Infrastructure Protection on Cisco Catalyst 6500 and 4500 Series Switches

A key element in an organization's overall security posture is the security of the network infrastructure. The network infrastructure is the foundation built with routers, switches, and other equipment that provide the fundamental network services that keep a network running. The infrastructure is often the target of denial of service (DoS) and other attacks that can directly or indirectly disrupt the network operation. In order to ensure the availability of the network, it is critical to implement the security tools and best practices that help protect each network element, and the infrastructure as a whole.

This document describes the tools that are currently available to protect Cisco Catalyst 6500 and 4500 Series switches from direct attacks. These tools can also help prevent accidental misconfiguration, which could present a risk to the infrastructure. This document also provides deployment guidelines to help implement these tools as an integrated security solution, rather than as isolated elements.

The first portion of this document provides an overview of the basic tools and technologies that are available on Catalyst switches for network device hardening. Subsequent sections provide a closer look at more advanced features that require additional explanation. Later sections provide deployment guidelines that describe how to implement these features in an integrated way, followed by additional reference information.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Basic Tools and Techniques for Device Hardening

Device hardening ensures the security of a device by controlling access to the device, disabling services that are not needed, and by establishing mechanisms to help control the use of system resources.

This section presents a compilation of best practices for device hardening on Cisco Catalyst 6500 and 4500 Series switches. Most of these best practices are based on tools and techniques that have been available for quite some time, and which can be considered reliable.

The following is a list of the recommended hardening best practices for Catalyst switches:

- Disabling Unneeded Services, page 5
- Controlling Switch Access, page 6
- Access Control Lists, page 7
- Locking Down Unused Ports, page 16

Disabling Unneeded Services

To facilitate deployment, Cisco Catalyst switches arrive with many services that are considered appropriate for most network environments already enabled. However, because not all networks have the same requirements, some of these services might not be needed and can be disabled. Disabling unneeded services has two benefits. It helps preserve system resources, and eliminates the potential of security exploits on the disabled services.

Disabling unneeded services becomes especially important for services that are known to be prone to being used for malicious purposes. Some services that are enabled by default can be used by attackers to obtain network and user information, bypass security controls, and even generate DoS attacks. A directed broadcast is a good example of a default service found in some switches and routers that could be used for DoS attacks. An IP-directed broadcast packet is an IP packet with a destination address that is a valid broadcast address for an IP subnet. When a directed broadcast packet reaches a switch that is directly connected to its destination subnet, and if the switch is configured to do so, that packet is *exploded* as a broadcast on the destination subnet. In this way, a single directed broadcast packet can reach multiple destinations, and can be used by programs such as smurf, to amplify the effects of an attack. A smurf attack, named after its exploit program, is a DoS attack that uses spoofed broadcast ping messages to flood a target system.

After you identify the services that are needed, it is a good practice to enable them only as they are needed. Most network devices allow a selective configuration of services. Some services can be activated either for the entire system, globally, or per component, typically at a module or interface level. Services known to be prone to abuse should be deployed selectively whenever this feature is available. For example, CDP is an available service on Cisco platforms that facilitates the administration and troubleshooting of network devices. CDP is globally enabled by default, but in most cases there is no real need to run this service on every interface. Running CDP on the wrong interface can give an attacker the chance to obtain sensitive device and network information. For this reason, CDP should be enabled only on the necessary interfaces.

Finally, disabling services is an activity that requires some planning. Prior to disabling any services, determine if there are any dependencies (some services depend on other services to work correctly). This helps avoid cases where one service unexpectedly breaks because another service was disabled.

The section, Unneeded Services, page 91 provides a list of available services on Catalyst switches that might not be needed and can therefore be disabled. This section also includes the commands to disable unneeded services, for both Cisco IOS and Catalyst OS.

Controlling Switch Access

Like other network infrastructure devices, switches often provide more access mechanisms than you might realize, from console to remote sessions based on protocols such as Telnet and SSH. Some of these mechanisms are enabled by default on some platforms, while other services need to be specifically configured. Establishing controls on these access mechanisms is fundamental to prevent unauthorized access and device misuse. Anyone who gains access to a switch can obtain critical information about the network, reconfigure the device, and even take the device out of service. Therefore, each switch in the infrastructure should be carefully configured to secure all the access mechanisms enabled on the system.

