

Understanding Policy Routing

Document ID: 10116

Introduction

Prerequisites

Requirements

Components Used

Conventions

Configurations

Network Diagram

Configuration for Firewall

Related Information

Introduction

Policy-based routing provides a tool for forwarding and routing data packets based on policies defined by network administrators. In effect, it is a way to have the policy override routing protocol decisions. Policy-based routing includes a mechanism for selectively applying policies based on access list, packet size or other criteria. The actions taken can include routing packets on user-defined routes, setting the precedence, type of service bits, etc.

In this document, a firewall is being used to translate 10.0.0.0/8 private addresses into Internet-routable addresses belonging to the subnet 172.16.255.0/24. See the diagram below for a visual explanation.

Refer to Policy-Based Routing for more information.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to any specific hardware or software versions.

The information shown in this document is based on the software and hardware versions below.

- Cisco IOS® Software Release 12.3(3)
- Cisco 2500 series routers

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

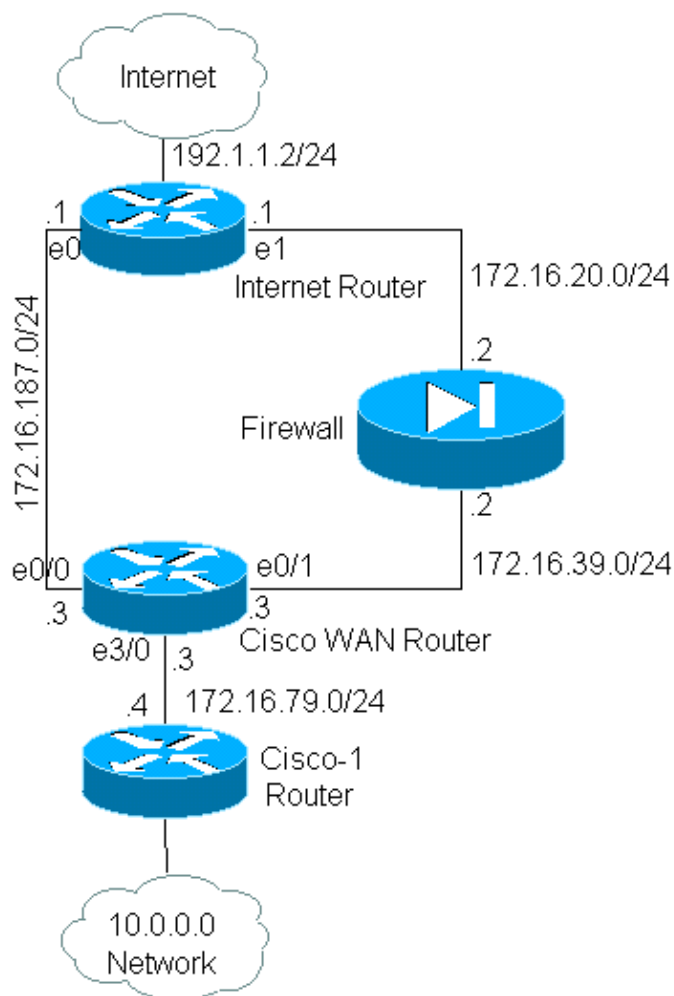
Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Configurations

In this example, with normal routing, all the packets from 10.0.0.0/8 network to the Internet will take the path through interface ethernet 0/0 of Cisco WAN Router (via 172.16.187.0/24 subnet) as it is the best path with least metric. With policy-based routing we want these packets to take the path through the Firewall to the Internet, normal routing behavior has to be overridden by configuring policy routing. The firewall translates all the packets from 10.0.0.0/8 network going to the Internet, which is however not necessary for policy routing to work.

Network Diagram



Configuration for Firewall

The firewall configuration below is included to provide a complete picture. However, it is not part of the policy routing issue explained in this document. The firewall in this example could easily be replaced by a PIX or another firewall device.

```
!  
ip nat pool net-10 172.16.255.1 172.16.255.254 prefix-length 24  
ip nat inside source list 1 pool net-10  
!  
interface Ethernet0  
ip address 172.16.20.2 255.255.255.0  
ip nat outside  
!
```

```

interface Ethernet1
 ip address 172.16.39.2 255.255.255.0
 ip nat inside
!
router eigrp 1
 redistribute static
 network 172.16.0.0
 default-metric 10000 100 255 1 1500
!
ip route 172.16.255.0 255.255.255.0 Null0
access-list 1 permit 10.0.0.0 0.255.255.255
!
end

```

Refer to IP Addressing and Services Commands for more information on **ip nat** related commands

In this example, the Cisco WAN Router is running policy routing to ensure that IP packets originating from the 10.0.0.0/8 network will be sent through the firewall. The configuration below contains an access list statement that sends packets originating from 10.0.0.0/8 network to the firewall.

