CHAPTER 34

Configuring Control Plane Policing

This chapter contains information on how to protect your Catalyst 4500 series switch using control plane policing (CoPP). The information covered in this chapter is unique to the Catalyst 4500 series switches, and it supplements the network security information and procedures in Chapter 37, "Configuring Network Security with ACLs."

Note For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

If the command is not found in the *Cisco Catalyst 4500 Command Reference*, you can locate it in the larger Cisco IOS library. Refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/ps6350/index.html

This chapter includes the following major sections:

Understanding How Control Plane Policing Works

The control plane policing (CoPP) feature increases security on the Catalyst 4500 series switch by protecting the CPU from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The Classification TCAM and QoS policers provide hardware support for CoPP. CoPP works with all supervisor engines supported by Cisco IOS Release 12.2(31)SG.

The traffic managed by the CPU is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

You can use CoPP to protect most of the CPU bound traffic and ensure routing stability, reachability and packet delivery. Most importantly, CoPP is often used to protect the CPU from the DoS attack. There is a list of pre-defined ACLs matching a selected set of Layer 2 and Layer 3 control plane packets. You can define your preferred policing parameters to each of these control packets but you cannot modify the matching criteria of these pre-defined ACLs. Following is the list of pre-defined ACLs:

Note As of Cisco IOS Release 12.2(31)SGA1, the GARP entry is no longer part of the CoPP. (though the system-cpp-garp-range entry still exists for the CPP, it is still idling and will be removed ina future release). Henceforward, GARP packets are considered regular user traffic and handled in the same way. If you want to handle GARP packets, you can still "police down" GARP packets using CPP once you define the user class for the GARP packet. (This is now possible because GARP is no longer part of the Static CAM area.) Moreover, you can manage GARP packet on a per-port basis with ACL and QoS (like any other regular traffic).

Pre-defined Named ACL	Description
system-cpp-dot1x	MacDA = 0180.C200.0003
system-cpp-bpdu-range	MacDA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MacDA = 0100.0CCC.CCCC (UDLD/DTP/VTP/Pagp)
system-cpp-sstp	MacDA = 0100.0CCC.CCCD
system-cpp-cgmp	Mac DA = 01-00-0C-DD-DD-DD
system-cpp-ospf	IP Protocol = OSPF, IPDA matches 224.0.0.0/24
system-cpp-igmp	IP Protocol = IGMP, IPDA matches 224.0.0.0/3
system-cpp-pim	IP Protocol = PIM, IPDA matches 224.0.0.0/24
system-cpp-all-systems-on-subnet	IPDA = 224.0.0.1
system-cpp-all-routers-on-subnet	IPDA = 224.0.0.2
system-cpp-ripv2	IPDA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP Protocol = UDP, L4SrcPort = 68, L4DstPort = 67
system-cpp-dhcp-sc	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 68
system-cpp-dhcp-ss	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 67

For the Data Plane and Management Plane traffic, you can define your own ACLs to match the traffic class that you want to police.

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. MQC uses class maps to define packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The control-plane global configuration command allows the CoPP service policy to be directly attached to the control plane.

The only policy-map that you can attach to the control-plane is *system-cpp-policy.* It must contain the pre-defined class-maps in the pre-defined order at the beginning of the policy map. The best way to create the system-cpp-policy policy-map is through the global macro *system-cpp*.

The system-cpp-policy contains the pre-defined class maps for the control plane traffic. The names of all system defined CoPP class maps and their matching ACLs contain the prefix “system-cpp-”. By default, no action is specified for each traffic class. You can define your own class maps matching CPU bound data plane and management plane traffic. You can add your defined class maps to the system-cpp-policy policy-map.

Caveat for Control Plane Policing

Port Security might cancel its effect for non-IP control packets.

