BGP FlowSpec Services

beyond DDOS mitigation

ITNOG7 - 10/05/2023

nicola modena - CCIE #19119 / JNCIE-SP #986
nicola@modena.to - @nmodena

Agenda

- BGP FlowSpec origins & configuration
- BGP-FS Service 1 – flow based egress engineering
- BGP-FS Service 2 – bidirectional traffic steering
- BGP-FS Service 3 - NFV

About me

Nicola Modena - CCIE #19119 R&S (15Y) / JNCIE-SP #986 Emeritus

Independent Network Architect

More than 25 years experience designing and implementing service provider and large enterprise networks.

https://linkedin.com/in/nmodena

1

BGP FlowSpec

Distributed Policy Based Routing

BGP FlowSpec

«Dissemination of Flow Specification Rules» [for IPv6]

Defined in RFC5575 (2009) up by RFC7674, RFC8955 for IPv4, RFC8955 for IPv6

some draft exist for specific functions (if-group / persistence / SR)

in a nutshell:

- Distributed PBR (Policy Based Routing)
- Signaled with BGP with a dedicated AFI/SAFI
- Mostly used for DDOS mitigation

NOTE: FlowSpec <is not> OpenFlow <and> <is not> NetFlow

BGP inet4/6 AFI/SAFI vs BGP FlowSpec

BGP inet	**Destination**	**Next-hop**
	vs	
BGP FS	**Flow Specification**	**Action**
	Src/Dst Address/Subnet Src/Dst Port/Range IP Protocol ICMP Type/Code TCP Flags Packet Lenght DSCP Value Fragment Bits	Traffic Rate BPS/PPS Drop [Rate = 0] Send to VRF Set DSCP Sample Redirect NH

Example: Drop all UDP traffic sourced from port 123 with dest IP 192.0.0.0/24

Router (client) configuration

IOS XR

```
!*** enable AFI/SAFI ***
router bgp $ASN$
   ...
   address-family ipv4 flowspec
   address-family ipv6 flowspec
   !
   neighbor $RR$
    ...
    address-family ipv4 flowspec
      route-policy FLOWSPEC4-FILTER-IN in
      maximum-prefix 1000 95 discard-extra-paths
    !
    address-family ipv6 flowspec
      route-policy FLOWSPEC6-FILTER-IN in
      maximum-prefix 1000 95 discard-extra-paths
!!
!*** activate on the platform ***

flowspec
   local-install interface-all
!
!*** disable on specific interfaces ***

interface XXXX
   ipv4 flowspec disable
   ipv6 flowspec disable
```

Junos

```
/*** enable AFI/SAFI ***/
protocols {
    bgp {
        group iBGP {
             import [.. FLOWSPEC-FILTER-IN ]
             family inet {
                flow {
                     accepted-prefix-limit {
                         maximum 1000;
                    }
                }
                family inet6 {
                   flow {
               [...]
}}
/*** activate on the platform ***/
routing-options {
    flow {
        interface-group 1 exclude;
        term-order standard;
}}

/*** disable on specific interfaces ***/

interfaces XXXX unit 0 family inet filter group 1
interfaces XXXX unit 0 family inet6 filter group 1
```