Configuration for Cisco_WAN_Router

```

!
interface Ethernet0/0
 ip address 172.16.187.3 255.255.255.0
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 172.16.39.3 255.255.255.0
 no ip directed-broadcast
!
interface Ethernet3/0
 ip address 172.16.79.3 255.255.255.0
 no ip directed-broadcast
 ip policy route-map net-10
!
router eigrp 1
 network 172.16.0.0
!

access-list 111 permit ip 10.0.0.0 0.255.255.255 any
!
route-map net-10 permit 10
 match ip address 111
 set interface Ethernet0/1
!
route-map net-10 permit 20
!
end

```

Refer to the **route-map** command documentation for more information on **route-map** related commands.

Note: The **log** keyword in **access-list** command is not supported by PBR. If the **log** keyword configured, it does not show any hits.

Configuration for Cisco-1 Router

```

!
version 12.3

!

interface Ethernet0

```

```
!-- Interface connecting to 10.0.0.0 network
```

```
ip address 10.1.1.1 255.0.0.0
```

```
!  
interface Ethernet1
```

```
!-- Interface connecting to Cisco_Wan_Router
```

```
ip address 172.16.79.4 255.255.255.0
```

```
!  
router eigrp 1  
network 10.0.0.0  
network 172.16.0.0  
no auto-summary  
!
```

```
!---Output Suppressed
```

Configuration for Internet_Router

```
!  
version 12.3
```

```
!  
interface Ethernet1
```

```
!-- Interface connecting to Firewall
```

```
ip address 172.16.20.1 255.255.255.0
```

```
interface Serial0
```

```
!--- Interface connecting to Internet
```

```
ip address 192.1.1.2 255.255.255.0  
clockrate 64000  
no fair-queue  
!  
interface Ethernet0
```

```
!--- Interface connecting to Cisco_Wan_Router
```

```
ip address 172.16.187.1 255.255.255.0  
!
```

```
!  
router eigrp 1  
redistribute static
```

```
!--- Redistributing the static default route for other routers to reach Internet
```

```
network 172.16.0.0
no auto-summary
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.1.1.1
```

!-- Static default route pointing to the router connected to Internet

!---Output Suppressed

In testing this example, a ping sourced from 10.1.1.1 on the Cisco-1 Router, using the **extended ping** command, was sent to a host on the Internet. In this example, 192.1.1.1 was used as the destination address. To see what is happening on the Internet Router, fast switching was turned off while the **debug ip packet 101 detail** command was used.



Warning: Using the **debug ip packet detail** command on a production router can cause high CPU

utilization, which can result in a severe performance degradation or a network outage. We recommend that you carefully read the Using the Debug Command section of Understanding the Ping and Traceroute Commands before you use debug commands.

Note: The **access-list 101 permit icmp any any** statement is used to filter the **debug ip packet** output. Without this access list, the **debug ip packet** command can generate so much output to the console that the router locks up. Use extended ACLs when you configure PBR. If no ACL is configured in order to establish the match criteria, it results in all traffic being policy-routed.

```
Results of ping from Cisco_1 to 192.1.1.1/internet taken from Internet_Router:
Packet never makes it to Internet_Router
```

```
Cisco_1# ping
Protocol [ip]:
Target IP address: 192.1.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header?[no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
.....
Success rate is 0 percent (0/5)
```

As you can see, the packet never made it to the Internet Router. The debug commands below, taken from the Cisco WAN Router, show why this happened.

```
Debug commands run from Cisco_WAN_Router:
"debug ip policy"
*Mar  1 00:43:08.367: IP: s=10.1.1.1 (Ethernet3/0), d=192.1.1.1, len 100, policy match
*Mar  1 00:43:08.367: IP: route map net-10, item 10, permit
```

```
!--- Packet with source address belonging to 10.0.0.0/8 network
!--- is matched by route-map "net-10" statement 10.
```

```
*Mar 1 00:43:08.367: IP: s=10.1.1.1 (Ethernet3/0), d=192.1.1.1 (Ethernet0/1), len 100, po
*Mar 1 00:43:08.367: Ethernet3/0 to Ethernet0/1 192.1.1.1
```

```
!--- matched packets previously are forwarded out of interface
!--- ethernet 0/1 by the set command.
```