The following is a collection of best practices to help secure access to Catalyst switches:

- Secure local password management—Access to switches is controlled primarily by the use of user names and passwords. The best way to handle most passwords is to maintain them on a centralized system such as a TACACS+ or RADIUS authentication server. Unfortunately, it is not always possible to handle all passwords on an authentication server, and often switches require the configuration of local passwords for privilege access. It is also common to find special-use user names, secret keys, and other password information in their configuration files. As a best practice, all passwords and secrets contained in configuration files must be encrypted. Default passwords are another concern. Switches might be delivered from the factory with default user names and passwords. Users should be forced to change these default passwords, and to use non-trivial passwords.
- Controlling interactive access—Switches can be accessed through a variety of interactive mechanisms, including telnet, rlogin, and SSH. These access mechanisms always involve sessions or lines that need to be properly protected. To that end, lines that are not going to be used need to be disabled, access should be configured only for the protocols actually needed, and all lines should be configured with authentication.

- Warning banners—Login banners should be used not only to dissuade possible attackers but also because in some jurisdictions they are required by law. Banners must give notice that any unauthorized use of the system is unlawful, and can be subject to civil or criminal penalties. Also important, banners should not reveal any platform or configuration-related information.
- Implementing role-based access—Role-based access allows administrators to define multiple users and groups, each of which can be associated with a list of permitted or denied commands. This feature is especially useful in environments where switches are administered by multiple groups of people with different access requirements.
- Securing web-based GUI access—Catalyst switches can be configured and monitored using a convenient web-based user interface. Whenever it is available, the web-based GUI should be configured with HTTPS rather than HTTP. HTTP does not provide encryption for client connections, which leaves communication between clients and servers vulnerable to interception and other attacks. Another good practice is to enable authentication for HTTP and HTTPS connections.
- Use secure access protocols (SSH) instead of clear text protocols (telnet)—SSH is a protocol that provides secure remote access, allowing you to issue commands remotely, and to transfer files. SSH implements strong authentication and encryption, which make it a better option over insecure protocols such as rlogin and telnet.
- Controlling SNMP access—Whenever available, use SNMPv3 rather than earlier versions of the protocol. SNMP versions 1 and 2c are weak in security. In these earlier versions of SNMP, access to MIB objects is primarily controlled by the use of community strings, but neither version provides authentication or encryption. SNMP version 3 incorporates security features, such as authentication, identity, and access control.

Access Control, page 98 provides the configuration guidelines to implement these best practices in Cisco IOS and Catalyst OS.

Access Control Lists

Catalyst 6500 and 4500 Series switches support several classes of Layer 2 and Layer 3 Access Control Lists (ACLs) that can be used to shield the infrastructure from DoS, source address spoofing and other attacks. The Layer 2 and Layer 3 ACLs available can help protect the infrastructure by filtering traffic destined to the management and control planes, and by blocking illegitimate packets, such as those containing private addresses or spoofed IP addresses.

The following types of ACLs are available:

- Router ACL, page 8
- VLAN ACL(VACL), page 8
- Port ACL (PACL), page 12
- Unicast MAC Address Filtering (MAC Address-Based Traffic Blocking), page 14
- IP Permit Lists, page 15
- Access-Class, page 15

Router ACL

Router ACLs, also known as Cisco IOS ACLs, are the standard and extended IP ACLs available on Cisco IOS Software. These ACLs are applied to Layer 3 interfaces and to VLAN interfaces, and affect only routed traffic. In addition, router ACLs can be applied in a specific inbound or outbound direction. As with IOS routers, standard IP access lists are based on source addresses, while extended IP ACLs can be based on source and destination addresses, and optionally on protocol type information.

Catalyst 6500 Series switches require the following hardware in order to run Router ACLs:

- Supervisor Engine 1 with a Policy Feature Card (PFC) and MSFC or MSFC2
- Supervisor Engine 2 with a PFC2 and MSFC2
- Supervisor Engine 720 with a PFC3A/PFC3B/PFC3BXL and MSFC3
- Supervisor Engine 32 with a PFC3B and MSFC2A

Catalyst 4500 Series switches require Cisco IOS Software in order to run Router ACLs.

The following example shows a router ACL configured to filter IP packets with source IP addresses falling within the private address space (as defined in RFC 1918):

```
!--- Filter RFC 1918 space.
access-list 110 deny ip 10.0.0.0 0.255.255.255 any
access-list 110 deny ip 172.16.0.0 0.15.255.255 any
access-list 110 deny ip 192.168.0.0 0.0.255.255 any
!
access-list 110 permit ip any any
!
```

Note Catalyst 6500 Series switches equipped with PFC3 support Optimized ACL Logging (OAL), a feature that provides hardware support for ACL logging. Unless you configure OAL, packets that require logging are processed entirely in software on the MSFC. OAL permits or drops packets in hardware on the PFC3 and uses an optimized routine to send information to the MSFC3 to generate the logging messages. For more information on OAL, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/acl.htm

For more information about configuring Cisco IOS ACLs, refer to "Traffic Filtering and Firewalls" in the *Cisco IOS Security Configuration Guide*, Release 12.2, at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/index.htm

More information on ACLs on the Catalyst 6500 is available at the following URL:

http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/tech/65acl_wp.pdf

For more information on ACLs on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/secure.htm

VLAN ACL(VACL)

VLAN ACLs (VACLs), also known as VLAN maps, are access lists that are applied to VLANs. Unlike router ACLs that are configured on Layer 3 interfaces and that affect routed packets only, VACLs affect all packets, bridged and routed, and can be applied to any VLAN. In addition, VACLs are not defined by direction (input or output) as router ACLs are. After a VACL is configured on a switch, the filtering rules apply to all packets that are routed in to or out of the associated VLAN or are bridged within the VLAN.