Although Source MAC Learning on the Catalyst 4500 series switch is performed in software, learning of source MAC addresses from control packets (e.g.: IEEE BPDU/CDP/SSTP BPDU/GARP/etc) is dis-allowed. Once you configure Port Security on a port where you expect to receive a high rate of such (possibly rogue) control packets, the system generates a copy of the packet to the CPU (until the source address is learned, how Port Security is implemented), rather than forward it.

The current architecture of the Catalyst 4500 switching engine does not allow you to apply policing on the copy of packets sent to the CPU; policing can only be applied on packets that are forwarded to CPU. So, copies of packets are sent to the CPU at the rate control packets arrive and Port Security is not triggered because learning from control packets is dis-allowed. Furthermore, policing will not be applied because the packet copy, not the original, is sent to the CPU.

CoPP Default Configuration

CoPP is disabled by default.

Configuring CoPP

This section includes the following tasks:

- Configure CoPP for Control Plan Traffic, page 34-3
- Configure CoPP for Data Plane and Management Plan Traffic, page 34-5

Configure CoPP for Control Plan Traffic

To configure CoPP for Control Plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch# config terminal`	Enters global configuration mode.
Step 2	`Switch(config)# qos`	(Optional) Enables QoS globally.
Step 3	`Switch(config)#` `macro global apply system-cpp`	(Optional) Creates the system-cpp-policy policy-map and attaches it to the control-plane.

	Command	Purpose
Step 4	Switch(config)# **policy-map system-cpp-policy** Switch(config-pmap)# **class** {**system-cpp-dot1x** | **system-cpp-bpdu-range** | **system-cpp-cdp** | **service** | **system-cpp-sstp** | **system-cpp-cgmp** | **system-cpp-ospf** | **system-cpp-igmp** | **system-cpp-pim** | **system-cpp-all-systems-on-subnet** | **system-cpp-all-routers-on-subnet** | **system-cpp-ripv2** | **system-cpp-ip-mcast-linklocal** | **system-cpp-dhcp-cs** | **system-cpp-dhcp-sc** | **system-cpp-dhcp-ss**} Switch(config-pmap-c)# **police** [**aggregate** *name*] *rate burst* [**conform-action** {**drop** | **transmit**}] [{**exceed-action** {**drop** | **transmit**}}]}	Associates actions to one or multiple system defined control plane traffic in the service policy map. Repeat this step if necessary.
Step 5	Switch# **show policy-map system-cpp-policy**	(Optional) Verifies the configuration

