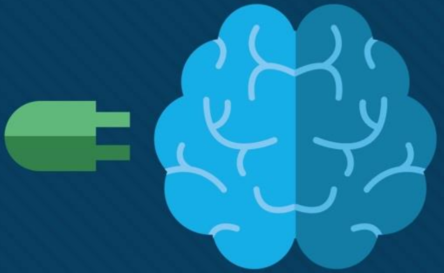
Chapter 22: Infrastructure Security

Instructor Materials

CCNP Enterprise: Advanced Routing



Chapter 22 Content

This chapter covers the following content:

- **Cisco IOS AAA Troubleshooting -** This section explains how to identify and troubleshoot issues related to AAA using the local database, a RADIUS server, and a TACACS+ server.
- **Troubleshooting Unicast Reverse Path Forwarding (uRPF) -** This section explores what to look for when having issues with uRPF.
- **Troubleshooting Control Plane Policing (CoPP) -** This section examines CoPP and the items you should be considering when troubleshooting issues related to CoPP.
- **IPv6 First-Hop Security -** This section describes IPv6 First-Hop Security features, such as RA Guard, DHCP Guard, ND inspection/snooping, and Source Guard.

Cisco IOS AAA Troubleshooting

- AAA is a framework that provides authentication, authorization, and accounting when securing the management plane.
- The first A in AAA stands for authentication, which is about identifying and verifying the user based on something she knows, something she has, or something she is.
- The second A in AAA stands for authorization, which is about determining and controlling what the authenticated user is permitted to do.
- The final A in AAA stands for accounting, which is about collecting information to be used for billing, auditing, and reporting.
- This section examines AAA, using the local database, a RADIUS server, and a TACACS+ server.

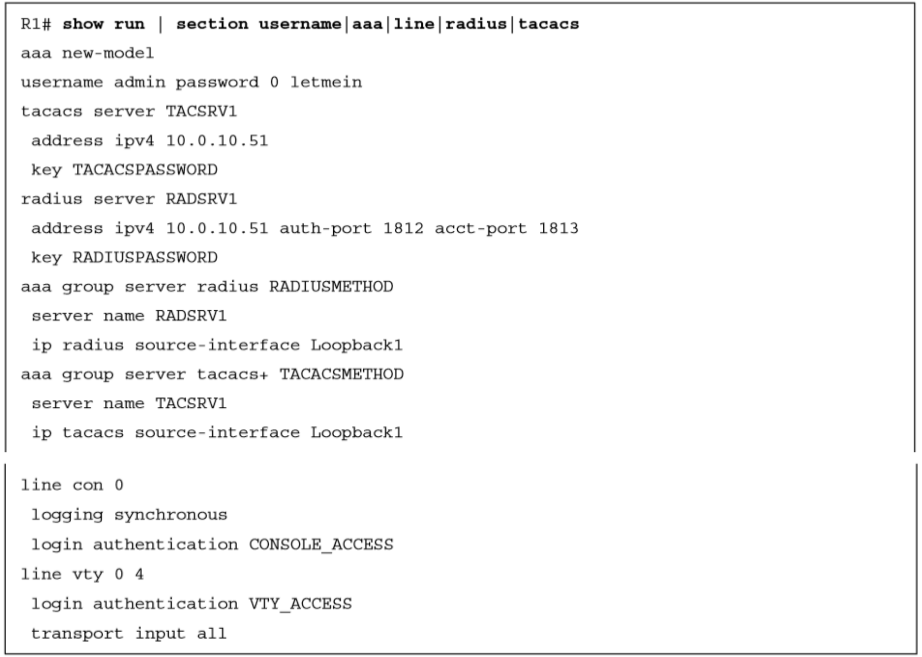
Cisco IOS AAA Troubleshooting

Cisco IOS AAA Basics

This section examines AAA, using the local database, a RADIUS server, and a TACACS+ server.

Example 22-1 provides a sample Cisco IOS AAA configuration for management access to the vty lines and the console port. The **aaa new-model** command is used to enable AAA services on the router. By default, AAA is disabled. AAA commands are not available on Cisco IOS products until you enable AAA.

Example 22-1 *Verifying Cisco IOS AAA Configuration*



```
R1# show run | section username|aaa|line|radius|tacacs
aaa new-model
username admin password 0 letmein
tacacs server TACSRV1
 address ipv4 10.0.10.51
 key TACACSPASSWORD
radius server RADSRV1
 address ipv4 10.0.10.51 auth-port 1812 acct-port 1813
 key RADIUSPASSWORD
aaa group server radius RADIUSMETHOD
 server name RADSRV1
 ip radius source-interface Loopback1
aaa group server tacacs+ TACACSMETHOD
 server name TACSRV1
 ip tacacs source-interface Loopback1

line con 0
 logging synchronous
 login authentication CONSOLE_ACCESS
line vty 0 4
 login authentication VTY_ACCESS
 transport input all
```

Cisco IOS AAA Commands

Refer back to Figure 22-1.

- The **username admin password 0 letmein** command is used in this case to create the username admin and the password letmein, which will be stored in the local username and password database.
- The **tacacs server TACSRV1** command is used to provide the settings needed to connect to a TACACS+ server.
- The **radius server RADSRV1** command is used to provide the settings needed to connect to a RADIUS server.
- The **aaa group server radius RADIUSMETHOD** command is used to group one or more RADIUS servers that will be used together within a distinct list or method and to specify any common settings that will be used.
- The **aaa group server tacacs+ TACACSMETHOD** command is used to group one or more TACACS servers that will be used together within a distinct list or method and to specify any common settings that will be used.

Cisco IOS AAA Troubleshooting

Cisco IOS AAA Commands (Cont.)

Refer back to Figure 22-1.

