Policy Based Routing

This chapter describes how to configure the Cisco ASA to support policy based routing (PBR). The following sections describe policy based routing, guidelines for PBR, and configuration for PBR.

- About Policy Based Routing, page 21-1
- Guidelines for Policy Based Routing, page 21-3
- Configure Policy Based Routing, page 21-4
- Examples for Policy Based Routing, page 21-6
- History for Policy Based Routing, page 21-13

About Policy Based Routing

Typical routing systems and protocols are based on routing the traffic based on the destination of the traffic. Destination based routing systems make it hard to change the routing behavior of specific traffic. With Policy Based Routing (PBR), you can define the routing behavior based on a number of different criteria other than destination network. These criteria include source or destination network, source or destination address, source or destination port, protocol, packet size, and packet classification among several others.

PBR has the ability to implement Quality of Service (QoS) by classifying and marking traffic at the network edge and then using PBR throughout the network to route marked traffic along a specific path. This permits routing of packets originating from different sources to different networks even when the destinations are the same and can be useful when interconnecting several private networks.

Why Use Policy Based Routing?

Consider a company that has two links between locations, one a high bandwidth, low delay expensive link and the other a low bandwidth, higher delay lower expense link. While using traditional routing protocols the higher bandwidth link would get most if not all of the traffic sent across it based on the metric savings obtained by the bandwidth and/or delay (using EIGRP or OSPF) characteristics of the link. PBR allows you to route higher priority traffic over the high bandwidth link and sends all other traffic over the low bandwidth link.

Some applications of policy based routing are listed below:

- Equal-Access and Source-Sensitive routing
- Quality of Service
- Cost Saving
- Load Sharing

**Equal-Access and Source-Sensitive Routing**

In this topology, traffic from HR network & Mgmt network can be configured to go through ISP1 and traffic from Eng network can be configured to go through ISP2. Thus, policy based routing enables the network administrators to provide equal-access and source-sensitive routing as shown below.

**Quality of Service**

By tagging packets with policy based routing, network administrators can classify the network traffic at the perimeter of the network for various classes of service and then implementing those classes of service in the core of the network using priority, custom or weighted fair queuing (as shown in the figure below). This setup improves network performance by eliminating the need to classify the traffic explicitly at each WAN interface in the core of backbone network.
Cost Saving

An organization can direct the bulk traffic associated with a specific activity to use a higher-bandwidth high-cost link for a short time and continues basic connectivity over a lower-bandwidth low-cost link for interactive traffic by defining the topology as shown below.

Load Sharing

In addition to the dynamic load-sharing capabilities offered by ECMP load balancing, network administrators can now implement policies to distribute traffic among multiple paths based on the traffic characteristics.

As an example, in the topology depicted in the Equal-Access Source Sensitive Routing scenario, an administrator can configure policy based routing to load share the traffic from HR network through ISP1 and traffic from Eng network through ISP2.

Implementation of PBR

The ASA uses ACLs to match traffic and then perform routing actions on the traffic. Specifically, you configure a route map that specifies an ACL for matching, and then you specify one or more actions for that traffic. Finally, you associate the route map with an interface where you want to apply PBR on all incoming traffic.

Guidelines for Policy Based Routing

Firewall Mode

Supported only in routed firewall mode. Transparent firewall mode is not supported.

Clustering

- Clustering is supported.
- In a cluster scenario, without static or dynamic routes, with ip-verify-reverse path enabled, asymmetric traffic may get dropped. So it is recommended to disable ip-verify-reverse path.
Additional Guidelines

- All existing route map related configuration restrictions and limitations will be carried forward.

Configure Policy Based Routing

A route map is comprised of one or more route-map statements. Each statement has a sequence number, as well as a permit or deny clause. Each route-map statement contains match and set commands. The match command denotes the match criteria to be applied on the packet. The set command denotes the action to be taken on the packet.