The following example shows how to police CDP packets:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-cdp
Switch(config-pmap-c)# police 32000 1000 conform-action transmit exceed-action drop
Switch(config-pmap-c)# end
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
*    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop *
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
Switch#
```

Configure CoPP for Data Plane and Management Plan Traffic

To configure CoPP for Data Plane and Management Plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch(config)# qos`	(Optional) Enables QoS globally.
Step 2	`Switch(config)#` `macro global apply system-cpp`	(Optional) Attaches the system-cpp-policy policy-map to the control-plane.

Step 3	`Switch(config)# {ip | mac} access-list extended {access-list-name}`
	
	`For an ip access list, issue`
	`Switch(config-ext-nacl)#{permit|deny} {protocol} source {source-wildcard} destination {destination-wildcard}`
	
	`For a mac access list, issue`
	`Switch(config-ext-macl)#{permit|deny} source {source-wildcard} destination {destination-wildcard} [protocol-family]`
	
	OR
	
	`Switch(config)# access-list {access-list-name} {permit | deny} {type-code wild-mask | address mask}`
		Defines ACLs to match traffic:
		• **permit** - sets the conditions under which a packet passes a named ACL
		• **deny** - sets the conditions under which a packet does not pass a name ACL
		Note You must configure ACLs in most cases to identify the important or unimportant traffic.
		• **type-code** - 16-bit hexadecimal number written with a leading 0x; for example, 0x6000. Specify either a Link Service Access Point (LSAP) type code for 802-encapsulated packets or a SNAP type code for SNAP-encapsulated packets. (LSAP, sometimes called SAP, refers to the type codes found in the DSAP and SSAP fields of the 802 header.)
		• **wild-mask** - 16-bit hexadecimal number whose ones bits correspond to bits in the type-code argument. The wild-mask indicates which bits in the type-code argument should be ignored when making a comparison. (A mask for a DSAP/SSAP pair should always be 0x0101 because these two bits are used for purposes other than identifying the SAP code.)
		• **address** - 48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. This field is used for filtering by vendor code.
		• **mask** - 48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. The ones bits in mask are the bits to be ignored in address. This field is used for filtering by vendor code.

	Command	Purpose

Step 4	`Switch(config)# class-map {traffic-class-name}`
	
	`Switch(config-cmap)# match access-group {access-list-number | name {access-list-name}}`
		Defines the packet classification criteria. Use the **match** statements to identify the traffic associated with the class.

Step 5	`Switch(config-cmap)# exit`	Returns to global configuration mode.

Step 6	`Switch(config)# policy-map system-cpp-policy`
	
	`Switch(config-pmap)# class <class-map-name>`
	
	`Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop | transmit}] [{exceed-action {drop | transmit}}]`
		Adds the traffic classes to the CoPP policy-map. Uses the **police** statement to associate actions to the traffic class.

Step 7	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show policy-map system-cpp-policy`	Verifies your entries.

The following example shows how to configure trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specific rate (this example assumes the global qos is enabled and the system-cpp-policy policy-map has been created):

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp

! Allow 10.1.1.1 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet

! Allow 10.1.1.2 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet

! Rate limit all other Telnet traffic.
Switch(config)# access-list 140 permit tcp any any eq telnet

! Define class-map "telnet-class."
Switch(config)# class-map telnet-class
Switch(config-cmap)# match access-group 140
Switch(config-cmap)# exit

! Add the class-map "telnet-class" to "system-cpp-policy" and define ! the proper action
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class telnet-class
Switch(config-pmap-c)# police 80000 1000 conform transmit exceed drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

! Verify the above configuration steps
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop
    Class system-cpp-sstp
```

```
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
*    Class telnet-class
      police 8000 bps 1000 byte conform-action drop exceed-action drop
```

CoPP Configuration Guidelines and Restrictions

When configuring CoPP, follow these guidelines and restrictions:

- Only ingress CoPP is supported. So only **input** keyword is supported in control-plane related CLIs.
- Use the system defined class maps for policing control plane traffic.
- Control plane traffic can be policed only using CoPP. Traffic cannot be policed at the input interface or VLAN even though a policy-map containing the control-plane traffic is accepted when the policy-map is attached to an interface or VLAN.
- System-defined class maps cannot be used in policy-maps for regular QoS.
- Use ACLs and class-maps to identify data plane and management plane traffic that are handled by CPU. User-defined class maps should be added to the **system-cpp-policy** policy-map for CoPP.
- The policy-map named **system-cpp-policy** is dedicated for CoPP. Once attached to the **control-plane**, it cannot be detached.
- The default **system-cpp-policy** map does not define actions for the system-defined class maps, which means **no policing**.
- The only action supported in **system-cpp-policy** policy-map is **police**.
- Do not use the **log** keyword in the CoPP policy ACLs.
- Both MAC and IP ACLs can be used to define data plane and management plane traffic classes. But if a packet also matches a pre-defined ACL for the control plane traffic, the **police** action (or no police action) of the control plane class will be taken as the control plane classes appear above user-defined classes in the service policy. This is the same MQC semantic.
- The exceeding action **policed-dscp-transmit** is not supported for CoPP.
- CoPP is not enabled unless the global QoS is enabled and **police** action is specified.

Monitoring CoPP

You can enter the **show policy-map control-plane** command for developing site-specific policies, monitoring statistics for the control plane policy, and troubleshooting CoPP. This command displays dynamic information about the actual policy applied including rate information and the number of bytes (and packets) that conformed or exceeded the configured policies both in hardware and in software.

