Revision 0515

NFV 362-WBT: AT&T Vyatta 5600 vRouter Policy-based Routing

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T."

AT&T

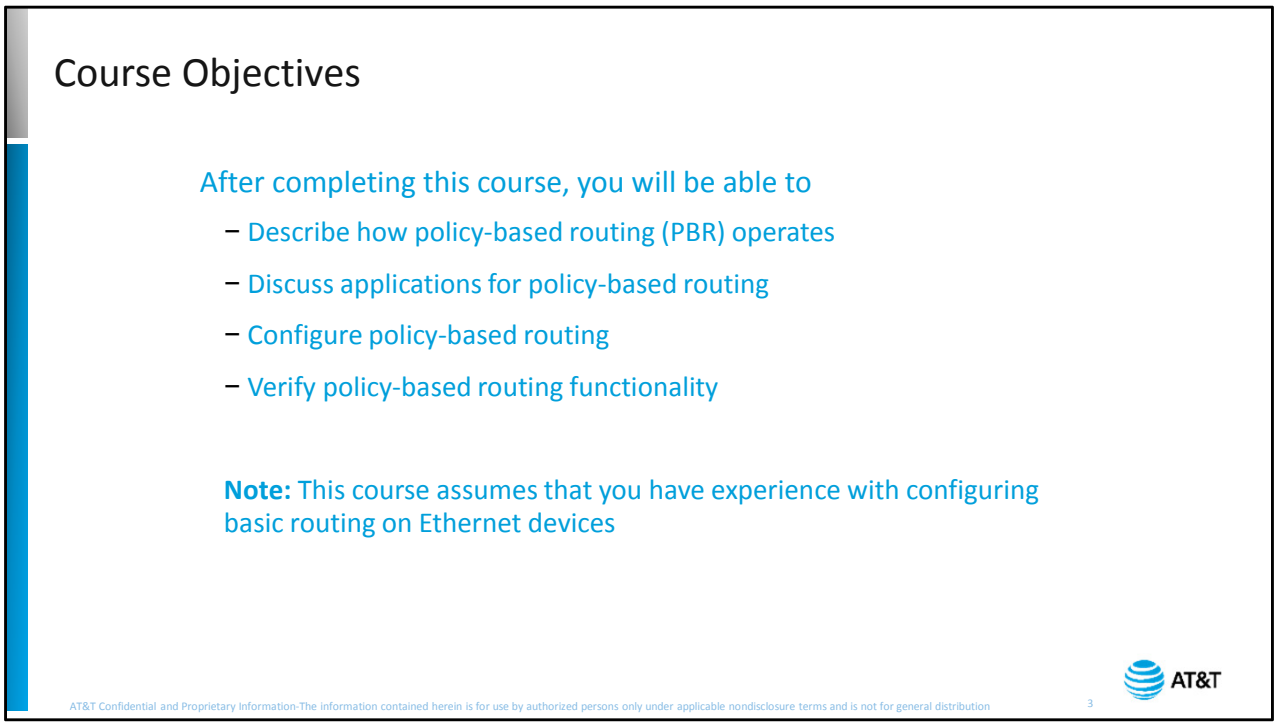
Welcome to the AT&T Vyatta 5600 vRouter Policy-Based routing course.

Legal Disclaimer

All or some of the products detailed in this presentation may still be under development and certain specifications may be subject to change.

Nothing in this presentation shall be deemed to create a warranty of any kind.