- When a route map is configured with both IPv4 and IPv6 match/set clauses or when a unified ACL matching IPv4 and IPv6 traffic is used, the set actions will be applied based on destination IP version.
- When multiple next-hops or interfaces are configured as set action, all these options are evaluated one after the other until a valid usable option is found. No load balancing will be done among the configured multiple options.
- When a route map with match ACLs containing IPv6 addresses is attached to an interface, all IPv6 related ACLs will be discarded with a warning.
- Verify-availability option uses track object configuration to track and verify the status of configured next-hop. Since track object configuration is not supported in multi-mode, we will not support verify-availability option too.

Procedure

Step 1 Define an access-list:

```plaintext
access-list access_list_name {permit | deny} ip {any4|ip_address...}
```

Example:
```
ciscoasa(config)# access-list testacl permit ip 10.1.1.0 255.255.255.0 20.1.1.0 255.255.255.0
```

Step 2 Create a route map entry:

```plaintext
route-map name {permit | deny} [sequence_number]
```

Example:
```
ciscoasa(config)# route-map testmap (permit) [12]
```

Route map entries are read in order. You can identify the order using the `sequence_number` argument, or the ASA uses the order in which you add route map entries.

Note When a route-map is configured without a permit or deny action and without a sequence-number, it by default will assume the action as permit and sequence-number as 10

Step 3 Define the match criteria to be applied using an access-list:

```plaintext
match ip-address <access-list name> [access-list name]
```

Example:
```
ciscoasa(config-route-map)# match ip-address testacl
```
**Step 4** Indicate where to output packets that pass a match clause of a route map for policy routing:

```
set ip default next-hop [ip_address/ ipv6_address]
```

Example:
```
ciscoasa(config-route-map)# set ip default next-hop 10.86.118.3
```

**Step 5** Configure one or more actions:

- Set a recursive next hop IP address:
  
  ```
  set ip next-hop recursive ip_address
  ```

  Example:
  ```
ciscoasa(config-route-map)# set ip next-hop recursive 10.86.118.1
  ```

- Verify if the next hops of a route map are Cisco Discovery Protocol (CDP) neighbors before policy routing to those next hops:
  
  ```
  set ip next-hop verify-availability next-hop-address sequence number track object
  ```

  Example:
  ```
ciscoasa(config-route-map)# set ip next-hop verify-availability 10.86.118.3 56 track 10
  ```

- Set the output interface for the packet:
  
  ```
  set interface interface_name
  ```
  or
  
  ```
  set interface null0
  ```

  Example:
  ```
ciscoasa(config-route-map)# set interface GigabitEthernet0/0
  ```

- Set the Don't Fragment (DF) bit value in the IP header:
  
  ```
  set ip df {0|1}
  ```

  Example:
  ```
ciscoasa(config-route-map)# set ip df 0
  ```

- Classify IP traffic by setting a Differentiated Services Code Point (DSCP) or an IP-precedence value in the packet:
  
  ```
  set ip dscp new-dscp
  ```

  Example:
  ```
ciscoasa(config-route-map)# set ip dscp af33
  ```

**Note**

When multiple set actions are configured, PBR will evaluate them in the following order:

```
set ip next-hop verify-availability
set ip next-hop
set ip next-hop recursive
set interface
set ip default next-hop
set default interface
```
Step 6  Configure an interface and enter interface configuration mode:

```
interface physical_interface
```

Example:
```
ciscoasa(config)# interface GigabitEthernet0/0
```

Step 7  Configure policy based routing for through-the-box traffic:
```
policy-route route-map route-map_name
```

Example:
```
ciscoasa(config-if)# policy-route route-map testmap
```

---

### Examples for Policy Based Routing

#### Examples for Route Map Configuration

In the following example, since no action and sequence is specified, an implicit action of permit and a sequence number of 10 is assumed:
```
ciscoasa(config)# route-map testmap
```

In the following example, since no match criteria is specified, an implicit match 'any' is assumed:
```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10
```