In Catalyst 4500 Series switches, VACLs are supported only on systems running Cisco IOS. In this platform, VACLs can be configured for IP and MAC-layer traffic. In the case of IP, the VACLs can be configured to map Layer 3 address information. All other non-IP protocols can be controlled with MAC-based ACLs, which use MAC address and Ethertype information to match packets.

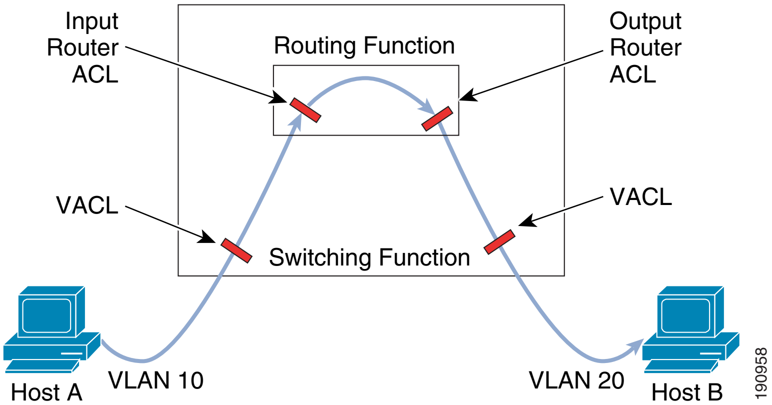
Catalyst 6500 Series switches support VACLs on both Catalyst OS and Cisco IOS. In addition, VACLs support IP and IPX-based VACLs, and MAC-based ACLs. IP and IPX traffic can be controlled by mapping Layer 3 address information, while other non-IP traffic can be filtered based on MAC address and Ethertype information using MAC-based ACLs.

Catalyst 6500 Series switches require the following hardware in order to support VACLs:

- Supervisor Engine 1 with a PFC
- Supervisor Engine 2 with a PFC2
- Supervisor Engine 720 with a PFC3A/PFC3B/PFC3BXL
- Supervisor Engine 32 with a PFC3B

VACLs can be used in conjunction with router ACLs. Figure 1 shows the logical relationship between VACLs and router ACLs. When used together with router ACLs, a VACL is first applied to incoming packets on the corresponding ingress VLAN. If the packet is Layer 3 forwarded and is permitted by the VACL, it is filtered by the Cisco IOS ACL on the same VLAN. The same process happens in reverse in the egress direction.

Figure 1 Logical Relationships Between VACLs and Router ACLs



Configuring VACLs in Catalyst OS

To configure a VACL in Catalyst OS software, perform the following steps:

Step 1 Create VACL and add entries using the **set security acl** command.

Step 2 Commit the VACL to NVRAM using the **commit** command.

Step 3 Map VACL to a VLAN using the **set security acl map** command.

This example shows an IP-based VACL called IPACL1 and that allows traffic from source address 172.20.53.4. This VACL is then mapped to VLAN 10:

```
Console> (enable) set security acl ip IPACL1 permit host 172.20.53.4 0.0.0.0
IPACL1 editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
Console> (enable) commit security acl all
ACL commit in progress.
ACL IPACL1 is committed to hardware.
Console> (enable)
Console> (enable) set security acl map IPACL1 10
ACL IPACL1 mapped to vlan 10
Console> (enable)
```

This example shows a MAC-based VACL called MACACL1 and that permits all traffic from 8-2-3-4-7-A. This VACL is then mapped to VLAN 20:

```
Console> (enable) set security acl mac MACACL1 permit host 8-2-3-4-7-A any
MACACL1 editbuffer modified. Use 'commit' command to apply changes.
Console> (enable)
Console> (enable) commit security acl all
ACL commit in progress.
ACL IPACL1 is committed to hardware.
Console> (enable)
Console> (enable) set security acl map MACACL1 20
ACL IPACL1 mapped to vlan 20
Console> (enable)
```

For more information on how to configure VACLs on Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/acc_list.htm

Configuring VACLs in Cisco IOS

To configure a VACL in Cisco IOS software, perform the following steps:

Step 1 Create a VLAN map using the **vlan access-map** command.

