What You Make Possible

BUILT FOR
THE HUMAN
NETWORK

CISCO

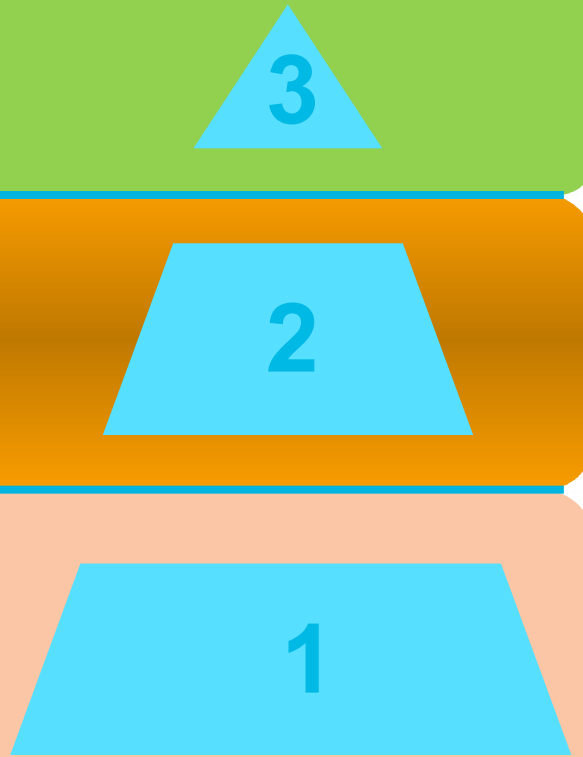
A simple DDoS mitigation mechanism explained

Bertrand Duvivier, bduvivie@cisco.com
Gunter Van de Velde,
gvandeve@cisco.com

BUILT FOR THE HUMAN NETWORK
CISCO

DDoS Mitigation Adoption Cycle

- Phase III
  - Dynamic application aware redirection and traffic handling
- Phase II
  - Malicious traffic mitigation
  - Cleaning of Malicious traffic
  - Dirty and clean traffic handling
  - Usage of Multi-instance BGP
- Phase I
  - ACL
  - RTBH
  - PBR
  - uRPF



DDoS Overview

Distributed denial-of-service (DDoS) attacks target network infrastructures or computer services by sending overwhelming number of service requests to the server from many sources.

Server resources are used up in serving the fake requests resulting in denial or degradation of legitimate service requests to be served

Addressing DDoS attacks

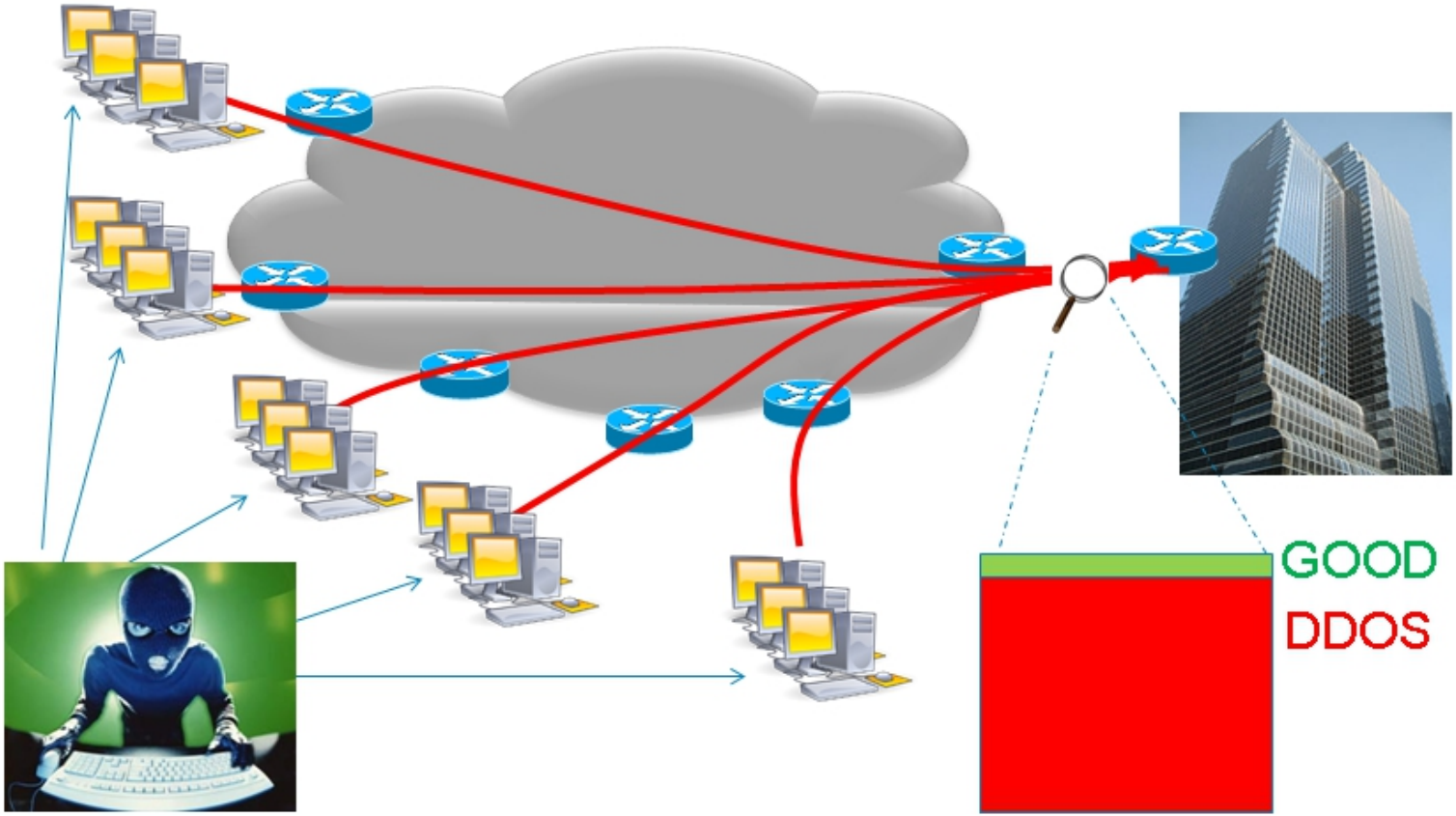
Detection – Detect incoming fake requests

Mitigation

Diversion – Send traffic to a specialized device that removes the fake packets from the traffic stream while retaining the legitimate packets