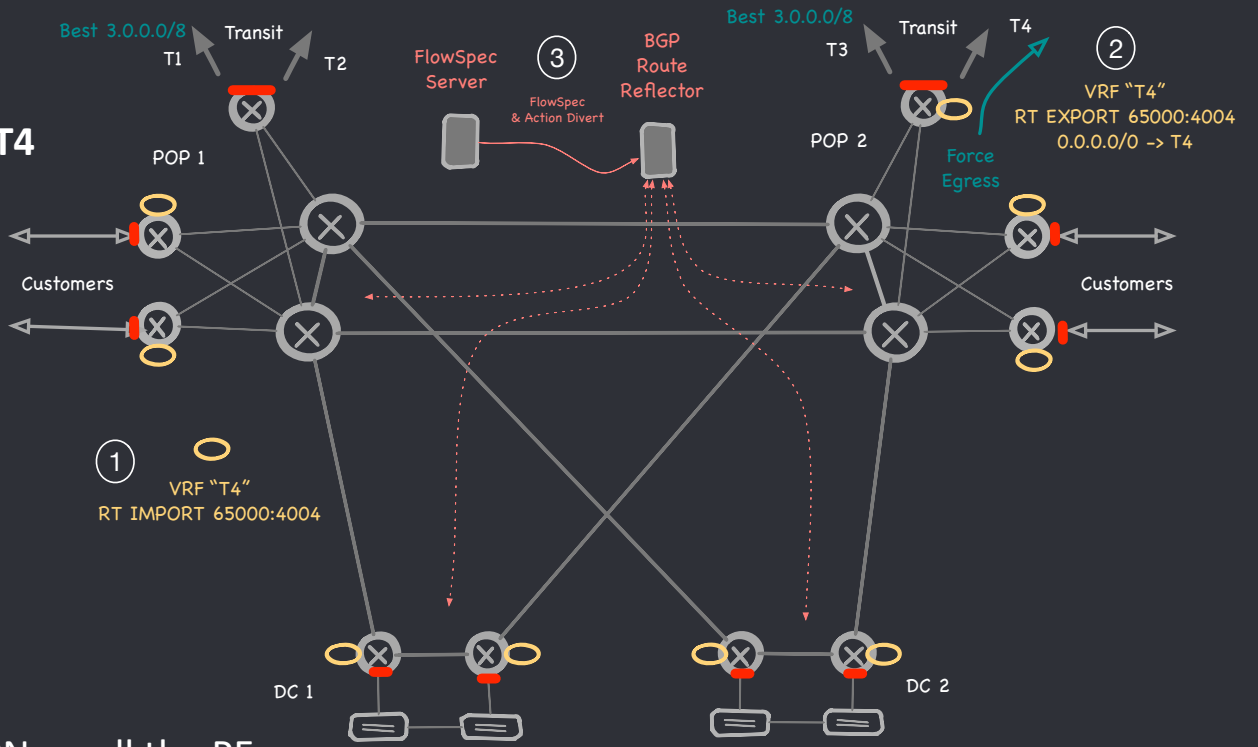
2

use case 1 : Flows-based egress engineering

bypass routing for specific traffic flows

Flows-based egress engineering

Force traffic out trough T4 regardless of routing



1) Configure «T4» L3VPN on all the PE
2) Create a «T4-EXIT» MPLS L3VPN exporting 0/0 pointing to T4 as next-hop
3) Distribute a FlowSpec definition to divert required traffic to VRF «T4»

Flows-based egress engineering

ExaBGP as Policy Injector

https://github.com/Exa-Networks/exabgp

ExaBGP

```
neighbor $route-reflector$ {
  local-as $ASN$;                          ## ( 1 ) ##
  peer-as $ASN$;
  [...]

  family {
    ipv4 flow;
  }

  flow {
    route DC2-to-AWS-via-T4 {
      match {                              ## ( 2 ) ##
        source 192.0.2.0/24;
        destination 3.0.0.0/8;
      }
      then {
        # redirect to vrf T4 (
        redirect 65000:4004;               ## ( 3 ) ##
      }
    }
}
```

diversion policy:

1) peering parameters
2) flow description
3) redirect to VRF with RT65000:4004

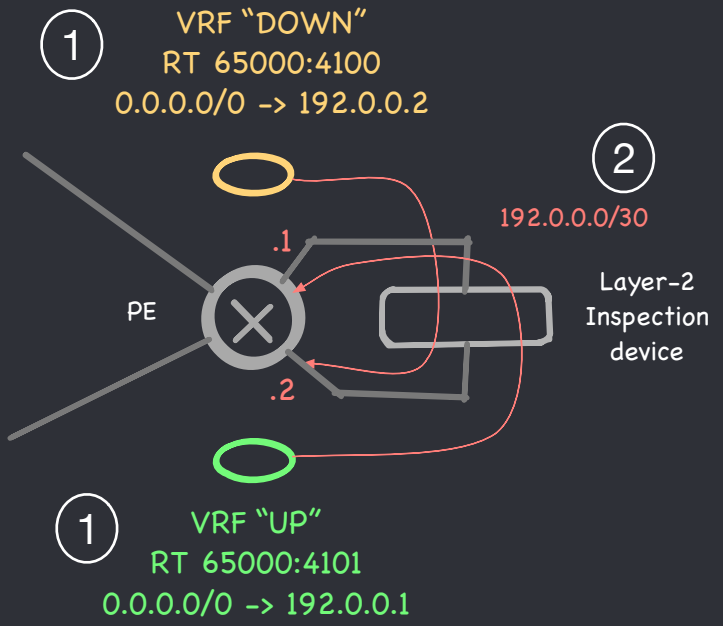
3 use case 2 : bidirectional traffic steering

