Practical everyday BGP filtering with AS_PATH filters: Peer Locking

job@ntt.net

Disclaimer: ISPs and their ASNs used in this talk are examples for discussion purpose only. NTT does not admit or deny any relationships with these entities.

Part 1

Anybody know http://puck.nether.net/bgp/leakinfo.cgi ?

https://www.nanog.org/meetings/nanog41/presentations/mauch-lightning.pdf

What are we talking about?

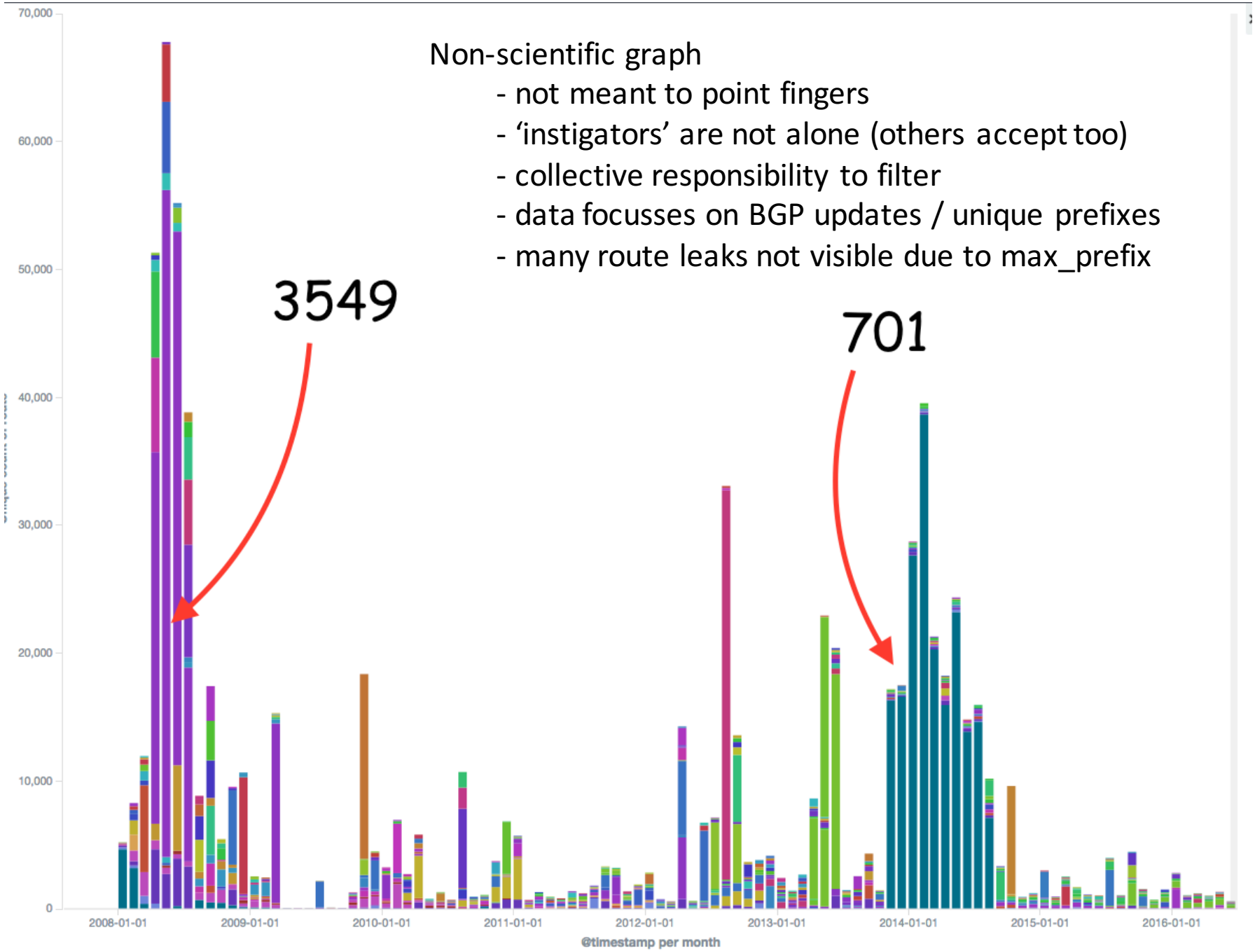
```
--[ Folder: =INBOX ]--~s Suspect ~b 3491--[*143080 messages ]----------
Date: Thu, 10 Mar 2016 11:05:15 +0100
From: Jared Mauch <jared@puck.nether.net>
To: job@ntt.net, mlong@us.ntt.net, tlyon@us.ntt.net, dpaxton@noc.us.ntt.net
Subject: Suspect BGP routes


        Greetings, this is an automated message regarding a BGP
routing leak that was detected by our automated system.

        The following leak(s) are related to your network.  If the
line says ANNOUNCED in it, it may list more than one prefix and
may reflect a transient leak.  These are still important to review
as they reflect poor bgp filtering or a possible routing architecture
flaw.  If an AS-PATH is legitimate that contains a sequence of ASNs
and should be exempted, please advise Jared Mauch.  (an example is
"1 2 3" - where ASN 1 buys transit from ASN 2 to get ASN 3 routes).

190.168.128.0/18 852 2914 6762 8048 23520 3491 3257 20312 27890 / Inspect 6762
190.168.128.0/18 2497 2914 6762 8048 23520 3491 3257 20312 27890 / Inspect 6762
~
```