The output of the **show policy-map control-plane** command is as follows:

```
Switch# show policy-map control-plane

Control Plane

Service-policy input: system-cpp-policy

    Class-map: system-cpp-dot1x (match-all)
      0 packets
      Match: access-group name system-cpp-dot1x

    Class-map: system-cpp-bpdu-range (match-all)
      0 packets
      Match: access-group name system-cpp-bpdu-range

*    Class-map: system-cpp-cdp (match-all)
      160 packets
      Match: access-group name system-cpp-cdp
**      police: Per-interface
        Conform: 22960 bytes Exceed: 0 bytes
*
    Class-map: system-cpp-sstp (match-all)
      0 packets
      Match: access-group name system-cpp-sstp

    Class-map: system-cpp-cgmp (match-all)
      0 packets
      Match: access-group name system-cpp-cgmp

    Class-map: system-cpp-ospf (match-all)
      0 packets
      Match: access-group name system-cpp-ospf

    Class-map: system-cpp-igmp (match-all)
      0 packets
      Match: access-group name system-cpp-igmp

    Class-map: system-cpp-pim (match-all)
      0 packets
      Match: access-group name system-cpp-pim

    Class-map: system-cpp-all-systems-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-systems-on-subnet

    Class-map: system-cpp-all-routers-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-routers-on-subnet

    Class-map: system-cpp-ripv2 (match-all)
      0 packets
      Match: access-group name system-cpp-ripv2

    Class-map: system-cpp-ip-mcast-linklocal (match-all)
      0 packets
      Match: access-group name system-cpp-ip-mcast-linklocal

    Class-map: system-cpp-dhcp-cs (match-all)
      83 packets
      Match: access-group name system-cpp-dhcp-cs

    Class-map: system-cpp-dhcp-sc (match-all)
      0 packets
```

```
      Match: access-group name system-cpp-dhcp-sc

    Class-map: system-cpp-dhcp-ss (match-all)
      0 packets
      Match: access-group name system-cpp-dhcp-ss

*    Class-map: telnet-class (match-all)
      0 packets
      Match: access-group 140
**      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes*

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
Switch#
```

To clear the counters on the control-plane, enter the **clear control-plane *** command:

```
Switch# clear control-plane *
Switch#
```

To display all the CoPP access list information, enter the **show access-lists** command:

```
Switch# show access-lists
Extended IP access list system-cpp-all-routers-on-subnet
10 permit ip any host 224.0.0.2
Extended IP access list system-cpp-all-systems-on-subnet
10 permit ip any host 224.0.0.1
Extended IP access list system-cpp-dhcp-cs
10 permit udp any eq bootpc any eq bootps Extended IP access list
system-cpp-dhcp-sc
10 permit udp any eq bootps any eq bootpc Extended IP access list
system-cpp-dhcp-ss
10 permit udp any eq bootps any eq bootps Extended IP access list
system-cpp-igmp
10 permit igmp any 224.0.0.0 31.255.255.255 Extended IP access list
system-cpp-ip-mcast-linklocal
10 permit ip any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ospf
10 permit ospf any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-pim
10 permit pim any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ripv2
10 permit ip any host 224.0.0.9
Extended MAC access list system-cpp-bpdu-range
permit any 0180.c200.0000 0000.0000.000f Extended MAC access list
system-cpp-cdp
permit any host 0100.0ccc.cccc
Extended MAC access list system-cpp-cgmp
permit any host 0100.0cdd.dddd
Extended MAC access list system-cpp-dot1x
permit any host 0180.c200.0003
system-cpp-sstp
permit any host 0100.0ccc.cccd
```

To display one CoPP access list, enter the **show access-lists system-cpp-cdp** command:

```
Switch# show access-list system-cpp-cdp
Extended MAC access list system-cpp-cdp
permit any host 0100.0ccc.cccc
Switch#
```