Unicast Reverse Path Forwarding

ISP Workshops

Last updated 16th September 2023

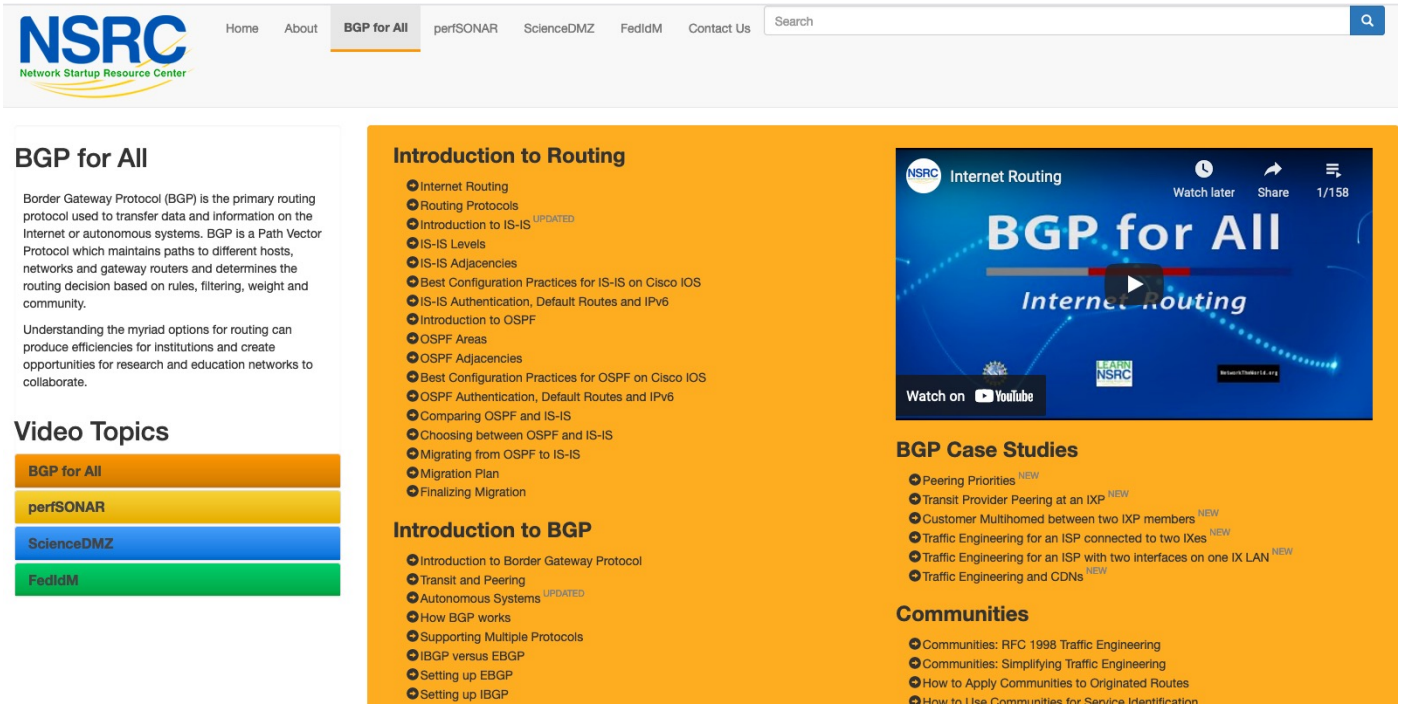
Acknowledgements

- This material originated from the Cisco ISP/IXP Workshop Programme developed by Philip Smith & Barry Greene
- Use of these materials is encouraged as long as the source is fully acknowledged and this notice remains in place
- Bug fixes and improvements are welcomed
  - Please email *workshop (at) bgp4all.com*

Philip Smith

BGP Videos

- NSRC has produced a library of BGP presentations (including this one), recorded on video, for the whole community to use
  - https://learn.nsrc.org/bgp