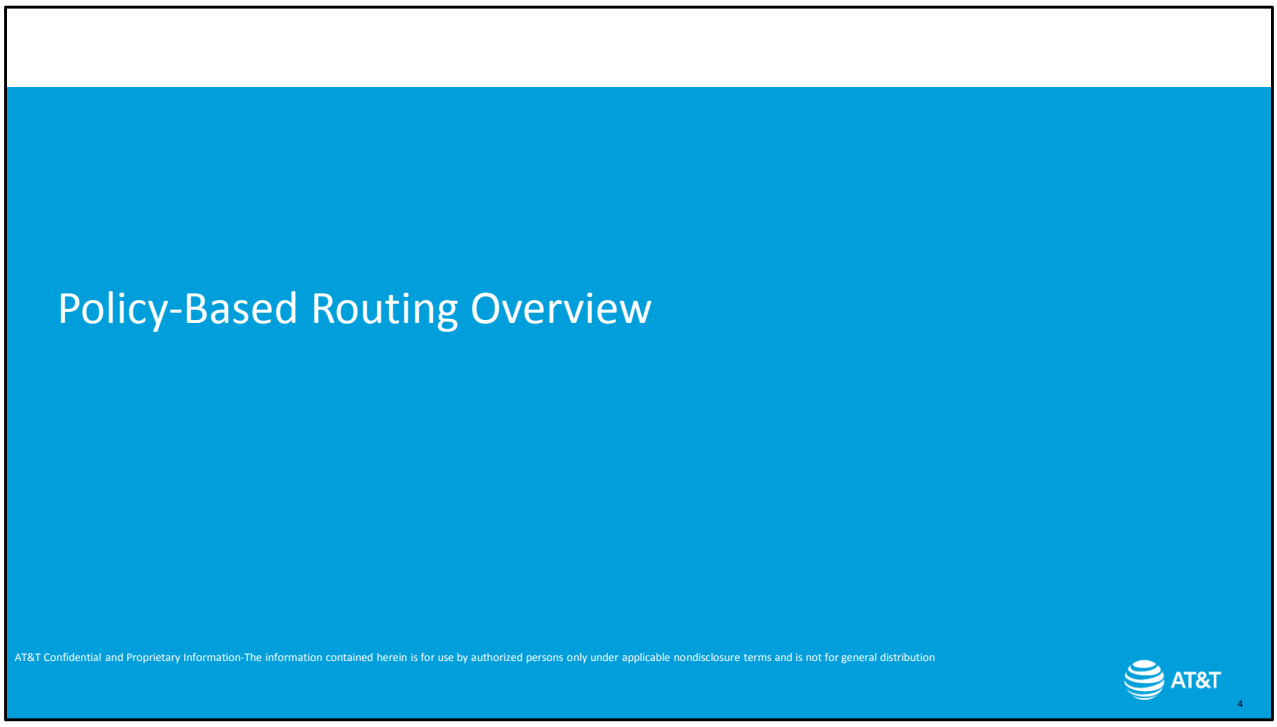
Please take a moment to read our legal disclaimer.



After completing this course, you will be able to:

- Describe how policy-based routing (PBR) operates
- Discuss applications for policy-based routing
- Configure policy-based routing
- Verify policy-based routing functionality

Please note that this course assumes that you have completed courses on basic routing protocol operations, or have equivalent experience configuring routing on a vRouter.

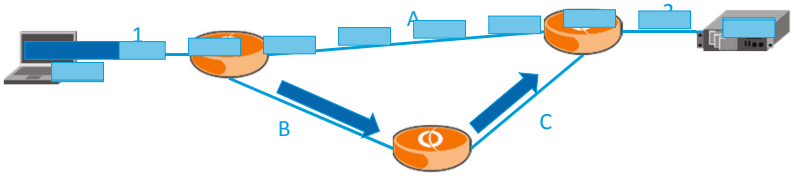


We'll begin with an overview of policy-based routing.

What is Policy-Based Routing?

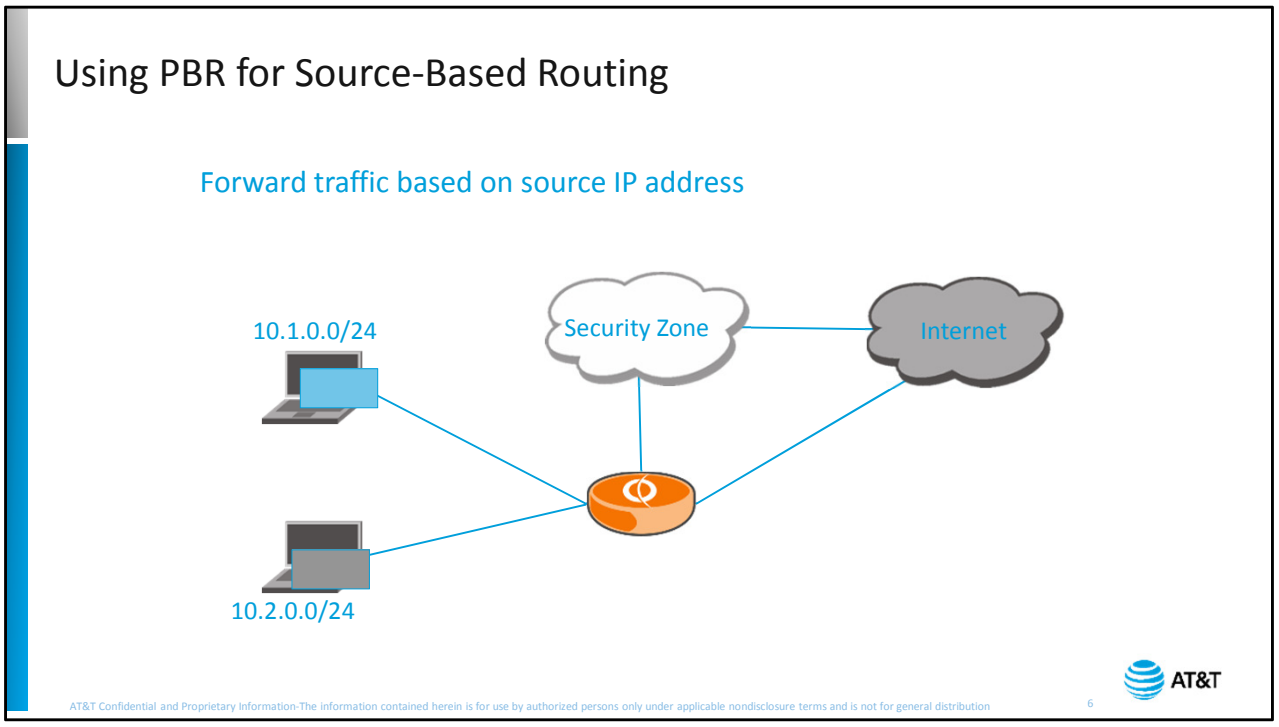
Policy-based Routing (PBR) allows administrators to override routing table