In this example, all traffic matching <acl> will be policy routed and forwarded through outside interface.
```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl>
ciscoasa(config-route-map)# set interface outside
```

In this example, since there are no interface or next-hop actions are configured, all traffic matching <acl> will have df bit and dscp fields modified as per configuration and are forwarding using normal routing
```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl>
sset ip df 1
set ip precedence af11
```

In the following example, all traffic matching <acl_1> is forwarded using next-hop 1.1.1.10, all traffic matching <acl_2> is forwarded using next-hop 2.1.1.10 and rest of the traffic is dropped. No "match" criteria implies an implicit match "any".
```
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address <acl_1>
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10
```
```
ciscoasa(config)# route-map testmap permit 20
ciscoasa(config-route-map)# match ip address <acl_2>
```
In the following example, the route-map evaluation will be such that (i) a route-map action permit and acl action permit will apply the set actions (ii) a route-map action deny and acl action permit will skip to normal route lookup (iii) a route-map action of permit/deny and acl action deny will continue with next route-map entry. When no next route-map entry available, we will fallback to normal route lookup.

```plaintext
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address permit_acl_1 deny_acl_2
ciscoasa(config-route-map)# set ip next-hop 1.1.1.10

ciscoasa(config)# route-map testmap deny 20
ciscoasa(config-route-map)# match ip address permit_acl_3 deny_acl_4
ciscoasa(config-route-map)# set ip next-hop 2.1.1.10

ciscoasa(config)# route-map testmap permit 30
ciscoasa(config-route-map)# match ip address deny_acl_5

ciscoasa(config-route-map)# set interface outside
```

In the following example, when multiple set actions are configured, they are evaluated in the order mentioned above. Only when all options of a set action are evaluated and cannot be applied, the next set actions will be considered. This ordering will ensure that the most available and least distant next-hop will be tried first followed by next most available and least distant next-hop and so on.

```plaintext
ciscoasa(config)# route-map testmap permit 10
ciscoasa(config-route-map)# match ip address acl_1

ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.10 1 track 1

ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.11 2 track 2

ciscoasa(config-route-map)# set ip next-hop verify-availability 1.1.1.12 3 track 3

ciscoasa(config-route-map)# set ip next-hop 2.1.1.10 2.1.1.11 2.1.1.12

ciscoasa(config-route-map)# set ip next-hop recursive 3.1.1.10

ciscoasa(config-route-map)# set interface outside-1 outside-2

ciscoasa(config-route-map)# set ip default next-hop 4.1.1.10 4.1.1.11

ciscoasa(config-route-map)# set default interface Null10
```
Example Configuration for PBR

This section describes the complete set of configuration required to configure PBR for the following scenario:

First, we need to configure interfaces.

```
(config)# interface GigabitEthernet0/0
(config-if)# no shutdown
(config-if)# nameif inside
(config-if)# ip address 10.1.1.1 255.255.255.0

(config)# interface GigabitEthernet0/1
(config-if)# no shutdown
(config-if)# nameif outside-1
(config-if)# ip address 192.168.6.5 255.255.255.0

(config)# interface GigabitEthernet0/2
(config-if)# no shutdown
(config-if)# nameif outside-2
(config-if)# ip address 172.16.7.6 255.255.255.0
```

Then, we need to configure an access-list for matching the traffic.

```
(config)# access-list acl-1 permit ip 10.1.0.0 255.255.0.0
(config)# access-list acl-2 permit ip 10.2.0.0 255.255.0.0
```

We need to configure a route-map by specifying the above access-list as match criteria along with the required set actions.

```
(config)# route-map equal-access permit 10
(config-route-map)# match ip address acl-1
(config-route-map)# set ip next-hop 192.168.6.6

(config)# route-map equal-access permit 20
(config-route-map)# match ip address acl-2
(config-route-map)# set ip next-hop 172.16.7.7

(config)# route-map equal-access permit 30
```
(config-route-map)# set ip interface Null0

Now, this route-map has to be attached to an interface.