Return – Send back the clean traffic to the server

DDOS impact on Customer Business

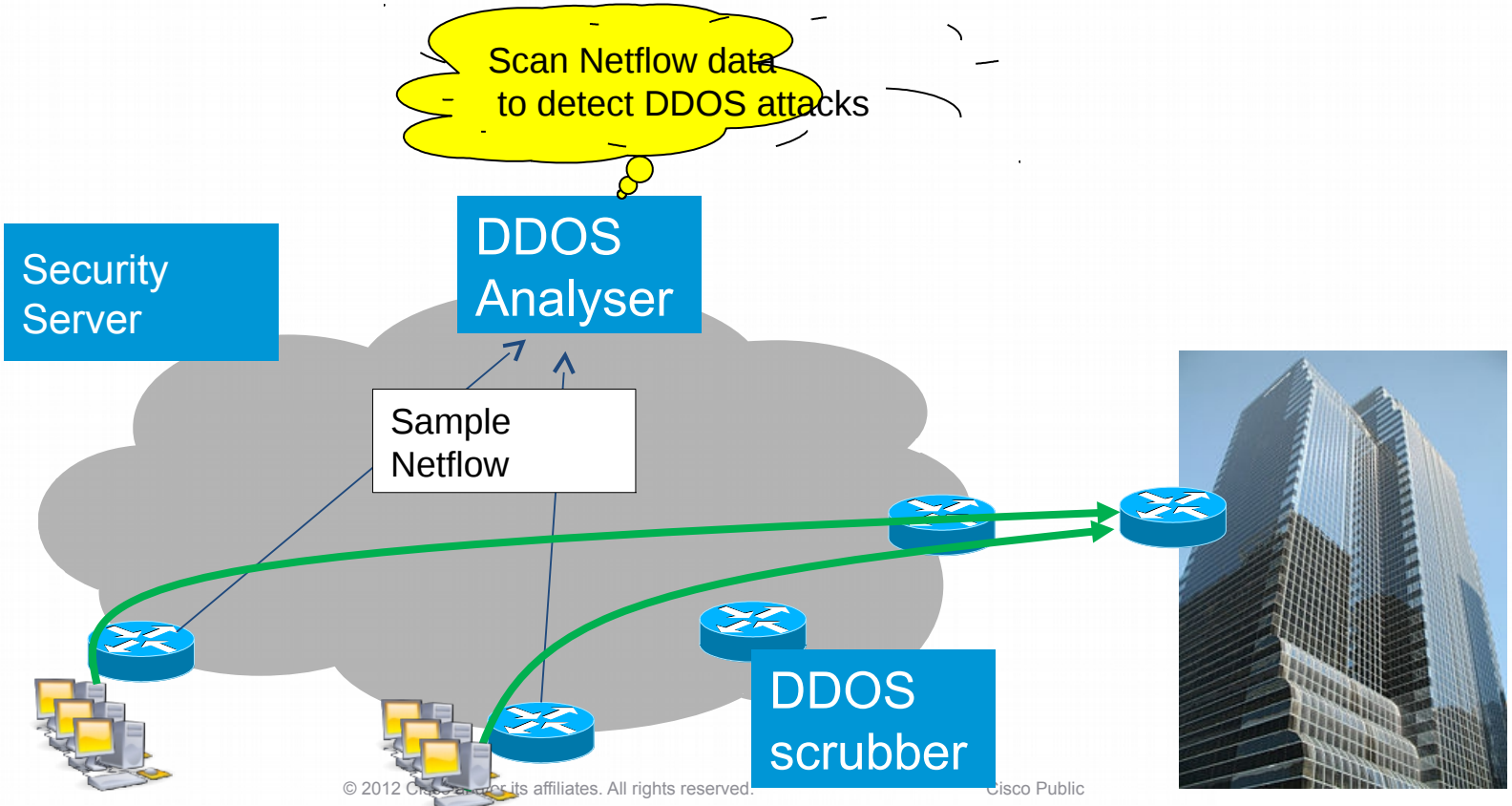


DDOS impact on customer Business

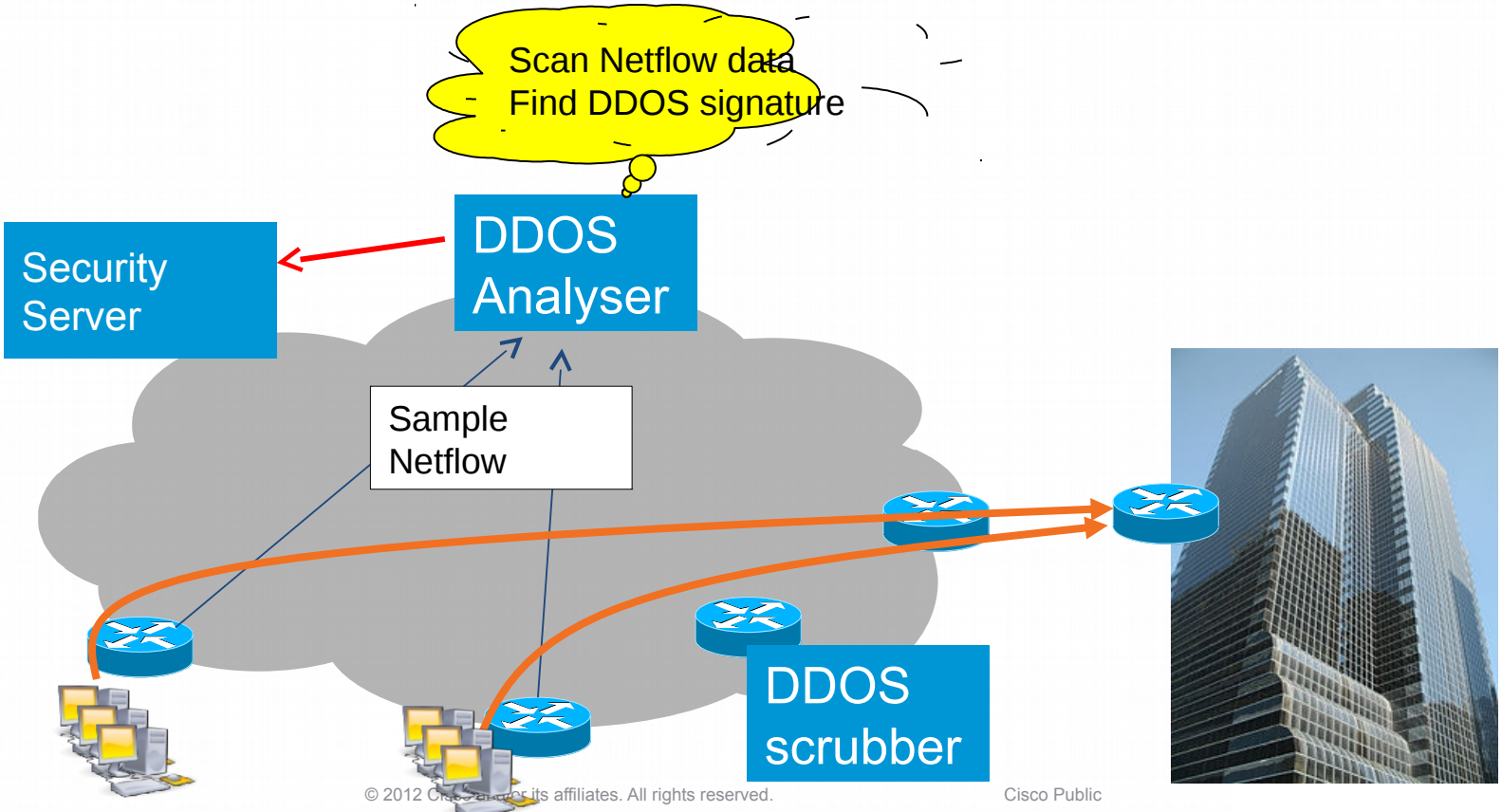
- Enterprise customer can’t defend themselve, when DDoS hit the FW… it’s already too late.
- SP could protect enterprise by cleaning DDoS traffic at ingress peering point.
- New revenue for SP.
- Mandated service to propose to Financial and visible customers.