- Static route based on traffic type
- Interactive traffic takes best path (learned by routing protocol)
- File transfers take slower path (defined by PBR)



5

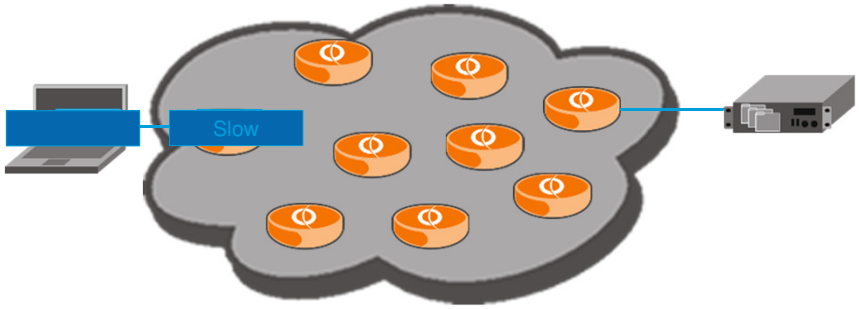
Policy-based routing (PBR) allows you, the administrator, to configure routes that override the device routing table.
You can think of policy-based routes as static routes for specific traffic. Rather than all traffic taking the same path through the network, you can direct traffic down different paths based on various identifiers.
So why would you want to do this? Well, let's take a look at a very simple network.
In this example, our dynamic routing protocol has determined that the best path between network 1 and network 2 is via link A, so all traffic between those two networks takes that path.
Link A is fully utilized. But you have built redundancy into your network, and you have another available path between networks 1 and 2, via links B and C. With standard routing, traffic will never take this path, because it is not the best path.
Policy-based routing allows you to differentiate traffic types, so you can use both paths. Interactive traffic, such as Web browsing, can continue to take the best path as learned by the routing protocol. However, "batch job" traffic such as background file transfers, which don't require the same level of performance, can be directed to take the slower path. This frees up some bandwidth on the best path link, and utilizes the backup path more fully.



One common implementation of policy-based routing is to make forwarding decisions based on the source IP address.
In this example, we want traffic from certain hosts to pass through a security zone before exiting our network. To implement this, we create a routing policy that directs traffic from hosts on subnet 10.1.0.0 to the security zone, while allowing traffic from other subnets to use the regular default route to exit the network.

Using PBR to Support End-to-End QOS

Modifies IP headers for upstream routing/quality of service



Note: Cannot use alternate forwarding and packet marking on same packets

Another function of policy-based routing is to modify IP headers to support end-to-end policy routing.
This can be done through packet marking, you would do this in conjunction with deploying quality of service routing throughout your enterprise. Packets are marked at the entry point to the policy network. Routers further along in the path use the markings to make queuing or forwarding decisions.
In this example we mark some packets to use a ‘fast path’ and others to use a ‘slow path’.
Note that the vRouter cannot perform alternate forwarding and packet marking on the same packet in a single device.