(config)# interface GigabitEthernet0/0
(config-if)# policy-route route-map equal-access

To display the policy routing configuration.

(config)# show policy-route
Interface Route map
GigabitEthernet0/0 equal-access

Policy Based Routing in Action

We will use the above test setup to configure policy based routing with different match criteria & set actions and see how they get evaluated and applied.

First, we will start with the basic configuration for all the devices involved in the setup. Here, A, B, C & D are assumed as ASA devices and H1, H2 as IOS routers.

ASA-A:

ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 15.1.1.60 255.255.255.0
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# no shut
ciscoasa(config)# interface GigabitEthernet0/1.1
ciscoasa(config-if)# vlan 391
ciscoasa(config-if)# nameif outside
  ciscoasa(config-if)# security-level 0
  ciscoasa(config-if)# ip address 25.1.1.60 255.255.255.0

ASA-B:

ciscoasa(config)# interface GigabitEthernet0/0
  ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/0.1
  ciscoasa(config-if)# vlan 291
  ciscoasa(config-if)# nameif outside
  ciscoasa(config-if)# security-level 0
  ciscoasa(config-if)# ip address 45.1.1.61 255.255.255.0

ASA-C:

ciscoasa(config)# interface GigabitEthernet0/0
  ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/0.2
  ciscoasa(config-if)# vlan 292
  ciscoasa(config-if)# nameif outside
  ciscoasa(config-if)# security-level 0
  ciscoasa(config-if)# ip address 55.1.1.61 255.255.255.0

ASA-D:

ciscoasa(config)# interface GigabitEthernet0/0
  ciscoasa(config-if)# no shut

ciscoasa(config) #interface GigabitEthernet0/0.1
  ciscoasa(config-if)# vlan 291
Examples for Policy Based Routing

ciscoasa(config-if)# nameif inside-1
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 45.1.1.62 255.255.255.0
ciscoasa(config)# interface GigabitEthernet0/0.2
ciscoasa(config-if)# vlan 292
ciscoasa(config-if)# nameif inside-2
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 55.1.1.61 255.255.255.0
ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 65.1.1.60 255.255.255.0

H1:

ciscoasa(config)# interface Loopback1
ciscoasa(config-if)# ip address 15.1.1.100 255.255.255.255

ciscoasa(config-if)# interface Loopback2
ciscoasa(config-if)# ip address 15.1.1.101 255.255.255.255

ciscoasa(config)# ip route 0.0.0.0 0.0.0.0 15.1.1.60

H2:

ciscoasa(config)# interface GigabitEthernet0/1
ciscoasa(config-if)# ip address 65.1.1.100 255.255.255.0
ciscoasa(config-if)# ip route 15.1.1.0 255.255.255.0 65.1.1.60

We will configure PBR on ASA-A to route traffic sourced from H1.

ASA-A:

ciscoasa(config-if)# access-list pbracl_1 extended permit ip host 15.1.1.100 any
ciscoasa(config-if)# route-map testmap permit 10
ciscoasa(config-if)# match ip address pbracl_1
ciscoasa(config-if)# set ip next-hop 25.1.1.61
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# policy-route route-map testmap
ciscoasa(config-if)# debug policy-route

H1: ping 65.1.1.100 repeat 1 source loopback1

pbr: policy based route lookup called for 15.1.1.100/44397 to 65.1.1.100/0 proto 1
sub_proto 8 received on interface inside
pbr: First matching rule from ACL(2)
pbr: route map testmap, sequence 10, permit; proceed with policy routing
pbr: evaluating next-hop 25.1.1.61
pbr: policy based routing applied; egress_ifc = outside : next_hop = 25.1.1.61
The packet is forwarded as expected using the next-hop address in the route-map.