DDoS mitigation architecture

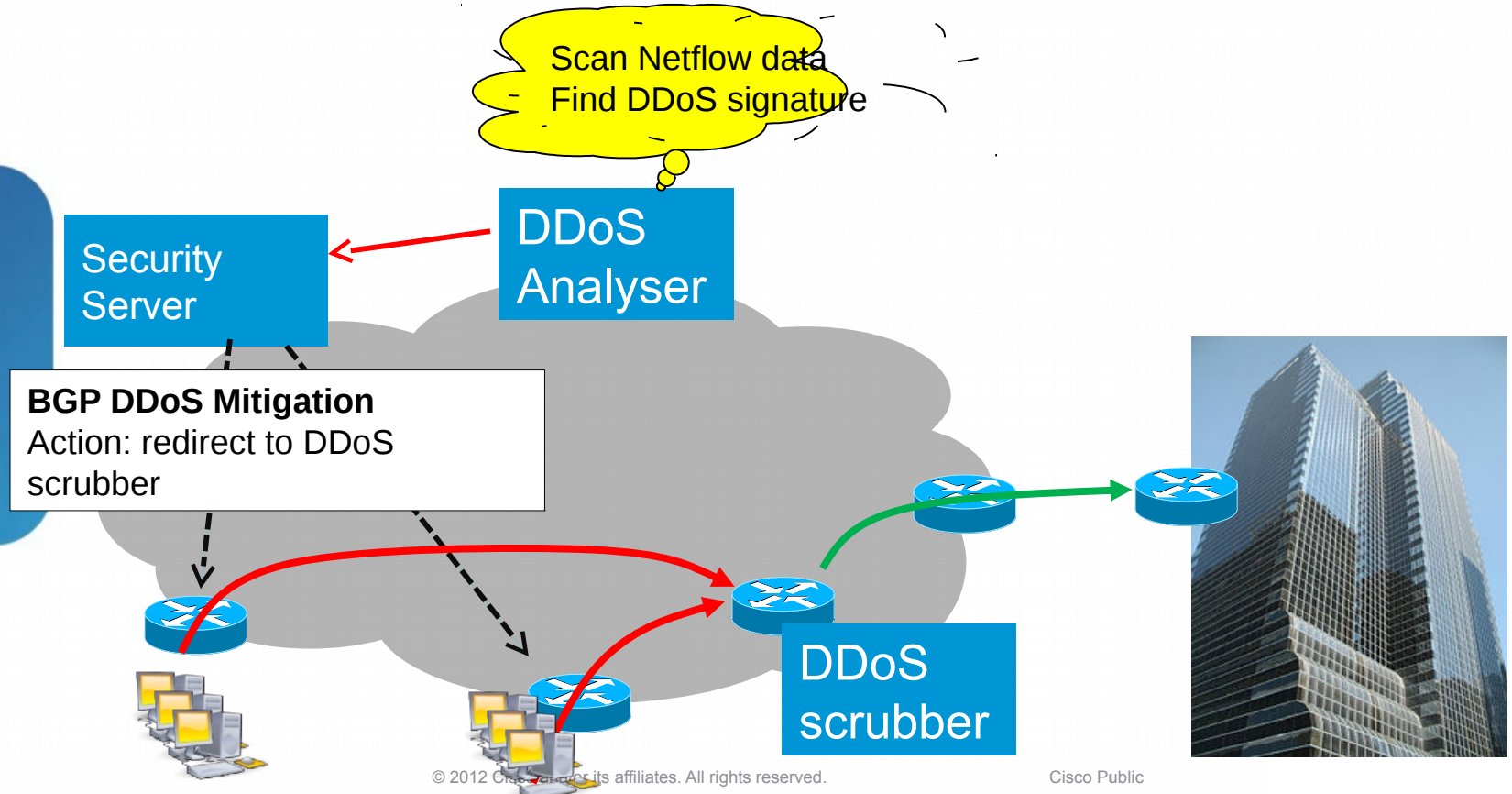
1. Detection (no DDoS)



DDoS mitigation architecture
2. Detection (DDOS)

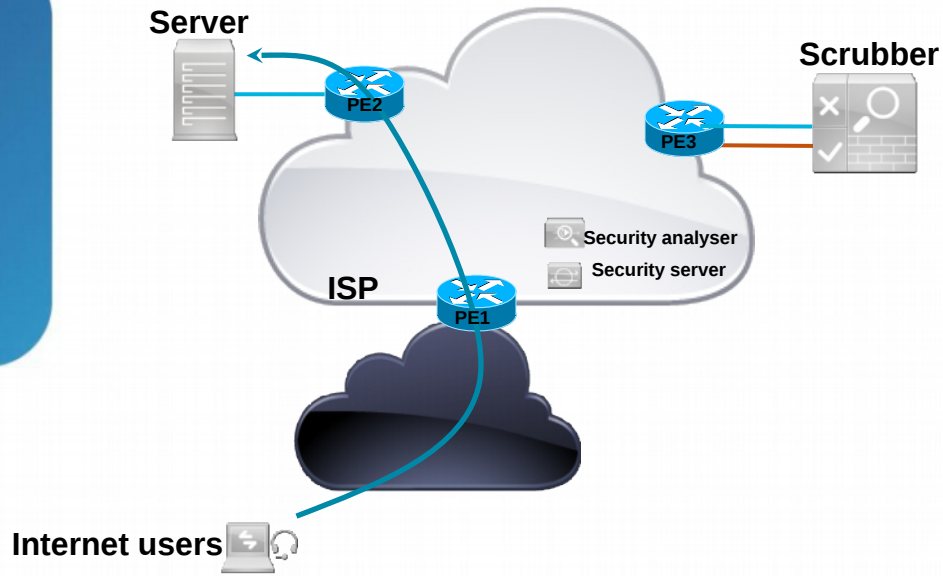


DDoS mitigation architecture
3. Redirect traffic to DDOS scruber



The concept

Traffic under normal conditions



Traffic under normalized conditions

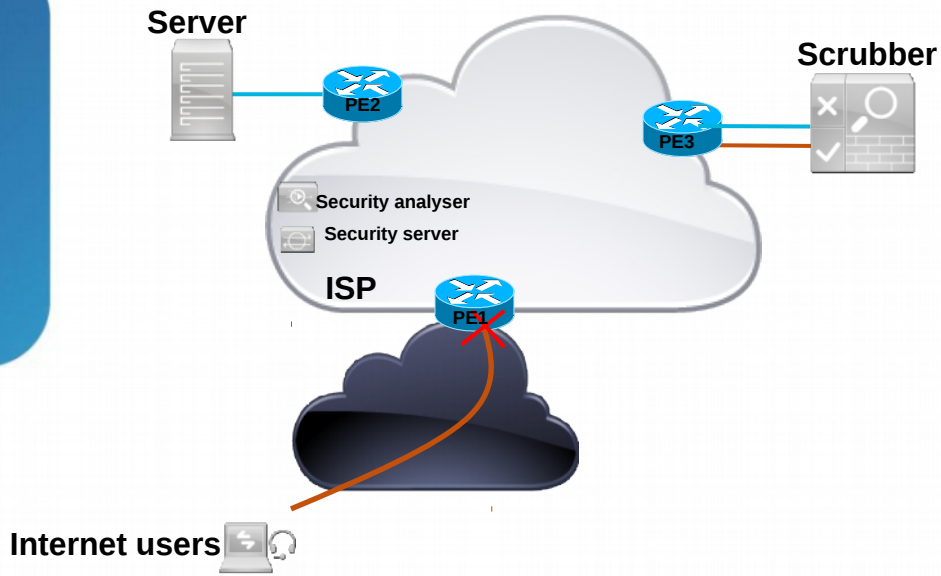
- Traffic takes shortest path
- Upstream and downstream traffic follow traditional routing
- ALL interfaces are in the GLOBAL routing table