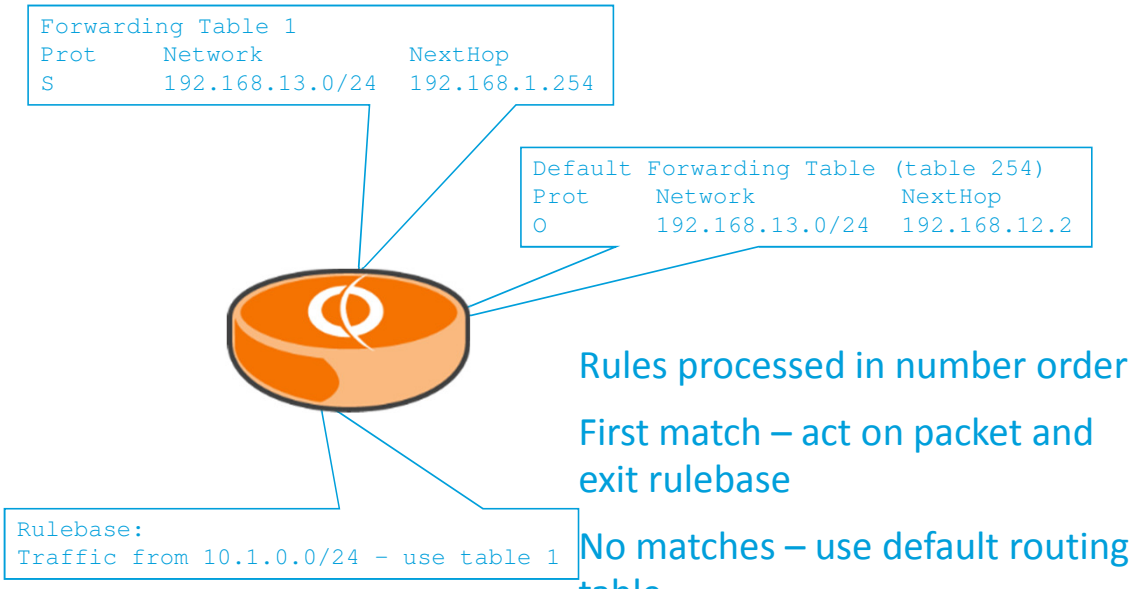
Options for Identifying Traffic

Criteria for Identifying Traffic

- Destination address or port
- Source address or port
- Protocol
- ICMP type
- IP state
- TCP flags
- DSCP matching and marking

You have a choice of several different criteria you can use to identify traffic for policy-based routing. In most cases, you will use information from the IP packet headers, such as source and destination addresses or ports. If you are doing PBR for quality of service applications, you may also use the TCP flags.

How PBR Works in the vRouter



Like any router, the vRouter builds a forwarding table based on information learned from routing protocols and locally configured routing entries. Unless otherwise configured, this is the table that the vRouter uses to make forwarding decisions.

When you configure policy-based routing, you create a separate forwarding table. This table holds static entries that you, the network administrator configure.

You then configure a set of rules in a rulebase that specify which traffic should use the alternate forwarding table. These rules are defined using a syntax similar to other vRouter rule-based features, such as firewalls.

As with other vRouter rulebases, each rule is numbered, and the rules are processed in number order.

As soon as a packet matches a rule, the vRouter takes the action defined in the rule. In this case, the action can be to send the packet using a different forwarding table. If the packet does not match any policy routing rules, the vRouter will forward the packet using the default forwarding table.

DSCP Matching

Incoming packets can be matched with a given DSCP value

- DSCP match values can range from 0 to 63

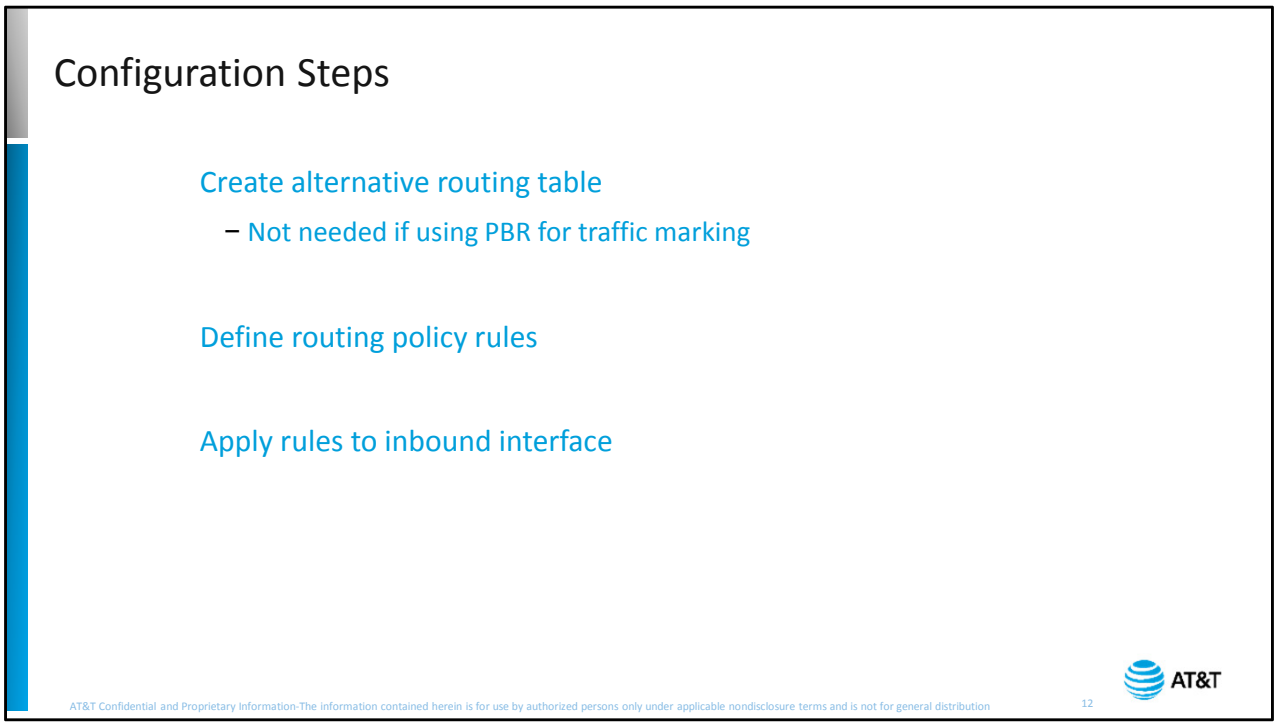
```
set policy route pbr policy_name rule rule_num match dscp
dscp_value
```


