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

About Policy Based Routing

Traditional routing is destination-based, meaning packets are routed based on destination IP address. However, it is difficult to change the routing of specific traffic in a destination-based routing system. With Policy Based Routing (PBR), you can define routing based on criteria other than destination network—PBR lets you route traffic based on source address, source port, destination address, destination port, protocol, or a combination of these.

Policy Based Routing:

- Lets you provide Quality of Service (QoS) to differentiated traffic.
- Lets you distribute interactive and batch traffic across low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost switched paths.
- Allows Internet service providers and other organizations to route traffic originating from various sets of users through well-defined Internet connections.

Policy Based Routing can implement 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 it 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, less-expensive 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/low-delay link, while sending all other traffic over the low-bandwidth/high-delay link.

Some applications of policy based routing are:

**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 here.

**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 show here.
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.

Per-flow Routing

Since the ASA performs routing on a per-flow basis, policy routing is applied on the first packet and the resulting routing decision is stored in the flow created for the packet. All subsequent packets belonging to the same connection simply match this flow and are routed appropriately.

PBR Policies Not Applied for Output Route Look-up

Policy Based Routing is an ingress-only feature; that is, it is applied only to the first packet of a new incoming connection, at which time the egress interface for the forward leg of the connection is selected. Note that PBR will not be triggered if the incoming packet belongs to an existing connection, or if NAT is applied.

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 disabling ip-verify-reverse path is recommended.
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 multiple next-hops or interfaces are configured as a set action, all 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.
- The verify-availability option is not supported in multiple context mode.

Procedure

Step 1 Define a standard or extended access-list:

```
access-list name standard {permit | deny} {any4 | host ip_address | ip_address mask}
access-list name extended {permit | deny} protocol source_and_destination_arguments
```

Example:

```
ciscoasa(config)# access-list testacl extended permit ip 10.1.1.0 255.255.255.0 10.2.2.0 255.255.255.0
```

If you use a standard ACL, matching is done on the destination address only. If you use an extended ACL, you can match on source, destination, or both.

IPv6 ACLs are not supported.

Step 2 Create a route map entry:

```
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.

The ACL also includes its own permit and deny statements. For Permit/Permit matches between the route map and the ACL, the Policy Based Routing processing continues. For Permit/Deny matches, processing ends for this route map, and other route maps are checked. If the result is still Permit/Deny, then the regular routing table is used. For Deny/Deny matches, the Policy Based Routing processing continues.

**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:

match ip address access-list_name [access-list_name...]  

Example:

ciscoasa(config-route-map)# match ip address testacl

Step 4 Configure one or more set actions:

• Set the next hop address:

  set ip next-hop ip_address  

  You can configure multiple next-hop IP addresses in which case they are evaluated in the specified order until a valid routable next-hop IP address is found. The configured next-hops should be directly connected; otherwise the set action will not be applied.

• Set the default next hop address:

  set ip default next-hop ip_address  

  If the normal route lookup fails for matching traffic, then the ASA forwards the traffic using this specified next-hop IP address.

• Set a recursive next hop IPv4 address:

  set ip next-hop recursive ip_address  

  Both set ip next-hop and set ip default next-hop require that the next-hop be found on a directly connected subnet. With set ip next-hop recursive, the next-hop address does not need to be directly connected. Instead a recursive lookup is performed on the next-hop address, and matching traffic is forwarded to the next-hop used by that route entry according to the routing path in use on the router.

• Verify if the next IPv4 hops of a route map are available:

  set ip next-hop verify-availability next-hop-address sequence_number track object  

  You can configure an SLA monitor tracking object to verify the reachability of the next-hop. To verify the availability of multiple next-hops, multiple set ip next-hop verify-availability commands can be configured with different sequence numbers and different tracking objects.

• Set the output interface for the packet:

  set interface interface_name  

  or

  set interface null0  

  This command configures the interface through which the matching traffic is forwarded. You can configure multiple interfaces, in which case they are evaluated in the specified order until a valid interface is found. When you specify null0, all traffic matching the route-map will be dropped. There must be a route for the destination that can be routed through the specified interface (either static or dynamic).

• Set the default interface to null0:

  set default interface null0  

  If a normal route lookup fails, the ASA forwards the traffic null0, and the traffic will be dropped.

• Set the Don't Fragment (DF) bit value in the IP header:
### Set IP DF {0|1}

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

```
set ip dscp new_dscp
```

**Note:** When multiple set actions are configured, the ASA evaluates 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 5
Configure an interface and enter interface configuration mode:

```
interface interface_id
```

**Example:**

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

### Step 6
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
```

To remove an existing Policy Based Routing map, simply enter the `no` form of this command.