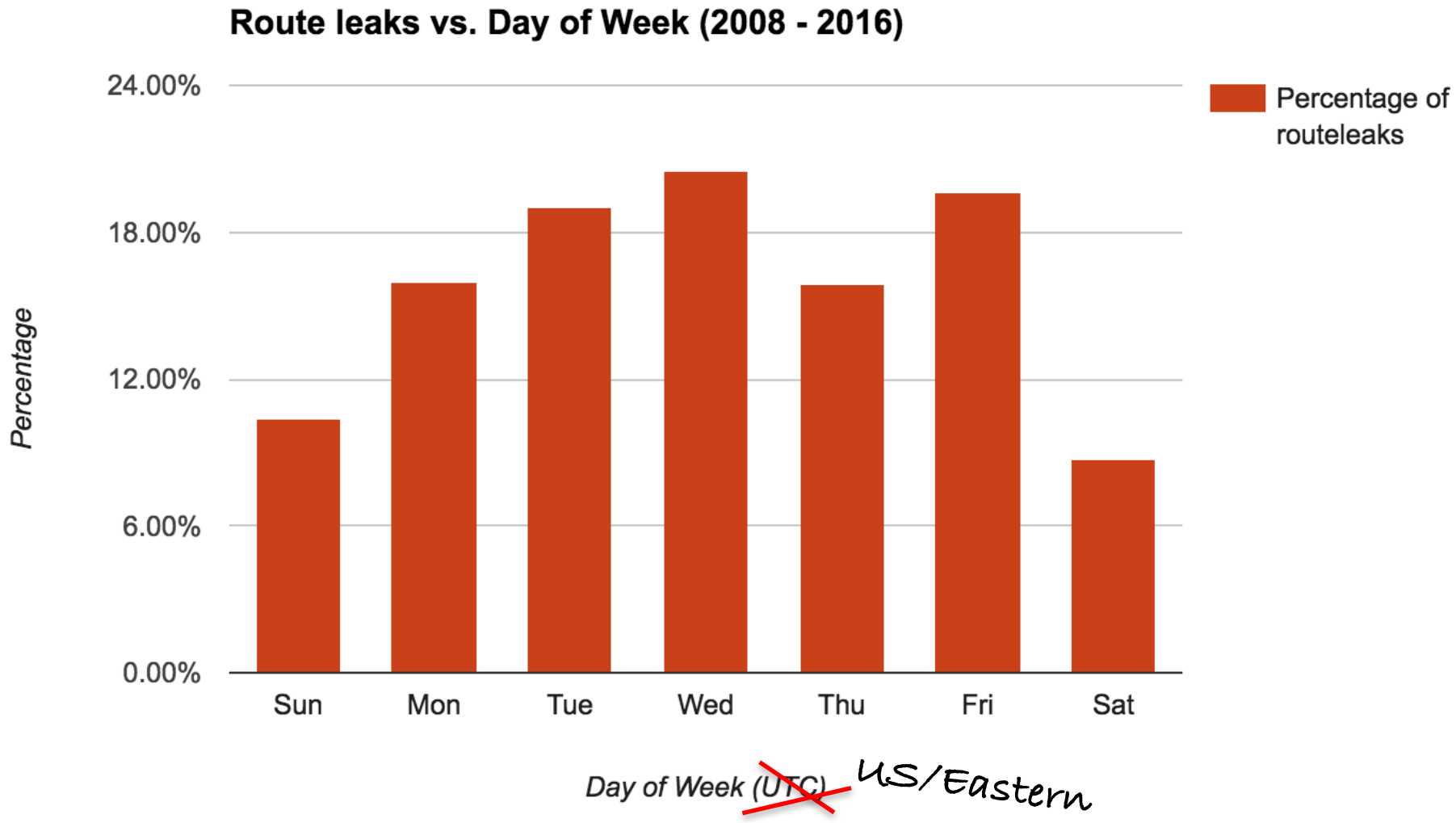
Wikipedia proclaimed “big boys”