AT&T Confidential and Proprietary Information-The information contained herein is for use by authorized persons only under applicable nondisclosure terms and is not for general distribution

vRouter PBR Configuration

AT&T

11



Configuring policy-based routing consists of three major components.
First, if you are using PBR to redirect traffic, you have to create the alternate routing table. If you are using PBR for traffic marking, you can skip this step.
Next, create the rules that define which traffic will use the alternative routes or be marked.
Finally, apply the rulebase to the interface where traffic that you want re-routed will arrive at the vRouter.

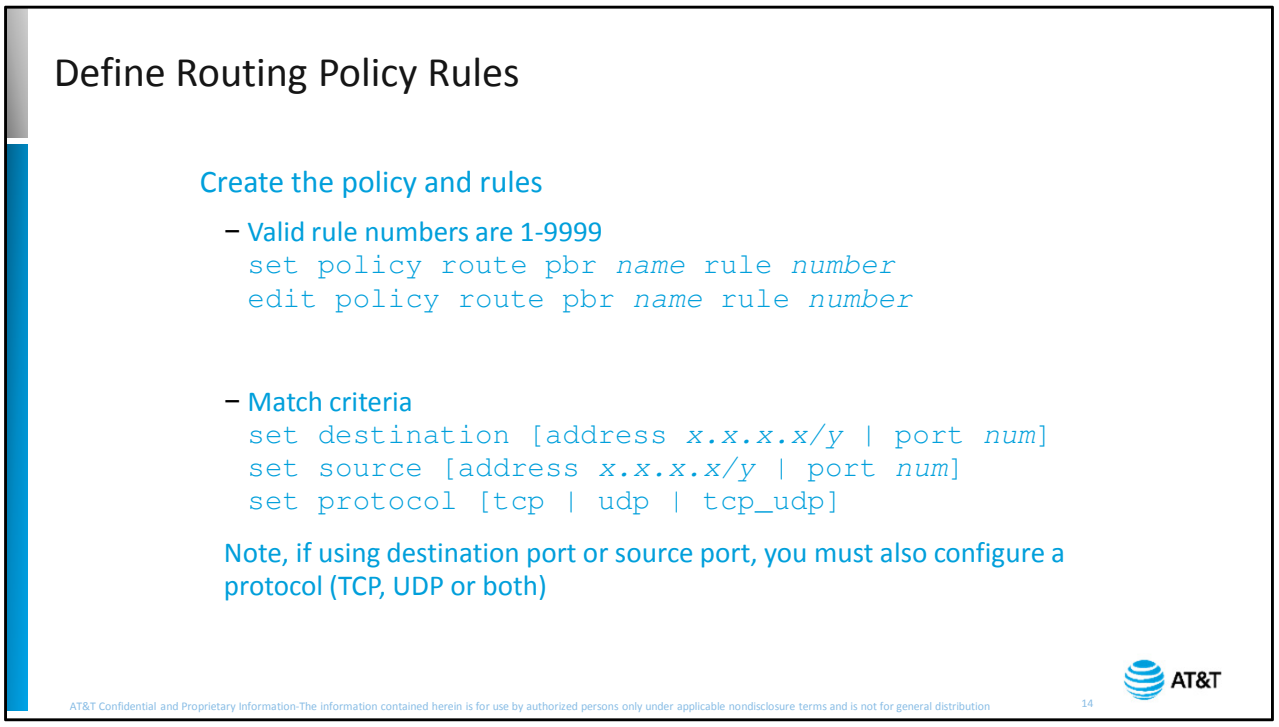
Create Alternative Routing Table

Begin by creating a alternative routing table

`set protocols static table` *`n`* `route` *`network/mask`*
`next-hop` *`address`*

- Table *n* is the identifier for the alternative routing table
  - Value is 1 - 128
  - Default routing able uses number 254

To create the alternative routing table, configure static routes, adding the parameter `table` and a number to identify the alternative routing table. You can create tables numbered 1 through 128, the default routing table uses number 254 as the internal identifier.
If you do not specify a table number, the route will be added to the default system routing table.