When a next-hop is configured, we do a lookup in input route table to identify a connected route to the configured next-hop and use the corresponding interface. The input route table for this example is shown below (with the matching route entry highlighted).

```
in   255.255.255.255 255.255.255.255 identity  
in   15.1.1.60       255.255.255.255 identity 
in   25.1.1.60       255.255.255.255 identity 
in   35.1.1.60       255.255.255.255 identity 
in   10.127.46.17    255.255.255.255 identity 
in   15.1.1.0        255.255.255.0   inside  
in   25.1.1.0        255.255.255.0   outside 
in   35.1.1.0        255.255.255.0   dmz    
```

Next let's configure ASA-A to route packets from H1 loopback2 out of ASA-A dmz interface.

```
ciscoasa(config)# access-list pbracl_2 extended permit ip host 15.1.1.101 any 
ciscoasa(config)# route-map testmap permit 20 
ciscoasa(config-route-map)# match ip address pbracl 
ciscoasa(config-route-map)# set ip next-hop 35.1.1.61 
ciscoasa(config)# show run route-map 
! 
route-map testmap permit 10 
  match ip address pbracl_1 
  set ip next-hop 25.1.1.61 
! 
route-map testmap permit 20 
  match ip address pbracl 
  set ip next-hop 35.1.1.61 
! 
```

**H1: ping 65.1.1.100 repeat 1 source loopback2**

The debugs are shown below:

```
pbr: policy based route lookup called for 15.1.1.101/1234 to 65.1.1.100/1234 proto 6 
sub_proto 0 received on interface inside 
pbr: First matching rule from ACL(3) 
pbr: route map testmap, sequence 20, permit; proceed with policy routing 
pbr: evaluating next-hop 35.1.1.61 
pbr: policy based routing applied; egress_ifc = dmz : next_hop = 35.1.1.61 
```

and, the route entry chosen from input route table is shown below:

```
in   255.255.255.255 255.255.255.255 identity  
in   15.1.1.60       255.255.255.255 identity 
in   25.1.1.60       255.255.255.255 identity 
in   35.1.1.60       255.255.255.255 identity 
in   10.127.46.17    255.255.255.255 identity 
in   15.1.1.0        255.255.255.0   inside  
in   25.1.1.0        255.255.255.0   outside 
```
## History for Policy Based Routing

### Table 21-1 History for Route Maps

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy based routing</td>
<td>9.4(1)</td>
<td>Policy Based Routing (PBR) is a mechanism by which traffic is routed through specific paths with a specified QoS using ACLs. ACLs let traffic be classified based on the content of the packet’s Layer 3 and Layer 4 headers. This solution lets administrators provide QoS to differentiated traffic, distribute interactive and batch traffic among low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths, and allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections. We introduced the following commands: <code>set ip next-hop verify-availability</code>, <code>set ip next-hop</code>, <code>set ip next-hop recursive</code>, <code>set interface</code>, <code>set ip default next-hop</code>, <code>set default interface</code>, <code>set ip df</code>, <code>set ip dscp</code>, <code>policy-route route-map</code>, <code>show policy-route</code>, <code>debug policy-route</code></td>
</tr>
<tr>
<td>IPv6 support for Prefix Rule</td>
<td>9.3.2</td>
<td>We introduced this feature.</td>
</tr>
<tr>
<td>Policy Based Routing for Route Maps</td>
<td>9.4.1</td>
<td>Policy Based Routing (PBR) is a mechanism by which traffic is routed through specific paths with a specified QoS using ACLs. ACLs let traffic be classified based on the content of the packet’s Layer 3 and Layer 4 headers. This solution lets administrators provide QoS to differentiated traffic, distribute interactive and batch traffic among low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths, and allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections. We introduced the following commands: <code>set ip next-hop verify-availability</code>, <code>set ip next-hop</code>, <code>set ip next-hop recursive</code>, <code>set interface</code>, <code>set ip default next-hop</code>, <code>set default interface</code>, <code>set ip df</code>, <code>set ip dscp</code>, <code>policy-route route-map</code>, <code>show policy-route</code>, <code>debug policy-route</code></td>
</tr>
</tbody>
</table>