Step 2 Set an action for the VLAN map (drop, forward) using the **action VLAN map** command.

Step 3 Define a match criteria based on either an IP or MAC-based ACL using the **match VLAN map** command.

Step 4 Apply the VLAN map to one or more VLANs using the **vlan filter** command.

In the following example, the VLAN map is configured to drop IP packets and to forward MAC packets by default. By applying standard ACL 101 and the extended named access lists igmp-match and tcp-match, the VLAN map is configured to do the following:

- Forward all UDP packets
- Drop all IGMP packets
- Forward all TCP packets
- Drop all other IP packets
- Forward all non-IP packets

```
!--- Initially create the IP ACLs used for the match criteria
```

```
Switch(config)# access-list 101 permit udp any any
Switch(config)# ip access-list extended igmp-match
Switch(config-ext-nacl)# permit igmp any any
Switch(config)# ip access-list extended tcp-match
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
!
!--- Create VLAN map and define actions per map instance
Switch(config)# vlan access-map drop-ip-default 10
Switch(config-access-map)# match ip address 101
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 20
Switch(config-access-map)# match ip address igmp-match
Switch(config-access-map)# action drop
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 30
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
!
!--- Apply VLAN map to actual VLAN
Switch(config)# vlan filter drop-ip-default vlan-list 110
```

In this next example, the VLAN map is configured to drop all packets (IP and non-IP). By applying **access lists tcp-match** and **good-hosts**, the VLAN map is configured to do the following:

- Forward all TCP packets
- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Drop all other IP packets
- Drop all other MAC packets

```
Switch(config)# ip access-list extended tcp-match
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
Switch(config)# mac access-list extended good-hosts
Switch(config-ext-macl)# permit host 000.0c00.0111 any
Switch(config-ext-macl)# permit host 000.0c00.0211 any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-all-default 10
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-all-default 20
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
Switch(config)# vlan filter drop-all-default vlan-list 110
```

For more information on how to configure VACLs on Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/vacl.htm

For more information on how to configure VACLs on Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/secure.htm

Port ACL (PACL)

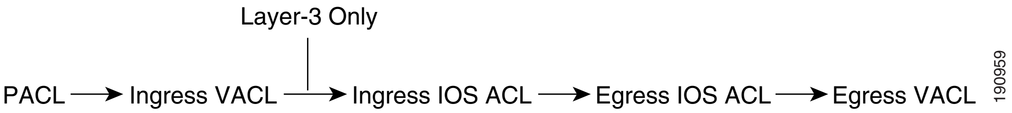
Port ACLs (PACLs) are IP based and MAC based access control lists applied to Layer 2 physical ports on a switch. PACLs are available on Catalyst 6500 Series switches running Catalyst OS and Catalyst 4500 Series switches running Cisco IOS.

Note PACLs are available only on Supervisor Engine 720 with PFC3A/PFC3B/PFC3BXL and Supervisor Engine 32 with PFC3B. Only input PACLs are supported on Catalyst 6500 Series switches equipped with those supervisor engines.

With PACLs, IP traffic is filtered with either IP standard or extended ACLs, while non-IP traffic is controlled with MAC based ACLs. Standard IP ACLs filter IP traffic solely based on source addresses. Extended IP ACLs filter traffic by source and destination addresses, and optionally by protocol type. MAC based ACLs filter non-IP traffic based on source and destination MAC addresses, and optionally by protocol type.

By configuring an IP-based ACL and a MAC-based ACL on the same physical port, it is possible to filter IP and non-IP traffic simultaneously on a Layer 2 physical port. However, only one IP ACL and one MAC ACL can be applied on a physical port at the same time. If a new IP ACL or MAC ACL is applied to a port that has already been configured with an IP ACL or MAC ACL, the new ACL will replace the previously configured one. Figure 2 shows the logical relationships between ACL types.

Figure 2 Logical Relationships Between ACL Types



PACLs can be used in conjunction with VACLs and router ACLs. There are three modes of operation that define the way PACLs interact with other ACLs, and which can be configured on a per-port basis:

- Prefer Port/Port-based mode—If a PACL is configured on a Layer 2 port the PACL takes effect and overrides other ACLs (Router ACL and VACL). If no PACL feature is configured on the Layer 2 port, other ACLs applicable to the interface are merged and applied on the interface. This is the default mode on Catalyst 4500 Series switches.
- Prefer VLAN/VLAN-based mode—VLAN-based ACLs (Router ACL and VACL) take effect on the port and override the PACL. The PACL only takes effect if no VLAN-based ACLs are applied to the Layer 2 interface. This is the default mode on Catalyst 6500 Series switches.
- Merge—With this mode, the ingress PACL, VACL, and Cisco IOS ACL are merged together following the logical serial model shown in Figure 2.