Bidirectional traffic steering

Force bidirectional transit trough L2 device

ExaBGP

```
[...]
flow {
  route CUST-UP {              <- UPSREAM TRAFFIC FLOW
    match {
      source 192.0.2.0/24;
      destination 100.0.2.0/24;
    }
    then {
      redirect 65000:4101;       // RT destination VRF
  }}
  route CUST-DOWN {            <- DOWNSTREAM TRAFFIC FLOW
    match {
      source 100.0.2.0/24;
      destination 192.0.2.0/24;
    }
    then {
      redirect 65000:4100;       // RT destination VRF
}}}
```



1) UP & DOWN L3VPN with default-route leaking and next-hop trough Layer-2 device
2) point-to-point link in Global Routing Table (Without IGP Adjecency!)

Flows-based steering & egress engineering

- Useful for temporary traffic diversion
- Quick solution without any backbone configuration change
- VRF for traffic diversion can be permanently defined

  -> (just 1 FIB entry x VRF)

advice:

- check/set default platform diversion action if vrf doesn't exist

  -> (drop -> forward)
- provide fallback if next-hop/interface goes down

  -> (floating default route)

4 use case 3 : traffic steering for NFV

NFV with BGP FlowSpec

example:

- Analyze ALL DNS traffic for selected customers

  (es: who have subscribed for parental-control)

but also valid for other scenario:

- Intercept all web traffic to trigger redirect to a captive portal for user activation/deactivation (and block the remaining traffic)
- Insert a pool of caching proxy/waf in front of web server
- as an infrastructure for almost any NFV solution

NFV with BGP FlowSpec

Requirement: Service Provider Class Solutions

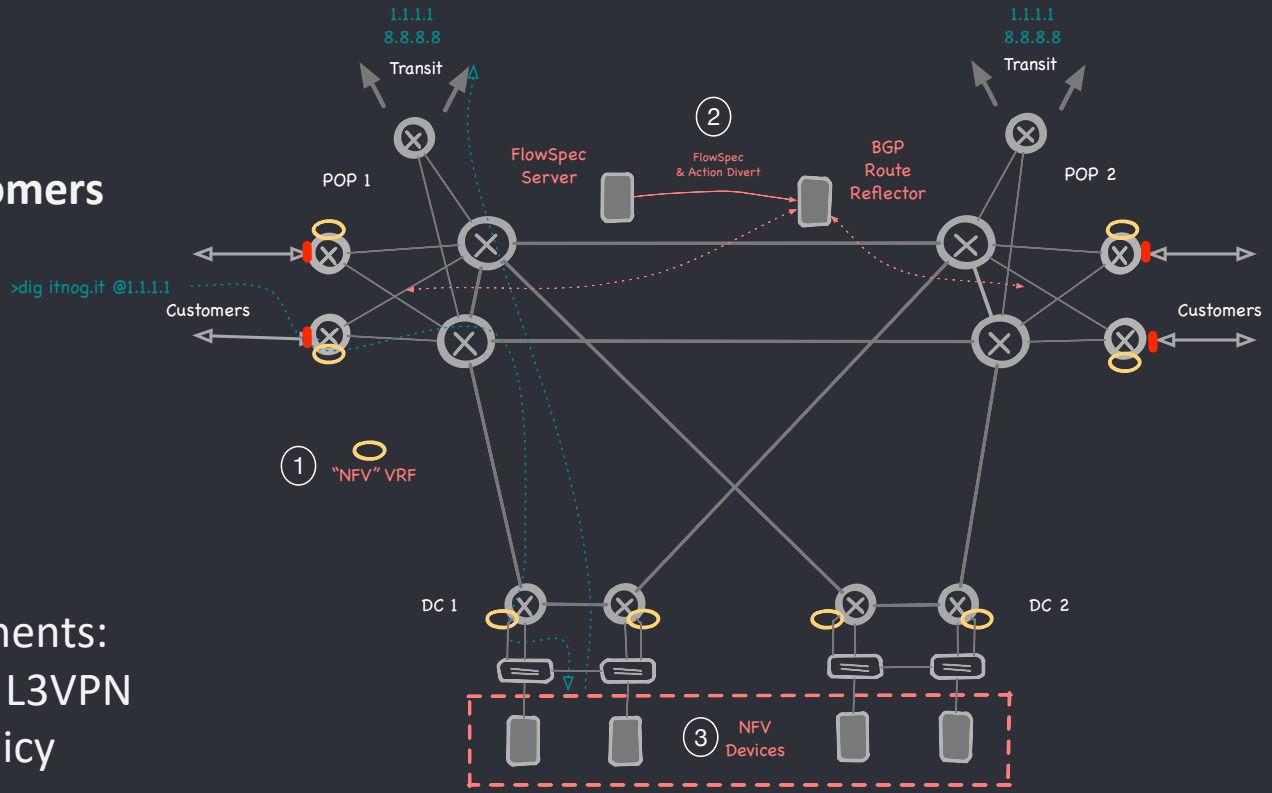
- Dynamic & Flexible -> BGP FlowSpec
- Load Balance -> BGP Multipath
- Proximity -> BGP path selection (IGP Metric)
- Reliable -> BGP for HA
- Scalable -> BGP can scale ?