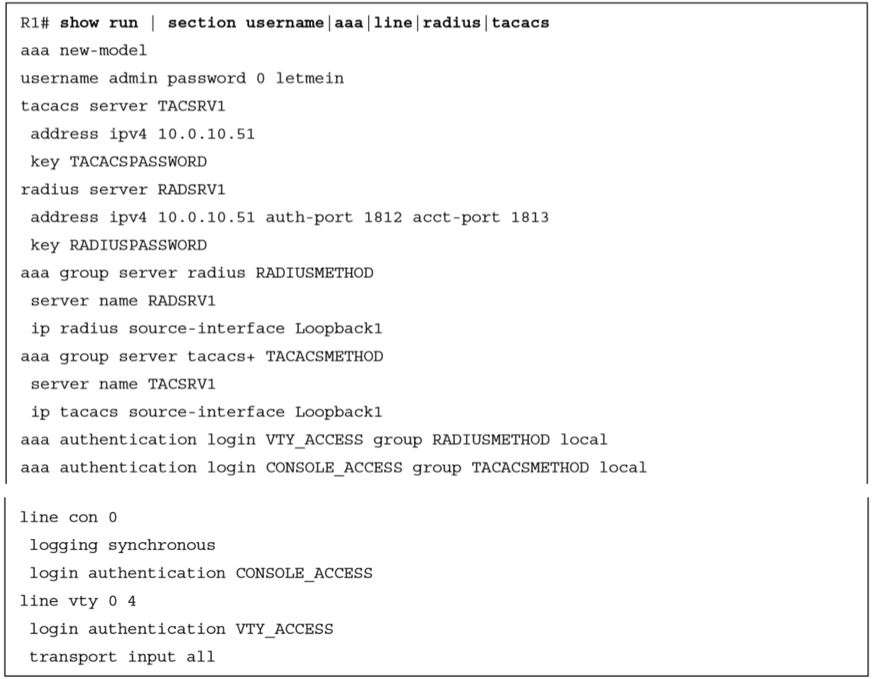
- The **aaa authentication login VTY_ACCESS group RADIUSMETHOD local** command creates a AAA method list called VTY_ACCESS for login authentication. The first method that will be used is the group of servers in the RADIUSMETHOD group and if the RADIUS servers are not available, and the second method that will be used is the local username and password database.
- The **aaa authentication login CONSOLE_ACCESS group TACACSMETHOD local** command creates a AAA method list called CONSOLE_ACCESS for login authentication.
- The command **login authentication CONSOLE_ACCESS** in line con 0 configuration mode configures the console port to use the AAA method list called CONSOLE_ACCESS for authenticating to the console port.
- The **command login authentication VTY_ACCESS** in line vty 0 4 configuration mode configures the vty lines to use the AAA method list called VTY_ACCESS for authenticating vty access, such as Telnet and SSH connections.

Cisco IOS AAA Troubleshooting
Verifying/Troubleshooting Cisco IOS AAA Configuration

Using Example 22-1, consider the following when troubleshooting Cisco IOS AAA authentication:

- **AAA needs to be enabled** - AAA is disabled by default on Cisco routers and switches. To enable AAA, use the **aaa new-model** command.
- **AAA relies on the local username and password database or an AAA server such as RADIUS or TACACS+**
- **A method list defines the authentication methods -** When no method list exists, the vty lines use the local username and password database by default.

Example 22-1 *Verifying Cisco IOS AAA Configuration*



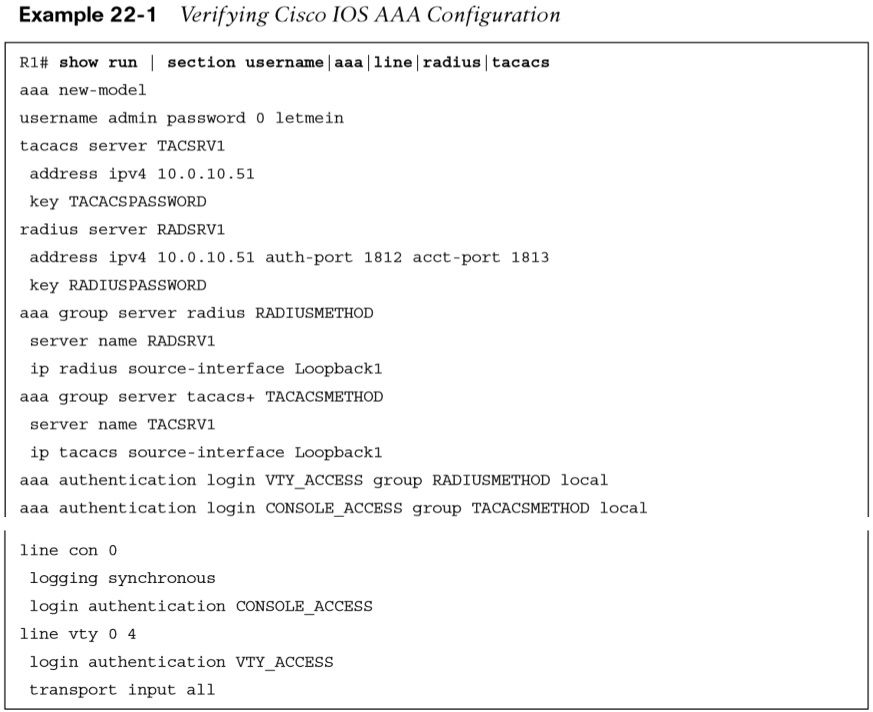
```
R1# show run | section username|aaa|line|radius|tacacs
aaa new-model
username admin password 0 letmein
tacacs server TACSRV1
 address ipv4 10.0.10.51
 key TACACSPASSWORD
radius server RADSRV1
 address ipv4 10.0.10.51 auth-port 1812 acct-port 1813
 key RADIUSPASSWORD
aaa group server radius RADIUSMETHOD
 server name RADSRV1
 ip radius source-interface Loopback1
aaa group server tacacs+ TACACSMETHOD
 server name TACSRV1
 ip tacacs source-interface Loopback1
aaa authentication login VTY_ACCESS group RADIUSMETHOD local
aaa authentication login CONSOLE_ACCESS group TACACSMETHOD local

line con 0
 logging synchronous
 login authentication CONSOLE_ACCESS
line vty 0 4
 login authentication VTY_ACCESS
 transport input all
```

Cisco IOS AAA Troubleshooting

Verifying/Troubleshooting Cisco IOS AAA Configuration (Cont.)

- **Method list service is incorrect** – The method list service must match the service for which you are creating the list.
- **AAA method lists are applied to the lines -** The method list that will be used to define how authentication will occur for the vty lines or console line needs to be applied with the **login authentication {default** | *list_name*} command.
- **The router needs to be able to reach the AAA server -** Use the **test aaa** command on the router or use Telnet to reach the authentication port number of the AAA server to verify connectivity.
- **The router needs to be configured with the correct pre-shared key -** Ensure that the router and the AAA server are configured with the same pre-shared key.