Note Supervisor Engines III and Supervisor Engine IV running on a Catalyst 4500 Series switch support both input and output PACLs on an interface.

The relationship between PACLs, Router ACLs and VACLs depends on the configuration of the PACL mode as summarized in Table 1.

Table 1 Interaction Between PACLs, VACLs, and Router ACLs

ACL Type	PACL Mode		
	Prefer Port	Prefer VLAN	Merge
Input router ACL	PACL applied	Input router ACL applied	PACL, Input router ACL (merged) applied in order (ingress)
VACL	PACL applied	VACL applied	PACL, VACL (merged) applied in order (ingress)
VACL + Input router ACL	PACL applied	VACL + Input router ACL applied	PACL, VACL, Input router ACL (merged) applied in order (ingress)

Configuring PACLs in Catalyst OS

To configure PACLs in Catalyst OS, perform the following steps:

Step 1 Specify the PACL mode using the **set port security-acl** command:

```
Console> (enable) set port security-acl mod/ports.. [port-based | vlan-based | merge]
```

Step 2 Map the PACL to ports or to a VLAN using the **set security acl map** command:

```
Console> (enable) set security acl map acl_name [mod/ports | vlans]
```

This example shows how to map a PACL to a port when the port is in VLAN-based mode:

```
Console> (enable) set port security-acl 3/1 vlan-based
ACL interface is set to vlan-based mode for port(s) 3/1.
Console> (enable) set security acl map ipacl1 3/1
Port 3/1 is set to vlan-based mode, config is saved in Nvram.
Config will be applied when the port is set to port-based/merge.
Console> (enable) set port security-acl 3/1 port-based
Warning: Vlan-based ACL features will be disabled on port(s) 3/1.
ACL interface is set to port-based mode for port(s) 3/1.
```

For more information on how to configure PACLs on Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/acc_list.htm#wp1203610

Configuring PACLs in Cisco IOS

To configure a PACL in Cisco IOS software, perform the following steps:

Step 1 Create the standard or extended IP ACLs or named MAC extended ACLs to be applied using the **ip access-list** or **mac access-list** commands:

Step 2 Set the mode in which the PACL will interact with other ACLS using the **access-group mode interface** command:

```
Switch(config-if)# access-group mode {prefer {port | vlan} | merge}
```

Step 3 Apply the previously defined ACLs to the desired Layer 2 port using the **access-group interface** configuration command:

```
Switch(config-if)# {ip | mac access-group {name | number| in | out}
```

In the following example an IP ACL and a MAC ACL are defined and applied to interface FastEthernet 6/1. The IP ACL, called simple-ip-acl, is configured to permit all TCP traffic and implicitly deny all other IP traffic. The MAC ACL, simple-mac-acl, is configured to permit source host 000.000.011 to any destination host. Finally, PACL mode is set prefer port:

```
Switch(config)# ip access-list extended simple-ip-acl
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# end
Switch(config)# mac access-list extended simple-mac-acl
Switch(config-ext-macl)# permit host 000.000.011 any
Switch(config-ext-macl)# end
Switch(config)# interface FastEthernet 6/1
Switch(config-if)# access-group mode prefer port
Switch(config-if)# ip access-group simple-ip-acl in
Switch(config-if)# mac access-group simple-mac-acl in
```

For more information on how to configure PACLs on Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/secure.htm#wp1071428

Unicast MAC Address Filtering (MAC Address-Based Traffic Blocking)

Unicast MAC address filtering, also known as MAC address-based traffic blocking, is an available feature on Catalyst 6500 and 4500 Series switches running Cisco IOS software that provides alternative filtering based on MAC addresses. Unicast MAC address filtering allows the definition of filter rules that are run in hardware and that block all unicast traffic to and from a particular MAC address in a given VLAN. This feature can be used in conjunction with other VLAN-based ACLs, but a unicast MAC address filter takes precedence.

To block all unicast traffic to and from a particular MAC address in a specific VLAN, use the **mac-address-table** command:

```
Switch(config)# mac-address-table static mac_address vlan vlan_ID drop
```

This example shows how to block all unicast traffic to and from MAC address 0050.3e8d.6400 in VLAN 12:

```
Switch# configure terminal
Switch(config)# mac-address-table static 0050.3e8d.6400 vlan 12 drop
```

For more information on how to configure MAC address-based traffic blocking on Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/secure.htm#wp1074871

For more information on how to configure unicast MAC address filtering on Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/secure.htm#wp1066708

IP Permit Lists

The IP permit list is an available feature in Catalyst OS and supported on Catalyst 6500 and 4500 Series switches. An IP permit list is an access-list that prevents inbound Telnet, SNMP, and SSH access to the switch from unauthorized source IP addresses. All other TCP/IP services (such as IP traceroute and IP ping) continue to work normally when you enable the IP permit list. Outbound Telnet, Trivial File Transfer Protocol (TFTP), and other IP-based services are unaffected by the IP permit list. IP permit lists can be replaced with IOS ACLs or VACLs.