The packet matched policy entry 10 in the net-10 policy map, as expected. So why didn't the packet make it to the Internet Router?

```
"debug arp"
*Mar 1 00:06:09.619: IP ARP: creating incomplete entry for IP address: 192.1.1.1 interface
*Mar 1 00:06:09.619: IP ARP: sent req src 172.16.39.3 00b0.64cb.eabl,
dst 192.1.1.1 0000.0000.0000 Ethernet0/1
*Mar 1 00:06:09.635: IP ARP rep filtered src 192.1.1.1 0010.7b81.0b19, dst 172.16.39.3
00b0.64cb.eabl wrong cable, interface Ethernet0/1
```

```
Cisco_Wan_Router# show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	172.16.39.3	-	00b0.64cb.eabl	ARPA	Ethernet0/1
Internet	172.16.39.2	3	0010.7b81.0b19	ARPA	Ethernet0/1
Internet	192.1.1.1	0	Incomplete	ARPA	

The **debug arp** output shows this. The Cisco WAN router attempts to do what it was instructed and tries to put the packets directly onto the ethernet 0/1 interface. This requires that the router send an Address Resolution Protocol (ARP) request for the destination address of 192.1.1.1, which the router realizes is not on this interface, and hence the ARP entry for this address is "Incomplete," as seen by the **show arp** command. An encapsulation failure then occurs as the router is unable to put the packet on the wire with no ARP entry.

By specifying the firewall as the next-hop, we can prevent this problem and make the route-map work as intended:

```
Config changed on Cisco_WAN_Router:
!
route-map net-10 permit 10
 match ip address 111
 set ip next-hop 172.16.39.2
!
```

Using the same **debug ip packet 101 detail** command on the Internet Router, we now see that the packet is taking the correct path. We can also see that the packet has been translated to 172.16.255.1 by the firewall, and that the machine being pinged, 192.1.1.1, has replied:

```
Cisco_1# ping
Protocol [ip]:
Target IP address: 192.1.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/76 ms
```

```
Results of ping from Cisco_1 to 192.1.1.1/internet taken from Internet_Router:
Internet_Router#
```

```
*Mar 1 00:06:11.619: IP: s=172.16.255.1 (Ethernet1), d=192.1.1.1 (Serial0), g=192.1.1.1,
*Mar 1 00:06:11.619: ICMP type=8, code=0
```

```
!--- Packets sourced from 10.1.1.1 are getting translated to 172.16.255.1 by
!--- the Firewall before it reaches the Internet_Router.
```

```
*Mar 1 00:06:11.619:
*Mar 1 00:06:11.619: IP: s=192.1.1.1 (Serial0), d=172.16.255.1 (Ethernet1), g=172.16.20.2
*Mar 1 00:06:11.619: ICMP type=0, code=0
```

```
!--- Packets returning from Internet arrive with the destination
!--- address 172.16.255.1 before it reaches the Firewall.
```

```
*Mar 1 00:06:11.619:
```

The **debug ip policy** command on the Cisco WAN Router shows that the packet was forwarded to the firewall, 172.16.39.2:

Debug Commands Run From Cisco_WAN_Router

```
"debug ip policy"
*Mar 1 00:06:11.619: s=10.1.1.1 (Ethernet3/0), d=192.1.1.1, len 100, policy match
*Mar 1 00:06:11.619: IP: route map net-10, item 20, permit
*Mar 1 00:06:11.619: s=10.1.1.1 (Ethernet3/0), d=192.1.1.1 (Ethernet0/1), len 100, policy
*Mar 1 00:06:11.619: Ethernet3/0 to Ethernet0/1 172.16.39.2
```

Policy Based Routing for encrypted traffic

Forward the decrypted traffic to a loopback interface in order to route the encrypted traffic based on policy routing and then do PBR on that interface. If the encrypted traffic is passed over a VPN tunnel then disable `ip cef` on the interface, and terminate the vpn tunnel.

Related Information

- [IP Routing Support Page](#)
- [NAT Support Page](#)
- [Technical Support Tools and Resources](#)
- [Policy-Based Routing](#)
- [Cisco IOS Technologies](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)