Example 22-1 *Verifying Cisco IOS AAA Configuration*

```
R1# show run | section username|aaa|line|radius|tacacs
aaa new-model
username admin password 0 letmein
tacacs server TACSRV1
 address ipv4 10.0.10.51
 key TACACSPASSWORD
radius server RADSRV1
 address ipv4 10.0.10.51 auth-port 1812 acct-port 1813
 key RADIUSPASSWORD
aaa group server radius RADIUSMETHOD
 server name RADSRV1
 ip radius source-interface Loopback1
aaa group server tacacs+ TACACSMETHOD
 server name TACSRV1
 ip tacacs source-interface Loopback1
aaa authentication login VTY_ACCESS group RADIUSMETHOD local
aaa authentication login CONSOLE_ACCESS group TACACSMETHOD local

line con 0
 logging synchronous
 login authentication CONSOLE_ACCESS
line vty 0 4
 login authentication VTY_ACCESS
 transport input all
```

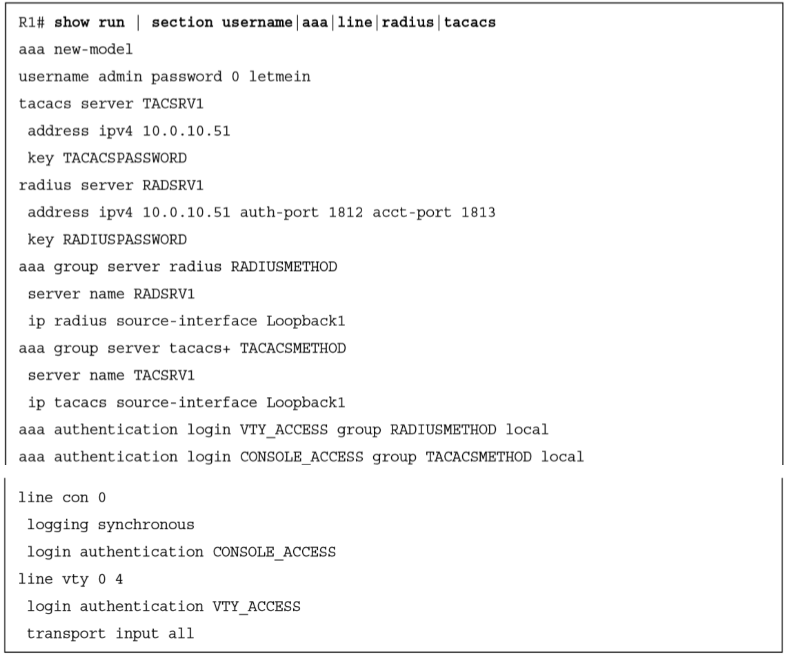
Cisco IOS AAA Troubleshooting

Verifying/Troubleshooting Cisco IOS AAA Configuration (Cont.)

- **The correct authenticating and accounting ports need to be configured -** RADIUS uses ports 1812 or 1645 (Cisco default) for authentication and 1813 or 1646 (Cisco default) for accounting.
- **Usernames and passwords need to be configured on the AAA server.**
- **The AAA server group needs to have the correct AAA server IP addresses**
- **User can authenticate but can’t execute any commands**
- **IP address of client configured on AAA server.**
- These commands can be used in real-time to verify the operation of the various AAA authentication processes:

debug aaa authentication
debug radius authentication
debug tacacs authentication
debug aaa protocol local

Example 22-1 *Verifying Cisco IOS AAA Configuration*



```
R1# show run | section username|aaa|line|radius|tacacs
aaa new-model
username admin password 0 letmein
tacacs server TACSRV1
 address ipv4 10.0.10.51
 key TACACSPASSWORD
radius server RADSRV1
 address ipv4 10.0.10.51 auth-port 1812 acct-port 1813
 key RADIUSPASSWORD
aaa group server radius RADIUSMETHOD
 server name RADSRV1
 ip radius source-interface Loopback1
aaa group server tacacs+ TACACSMETHOD
 server name TACSRV1
 ip tacacs source-interface Loopback1
aaa authentication login VTY_ACCESS group RADIUSMETHOD local
aaa authentication login CONSOLE_ACCESS group TACACSMETHOD local
line con 0
 logging synchronous
 login authentication CONSOLE_ACCESS
line vty 0 4
 login authentication VTY_ACCESS
 transport input all
```

Troubleshooting Unicast Reverse Path Forwarding (uRPF)

- uRPF is a security feature that helps limit or even eliminate spoofed IP packets on a network. This is accomplished by examining the source IP address of an ingress packet and determining whether it is valid. If it is valid, the packet will be forwarded. If it is not valid, the packet will be discarded.

- CEF (Cisco Express Forwarding) must be enabled on the IOS device for uRPF to work.

Troubleshooting Unicast Reverse Path Forwarding (uRPF)

uRPF can operate in three different modes. The mode you choose determines how the packet is identified as being valid or not valid:

- **Strict –** The router reviews the source IP address of the packet and notes the ingress interface. It then looks at the routing table to identify the interface (other than a default route) that would be used to reach the source IP address of the packet.
- **Loose –** The router reviews only the source IP address of the packet. It then looks in the routing table to identify whether there is any interface (other than a default route) that can be used to reach the source IP address listed in the packet. If there is and it is not the default route, the packet is valid and is forwarded. If not, the packet is discarded.
- **VRF -** VRF mode is the same as loose mode; however, it only examines interfaces that are in the same VRF as the interface on which the packet was received.

Troubleshooting Unicast Reverse Path Forwarding (uRPF) Modes (Cont.)

Because uRPF is configured on an interface-by-interface basis with the **command ip verify unicast source reachable-via {rx | any} [allow-default] [allow-self-ping]** *[list*], choosing the correct mode is important. (**rx** is for strict mode, and **any** is for loose mode.) Choosing the wrong mode can cause dropping of valid packets because of symmetric versus asymmetric routing.

You use the **allow-default** option when the return path is associated with an interface that is chosen based on a default route. By default, it is discarded. However, in cases where you need to override this behavior, you use the **allow-default** option in the command.

Another consideration is the *list* option, which allows you to attach an ACL that identifies which packets are subject to a uRPF check and which ones are not.

Pings from the router to one of its own IP addresses are blocked by uRPF. To self ping, use the **allow-self-ping** option.

Use the **show cef interface** *interface_name interface number* command in privileged EXEC mode to verify that uRPF and CEF are enabled on an interface.

Troubleshooting Control Plane Policing (CoPP)

CoPP varies based on IOS version and platform version. Therefore, this section covers the general elements that apply to all versions.

When configuring CoPP, you need to do the following:

- Create ACLs to identify the traffic.
- Create class maps to define a traffic class.
- Create policy maps to define a service policy.
- Apply the service policy to the control plane.