To configure IP permit lists, perform the following steps:

Step 1 Define IP permit list using the **set ip permit** command:

```
Console> (enable) # set ip permit ip_address [mask] [all | snmp | telnet | ssh]
```

Step 2 Activate the IP permit list using the **set ip permit enable** command:

```
Console> (enable) # set ip permit enable [telnet | snmp | ssh]
```

In this example SNMP access is granted to host 172.16.52.32, and SSH connections will be allowed only from the172.16.52.0/24 network.

```
Console> (enable) set ip permit 172.16.52.32 255.255.255.255 snmp
172.16.52.32 with mask 255.255.255.255 added to Snmp permit list.
Console> (enable) set ip permit 172.16.52.0 255.255.255.0 ssh
172.16.52.0 with mask 255.255.255.0 added to Ssh permit list.
Console> (enable) set ip permit enable
IP permit list enabled.
```

For more information on how to configure IP permit lists on Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/ip_perm.htm

For more information on how to configure IP permit lists on Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/configur/ip_perm.htm

Access-Class

Access-class is an available feature in Cisco IOS software that is supported on Catalyst 6500 and 4500 Series switches. Similar to IP permit lists, access-class prevents Telnet and SSH access to the switch from unauthorized IP addresses. With this feature, a standard or extended IP ACL is configured with a list of addresses to be permitted or blocked, then the access-class maps the ACL to one or more VTYs.

To configure an access-class, perform the following steps:

Step 1 Create the standard or extended IP ACL defining the IP addresses or subnets to be blocked or permitted using the **access-list** command.

Step 2 Access the VTYs to which you want to control access. Using the **access-class line** command, associate the previously configured IP ACL:

```
Switch(config)# line vty <0-15> [<0-15>]
Switch(config-line)# access-class access-list-number in
```

In the following example, Telnet and SSH access to VTYs 0 to 4 is restricted to host 172.16.1.1 only.

```
Switch(config)# access-list 10 permit 172.16.1.1
Switch(config)# line vty 0-4
Switch(config-line)# access-class 10 in
```

For more information on how to configure an access-class, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/1rfip1.htm#wp1017389

Locking Down Unused Ports

By default, all Ethernet ports on Catalyst switches running Catalyst OS are set to VLAN 1. In addition, by default, many control protocols such as CDP, PAgP and VTP, use VLAN 1 to transmit and receive packets across the network topology. Leaving all unused ports configured in VLAN 1 opens the chance for unauthorized access. Anyone connecting to an unused port would gain access to the entire VLAN 1 and all the resources in it. For this reason, all unused ports should be disabled and put in an unused, isolated VLAN. Disabling unused ports and placing them in an isolated VLAN helps contain any unauthorized access attempts from ports not in use.

The same recommendations also apply to Catalyst switches running Cisco IOS software. However, there are some differences in the default settings that should be noted. First, by default, in Cisco IOS all ports are disabled on Catalyst 6500s Series switches and enabled on Catalyst 4500 Series switches. Second, by default, all ports are set as routed interfaces, and in consequence they are not associated to a default VLAN. However, as soon as a port is configured as a Layer 2 switched interface, the port is automatically assigned to default VLAN 1.

In Catalyst OS, a port can be disabled using the **set port disable** command:

```
Console> (enable) set port disable mod/port
```

This example shows how to disable a port using the **set port disable** command:

```
Console> (enable) set port disable 5/10
Port 5/10 disabled.
Console> (enable)
```

For more information on the set port disable command for the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_po_r.htm#wp1468799

For more information on the set port disable command for the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_l_q.htm#wp1025937

In Catalyst OS, a port can be set to an unused VLAN using the **set vlan** command:

```
Console> (enable) set vlan vlan mod/port
```

This example shows how the unused port is set to unused VLAN 560:

```
Console> (enable) set vlan 560 4/10
VLAN 560 modified.
VLAN 1 modified.
VLAN  Mod/Ports
---- -----------------------
560   4/10
```

For more information on the **set vlan** command for the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_v.htm#wp1058935

For more information on the **set vlan** command for the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_s_z.htm#wp1052100

By default, in Cisco IOS all interfaces are disabled on Catalyst 6500 Series switches, and enabled on Catalyst 4500 Series switches. A previously enabled interface can be disabled using the **shutdown interface** command:

```
Switch(config)# interface type slot/port
Switch(config-if)# shutdown
```