7018, 174, 209, 3320, 3257, 286, 3356, 3549, 2914, 5511, 1239, 6453, 6762, 12956, 1299, 701, 2828, 6461

No more then two of these should show up in a given AS_PATH, following the “Transit-Free” paradigm.

https://en.wikipedia.org/wiki/Tier_1_network#List_of_tier_1_networks


Non-scientific graph
- not meant to point fingers
- ‘instigators’ are not alone (others accept too)
- collective responsibility to filter
- data focusses on BGP updates / unique prefixes
- many route leaks not visible due to max_prefix
3549
701
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
2008-01-01
2009-01-01
2010-01-01
2011-01-01
2012-01-01
2013-01-01
2014-01-01
2015-01-01
2016-01-01
@timestamp per month


Route leaks vs. Day of Week (2008 - 2016)
24.00%
18.00%
12.00%
6.00%
0.00%
Percentage
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Day of Week (UTC)
US/Eastern
Percentage of routeleaks

Peerlock-lite aka “bignetworks filter”

Assuming you’ll not sell transit to one of those big networks in the foreseeable future: **reject** any prefixes you receive from your customers which contain a $bignetwork ASN anywhere in the AS_PATH.

```
ip as-path access-list 99 permit \
     _(174|209|286|701|1239|1299  \
       |2828|2914|3257|3320|3356  \
       |3549|5511|6453|6461|6762  \
       |7018|12956)_

route-map ebgp-customer-in deny 1
  match as-path 99
```

Approaches to prevent route leaks #1

- Networks should not announce received prefixes over peering to other peers
  - **Fix**: Tag routes with BGP communities on ingress, execute on egress (recent NANOG thread)
  - **Note**: Always set egress filters to REJECT prefixes without any/the proper communities (failsafe)

Approaches to prevent route leaks #2

- One must apply a “whitelist” of prefixes a customer may announce on every customer session
  - **Fix:** use bgpq3 or some other prefix filter generator
- **Con:**
  - Customer’s AS-SET might contain the entire internet – thus when leaking a full table still allowing a lot to pass
    - https://github.com/job/irrtree
    - http://irrexplorer.nlnog.net/

Approaches to prevent route leaks #3

- Maximum prefix settings on peers + customers
  - **Fix:** if unsure: just do it
  - **Note:** automate the adjustment of max_prefix settings for your peers! Only email your peer when absolutely unsure what to configure.
- **Con:** does not help against small/partial route-leaks

Peer Lock

The Human Network: Peer locking in a nutshell

We know PCCW is not an upstream for AT&T, we know AT&T is not an upstream for PCCW, etc, etc etc.