When troubleshooting, look for issues in the ACLs, the class maps, the policy maps, and the application of the service policy.

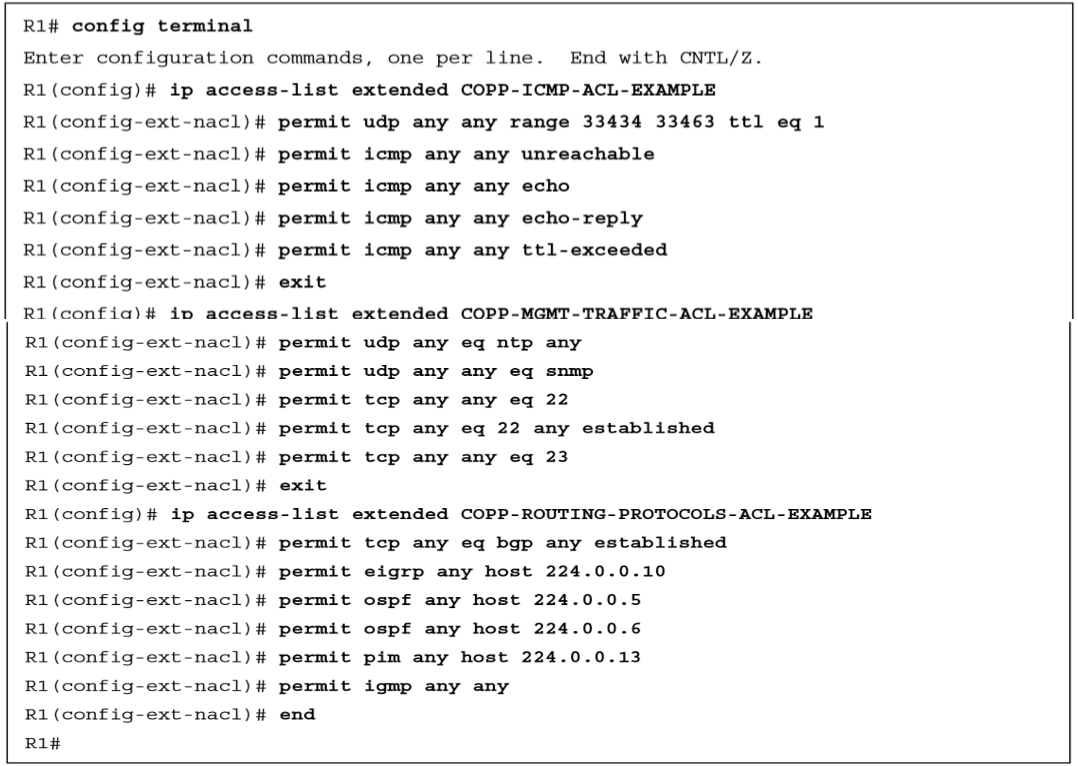
Troubleshooting Control Plane Policing (CoPP)

Creating ACLs to Identify the Traffic

ACLs are used with CoPP for identifying traffic. When the traffic is matched, it becomes the object of the policy action. So, defining the ACLs is the most critical step of the CoPP process as it is the foundation of CoPP. If the ACL is not created correctly, the traffic will not be matched, and therefore the policies will not be correctly applied.

Example 22-2 shows three ACLs defined. Each ACL was created with a specific purpose in mind.

Example 22-2 *A Sample ACL Configuration for CoPP*



```
R1# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list extended COPP-ICMP-ACL-EXAMPLE
R1(config-ext-nacl)# permit udp any any range 33434 33463 ttl eq 1
R1(config-ext-nacl)# permit icmp any any unreachable
R1(config-ext-nacl)# permit icmp any any echo
R1(config-ext-nacl)# permit icmp any any echo-reply
R1(config-ext-nacl)# permit icmp any any ttl-exceeded
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended COPP-MGMT-TRAFFIC-ACL-EXAMPLE
R1(config-ext-nacl)# permit udp any eq ntp any
R1(config-ext-nacl)# permit udp any any eq snmp
R1(config-ext-nacl)# permit tcp any any eq 22
R1(config-ext-nacl)# permit tcp any eq 22 any established
R1(config-ext-nacl)# permit tcp any any eq 23
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE
R1(config-ext-nacl)# permit tcp any eq bgp any established
R1(config-ext-nacl)# permit eigrp any host 224.0.0.10
R1(config-ext-nacl)# permit ospf any host 224.0.0.5
R1(config-ext-nacl)# permit ospf any host 224.0.0.6
R1(config-ext-nacl)# permit pim any host 224.0.0.13
R1(config-ext-nacl)# permit igmp any any
R1(config-ext-nacl)# end
R1#
```

Troubleshooting ACLs for CoPP

When troubleshooting ACLs for CoPP, focus on the following:

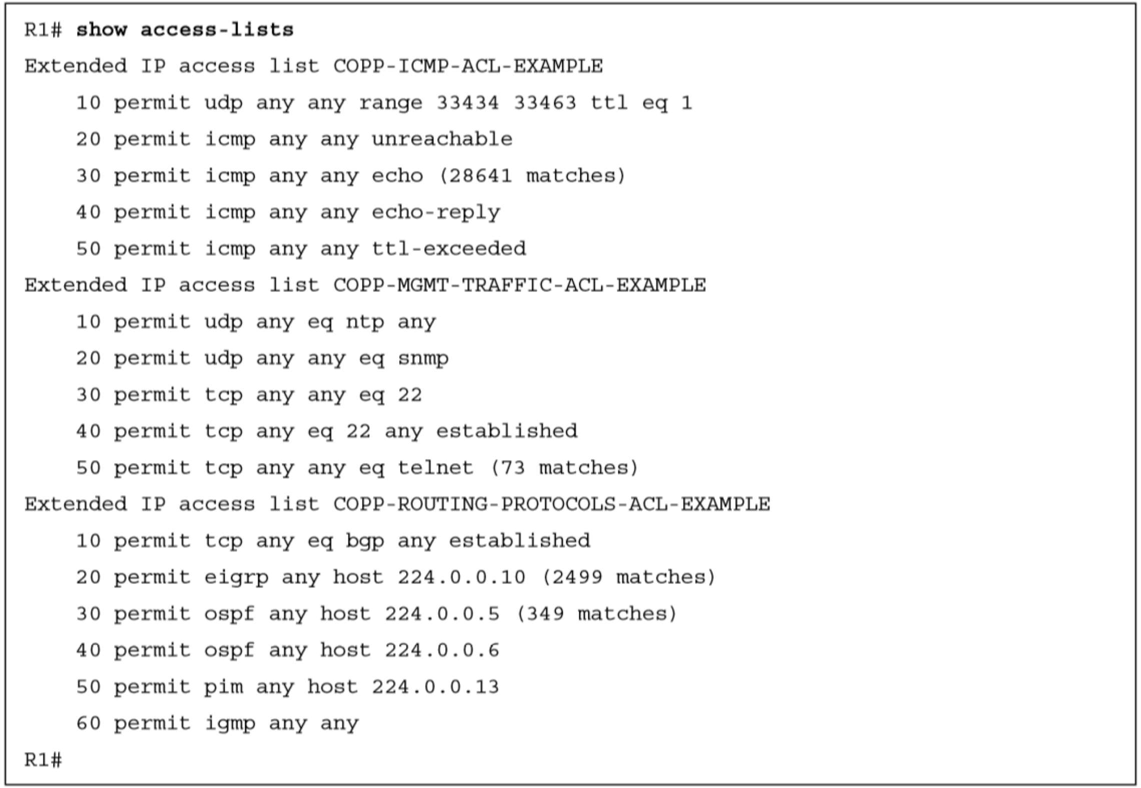
- **Grouping -** When grouping traffic types together, ensure that they are grouped based on function within the network. If you mix and match various protocols, the policies you apply later may not work for the type of traffic in question.
- **Action -** With ACLs, you can specify a permit or deny action. For CoPP, permit means to match the traffic and apply the policy. Deny means to exclude that traffic from the class and move on to the next class.
- **Protocol -** In an ACL, you can define a protocol that you want to match. If the wrong protocol is specified in the ACL, then the wrong type of traffic will be matched in the class. So, when troubleshooting, verify that the correct protocol is being specified in the ACL.
- **Source and destination -** Because ACLs allow you to specify source and destination addresses, you can be granular with your CoPP policies and match only if the traffic is from a specific source or destination. Therefore, it is imperative that the ACLs have the correct source and destination IP addresses applied, or the traffic will not be matched when it should be.
- **Operators and ports -** ACLs allow you to define operators such as greater-than, less-than, and equal-to and port numbers such as 179, 21, 22, 23, 80, and 443. Do not use the log or log-input keywords with CoPP ACLs, as logging can cause unexpected results with CoPP functionality.

Troubleshooting Control Plane Policing (CoPP)

Verifying ACLs

You can verify ACLs by using the show access-lists command, as shown in Example 22-3.

Example 22-3 *Verifying ACLs with the show access-lists Command*



```
R1# show access-lists
Extended IP access list COPP-ICMP-ACL-EXAMPLE
    10 permit udp any any range 33434 33463 ttl eq 1
    20 permit icmp any any unreachable
    30 permit icmp any any echo (28641 matches)
    40 permit icmp any any echo-reply
    50 permit icmp any any ttl-exceeded
Extended IP access list COPP-MGMT-TRAFFIC-ACL-EXAMPLE
    10 permit udp any eq ntp any
    20 permit udp any any eq snmp
    30 permit tcp any any eq 22
    40 permit tcp any eq 22 any established
    50 permit tcp any any eq telnet (73 matches)
Extended IP access list COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE
    10 permit tcp any eq bgp any established
    20 permit eigrp any host 224.0.0.10 (2499 matches)
    30 permit ospf any host 224.0.0.5 (349 matches)
    40 permit ospf any host 224.0.0.6
    50 permit pim any host 224.0.0.13
    60 permit igmp any any
R1#
```

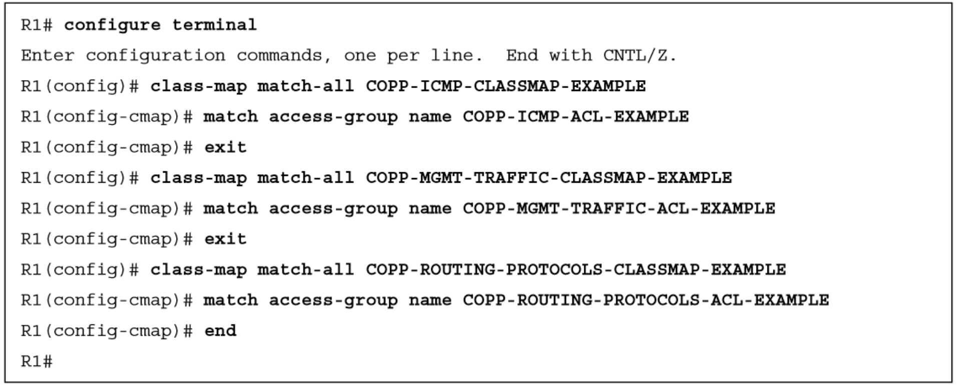
Creating Class Maps to Define a Traffic Class

Class maps are used to define a traffic class that is composed of three different elements:

- There is a name.
- One or more match commands are used to identify the packets that are part of the class.
- There are instructions on how the match commands will be evaluated.

As shown in Example 22-4, the name of the first class map listed is COPP-ICMP-CLASSMAP-EXAMPLE, with the instructions to match all.

Example 22-4 *A Sample Class Map Configuration for CoPP*



```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# class-map match-all COPP-ICMP-CLASSMAP-EXAMPLE
R1(config-cmap)# match access-group name COPP-ICMP-ACL-EXAMPLE
R1(config-cmap)# exit
R1(config)# class-map match-all COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE
R1(config-cmap)# match access-group name COPP-MGMT-TRAFFIC-ACL-EXAMPLE
R1(config-cmap)# exit
R1(config)# class-map match-all COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE
R1(config-cmap)# match access-group name COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE
R1(config-cmap)# end
R1#
```

The syntax of a class map is as follows:

router(config)# **class-map [match-any | match-all]** *class-name*
router(config-cmap)# **match [access-group | protocol | ip prec | ip dscp]**

Troubleshooting Class Maps

When troubleshooting class maps, focus on the following:

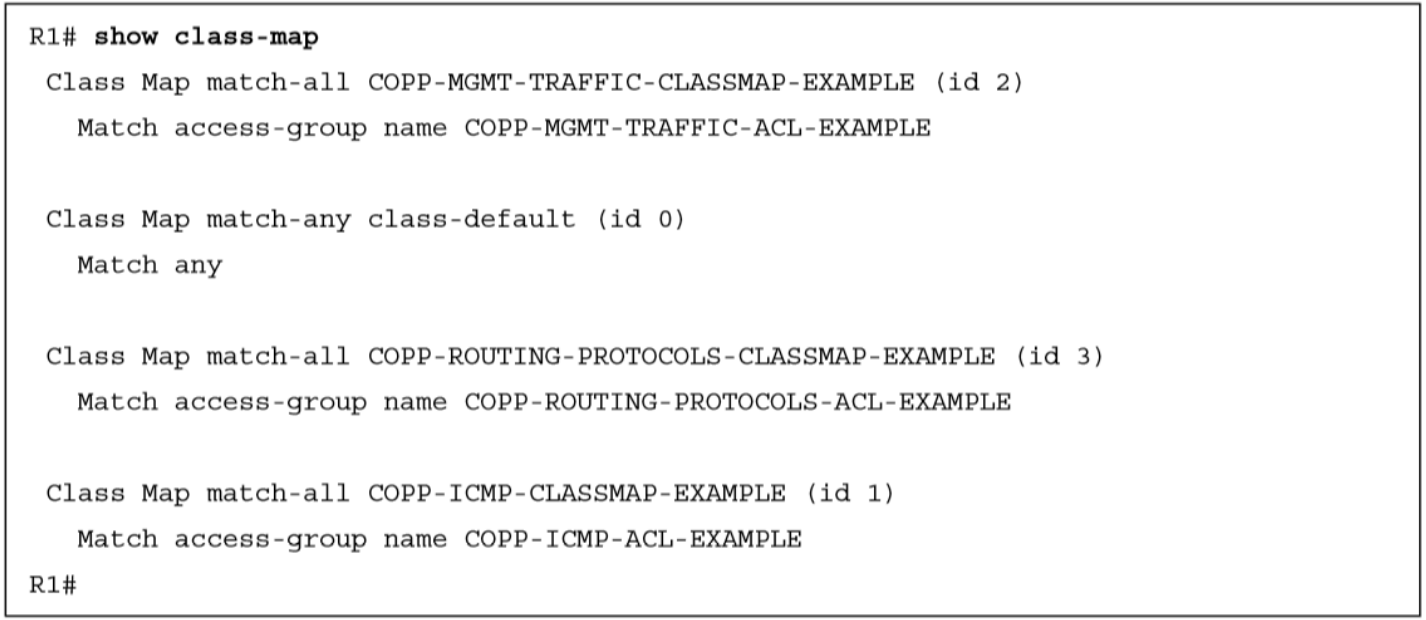
- **Access group -** The ACL in the class map is responsible for defining the interesting traffic (packets) that must be matched. If matched, the packets are classified as being members of the class, and the correct service policy applies.
- **Instruction -** A class map may contain one of two instructions: **match-any** or **match-all**. Using the correct instruction is important only if you have multiple **match** commands in the class map. If you have only one **match** command in the class map, the instruction does not matter.
- **Protocol -** If you choose not to use an ACL for matching, you can use the built-in protocol options of the **match** command. When using the **protocol** option, you must ensure that the correct protocol has been specified.
- **IP PREC/IP DSCP -** If you only need to match based on IP precedence or IP DSCP (Differentiated Services Code Point) values, you can use the **ip prec** or **ip dscp** options of the match command.
- **Case -** ACL names are case sensitive.

Troubleshooting Control Plane Policing (CoPP)

Verifying Class Maps

To verify all configured class maps, use the **show class-map** command, as shown in Example 22-6.

Example 22-6 *Verifying Class Maps with the show class-map Command*



```
R1# show class-map
 Class Map match-all COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE (id 2)
   Match access-group name COPP-MGMT-TRAFFIC-ACL-EXAMPLE

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE (id 3)
   Match access-group name COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE

 Class Map match-all COPP-ICMP-CLASSMAP-EXAMPLE (id 1)
   Match access-group name COPP-ICMP-ACL-EXAMPLE
R1#
```

Troubleshooting Control Plane Policing (CoPP)

Creating Policy Maps to Define a Service Policy

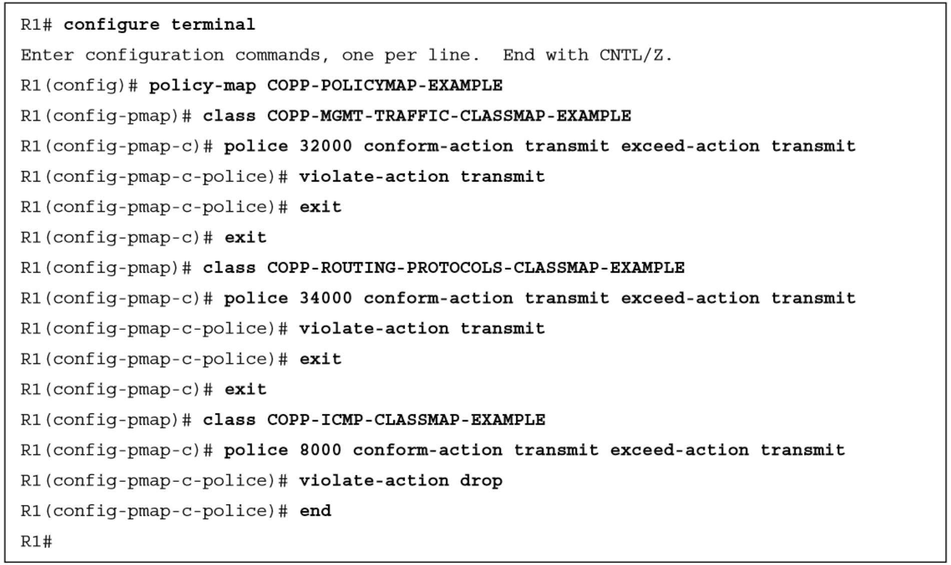
Policy maps are used with CoPP to associate the traffic class (as defined by the class map) with one or more policies, resulting in a service policy. The three elements are a name, a traffic class, and a policy.

In Example 22-7, there is a policy map named COPP-POLICYMAP-EXAMPLE that identifies multiple classes and the policy that is applied if the traffic matches.