This example shows how to disable an interface using the **shutdown interface** command:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# shutdown
Switch(config-if)# end
```

For more information on the **shutdown** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_r/irfshoip.htm#wp1018004

In Cisco IOS, by default, all ports are configured as routed interfaces, and because they are not Layer 2 ports they are not associated to a default VLAN. However, as soon as a port is configured as a Layer 2 switched interface using the **switchport** command, the port is automatically set to default VLAN 1. To change the default VLAN, use the **switchport access vlan** command:

```
Switch(config-if)# switchport access vlan vlan_ID
```

This example shows how the unused port is set to unused VLAN 560:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# switchport access vlan 560
Switch(config-if)#
```

For more information on the **switchport access vlan** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1088860

Note Catalyst 6500 and 4500 Series switches provide other security services that are not directly related to infrastructure protection, but that help secure the network. Refer to Other Security Services, page 109 to learn more about these security services.

Spanning Tree Protocol Security

As defined in the IEEE 802.1D standard, the Spanning Tree Protocol (STP) is a link management protocol for bridged networks that provides path redundancy while preventing undesirable loops in networks built of multiple active paths. Loops occur when multiple active paths exist between hosts, and which could result in an endless loop of traffic in the LAN that could bring the network down. STP implements an algorithm that guarantees a loop-free topology. With STP, all switches and bridges in the LAN exchange BPDU messages containing topology information. The STP algorithm uses the topology information to build a topological tree where only one active path at a time exists between any two hosts. Redundant paths are shut down and used as backups in case the primary paths fail. Changes to the physical topology normally trigger a recalculation of the topological tree.

A newer version of STP, called Rapid-STP (RSTP), is defined in IEEE 802.1w. RSTP works similarly to STP, but provides better convergence after a failure of a switch, switch port, or a LAN. RSTP significantly reduces the time to reconfigure the active topology of the network when changes to the physical topology or its configuration parameters occur. RSTP supersedes STP specified in 802.1D, but remains compatible with STP.

STP is a useful protocol, but unfortunately both versions of the protocol were conceived with no security in mind and, as a result, they are vulnerable to several types of attacks. STP does not implement any authentication and encryption to protect the exchange of BPDUs. Because of the lack of authentication, anyone can speak to an STP-enabled device. An attacker could very easily inject bogus BPDUs, triggering a topology recalculation. A forced change to the STP topology could lead into a DoS condition, or leave the attacker as a man-in-the-middle. In addition, because BPDUs are not encrypted, it is fairly simple to intercept BPDUs in transit, revealing important topology information.

Fortunately, Catalyst 6500 and 4500 Series switches support a set of features that help protect bridged networks using STP, and these are covered in this section (with exceptions noted). The following are the recommended best practices:

- Disable VLAN auto-negotiated trunking on user ports.
- Disable unused ports and put them into an unused VLAN (as covered in the previous section).
- Use Per-VLAN Spanning Tree (PVST).
- Implement port security (refer to Port Security, page 27).
- Enable traffic storm control (refer to Traffic Storm Control, page 32).
- Configure BPDU guard.
- Configure STP root guard.

Note Catalyst 6500 and 4500 Series switches provide other Layer 2 services that are not directly related to infrastructure protection but that help secure the network. Refer to Other Security Services, page 109 to learn more about these security services.

Disabling Auto-Negotiated Trunking

By default, all Ethernet ports on Catalyst switches are set to auto-negotiated trunking mode. Auto-negotiated trunking allows switches to automatically negotiate ISL and 802.1Q trunks. The negotiation is managed by the Dynamic Trunking Protocol (DTP). Setting a port to auto-negotiated trunking mode makes the port willing to convert the link into a trunk link, and the port becomes a trunk port if the neighboring port is set as a trunk, or configured in desirable mode. At the same time, a port configured in desirable mode becomes a trunk if the neighboring port is set to trunk, desirable, or auto mode.

While the auto-negotiation of trunks facilitates the deployment of switches, anyone can take advantage of this feature and easily set up an illegitimate trunk. For this reason, auto-negotiation trunking should be disabled on all ports connecting to end users.