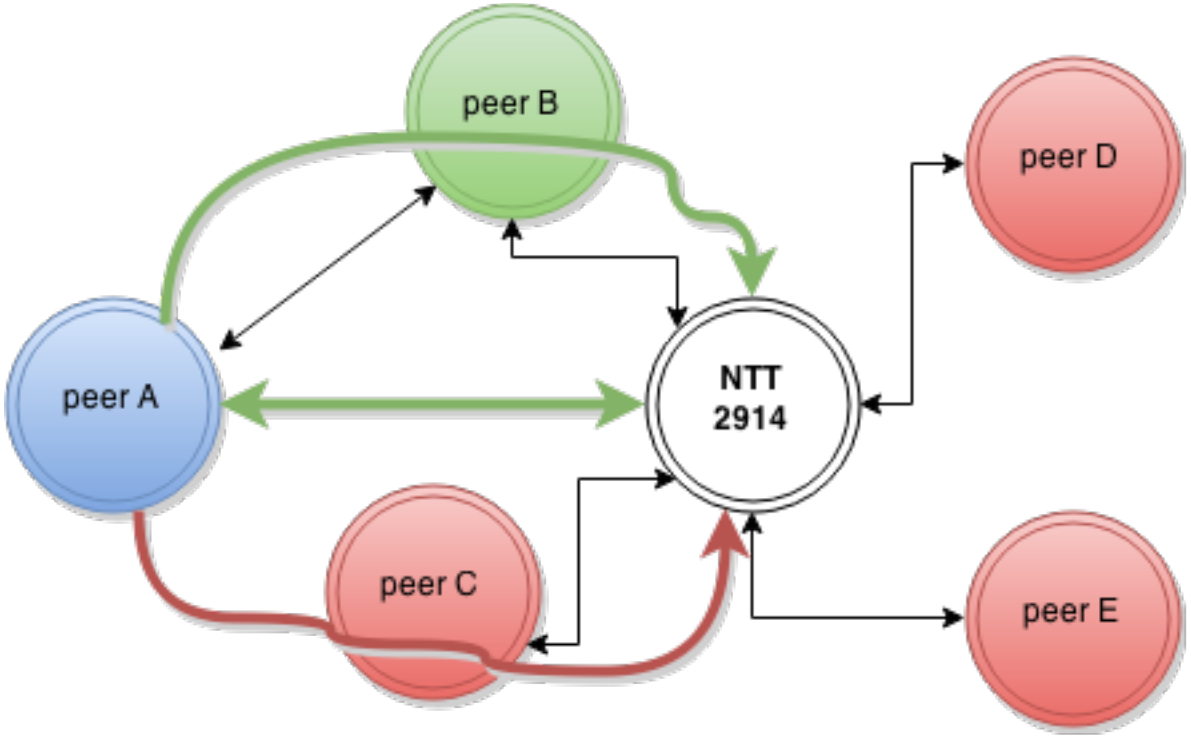
How do we know this? **We emailed them.**

example:

AS_PATH **2914_3491_7018** would be garbage!

Peerlock schematic goal

Given ASNs A, B, C, D, and E as our peers. Peer A subscribes to the peerlock idea (`Protected ASN`) and indicates that peer B is an "`Allowed Upstream`"



OK: ^A_

OK: ^B_A_

NOT OK: ^C_A_

NOT OK: ^D_A_

NOT OK: ^E_A_

Example cases:

- Prevent _7018_ routes from being accepted anywhere except on direct 7018 peering
- Allow only AS 3356 as upstream for peer PCCW globally (we don’t, but we could)

Deploying & Managing Peerlock

- “peer lock” is applied on **ALL eBGP sessions** (both customer sessions and peering sessions)
- “peer lock” is **entirely dynamic** through NTT’s network management web interface
- “peer lock” allows for **advanced** regional exceptions/rules
- **IT IS RECOMMENDABLE THAT BOTH PARTIES CONSENT TO PEERLOCK**

UI/table Mockup
Rules based approach

Protected ASN	Allowed Upstream	In What Region	Ignore Constraints	Active
3491	None	Everywhere	False	True
7018	None	Everywhere	True	True
65123	7018	US	False	True
4200000000	3491	Europe	False	True
4200000000	7018	US	False	True

Rule Constraints (unless overridden)

1. Both the `Protected ASN` and `Allowed Upstream` MUST be directly connected with eBGP sessions to the AS2914 backbone.

2. Only ASNs that connect with AS2914 in multiple regions are eligible to be used as an `Allowed Upstream`.

3. The `Allowed Upstream` field can only be set to ”None" in combination with in_what_region ”Everywhere”, if the `Protected ASN` connects with AS2914 in multiple regions.

4. An `Allowed Upstream` can only be specified for a region if the `Allowed Upstream` connects with AS2914 within that region.