Guess what my favorite protocol is?

NFV with BGP FlowSpec

Analyze all customers

DNS traffic



Solution Components:

1) «NFV» MPLS L3VPN
2) FlowSpec policy
3) NFV Devices

NFV with BGP FlowSpec

ExaBGP

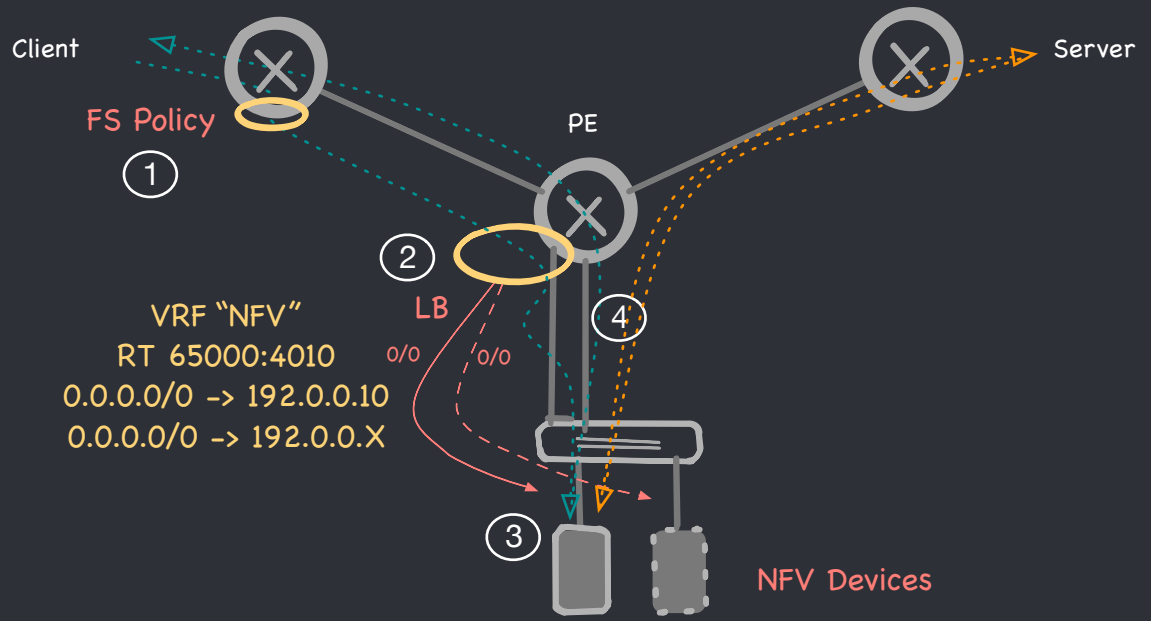
```
[...]
 flow {
    route  parental-control-pool-1 {
        match {
            source 100.64.0.0/16;              ## ( 1 ) ##
            destination-port 53;
            protocol udp;
        }
        then {
            # install on BNG 1 & BNG 3         ## ( 2 ) ##
            community [65000:48011 65000:48012];

            # redirect to NFV                  ## ( 3 ) ##
            redirect 65000:4010;
}}}}
```

Activate the diversion defining the policy

1) flow description
2) optional community to control distribution
3) redirect flow pointing to VRF RT 65000:4010