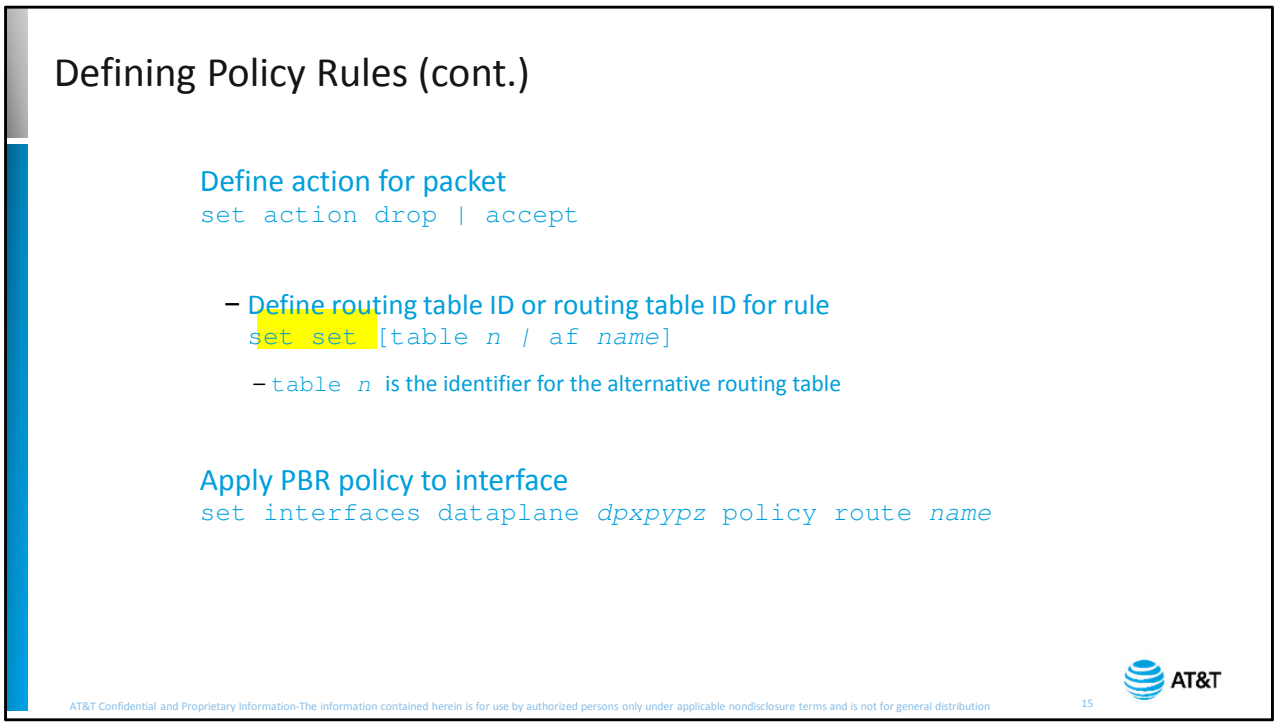
Next, define your policy, rules and match criteria.

To create a PBR policy, use the command `set policy route pbr` and give the policy a name. Each policy can consist of one or more rules; we recommend leaving space between your rule numbers to allow you flexibility if you need to add rules later on.

Because each rule has multiple parameters, we recommend using the `edit` command to move you within the configuration hierarchy and save you some typing. The syntax shown for the next few commands assumes you are using the `edit` command.

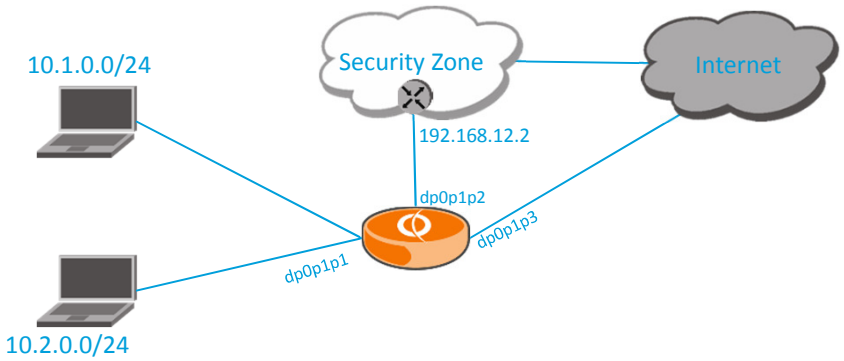
Each rule must include some type of match criteria. If you do not include a match criteria, all traffic will match the rule and have the defined action applied to it. The slide shows the most commonly used match criteria – source IP address and/or port, destination IP address and/or port, and IP protocol.