Pre-provisioned DDoS instrumentation

- Traffic Scrubber
  Separate clean and malicious traffic
- Security Analyser
  Analyses Netflow/IPFIX statistics from the traffic flows
- Security server
  Actions upon traffic analysis by communication to infrastructure routers

Phase-1: Traditional DDoS mitigation

Traffic under DDoS condition - RTBH

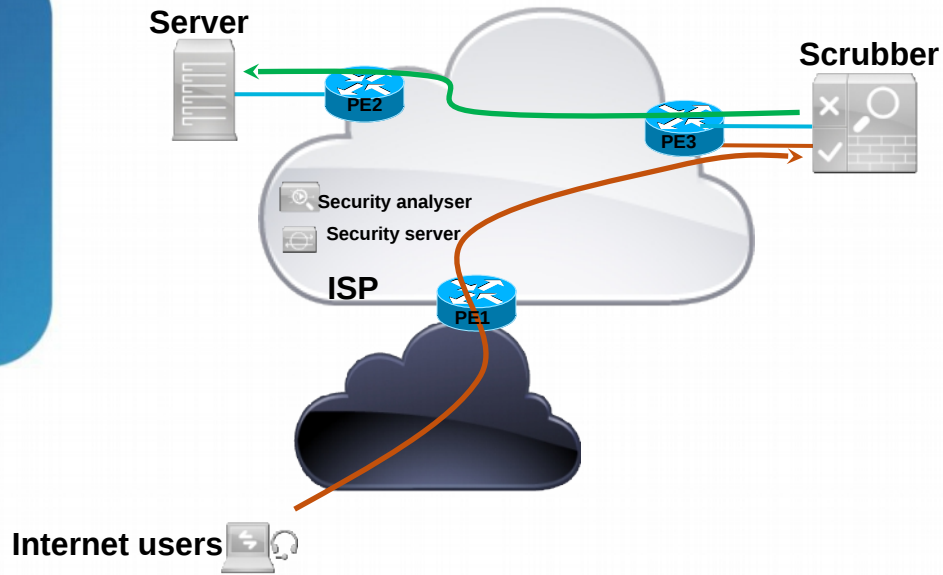


Traffic under DDoS condition

- Security analyser detected that the traffic flow is dirty
- Security server installs a filter upon ISP ingress router
- **All** (good and malicious) traffic is dropped at network ingress
- Operationally simple method
- Easy to remove filter if traffic normalizes
- Simple to debug and troubleshoot

Phase-2: BGP based DDoS

Traffic under DDoS condition



Traffic under DDoS condition

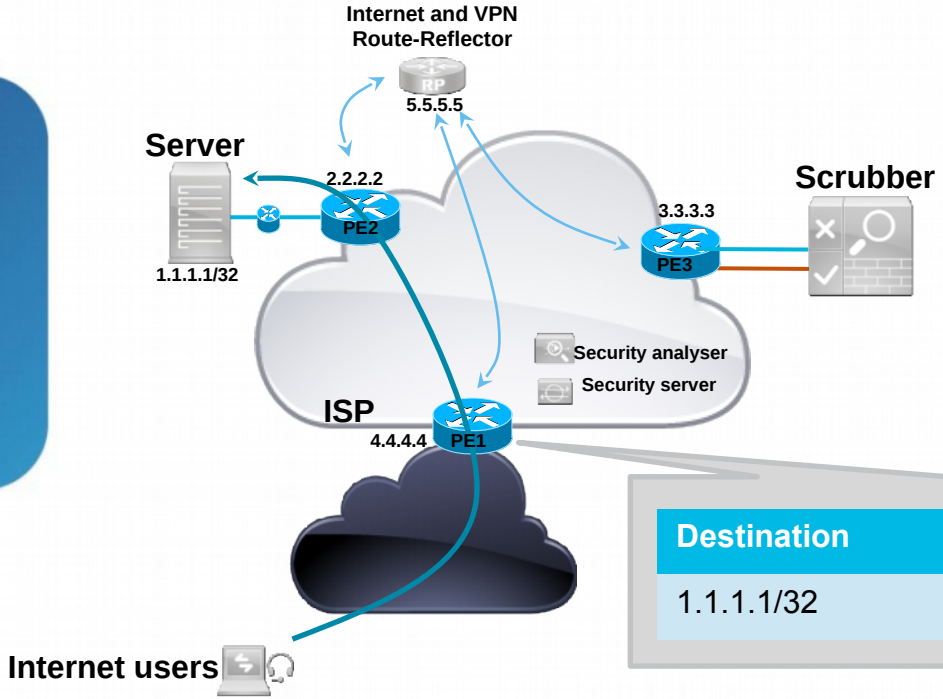
- Traffic is redirected to a scrubber
- Scrubber separates the clean from the malicious traffic
- Clean traffic is returned to original destination server

Goal

- **Do not drop all traffic**
- Collect traffic intelligence
- Operational simplicity
- Easy to remove redirect when traffic normalizes

How does it work?

Normal traffic condition

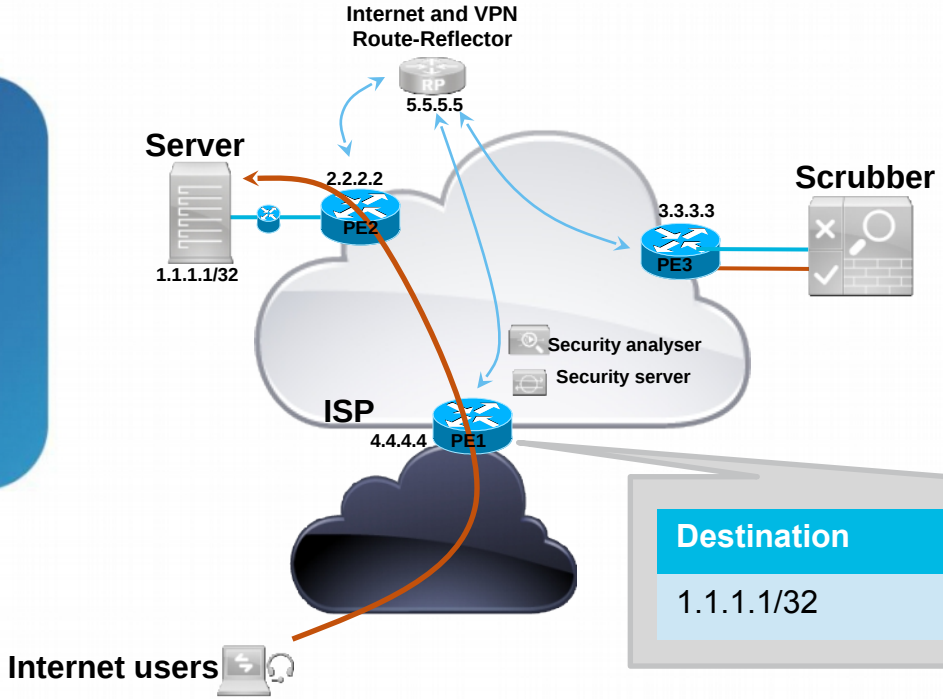


- All PE's peer with the RR
- All PE's exchange both Global Internet and VPN prefixes
- All PE interfaces are non-VPN
- Security analyser is performing doing analyses

Destination	Next-hop
1.1.1.1/32	2.2.2.2

How does it work?

