Monitoring and Securing BGP on Mikrotik Routers

Ben Ryall | Lee Hetherington
Teraco Virtual Tech Day - October 12th 2023

What we're talking about

- BGP session monitoring on MikroTik routers
  - You want to know sessions are down, right?
- Enabling RPKI on your external BGP sessions
- Generation and deployment of strict IRR filters
  - Mis-configured sessions can cause pain

Why did we do this?

- Everyone needs a hobby
- Lee built as35008, using MikroTik routers
- Lightweight Operations, but MANRS compliant
  - Strict IRR Filters, RPKI, Anti-Spoofing etc
  - Prove how easy it is todo the ‘right thing’ without big vendor iron
- Where tools didn’t exist, coded them!

MANRS

What do I need?

- Mikrotik router(s) running ROS 7+ (We've tested against 7.7 through 7.11.2) and speaking BGP
- API configured and accessible on your router(s)
- Mikrotik RouterOS API Python Packages
- BGPQ4 installed on your management host

Monitoring

BGP Monitoring

- There are two scripts in this collection
  - Which will help you monitor BGP sessions on your Mikrotik Routers
  - MikroTik is **STILL** lacking BGP support in their SNMP implementation
- mikrotik_bgpmon.py
- mikrotik_bgpmon_print.py

BGP Monitoring…part 2

- mikrotik_bgpmon.py
  - This script will look at configured (ignoring disabled) peers under /routing/bgp/connection and compare them with the items under /routing/bgp/session to see if they match
  - It'll also look at the status of sessions under /routing/bgp/session and alert you of any status that isn't established
  - It'll also send you an email with the output each time you run the script if you specify an email address

```
Skipping disabled connection: ipv6.sfmix.lg
Alerts generated:
Alert: BGP connection ipv4.sfmix.as8674 with 8674 is configured but not found in running sessions.
Alert: BGP connection ipv6.sfmix.as8674 with 8674 is configured but not found in running sessions.
Alert: BGP connection ipv4.sfmix.as21928 with 21928 is configured but not found in running sessions.
Alert: BGP connection ipv6.sfmix.as21928 with 21928 is configured but not found in running sessions.
```

BGP Monitoring…part 3

- mikrotik_bgpmon_print.py
  - This script will display the sessions currently running on the router
  - It doesn't look at things which are in /routing/bgp/connection that are not also in /routing/bgp/session - so it shouldn't be used to monitor the health
  - If you supply the routerIP, then up or down to the script at the command line, it'll show you the status

```
Session: ipv6.sfmix.as32934-1-1, AS: 32934, Peer IP: 2001:504:30::ba03:2934:1, Status: Established, Uptime: 1w1d17h49m16s880ms, Prefixes: 24
Session: ipv4.sfmix.as32934-2-1, AS: 32934, Peer IP: 206.197.187.92, Status: Established, Uptime: 2w2d18h26m16s930ms, Prefixes: 41
Session: ipv6.sfmix.as32934-2-1, AS: 32934, Peer IP: 2001:504:30::ba03:2934:2, Status: Established, Uptime: 2w2d18h26m15s40ms, Prefixes: 14
```

Securing BGP

Part 1 - Deploying Strict IRR Filters

- This collection of scripts will take your list of peers and generate filters, and push them to your router(s)
- Expects (can change) to be installed into /usr/share/mikrotik-irrupdater/
- Runs bgpq4 for you, to generate ASN/AS-SET pair prefix filters for your defined peers/customers
- Configure it, run the two wrappers in cron and you're good to go!
- Be conscious of Flash/SSD wear
    - Lots of config updates, depending on how regularly you run this

Deploying Strict IRR Filters… Part 2, Configuration

- **config/routers.conf**
  - Username and Password for the Mikrotik API
- **config/peers.conf**
  - Comma separated list of ASN,AS-SET pairs
  - Used by the wrapper to call bgpq4 to generate prefix lists
- **config/sessions.conf**
  - Comma separated list of ASN, SLUG, ROUTER-IP
  - Used to build the actual filters, ensuring the right naming etc

Deploying Strict IRR Filters… Part 3, Automation

- **buildprefixes.sh** - This will run bgpq4 based on the contents of your peers.conf, and then after run a filtergen script, to take the output of bgpq4 and turn it into the right format to build policy
- **pushfilters.sh** - This pushes the filters to your router, checks for required updates, if needed, removes old filter and replaces with a new version
- Run these from cron on your desired schedule – **buildprefixes.sh** takes some time to run, depending on your peers and the size of the as-set!