Note that if you are configuring your match criteria to include source or destination port, you must also set a protocol – TCP, UDP, or both.



Finally, specify the action to be taken on the matching packet. If a rule does not explicitly drop a packet in the action, the PBR action is to accept the packet, which causes it to be sent to the specified alternate routing table for lookup and forwarding.
The `set set` syntax shown here is not an error; you need to enter `set` twice.
Remember, we are several levels in to the configuration hierarchy. The full command syntax would be `set policy route pbr rule number set table.`
The final step is to apply the policy to the inbound interface.

Source-Based Routing Scenario



Redirect traffic from 10.1.0.0/24 to next-hop 192.168.12.2

All other traffic uses default routing table

Apply policy inbound on dp0p1p1

16

AT&T

We will use this scenario in our configuration.
The objective is to redirect traffic from subnet 10.1.0.0 to the next-hop router at 192.168.12.2.
All other traffic will use the vRouter default routing table.
Because traffic from subnet 10.1.0.0 arrives at this vRouter on data plane interface 1, that is where we need to apply the policy.

Source-Based Scenario Configuration

```
[edit]
vyatta@vyatta#  set protocol static table 1 route 0.0.0.0/0 next-hop 192.168.12.2
[edit]
vyatta@vyatta# edit policy route pbr Redirect10 rule 10
[edit policy route pbr Redirect10 rule 10]
vyatta@vyatta# set source address 10.1.0.0/24
[edit policy route pbr Redirect10 rule 10]
vyatta@vyatta# set set table 1
[edit policy route pbr Redirect10 rule 10]
vyatta@vyatta# top
[edit]
vyatta@vyatta# set interface dataplane dp0p1p1 policy route Redirect10
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta# save
[edit]
vyatta@vyatta#
```


17

AT&T

First, we create the alternate routing table entry. In this case, we are creating an alternative default route in table 1. Remember, our rules will ensure that only traffic from subnet 10.1.0.0 will use this routing entry, and in any case, we want all traffic from that subnet to be sent to this different next-hop.

Next, we create the routing policy and the first rule in the policy. Note that the prompt changes to indicate where we are in the configuration hierarchy.

Next, we specify the match criteria – in this case, traffic from network 10.1.0.0.

Next, we specify the action. We want matching traffic to be forwarded using table 1.

Next we go to the top of the configuration hierarchy so we can apply the policy to data plane interface 1.

We commit our changes to make them active. Then save them to make them permanent.

Verifying Operations

18

Verifying Routing Tables

Display a specified route table by number

`show ip route table num`

```
vyatta@vyatta:~$ show ip route table 1
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       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
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       > - selected route, * - FIB route, p - stale info

<Truncated Output>

S    *> 0.0.0.0/0 [1/0] via 192.168.12.2, dp0p1p2 table 1

vyatta@vyatta:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route


S    *> 0.0.0.0/0 [1/0] via 172.24.42.1, dp0p1p3
<Truncated Output>
```


AT&T

To verify the contents of an alternate routing table, use the command `show ip route` followed by the `table` parameter and the table number.

In this example, we are looking at table 1, which we created in our source-based routing scenario. The output displays the static default route that we configured, and the asterisk (*) and angle bracket (>) shows that the route is active and *in* the Forwarding Database (FIB).

If we do not specify a table number, the `show ip route` output displays the contents of the default routing table. In this case, we can see the static default route that will be used by all traffic that is not directed to table 1 by our routing policy.

Verifying PBR Policies

Display PBR rules and statistics active on an interface

`show policy route dpxpypz`

```
vyatta@vyatta:~$  show policy route dp0p1p1

----------------------------
Rulesets Information
----------------------------

IPv4 Policy Route "Redirect10":

 Active on (dp0p1p1,ROUTE)

rule  action   proto     packets  bytes
----  ------   -----     -------  -----
10    set      all       15       1104
  condition - saddr 10.1.0.0/24 daddr 0.0.0.0/0

10000 drop     all       1228     80360
  condition - saddr 0.0.0.0/0 daddr 0.0.0.0/0

vyatta@vyatta:~$
```