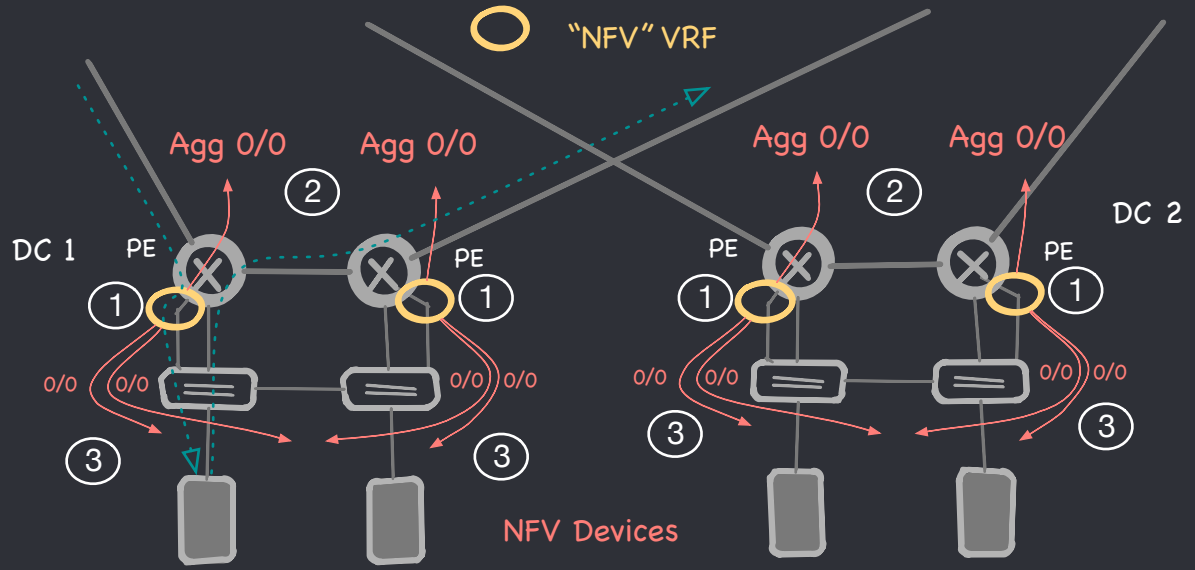
NFV with BGP FlowSpec – traffic flow



Traffic Flow

1) FlowSpec policy divert upstream traffic
2) Traffic exit from NFV vrf on dedicated PE interface and distributed trough NFV devices
3) Devices receive traffic and perform DNAT for «catch all» services
4) Return traffic and sessions to real destinations uses PE interface in Global Routing Table

NFV with BGP FlowSpec

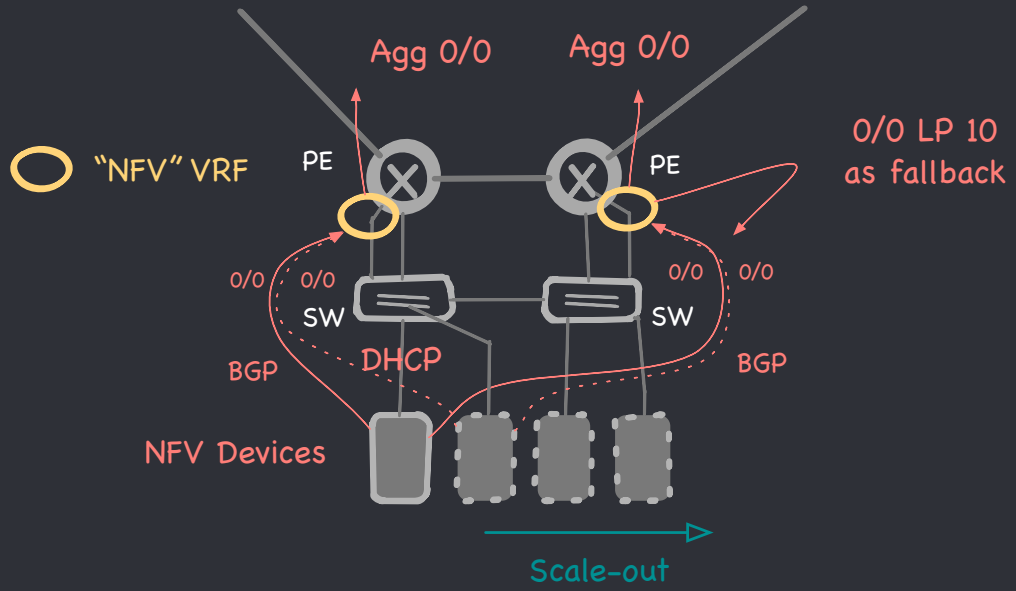


NFV VRF exit points:

1) At least 2 PE routers in any DC with dedicated [sub] interface
2) Only an «aggregate» default-route advertised from each PE in «NFV» VPN
3) Remote PE will select the closer exit-point using IGP cost (proximity)
   - IP lookup and load balacing it's performed only on exit-point
   - Less routing information is distributed to remote PE

