inter.link

Sorry we messed up

How we route-leaked everything to everyone due to a fun Arista bug

Everyone is doing their job to keep the internet safe, right?

... check for RPKI invalids
... filter based on AS-SETS
... have max prefix-limits

Everyone is doing their job to keep the internet safe, right?

Everyone is doing their job to keep the internet safe, right?

... not every downstream

Everyone is doing their job to keep the internet safe, right?

... not every downstream

... not every Tier1 with Swedish roots

Everyone is doing their job to keep the internet safe, right?

... not every downstream

... not every Tier1 with Swedish roots

... and if you frick up, people will notice, very fast

(not mentioning names, sorry for the leak Ben)

We are very sorry, that our devices didn't do, what we told them to do.

- A story about Arista Routing Control Functions (RCF)
- BUG 1060542

What is RCF?

- Routing control functions (RCF) is a language that can be used to express route filtering and attribute modification logic in a powerful and programmatic fashion.

We use RCF to control bgp sessions & apply filters

DRAFT

- check for prefix lengths
- control/manipulate/change bgp communities
- control/manage bgp flowspec rules
- set next hop
- check for bogons
- drop RPKI invalids
- check as prefix and asn path lists
- filter TIER1 paths
- etc.

Every peer had a designated RCF code/rule

```
address-family ipv4
  neighbor 192.0.2.217 activate
  neighbor 192.0.2.217 rcf in V4_65536_IN()
  neighbor 192.0.2.217 rcf out V4_65536_OUT()
```

This was fine until some ASN wanted to have multiple bgp session on the same device, with different configs

... one session with a default route
... one session without a default route
... another session with a different AS-SET

So ... let's rename all RCF functions

New function names:

- Include IP Version
- Include Peer IP
- Include BGP session type / function
- ... do some string manipulation, nasty IPs have . and some people even use : in their IPs

Sessions with new functions names for rcf code

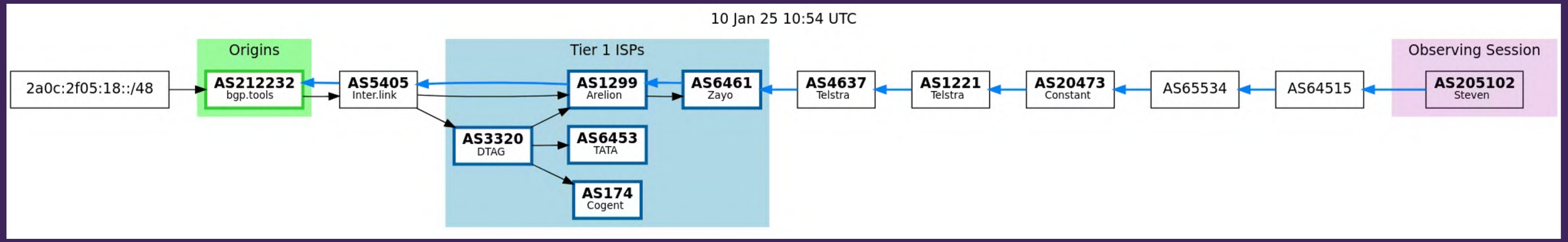
```
address-family flow-spec ipv4
   neighbor 192.0.2.217 activate
   neighbor 192.0.2.217 rcf in FLOWSPEC_V4_65536_192_0_2_217_IN()
   neighbor 192.0.2.217 rcf out FLOWSPEC_V4_65536_192_0_2_217_OUT()

address-family ipv4
   neighbor 192.0.2.217 activate
   neighbor 192.0.2.217 rcf in V4_65536_192_0_2_217_IN()
   neighbor 192.0.2.217 rcf out V4_65536_192_0_2_217_OUT()
```

And apply the new functions to bgp sessions

- Generate config via ansible
- copy generated config to device
- get a configuration session on the device
- copy the generated config to the current config
- generate diff
- do a configuration checkpoint
- commit the new config with a commit timer

Add any https://www.reddit.com/r/Whatcouldgowrong posting here


10 Jan 25 10:54 UTC
Origins
Tier 1 ISPs
Observing Session
2a0c:2f05:18::/48
AS212232
bgp.tools
AS5405
Inter.link
AS1299
Arelion
AS6461
Zayo
AS3320
DTAG
AS6453
TATA
AS174
Cogent
AS4637
Telstra
AS1221
Telstra
AS20473
Constant
AS65534
AS64515
AS205102
Steven


THE WORLD IS
TOTALLY SCREWED!

WHAT HAPPENED

Expected behavior according to vendo :

DRAFT

When renaming an RCF function (simultaneous modification in BGP config and RCF function name),
we expect the following steps to take place:

- After the config session has been committed, the RCF agent will take some time to compile the new function.
- While this is pending, the BGP neighbours using this RCF function as a filtering rule, are linked with an undefined function from a BGP agent perspective. We therefore expect to withdraw prefixes to these peers during the function compilation. So we expect this to be disruptive, but we don't expect to leak all prefixes during this phase (which is the behaviour you experienced).
- Once the RCF function is compiled and active, we are advertising/receiving only the prefixes allowed by this function.

What actually happened, according to vendor:

- While the RCF agent was compiling, BGP would start using the new function names, which were undefined.
- This meant that all of these peers intermittently shared the same undefined RCF config until RCF finished compiling.
- Some of these peers were low scale and some were high scale.
- The peers all started to join the same update-group.
- The update-group for one of the high scale peers was reused.
- Two of the low scale peers joined the high scale peers update-group and started reconciling with that update-groups advertised state, before the update-group finished processing the new undefined RCF function state which would withdraw all routes.
- As a consequence, we were leaking prefixes toward these 2 low scale peers and causing the sessions to flap as downstream sent a notification that we exceeded the route limit.

Obviously we tested the changes before deployment

At the time we assumed it would be hit-less, we started rollout in

- 34 Pops, 15 Countries, about 70 devices
- Some devices have:
  - a few customers or
  - many customers
  - IXPs
  - Upstreams
- In the beginning we saw no impact, we started at lower populated devices. Outcome varied based on device population.

Workaround / Fix

- Duplicate the RCF function or have the new function map to the old one
- After commiting wait for new RCF function to be added
- Update the BGP configuration to use the new RCF function
- Remove the old unused RCF function

or

- Shut down peers before renaming associated RCF functions

Fix

- BUG1076200 has been fixed by development and merged into the EOS code. Fix will be available in the next maintenance releases for EOS 4.31, 4.32 and 4.33 and in any release train above that (>= 4.34).

Again, we are very sorry, that our devices didn't do, what we told them to do.

- It will probably happen again.