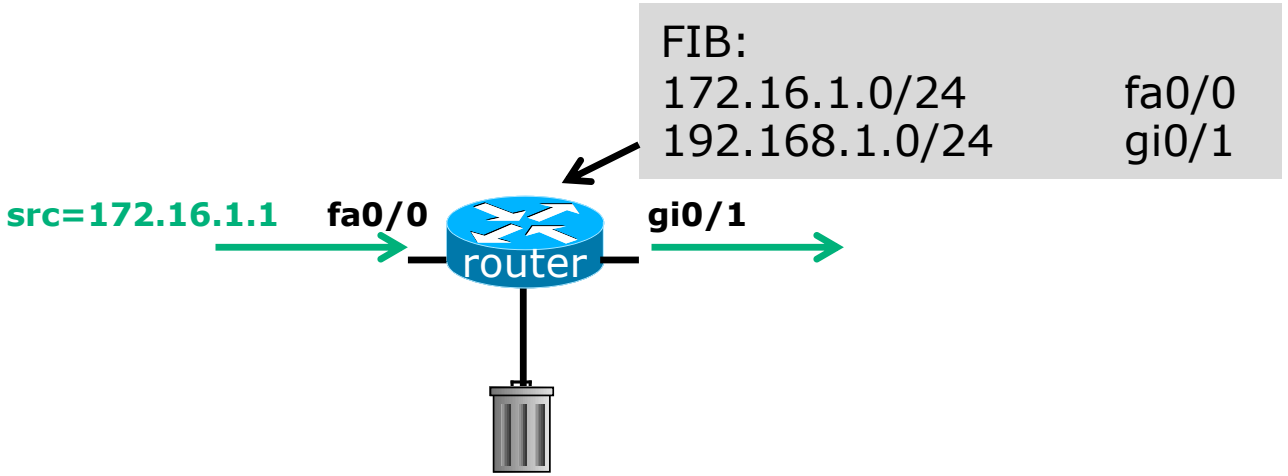
Unicast Reverse Path Forwarding

- uRPF is a technique where the router can discard packets with invalid/fake/incorrect source addresses by a simple check against the Forwarding Table (FIB)
  - More efficient than implementing ingress packet filters
- Part of BCP 38
  - https://tools.ietf.org/html/bcp38
- uRPF is a very effective tool to assist with defeating Denial of Service attacks, at source
  - Implemented by network operators on access devices, where end-users and end-devices connect to their network

uRPF

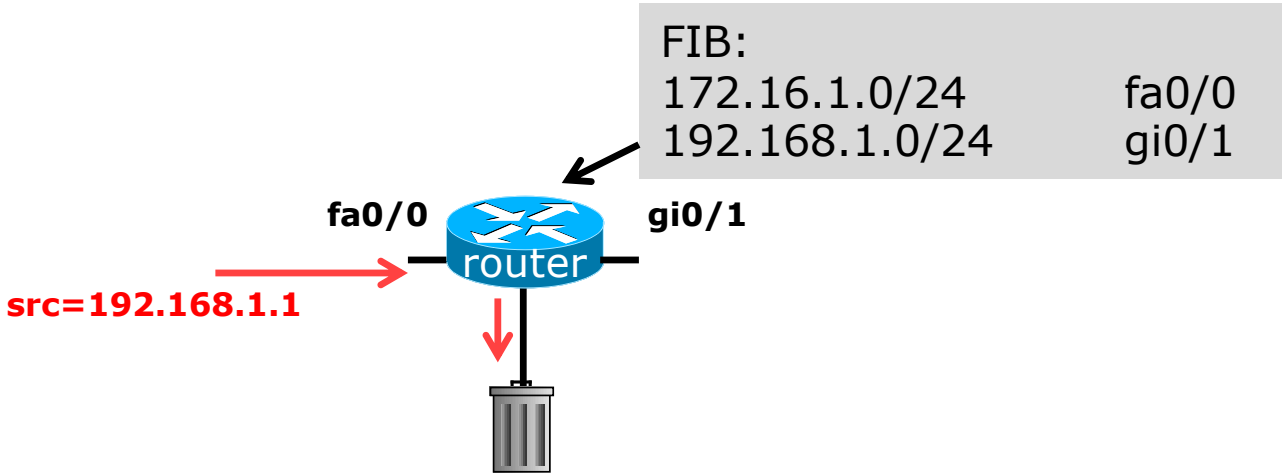
- There are two modes for uRPF:
  - Strict Mode
    - Source address must be reachable via the source (incoming) interface
    - Typically used in Access Networks
  - Loose Mode
    - Source address must be in the FIB
    - Typically used to drop non-routed address space
    - Used when asymmetric traffic flows are present (for example, when multihoming)
    - Used to implement source-based Remotely Triggered Blackhole Filtering (S/RTBH)