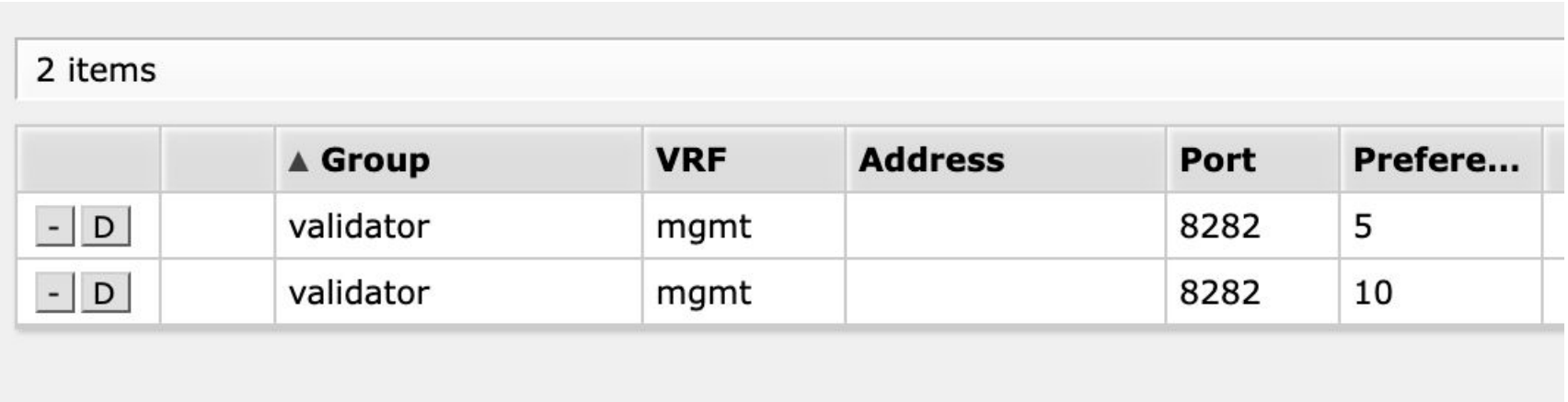
Deploying Strict IRR Filters… Part 4, Slugs?

- Think of a slug like a short name, to classify a group of peers who have similar policy applied
- Really, we're using this to avoid duplication of policy config, by using this slug to jump between policies - much like you can with other vendors.
- Example here, shows the end of the Meta AS32934 policy, jumping to the sfmix-import policy. **sfmix** is the slug in this case

- D		6352		as32934-sfmix-import-ipv6	if (dst==2c0f:ef78:6::/48) { jump sfmix-import }
- D		6353		as32934-sfmix-import-ipv6	if (dst==2c0f:ef78:9::/48) { jump sfmix-import }
- D		6354		as32934-sfmix-import-ipv6	if (dst in 2c0f:ef78:c::/46 && dst-len<=48) { jump sfmix-import }
- D		6355		as32934-sfmix-import-ipv6	if (dst in 2c0f:ef78:10::/47 && dst-len<=48) { jump sfmix-import }
- D		6356		as32934-sfmix-import-ipv6	if (dst==2c0f:ef78:12::/48) { jump sfmix-import }
- D		6357		as32934-sfmix-import-ipv6	reject

Part 2 - RPKI Origin Validation

- Need to have access to a validator – ideally more than 1 for resilience
  - We run these in docker containers on our management hosts
- Configuring the router is pretty straightforward
  - Everything is under Routing > RPKI in the WebUI