The syntax used when creating a policy map for CoPP is as follows:

router(config)# **policy-map** *service_policy_name*
router(config-pmap)# **class** *traffic_class_name*
router(config-pmap-c)# **police [cir | rate] conform-action [transmit | drop] exceed-action [transmit | drop]**

Example 22-7 *A Sample Policy Map Configuration for CoPP*



```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# policy-map COPP-POLICYMAP-EXAMPLE
R1(config-pmap)# class COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE
R1(config-pmap-c)# police 32000 conform-action transmit exceed-action transmit
R1(config-pmap-c-police)# violate-action transmit
R1(config-pmap-c-police)# exit
R1(config-pmap-c)# exit
R1(config-pmap)# class COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE
R1(config-pmap-c)# police 34000 conform-action transmit exceed-action transmit
R1(config-pmap-c-police)# violate-action transmit
R1(config-pmap-c-police)# exit
R1(config-pmap-c)# exit
R1(config-pmap)# class COPP-ICMP-CLASSMAP-EXAMPLE
R1(config-pmap-c)# police 8000 conform-action transmit exceed-action transmit
R1(config-pmap-c-police)# violate-action drop
R1(config-pmap-c-police)# end
R1#
```

Troubleshooting Control Plane Policing (CoPP)

Troubleshooting Policy Maps

When troubleshooting policy maps, consider the following:

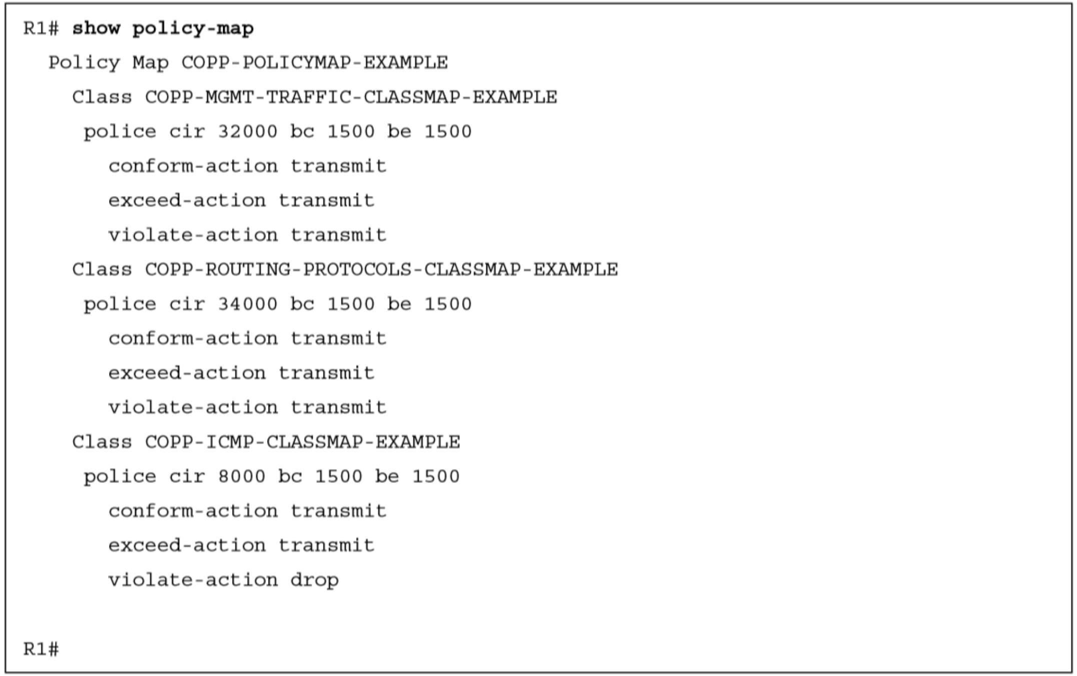
- **Order of operations -** Policy maps are processed from the top down. If there is more than one class specified, the first class listed is evaluated first, then the second, then the third, and so on down the list until you reach the default class at the end. If at any point the traffic matches any of the classes from the top down, the evaluation stops and the policy along with its actions is applied to the traffic.
- **Class map** - If the class map has been applied properly, you still need to check that the class map has been constructed properly and you need to ensure that the ACLs are being crafted correctly for the desired outcome.
- **Policy -** In the policy, make sure the correct CIR (in bits per second) has been applied or the correct RATE (in packets per second) has been applied. For conform-action, you can either transmit or drop. For the exceed-action, you can either transmit or drop.
- **Default class -** If a packet does not match any of the defined classes, it is subject to the conditions laid out in the default class.
- **Case -** Class map names are case sensitive. When specifying the class map in the policy map, double-check to make sure the name matches exactly.

Troubleshooting Control Plane Policing (CoPP)

Verify Policy Maps

To verify all configured policy maps, use the command show policy-map, as shown in Example 22-8.

Example 22-8 *Verifying Policy Maps with the **show policy-map** Command*



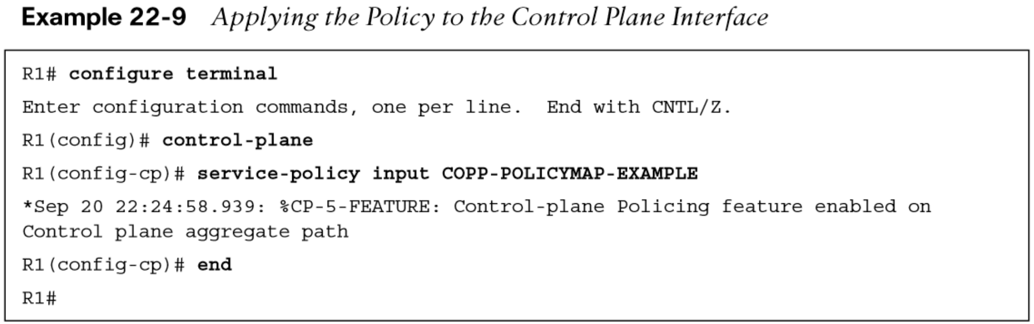
```
R1# show policy-map
  Policy Map COPP-POLICYMAP-EXAMPLE
    Class COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE
     police cir 32000 bc 1500 be 1500
       conform-action transmit
       exceed-action transmit
       violate-action transmit
    Class COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE
     police cir 34000 bc 1500 be 1500
       conform-action transmit
       exceed-action transmit
       violate-action transmit
    Class COPP-ICMP-CLASSMAP-EXAMPLE
     police cir 8000 bc 1500 be 1500
       conform-action transmit
       exceed-action transmit
       violate-action drop

R1#
```

Troubleshooting Control Plane Policing (CoPP)

Verify the Service Policy to the Control Plane

The service policy (as specified in the policy map) needs to be attached to the correct interface (see Example 22-9).