uRPF: Strict Mode



- Router compares source address of incoming packet with FIB entry
  - If FIB entry interface matches incoming interface, the packet is forwarded
  - If FIB entry interface does not match incoming interface, the packet is dropped

uRPF: Strict Mode



- Router compares source address of incoming packet with FIB entry
  - If FIB entry interface matches incoming interface, the packet is forwarded
  - If FIB entry interface does not match incoming interface, the packet is dropped

uRPF: IOS Configuration

- Configuring Strict Mode uRPF:

```
interface FastEthernet 0/1
 ip address 192.168.0.254 255.255.255.0
 ip verify unicast source reachable-via rx allow-self-ping
 ipv6 address 2001:DB8:0:1::FF/64
 ipv6 verify unicast source reachable-via rx
!
ip route 192.168.1.0 255.255.255.0 192.168.0.1
ipv6 route 2001:DB8:1:1::/64 2001:DB8:0:1::1
!
```

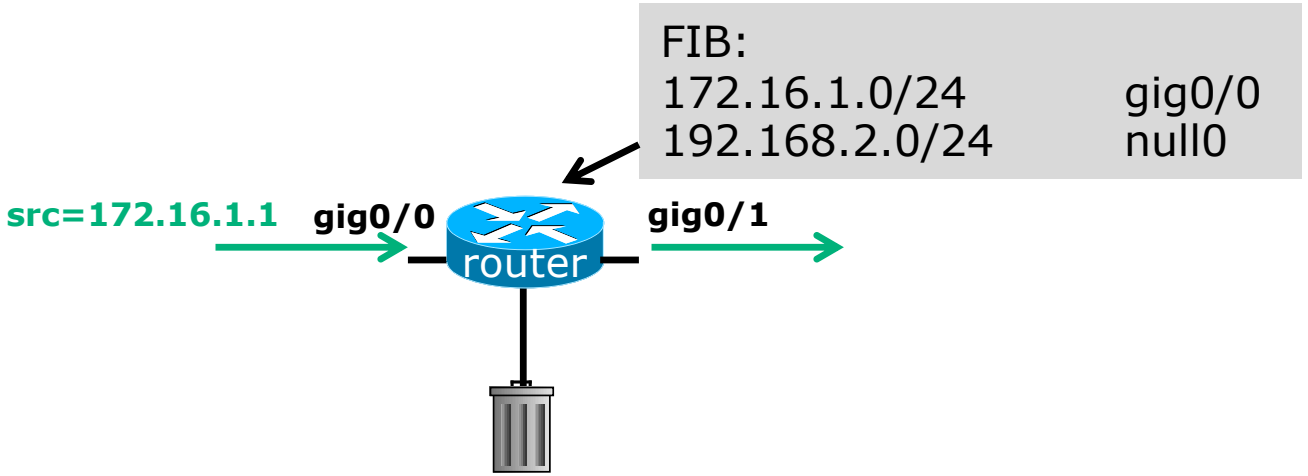
- This shows an ethernet LAN with uRPF configured
  - For IPv4 and IPv6
  - For both the direct LAN, *and*
  - For another network connected to the LAN

uRPF: IOS Configuration

- The router's IPv4 and IPv6 FIBs would look something like this:

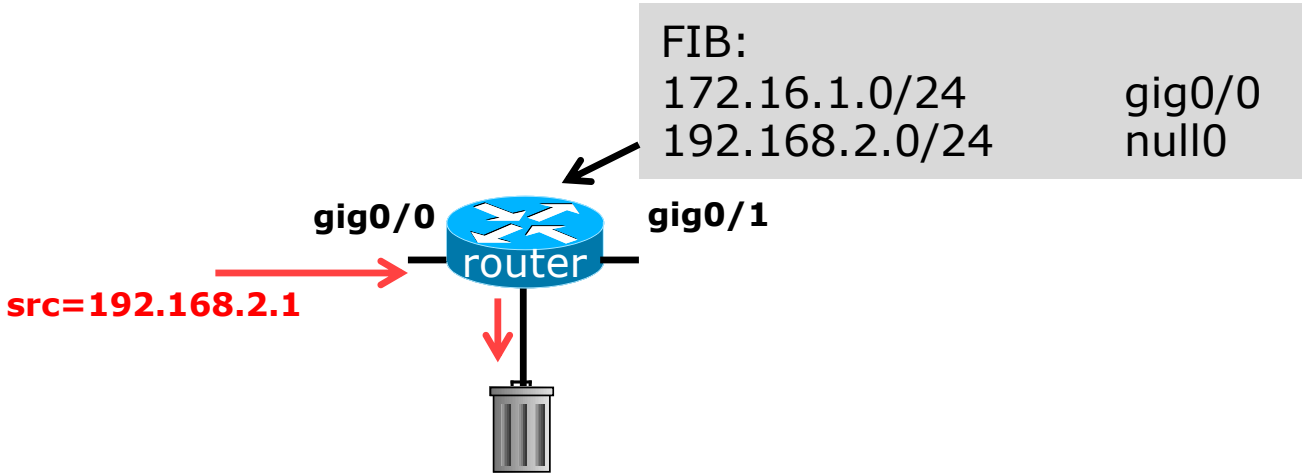
```
router# sh ip fib
...
192.168.0.0/24        attached              FastEthernet0/1
192.168.1.0/24        192.168.0.1           FastEthernet0/1
...
router# sh ipv6 fib
...
2001:DB8:0:1::/64
  attached to FastEthernet0/1
2001:DB8:1:1::/64
  nexthop FE80::6EB2:AEFF:FE6F:A508 FastEthernet0/1
...
```

uRPF: Loose Mode



- Router compares source address of incoming packet with FIB entry
  - If FIB entry exists and is a non-Null interface, the packet is forwarded
  - If FIB entry does NOT exist, or the interface is Null, the packet is dropped

uRPF: Loose Mode



- Router compares source address of incoming packet with FIB entry
  - If FIB entry exists and is a non-Null interface, the packet is forwarded
  - If FIB entry does NOT exist, or the interface is Null, the packet is dropped

uRPF: IOS Configuration

- Configuring Loose Mode uRPF on Cisco IOS:

```
interface FastEthernet 0/1
 ip address 192.168.0.254 255.255.255.0
 ip verify unicast source reachable-via any allow-self-ping
 ipv6 address 2001:DB8:0:1::FF/64
 ipv6 verify unicast source reachable-via any
!
ip route 192.168.1.0 255.255.255.0 192.168.0.1
ipv6 route 2001:DB8:1:1::/64 2001:DB8:0:1::1
!
```

  - The router will check the entire FIB for the destination

uRPF: IOS Configuration

- Cisco IOS allows various options:
  - reachable-via allows either
    - strict mode using the rx keyword *or*
    - loose mode using the any keyword
  - allow-self-ping enables the operator to use ping on the local interface to check local link connectivity
    - Without allow-self-ping it would not be possible to ping the local interface address from the router
  - In loose mode, the allow-default option allows a successful match against the default route
  - Access-lists can be used to cover selective uRPF checks

Deployment advice

- Implement uRPF on ***all*** single-homed customer facing interfaces
  - Cheaper (CPU & RAM) than implementing packet filters
- Make uRPF a default setting in all access router templates

- In the case of Multihomed connections, the deployment of strict uRPF needs very careful planning
  - Asymmetric traffic flows are common
  - Strict mode needs the BGP Weight feature (at minimum)
  - Loose mode ensures uRPF can be implemented

Summary

- uRPF has been available in major vendor implementations since the late 1990s
- More documentation contained in BCP38
  - https://tools.ietf.org/html/bcp38

- Implementation of uRPF is an essential technique for assisting with defeating Denial of Service attacks
- One of the principles in the MANRS initiative
  - https://www.manrs.org/manrs

Unicast Reverse Path Forwarding

ISP Workshops