2 items

		▲ Group	VRF	Address	Port	Prefere...
- D		validator	mgmt		8282	5
- D		validator	mgmt		8282	10

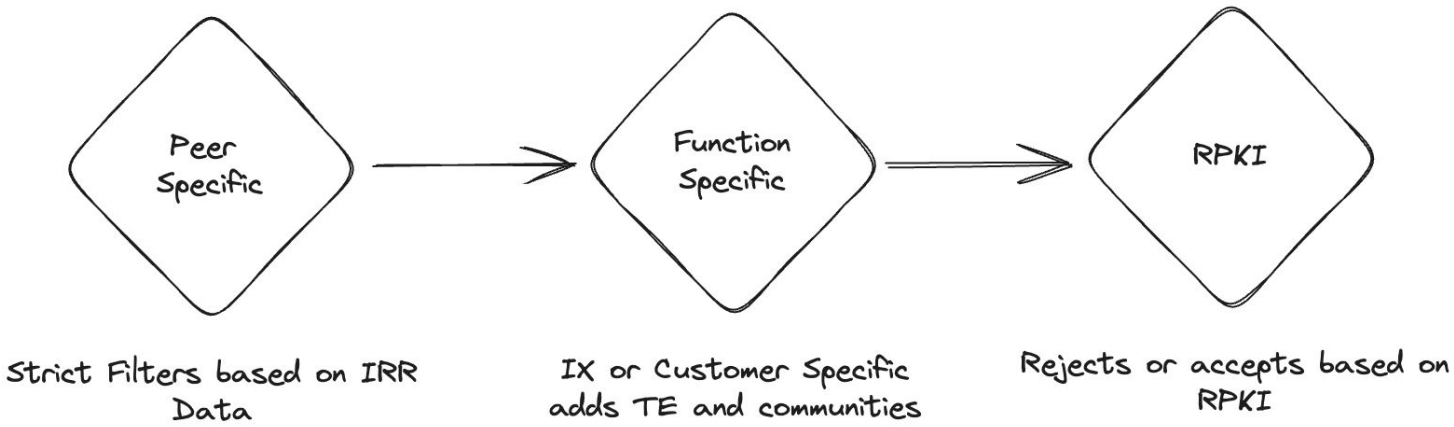
RPKI Origin Validation… Part 2

- BGP import policy required to perform the desired action
  - For us, we're rejecting invalids from all session types (Transit, Peers, Customers)
  - We're calling this rpki-import filter from other filters as the last step

	36	RPKI Imp…	rpki-import	rpki-verify validator
	37		rpki-import	if (rpki invalid) { reject } else { accept }

Part 3 - Bringing it all together with filters

- Strict IRR filters on Peers and Customers
- RPKI validation on all external BGP sessions
- Using the jump expression in filters, to hop between policies, avoiding duplicate config



What does that look like?

- Example here of Meta as32934 at SFMIX in San Francisco, USA

- The as32934 specific filter then calls the sfmix-import filter rather than ‘accept’

	8	SFMIX Im…	sfmix-import	set bgp-communities 35008:12276,35008:12,35008:51; set bgp-local-pref 95; set bgp-med 5; jump rpki-import

- The sfmix-import filter then calls the rpki-import filter

	36	RPKI Imp…	rpki-import	rpki-verify validator
	37		rpki-import	if (rpki invalid) { reject } else { accept }

BGP Process Tuning

- By default, BGP peers share a single process - which ends up flattening a single CPU core
  - Our peering boxes have 16 Cores and 16Gb Ram (CCR2116-12G-4S+)
- Sessions can be isolated into their own process - slightly increases memory usage, but helps with CPU usage during BGP events
- Particularly helpful with IRR filter updates due to the route refresh when the filter changes
- Find this under “extra” on BGP session configuration in the WebUI

Questions?

Github Links

BGP Monitoring Scripts

IRR Updating Scripts

We also have IRR Updating Scripts for Juniper too!