**Example:**

```
ciscoasa(config-if)# no policy-route route-map testmap
```

---

### Examples for Policy Based Routing

The following sections show examples for route map configuration, policy based routing, and a specific example of PBR in action.

### 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
```
In this example, all traffic matching <acl> will be policy routed and forwarded through outside interface.

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.

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".

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.

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.
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.

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

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

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

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

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

```
ciscoasa(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.

```
ciscoasa(config)# route-map equal-access permit 10
```

```
ciscoasa(config-route-map)# match ip address acl-1
```

Policy Based Routing in Action

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.

First, we will start with the basic configuration for all the devices involved in the set-up. Here, A, B, C, and D represent ASA devices, and H1 and H2 represent IOS routers.

ASA-A:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

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

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

To display the policy routing configuration.

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

---

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.

First, we will start with the basic configuration for all the devices involved in the set-up. Here, A, B, C, and D represent ASA devices, and H1 and H2 represent IOS routers.

ASA-A:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

Policy Based Routing in Action

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

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

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

To display the policy routing configuration.

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

---

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.

First, we will start with the basic configuration for all the devices involved in the set-up. Here, A, B, C, and D represent ASA devices, and H1 and H2 represent IOS routers.

ASA-A:

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

Policy Based Routing in Action

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.

```
ciscoasa(config)# interface GigabitEthernet0/0
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
```

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

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

To display the policy routing configuration.

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

---

We will use this test setup to configure policy based routing with different match criteria and set actions to see how they are evaluated and applied.
ciscoasa(config-if)# ip address 10.1.1.60 255.255.255.0

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

ciscoasa(config)# interface GigabitEthernet0/1.2

ciscoasa(config-if)# vlan 392

ciscoasa(config-if)# nameif dmz

ciscoasa(config-if)# security-level 50

ciscoasa(config-if)# ip address 35.1.1.60 255.255.255.0

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

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 inside

ciscoasa(config-if)# security-level 100

ciscoasa(config-if)# ip address 25.1.1.61 255.255.255.0

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

ciscoasa(config)# interface GigabitEthernet0/1

ciscoasa(config-if)# no shut

ciscoasa(config)# interface GigabitEthernet0/1.2

ciscoasa(config-if)# vlan 392

ciscoasa(config-if)# nameif inside

ciscoasa(config-if)# security-level 0

ciscoasa(config-if)# ip address 35.1.1.61 255.255.255.0

ASA-D:

ciscoasa(config)# interface GigabitEthernet0/0

ciscoasa(config-if)# no shut
ciscoasa(config) #interface GigabitEthernet0/0.1
ciscoasa(config-if)# vlan 291
  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 55.1.1.62 255.255.255.0

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)# 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 10.1.1.60

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
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 here (with the matching route entry highlighted).

```
in 255.255.255.255 255.255.255.255 identity
in 10.1.1.60 255.255.255.255 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 10.1.1.0 255.255.255.0 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_2
    set ip next-hop 35.1.1.61
```

H1: ping 65.1.1.100 repeat 1 source loopback2

The debugs are shown here:

```
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 here:

```
in 255.255.255.255 255.255.255.255 identity
in 10.1.1.60 255.255.255.255 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 10.1.1.0 255.255.255.0 inside
in 25.1.1.0 255.255.255.0 outside
in 35.1.1.0 255.255.255.0 dmz
```
History for Policy Based Routing

Table 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: 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, set ip df, set ip dscp, policy-route route-map, show policy-route, debug policy-route</td>
</tr>
</tbody>
</table>