Server is under DDoS

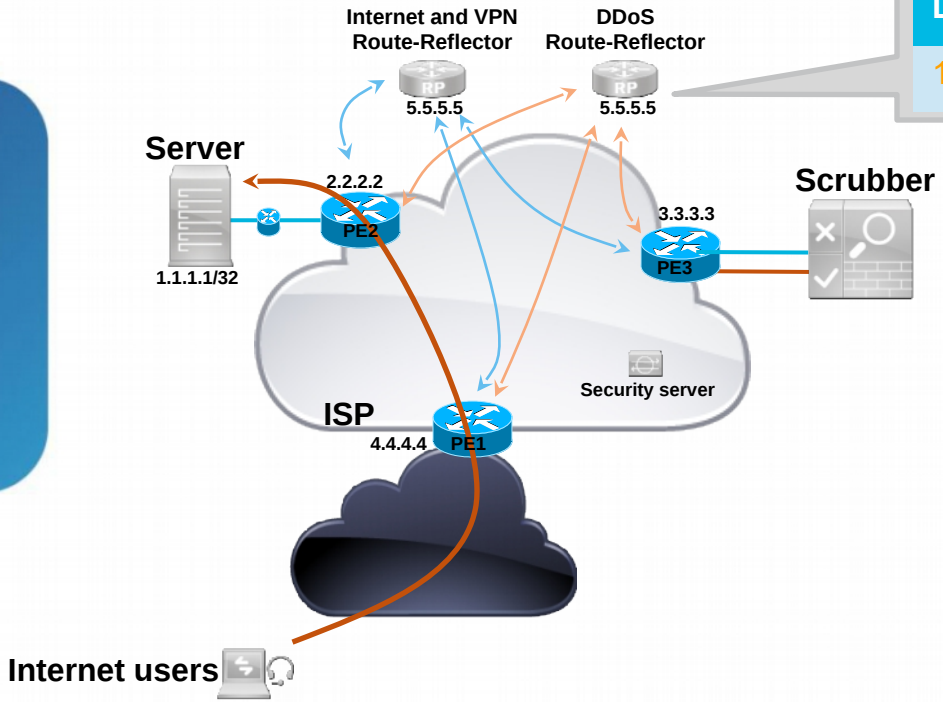


- **Flow is detected as dirty** by Security analyser
- **Result: Server is under attack**
- Traffic needs to be redirected to the scrubber to mitigate the attack

Destination	Next-hop
1.1.1.1/32	2.2.2.2

How does it work?

Server is under DDoS



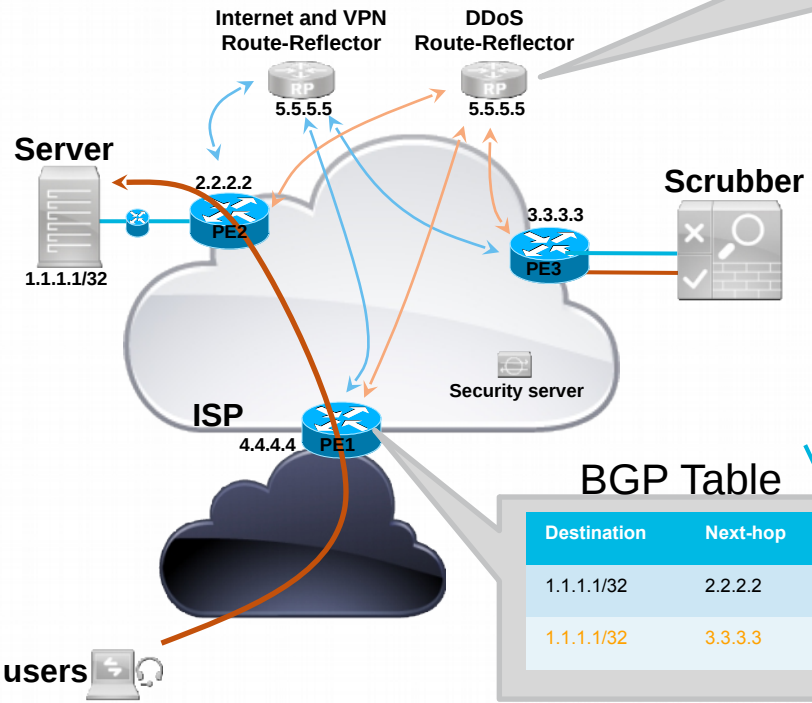
Destination	Next-hop
1.1.1.1/32	3.3.3.3

- DDoS Route-Reflector was pre-visioned
- Mitigation route to 1.1.1.1/32 is injected on the DDoS RR by the Security server
- Mitigation route to ***1.1.1.1/32*** is pointing to ***3.3.3.3*** on DDoS mitigation RR

How does it work?

Server is under DDoS

Destination	Next-hop
1.1.1.1/32	3.3.3.3



- Mitigation route to ***1.1.1.1/32*** is pointing to ***3.3.3.3*** is signalled to all PE's
- All PE's receive the mitigation route from the DDoS Mitigation RR
- Each PE will now have 2 routes to reach ***1.1.1.1/32***
- ***Which route will the PE use?***

BGP Table

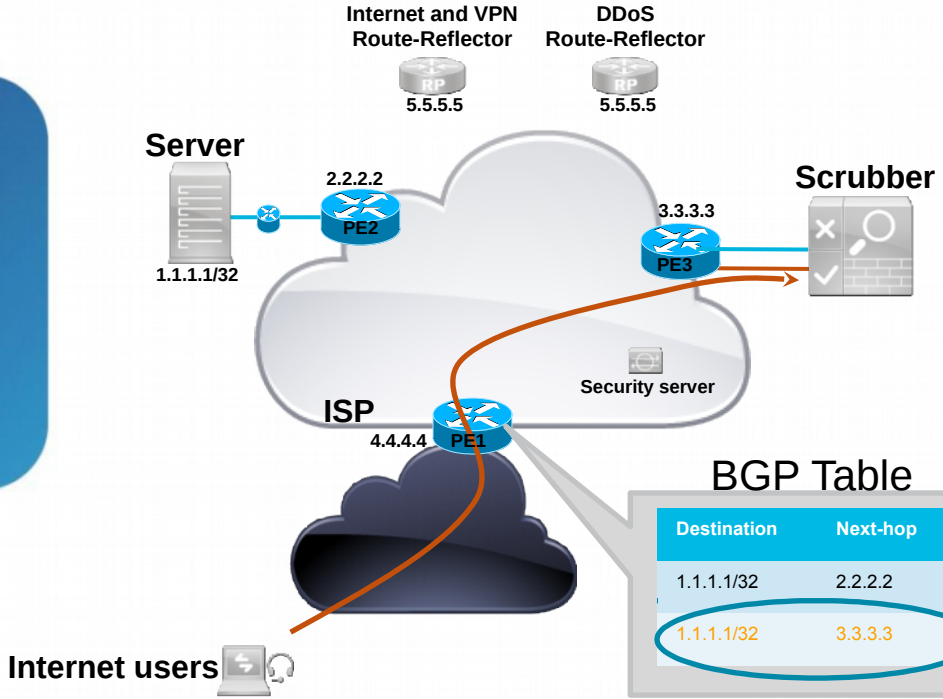
Destination	Next-hop
1.1.1.1/32	2.2.2.2
1.1.1.1/32	3.3.3.3

Routing Table

Destination	Next-hop
1.1.1.1/32	????????????