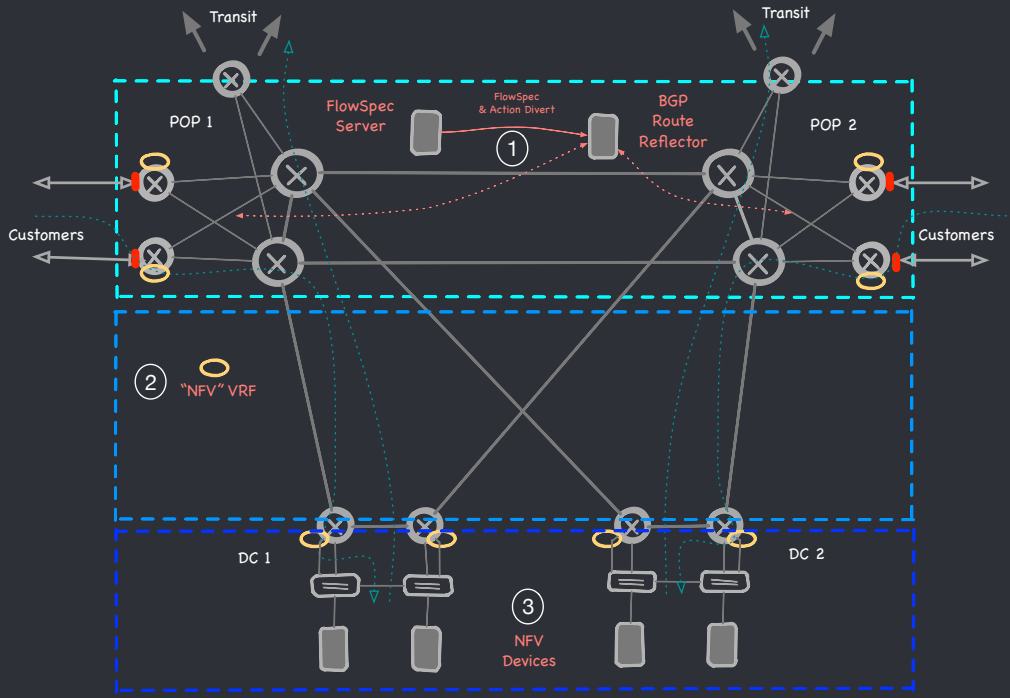
NFV with BGP FlowSpec

Scale-out NFV solution with BGP



- NFV as VM using dynamic IP via DHCP
- Setup 2 BGP session with PE interfaces in VRF NFV (hint: ExaBGP)
- Advertise default-route to PE in NFV vrf pointing to the NFV device (not installed in NFV RIB)
- NFV uses default-gw in Global Routing Table
- Ready to migrate to container and K8S

NFV with BGP FlowSpec



The solution is divided into 3 layer:

1 - Traffic diversion (BGP FlowSpec)
2 - Optimal traffic distribution & fallback (MPLS L3VPN)
3 - High Availability, Load Balancing and Scale-Out (BGP Session & Multipath)
Each layer is **independent** and self-contained in providing the required functionality
The common thread is BGP but used in three different ways

5 Summary

Best Practice

Apply policy only on edge interface / exclude core interfaces

Policies must be applied only once to avoid traffic loops

Implement import policy to prevent Control-Plane interruptions

ML, AI and expecially humans can be very smart creating policy ☺

es. prevent traffic filtering to TCP 179 from trusted source.. (Bridging Gap Protocol ☺)

Organize and tag FlowSpec policies with custom communities

in order to filter/apply policy only on specific devices type (es: internal, external)

Read carefully device capacity and limit the number of entry accepted

typically from a few hundred to a few thousand entries

flowspec rules are implemented in HW like ACL

limit max accepted prefix per AFI/SAFI **AFTER** import-policy enforcement

Summary

- BGP FlowSpec it’s a powerful toolset
- Misconsidered exclusively as a component for DDOS
- Flexible services can be created with just a few configuration lines
- NFV with Flowspec it’s more flexible & controllable than plain anycast

CONS

- it’s still PBR -> does not scale on device
- HW dependent -> check support & limits on each platform
- use with care, traffic loops are lurking

- Is this enough SDN ? ☺

a special thank to:
Ivan Pepelnjak for invaluable input

0

THANK YOU

Questions ?

An extended version of this presentation (and future updates) at https://github.com/nmodena/blog