Example 22-9 *Applying the Policy to the Control Plane Interface*

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# control-plane
R1(config-cp)# service-policy input COPP-POLICYMAP-EXAMPLE
*Sep 20 22:24:58.939: %CP-5-FEATURE: Control-plane Policing feature enabled on
Control plane aggregate path
R1(config-cp)# end
R1#
```

Troubleshooting the Application of the Service Policy

When troubleshooting the application of the service policy, consider the following:

The correct interface - There is only one interface to which you can apply CoPP, the control plane interface, so this one is easy to troubleshoot: It is either applied or not applied. You can verify this with the **show policy-map control-plane** [**input** | **output**] command.

Direction - CoPP can be applied to packets entering or leaving the control plane interface. Therefore, the correct direction (input/output) needs to be specified. Direction can be verified with the output of **show policy-map control-plane**.

Case - Policy map names are case sensitive. When attaching a policy map to the control plane interface, double-check to make sure the names match exactly.

In addition, the output of the **show policy-map control-plane** command provides you with a large amount of information in one section that can assist you with your troubleshooting efforts.

CoPP Summary

When troubleshooting CoPP, consider the following steps:

Step 1. Verify that the service policy is applied and in the correct direction by using the **show policy-map control-plane** command. If it is not, fix the issue. If it is, move on to step 2.

Step 2. Verify that the policy map is configured correctly with either the **show policy-map control-plane** command or the **show policy-map** command. Check that the correct class-map, rate/cir, conform-action, and exceed-action options have been applied. Also ensure that the classes are defined in the correct order, based on top-down processing. If not, fix the issues. If everything is correct, move on to step 3.

Step 3. Verify that the class map is configured correctly by using the **show class-map** command. Verify that the correct instructions (**match-any, match-all**) are applied, along with the correct ACL, protocol, IP precedence, or IP DSCP values. If not, fix the issues. If everything is correct, move on to step 4.

Step 4. Verify that the ACLs have been configured correctly for the types of traffic you want to match by using the **show access-list** command. This requires you to verify the action (permit, deny), protocol, addresses, operators, port numbers, and anything else you can use to identify interesting traffic. If you find any issues, fix them.

IPv6 First-Hop Security

This section focuses on providing you with information that will assist you in describing RA Guard, DHCPv6 Guard, the binding table, IPv6 ND inspection/snooping, and Source Guard.

IPv6 First Hop Security

Router Advertisement (RA) Guard - RA Guard is a feature that analyzes RAs and can filter out unwanted RAs from unauthorized devices. It is possible that some RAs will be unwanted or “rogue”, meaning you wouldn’t want them on the network. You can use RA Guard to block or reject these unwanted RA messages. RA Guard requires a policy to be configured in RA Guard policy configuration mode, and RA Guard is enabled on an interface-by-interface basis by applying the policy to the interface with the **ipv6 nd raguard attach-policy** [*policy-name* [**vlan {add | except | none | remove | all}** *vlan [vlan1, vlan2, vlan3*...]]] command.

DHCPv6 Guard - DHCPv6 Guard is a feature very similar to DHCP snooping for IPv4. It is designed to ensure that rogue DHCPv6 servers are not able to hand out addresses to clients, redirect client traffic, or starve out the DHCPv6 server and cause a DoS attack. For IPv6, DHCPv6 Guard can block reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents. DHCPv6 Guard requires a policy to be configured in DHCP Guard configuration mode, and DHCPv6 Guard is enabled on an interface-by-interface basis by applying the policy to the interface with the **ipv6 dhcp guard attach-policy** [*policy-name* [**vlan {add | except | none | remove | all**} *vlan [vlan1, vlan2, vlan3*...]]] command.

IPv6 First Hop Security (Cont.)

Binding Table - The binding table is a database that lists IPv6 neighbors that are connected to a device. It contains information such as the link-layer address, the IPv4 or IPv6 address, and the prefix binding. Other IPv6 First-Hop Security features use the information in this table to prevent snooping and redirect attacks.

IPv6 Neighbor Discovery Inspection/IPv6 Snooping - IPv6 neighbor discovery inspection/snooping is a feature that learns and populates the binding table for stateless autoconfiguration addresses. It analyzes ND messages and places valid bindings in the binding table and drops all messages that do not have valid bindings. A valid ND message is one where the IPv6-to-MAC mapping can be verified.

Source Guard – IPv6 Source Guard is a Layer 2 snooping interface feature for validating the source of IPv6 traffic. If the traffic arriving on an interface is from an unknown source, IPv6 Source Guard can block it. For traffic to be from a known source, the source must be in the binding table. The source is either learned using ND inspection or IPv6 address gleaning and placed in the binding table.

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 22

Description	
Troubleshooting AAA issues	Troubleshooting class map issues related to CoPP
Using uRPF strict versus loose mode to avoid issues	Troubleshooting policy map issues related to CoPP
Using the **allow-default** option with uRPF	Output of the **show policy-map control-plane** command
Troubleshooting ACL issues related to CoPP	IPv6 First-Hop Security

Prepare for the Exam

Key Terms for Chapter 22

Terms	
AAA	class map
method list	policy map
RADIUS	ACL
TACACS+,	access control list
uRPF	RA Guard
CoPP	DHCPv6 Guard
IPv6 neighbor discovery	IPv6 snooping
Source Guard	

Prepare for the Exam

Command Reference for Chapter 22

Task	Command Syntax
Display the configuration of the local usernames and passwords on the device, the AAA commands that have been configured, and the vty line configuration	**show run | section username | aaa | line vty**
Display the authentication process in real time	**debug aaa authentication**
Display the RADIUS authentication process in real time	**debug radius authentication**
Display the local authentication process in real time	**debug aaa protocol local**
Enable AAA services on the router	**aaa new-model**
Create a username and password in the local username and password database	**username** *name* **password** *password*
Create a method list for AAA login purposes using a group of AAA servers for authentication with fallback to the local username and password database if the servers are not reachable	**aaa authentication login** *method-list-name* **group** *server-group* **local**

Prepare for the Exam

Command Reference for Chapter 22 (Cont.)

Task	Command Syntax
Apply an AAA method list to a vty or console line in line configuration mode	**login authentication** *method-list-name*
Configure uRPF on an interface in interface configuration mode	**ip verify unicast source reachablevia** {rx | any} [allow-default] [allowself-ping] [*list*]
Display all configured ACLs on the router	**show access-lists**
Display all configured class maps on the router	**show class-map**
Display all configured policy maps on the router	**show policy-map**
Verify the applied policy map, the class maps in the order in which they will be applied, the match conditions of the class maps, and the policies that are applied to the traffic that is matched	**show policy-map control-plane**

cisco