How does it work?

Server is under DDoS

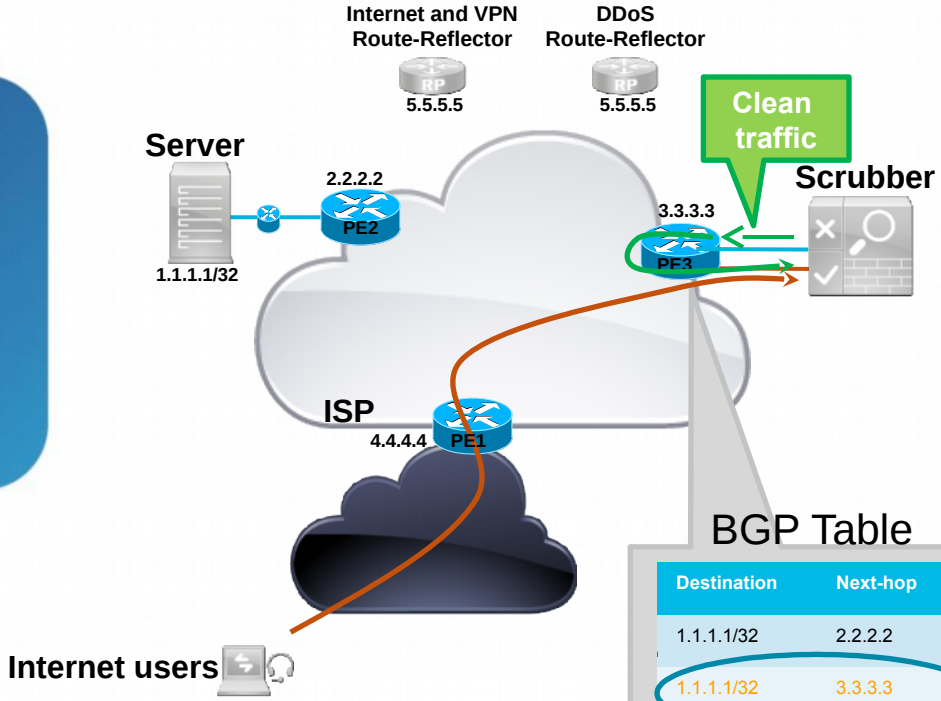


Trick # 1

- **The DDoS mitigation route will ALWAYS be preferred**, even if
  - Both prefix lengths are the same
  - DDoS prefix is shorter
  - Original prefix has better administrative distance

How does it work?

Server is under DDoS



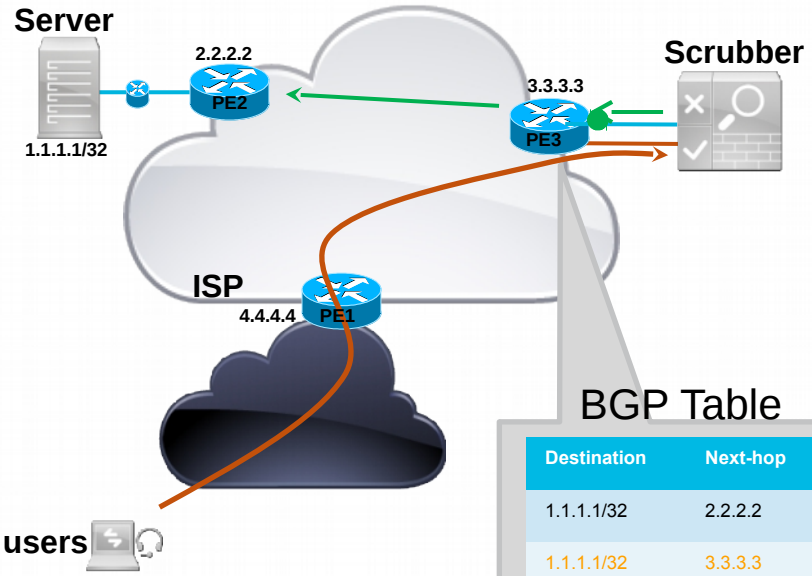
Problem

- Scrubber sends traffic to PE3
- PE3 does routing lookup for 1.1.1.1 and finds that it is directly attached
- ROUTING LOOP!!!
- How do we fix this?
  - Often this is fixed with true routing clutches:
  - Back-to-back cables
  - L2-VPN's
  - Manual tunnels
  - Etc… (all operational hard to maintain)
  - Better to use a new isolated dynamic routing table for the clean traffic
  - Pre-provision this Clean table

How does it work?

Server is under DDoS

- The clean traffic will be injected upon PE3 on an interface member of ***VPN Clean***
- PE3 will now do a routing destination lookup for 1.1.1.1 in VPN Clean
- The matching routing table entry is pointing towards PE2 at 2.2.2.2
- The clean flow, which is ***now part of VPN Clean*** is sent towards PE2 reachable at 2.2.2.2



BGP Table

Destination	Next-hop
1.1.1.1/32	2.2.2.2
1.1.1.1/32	3.3.3.3

Routing Table

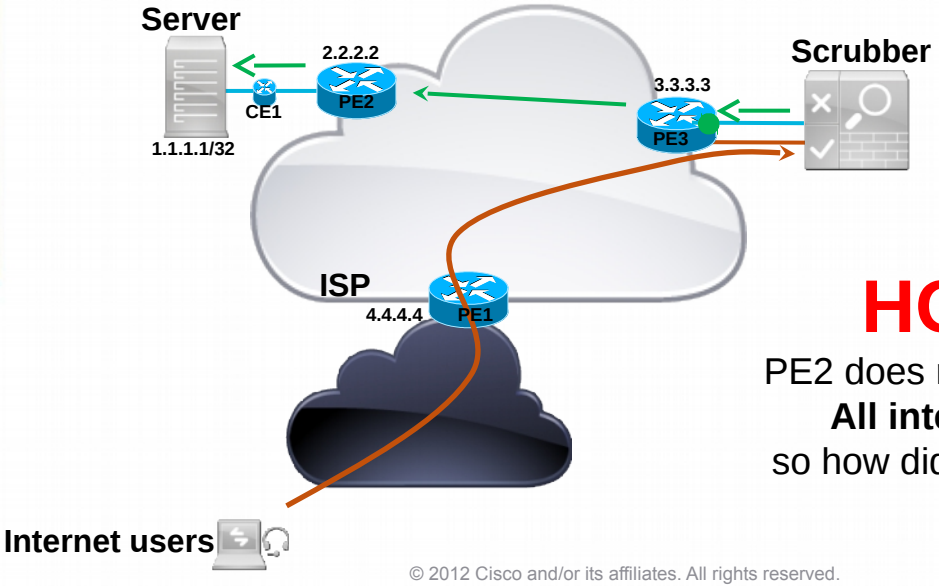
Destination	Next-hop	VPN
1.1.1.1/32	3.3.3.3	Global
1.1.1.1/32	2.2.2.2	Clean

How does it work?