20

AT&T

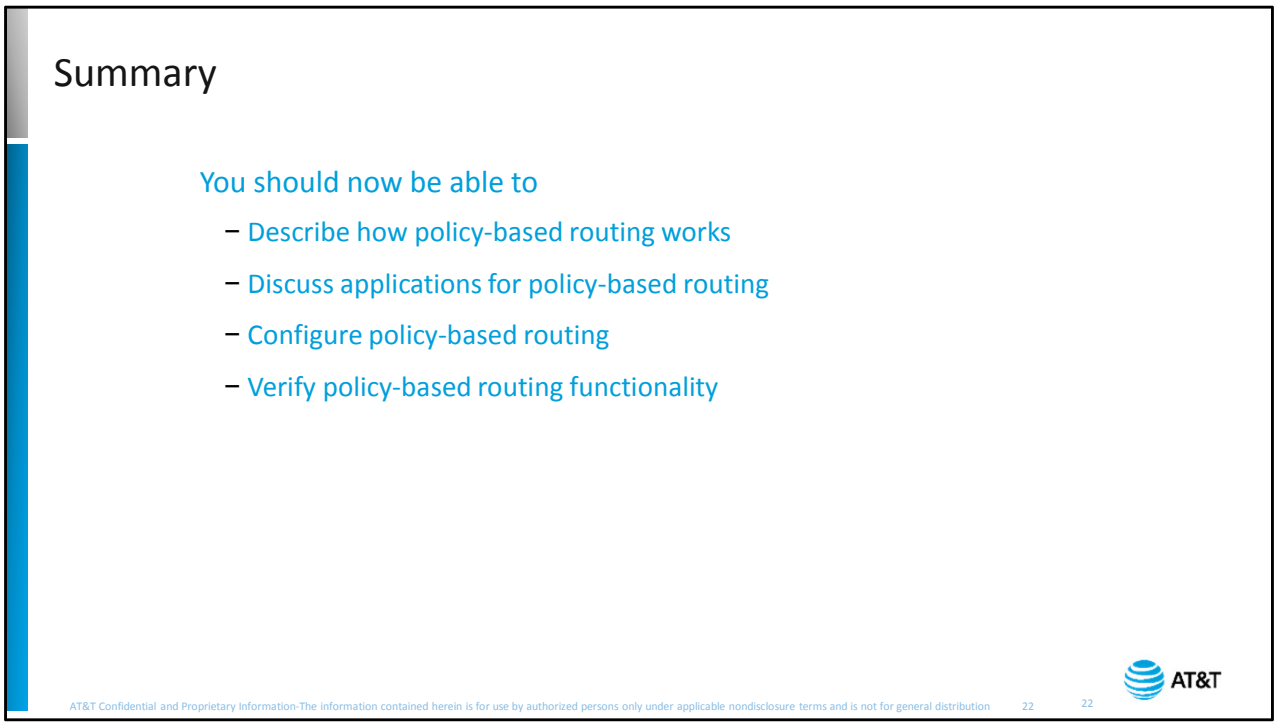
To verify your policy configuration, use the command `show policy route` followed by the data plane interface number.
In this example, we are looking at our rulebase from the source-based routing scenario.
Note that the output does not indicate the action – only that there is a match condition set.
We would have to look at the actual policy to see what is happening to packets matching the rule.

Testing Policies

- Use an external host to test
  - vRouter cannot use alternate tables for locally-generated traffic

```
[admin@localhost ~]$ ifconfig
dp0p1p1   Link encap:Ethernet  HWaddr 00:0C:29:38:A6:1C
          inet addr:10.1.0.10  Bcast:10.1.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe38:a61c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5375 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3403 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7681822 (7.3 MiB)  TX bytes:239575 (233.9 KiB)
          Interrupt:18 Base address:0x2000
[admin@localhost ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.101.1 (192.168.101.1)  0.485 ms  0.236 ms  0.158 ms
 2  192.168.12.2 (192.168.12.2)  0.457 ms  0.722 ms  0.617 ms
 3  172.24.42.1 (172.24.42.1)  0.951 ms  0.775 ms  0.654 ms
<Truncated Output>
```


In order to actually test your policies, you will need an external host.
The vRouter cannot use alternate tables to forward traffic generated by the vRouter itself, so using commands like `ping -I` or `traceroute` from a local interface will not validate a PBR configuration.
In this case, our external host has an IP address within the range we have defined as using the alternate routing table.
When we issue the `traceroute` command, we see that the first hop is the router with the configured policy, and the second hop is the next-hop IP address that we specified in the alternate routing table. This verifies that our traffic is using the alternate routing table.



This concludes the AT&T Vyatta 5600 vRouter Policy-Based Routing course.
You should now be able to:

- Describe how policy-based routing operates
- Discuss applications for policy-based routing
- Configure policy-based routing
- Verify policy-based routing functionality

We hope that this information has been useful, and that you will take additional AT&T Vyatta courses in the future.
Thank you.

End Of: Vyatta 5600 vRouter
Policy-based Routing

AT&T

23