Open Source Proof of Concept configuration generator

To facilitate in calculating what the proper as-path-sets are – I've published some python code. This is a variant what we used to validate the production implementation.

https://github.com/job/peerlock

WARNING: code is of Hazy Engineering Quality
WIN THE PRIZE: I've hidden one bug in the script

```
RP/0/RSP0/CPU0:r04.miamfl02.us.bb#show run as-path-set lock-AS7018-in
as-path-set lock-AS7018-in
  ios-regex '_174_',
  ios-regex '_701_',
  ios-regex '_1239_',
  ios-regex '_1299_',
  ios-regex '_2828_',
  ios-regex '_3257_',
  ios-regex '_3356_',
  ios-regex '_3491_',
  ios-regex '_3549_',
  ios-regex '_6762_',
  ios-regex '_6830_',
  ios-regex '_6939_',
  ios-regex '_7922_',
  ios-regex '_8283_'
end-set
!
```

These are generated

- per peer
- per region

```
{master}
job@r27.tokyjp05.jp.bb-re0> show configuration policy-options as-path lock-AS3491-in
".* (174|701|1239|1299|2828|3257|3356|3549|6762|6830|6939|7018|7922|8283) .* ";

{master}
job@r27.tokyjp05.jp.bb-re0>
```

Example workflow

1. Peering team engages with peer and **seeks permission, proposes initial ruleset**
2. Engineering evaluates if the initial proposed peer lock rules will **break the internet or not**
3. Deploy the ruleset in coordination with peer
4. Peers can contact your NOC for change requests, you commit to timely responses
5. Engineering approves/denies change requests to peer-lock rules

Example Technical Documentation for our eBGP peers

1. Contains configuration examples
2. Terminology
3. Disclaimer
4. Default operating mode
5. How to request changes / Who to contact

http://instituut.net/~job/peerlock_manual.pdf

Part 2

Dropping Bogon ASNs

Motivation:

- Occurrences of AS 23456 are misconfigurations or software bugs.
- Private/Reserved ASNs have no place in the global routing table

We should not reward misconfigurations by accepting these routes. The new paradigm: **fail hard & fail fast**.

NTT is not the only one: GTT, AT&T, KPN & DE-CIX have committed too for June/July 2016.

What Bogon ASNs to drop?

AS2914 will **NOT** accept route announcements from **ANY** eBGP neighbors which contain a “Bogon ASN” **anywhere** in the AS_PATH or its aggregate at.

Bogon ASNs are defined as:

0
23456
64496 – 131071
4200000000 – 4294967295

Based on: RFC5398, RFC6996, RFC7300

This policy is effective starting July 2016.
http://www.us.ntt.net/support/policy/routing.cfm#bogon

Config examples

http://as2914.net/bogon_asns/configuration_examples.txt

Currently have configs for BIRD, IOS XR, JunOS, IOS (yuck)

```
policy-options {
    as-path-group bogon-asns {
        as-path begin ".* 0 .*";
        as-path as_trans ".* 23456 .*";
        as-path reserved1 ".* [64496-131071] .*";
        as-path reserved2 ".* [4200000000-4294967295] .*";
    }
    policy-statement import_from_ebgp {
        term bogon-asns {
            from as-path-group bogon-asns;
            then reject;
        }
        term .....
    }
}
```

Part 3

Putting it all together: **Ingress**

1. Dynamic maximum prefix settings
2. Reject Bogon prefixes (RFC1918, etc)
3. Reject Bogon ASNs (AS0 / AS23456 etc)
4. Reject IXP prefixes (Some IXP subnets)
5. Reject leakage with the Peerlock filter
6. Match against IRR whitelist (only customers)
7. Mark as customer route (or as peer route)
8. Scrub internally significant BGP communities
9. Apply Features
   - (blackholing, traffic engineering, etc, only for customers)

Putting it all together: **egress**

1. Reject Bogon prefixes
2. remove-private-AS
3. Reject “bad” routes
4. Accept peer routes(on customer session)
5. Accept customer routes (on every session)
6. Do prepending (if requested & applicable)
7. Scrub internal communities
8. Set next-hop-self
9. Normalize Med

Questions, anytime, anywhere

job@ntt.net

Disclaimer: ISPs and their ASNs used in this talk are examples for discussion purpose only. NTT does not admit or deny any relationships with these entities.