Server is under DDoS

Routing Table

Destination	Next-hop	VPN
1.1.1.1/32	3.3.3.3	Global
1.1.1.1/32	CE1	Clean



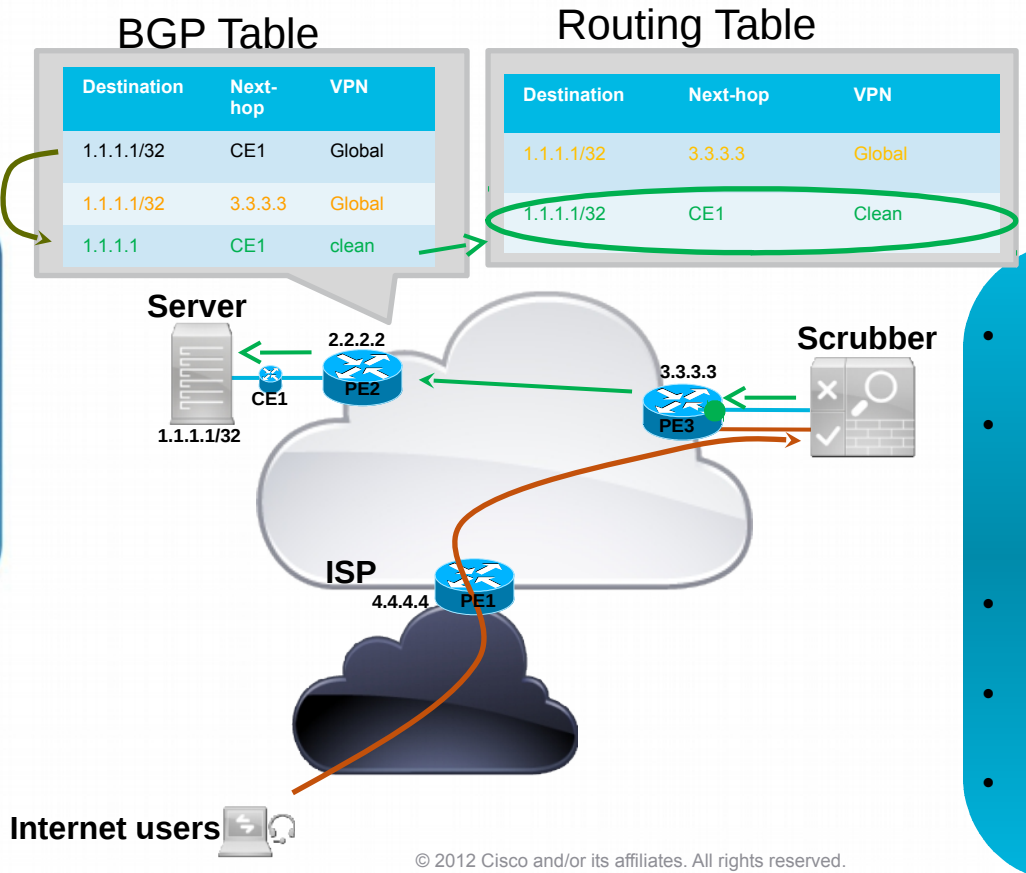
- PE2 receives the clean flow within ***VPN clean***
- PE2 does a destination address routing lookup in ***VPN clean***
- A matching route is found in ***VPN clean***
- Flow is forwarded towards CE1 onwards to Server

HOLD on a minute!

PE2 does **not** have **any interface** part of ***VPN clean***
All interfaces on PE2 are **global interfaces**
so how did that clean route for 1.1.1.1 get into VPN clean?

How does it work?

Server is under DDoS



Trick # 2

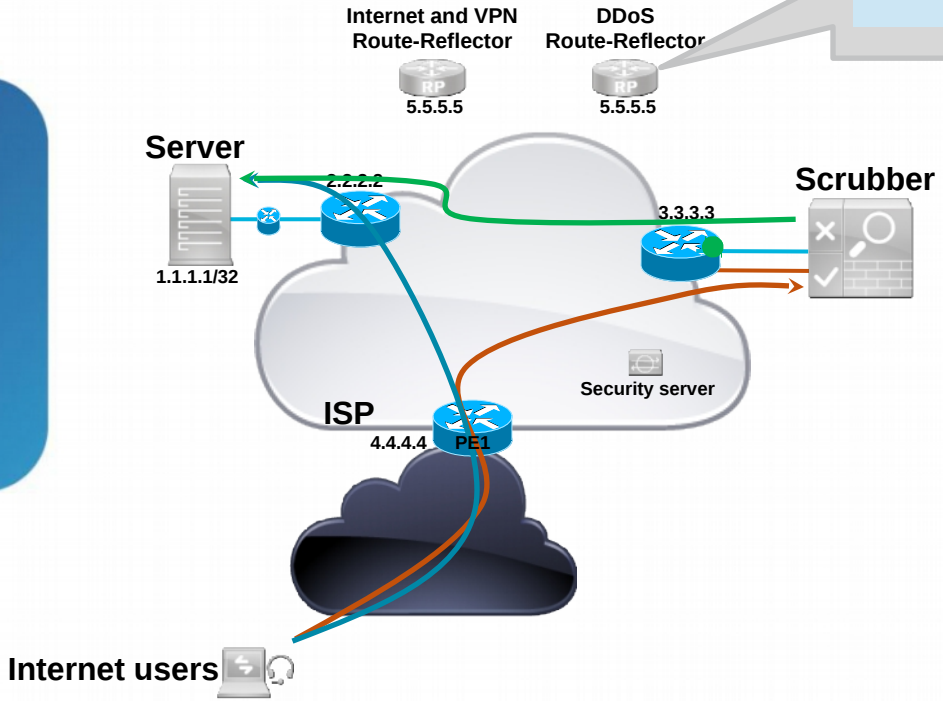
- Copy the locally BGP inserted route directly into VPN clean BGP table
- Neighbour details are inherited from the global table (i.e.)
  - Outgoing interface
  - Next-hop
- ***Interface pointing towards CE1 is NOT VPN aware***
- This VPN clean distributed as normal VPN
- New CLI command to do that

