home Limited	
<u>163.47.156.0/22</u>	 	Fiber @ Home Limited	
<u>163.47.156.0/23</u>	 	Fiber @ Home Limited	
<u>163.47.158.0/24</u>	 	Fiber @ Home Limited	
<u>163.47.159.0/24</u>	 	Fiber @ Home Limited	

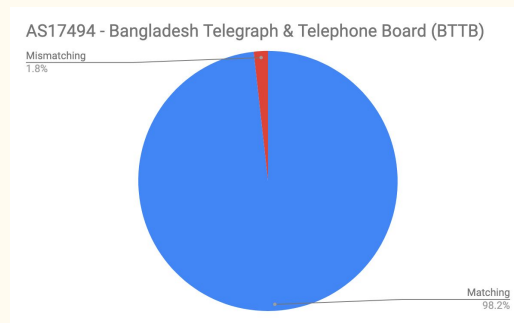
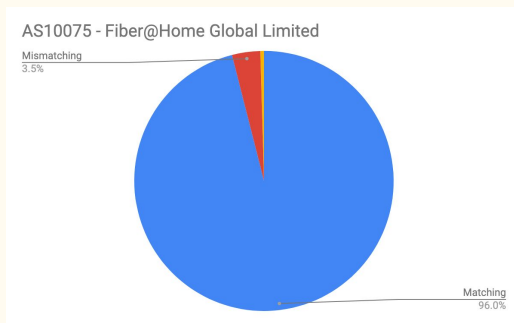
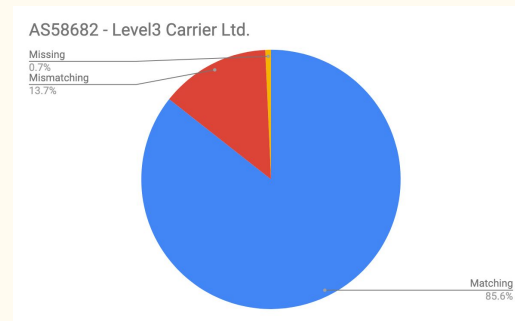
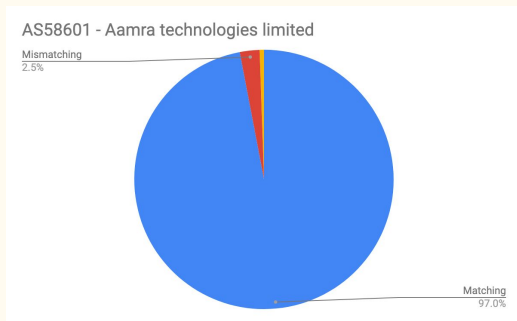
https://bgp.he.net/AS10075#_prefixes

Check for IRR / RPKI ROA validation

<u>103.9.112.0/24</u>	 	Aamra technologies limited	
<u>103.9.113.0/24</u>	 	Aamra technologies limited	
<u>103.9.114.0/24</u>	 	Aamra technologies limited	
<u>103.9.115.0/24</u>	 	Aamra technologies limited	
<u>103.206.47.0/24</u>			
<u>103.243.80.0/22</u>		Aamra Outsourcing Ltd.	
<u>103.243.80.0/24</u>		Aamra Outsourcing Ltd.	
<u>103.243.83.0/24</u>		Aamra Outsourcing Ltd.	
<u>116.206.60.0/24</u>		aamra Outsourcing Ltd.	

https://bgp.he.net/AS58601#_prefixes

Stats for top 5 BD ASNs



Contribute in the cleanup!

How can you contribute?

- If you maintain resources (IPv4, IPv6 or AS numbers) then ensure to register route objects for them in either of databases - database of your RIR (in Asia - do via My APNIC portal, in US - use ARIN portal).
- Create ROAs with your origin ASN and prefix length you intend to announce.
- Report all incorrect IRR entries you encounter to those registries to help them in removing old junk.
- If your provider has invalid route object or missing route object - suggest them to create one.
- DO NOT register route object on behalf of someone as a proxy entry as that has been a bad practice.
- If you have downstream ASNs behind you, register a AS SET.
- Register yourself on peeringdb.com portal and remember to mention your AS SET in the IRR section.

References

1. Tier 1 Networks Wikipedia Page - https://en.wikipedia.org/wiki/Tier_1_network#List_of_Tier_1_networks
2. BGP Version 4 RFC - <https://tools.ietf.org/html/rfc4271>
3. Root DNS servers list/locations - <https://root-servers.org/>
4. BGP in 2017 (APNIC Blog) - <https://blog.apnic.net/2018/01/10/bgp-in-2017/>
5. RSPL - <http://www.irr.net/docs/rpsl.html>
6. bgpq3 - <https://github.com/snar/bgpq3>
7. Hurricane Electric's route filtering algorithm - <http://routing.he.net/algorithm.html>
8. Google route filtering announcement NANOG75 - https://pc.nanog.org/static/published/meetings/NANOG75/1959/20190220_Morrow_Li_ghtning_Talk_Prefix_v1.pdf
9. IRR (present one) - <https://github.com/irrdnet/irrd>
10. IRR v4 (in deveopment) - <https://github.com/irrdnet/irrd4>
11. AT&T drops RPKI invalid for peers - <https://mailman.nanog.org/pipermail/nanog/2019-February/099501.html>

Questions/Comments?

Anurag Bhatia,
Hurricane Electric (AS6939)
anurag@he.net
Twitter: @anurag_bhatia
Web: <https://he.net>