  import from default-vrf route-policy ddos advertise-as-vpn

Going back to traditional traffic flow

DDoS attack has ended

Destination	Next-hop
~~1.1.1.1/32~~	~~3.3.3.3~~

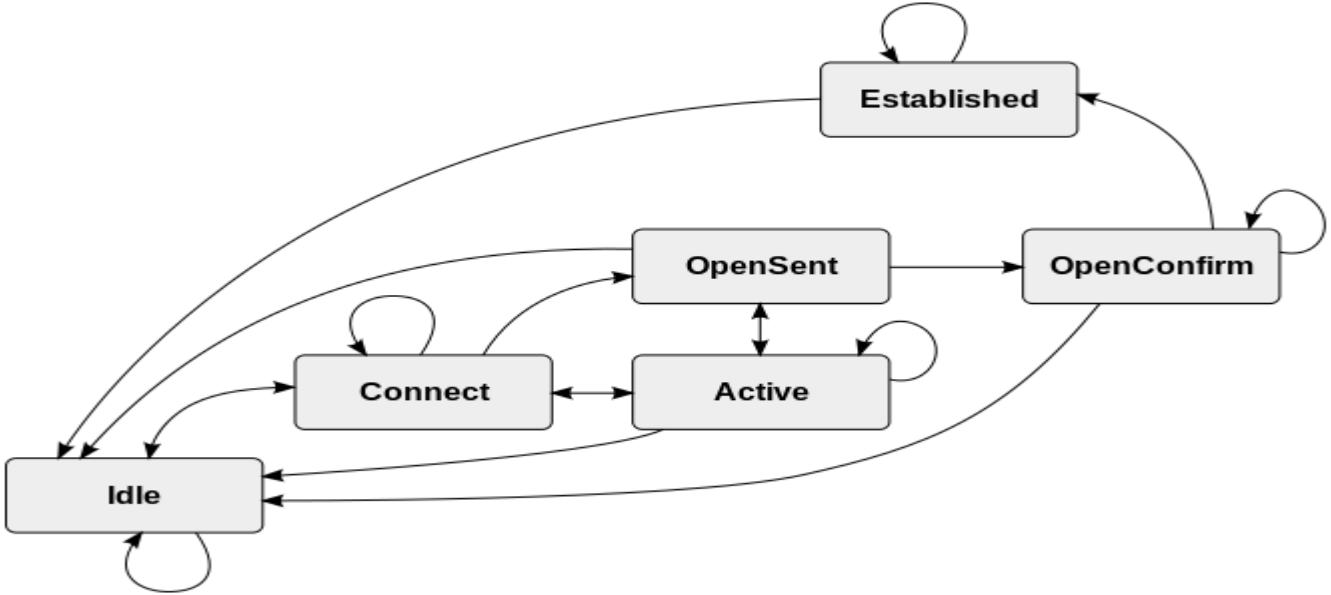


- Remove the routing entry on the Mitigation DDoS RR
- No more route is remaining on the DDoS Mitigation RR
- Traffic flows normally again

Why injecting DDoS in separate BGP instance ?

- Solution support redirection of BGP less/more specific prefixes or local originated prefixes (static route, redistributed route)
- Indepenant Inter-Domain control plane and DDoS plane
  - No need to withdraw and re-signal Inter-Domain prefixes, keep internet route intacts in control plane.
  - Easy to troubleshout

Any Questions?



BUILT FOR
THE HUMAN
NETWORK

Backup Slides

Technical details

Configuration (1)

```
router bgp 99 instance ddos
 bgp router-id 3.3.3.3
 bgp read-only
 bgp install diversion
 address-family ipv4 unicast
 !

router bgp 99
 bgp router-id 2.2.2.2
 address-family ipv4 unicast
 !
```

Creation of DDoS BGP instance

Allows config of 2th IPv4 or IPv6 instance
Suppresses BGP Update Generation

Triggers BGP ddos instance to install diversion path to RIB, so that the paths are pushed down to FIB

Configuration (2)

Importing the global route's in the clean VRF

```
vrf clean
address-family ipv4 unicast
  import from default-vrf route-policy ddos advertise-as-vpn
  export route-target
   111:1
  !
!
address-family ipv6 unicast
  import from default-vrf route-policy ddos advertise-as-vpn
  export route-target
   111:1
  !
!
!
```

“show” commands

```
RP/0/0/CPU0:hydra-prp-A# show route

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR
       A - access/subscriber, a - Application route, (!) - FRR Backup path

Gateway of last resort is not set

O    1.0.11.0/24 [110/2] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
O    1.1.1.1/32 [110/2] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
L    2.2.2.2/32 is directly connected, 00:37:24, Loopback0
O    3.3.3.3/32 [110/2] via 87.0.1.2, 00:36:19, GigabitEthernet0/2/1/9
O    4.4.4.4/32 [110/3] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
                [110/3] via 87.0.1.2, 00:36:19, GigabitEthernet0/2/1/9
B    5.5.5.5/32 [200/0] via 1.1.1.1, 00:34:22
             B > [200/0] via 123.0.0.2, 00:34:22
[...]
```

“show” commands (1)

```
RP/0/0/CPU0:hydra-prp-A#show route

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR
       A - access/subscriber, a - Application route, (!) - FRR Backup path

Gateway of last resort is not set

O    1.0.11.0/24 [110/2] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
O    1.1.1.1/32 [110/2] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
L    2.2.2.2/32 is directly connected, 00:37:24, Loopback0
O    3.3.3.3/32 [110/2] via 87.0.1.2, 00:36:19, GigabitEthernet0/2/1/9
O    4.4.4.4/32 [110/3] via 13.0.3.1, 00:36:19, GigabitEthernet0/2/1/5
                [110/3] via 87.0.1.2, 00:36:19, GigabitEthernet0/2/1/9
B    5.5.5.5/32 [200/0] via 1.1.1.1, 00:34:22
             B > [200/0] via 123.0.0.2, 00:34:22
[...]
```

“show” commands (2)

```
RP/0/0/CPU0:hydra-prp-A#show route 5.5.5.5/32

Routing entry for 5.5.5.5/32
  Known via "bgp 2394-ro", distance 200, metric 0, type internal
  Installed Feb 19 22:56:45.896 for 00:34:33
  Routing Descriptor Blocks
    1.1.1.1, from 1.1.1.1
      Route metric is 0
    123.0.0.2, from 101.0.0.4, Diversion Path (bgp)
      Route metric is 0
  No advertising protos.

RP/0/0/CPU0:hydra-prp-A#show cef 5.5.5.5/32 det
5.5.5.5/32, version 60652, internal 0x14000001 (ptr 0xaf6e3840) [1], 0x0 (0x0), 0x0 (0x0)
Updated Feb 19 22:56:46.723
 local adjacency 87.0.1.2
Prefix Len 32, traffic index 0, precedence n/a, priority 4
  gateway array (0xae07a310) reference count 2, flags 0x8020, source rib (5), 0 backups
                [1 type 3 flags 0xd0141 (0xae10f8c0) ext 0x420 (0xaec261e0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
   via 123.0.0.2, 2 dependencies, recursive [flags 0x6000]
    path-idx 0 [0xaf6e3c00 0x0]
    next hop 123.0.0.2 via 123.0.0.0/24


    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   GigabitEthernet0/2/1/9    87.0.1.2
```

“show” commands (3)

```
RP/0/0/CPU0:hydra-prp-A# show route 123.0.0.2

Routing entry for 123.0.0.0/24
  Known via "ospf 100", distance 110, metric 2, type intra area
  Installed Feb 19 22:54:48.363 for 00:39:01
  Routing Descriptor Blocks
    87.0.1.2, from 3.3.3.3, via GigabitEthernet0/2/1/9
      Route metric is 2
  No advertising protos.
RP/0/0/CPU0:hydra-prp-A#

RP/0/0/CPU0:hydra-prp-A#show route 1.1.1.1

Routing entry for 1.1.1.1/32
  Known via "ospf 100", distance 110, metric 2, type intra area
  Installed Feb 19 22:54:49.259 for 00:49:20
  Routing Descriptor Blocks
    13.0.3.1, from 1.1.1.1, via GigabitEthernet0/2/1/5
      Route metric is 2
  No advertising protos.
```