In Catalyst OS, auto-negotiated trunking can be disabled on a port using the **set trunk off** command. By default, auto-negotiated trunking is set to **auto**, which causes the port to become a trunk port if the neighboring port tries to negotiate a trunk link. Using the **off** keyword forces the port to become a nontrunk port and persuades the neighboring port to become a nontrunk port:

```
Console> (enable) set trunk mod/ports {on | off | desirable | auto | nonegotiate} [vlans |
none] [isl | dot1q | dot10 | lane | negotiate]
```

This example shows how to disable auto-negotiation on port 2 on module 1:

```
Console> (enable)#set trunk 1/2 off
Port(s) 1/2 trunk mode set to off.
Console> (enable)
```

For more information on the **set trunk** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsy_tr.htm#wp1170006

For more information on the **set trunk** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_s_z.htm#wp1025473

To disable auto-negotiated trunking in Cisco IOS software, use the switchport mode access command. Setting the port mode to access makes the port a nontrunking, nontagged single VLAN Layer 2 interface:

```
Switch(config-if)# switchport mode {access | trunk | {dynamic {auto | desirable}} |
dot1q-tunnel}
```

This example shows how to set a port as nontrunking, nontagged single-VLAN Layer 2:

```
Switch(config)# interface type slot/port
Switch(config-if)# switchport mode access
Switch(config-if)#
```

Note Catalyst 6500 switches running Cisco IOS software support the **switchport host macro** command. The **switchport host macro** command was designed to expedite the configuration of switch ports that connect to end stations. Using the **switchport host macro** command sets the switch port mode to **access**, enables spanning tree PortFast, and disables channel grouping, all at the same time. The **switchport host macro** command can be used as an alternative to the **switchport mode access** command.

For more information on the **switchport mode access** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1022676

For more information on the **switchport mode access** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/cmdref/snmp_vtp.htm#wp1210450

Per VLAN Spanning Tree (PVST)

Per-VLAN Spanning Tree (PVST) is an available feature on Catalyst 6500 and 4500 Series switches that implements a separate instance of spanning tree for each VLAN configured in the network. PVST is available on switches running Cisco IOS and Catalyst OS software, and it is enabled by default. It is recommended to leave PVST always enabled. Having a separate instance of STP per VLAN makes the network more resilient to attacks against spanning tree. If a problem occurs in one VLAN, the effects are contained in that VLAN, shielding the rest of the network.

There are different versions of PVST that all maintain separate spanning tree instances per VLAN, and work in a similar fashion. Per VLAN Spanning Tree (PVST) is the original version, which uses ISL trunking. Per VLAN Spanning Tree Plus (PVST+) provides the same functionality as PVST using 802.1Q trunking technology rather than ISL. PVST+ is an enhancement to the 802.1Q specification and is not supported on non-Cisco devices. Rapid-Per-VLAN-Spanning Tree (Rapid-PVST+) is another version of PVST that provides faster convergence of the spanning tree by using Rapid Spanning Tree Protocol (RSTP) with the existing configuration for PVST+.

PVST+ and Rapid-PVST+ are available on Catalyst 6500 and 4500 Series switches for both Catalyst OS and Cisco IOS. The default spanning tree protocol for Catalyst 6500 Series switches is Rapid-PVST+ for systems running Catalyst OS software, and PVST+ for systems running Cisco IOS software. PVST+ is the default spanning tree for Catalyst 4500 Series switches for both, Catalyst OS and Cisco IOS software.

To modify the spanning tree mode on a system running Catalyst OS, use the **set spantree mode** command.

```
Console> (enable) set spantree mode {mistp | pvst+ | mistp-pvst+ | mst | rapid-pvst+}
```

This example shows how to configure Rapid-PVST+:

```
Console> (enable) set spantree mode rapid-pvst+
Warning!! Changing the STP mode from a telnet session will disconnect the
session because there are no VLANs mapped to any RAPID-VST+ instance.
Do you want to continue [n]?
```

For more information on the **set spantree mode** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1058609

For more information on the **set spantree mode** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_q_s.htm#wp1085096

To modify the spanning tree mode on a system running Cisco IOS, use the **spanning-tree mode** command:

```
Switch(config)# spanning-tree mode [pvst | mst | rapid-pvst]
```

This example shows how to configure Rapid-PVST+:

```
Switch(config)# spanning-tree mode rapid-pvst
Switch(config)#
```

For more information on the **spanning-tree mode** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1087162

For more information on the **spanning-tree mode** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/cmdref/snmp_vtp.htm#wp1144173

BPDU Guard

BPDU guard is an available feature on Catalyst 6500 and 4500 Series switches running Catalyst OS and Cisco IOS software that prevents a host port from participating in spanning tree. Under normal circumstances, Layer 2 access ports connected to a single workstation or server should not participate in spanning tree. When enabled on a port, BPDU guard shutdowns the port as soon as a BPDU is received in that port. In this way, BPDU guard helps prevent unauthorized access and the illegal injection of forged BPDUs.

BPDU guard requires STP PortFast to be configured on the port first. STP PortFast causes a Layer 2 LAN port configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. PortFast can be used on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge.

BPDU can be configured per port, or globally. When configured globally, BPDU guard is effective only on ports in the operational PortFast state.

To enable BPDU guard on a port of a system running Catalyst OS, use the **set spantree bpdu-guard** command. You must first enable PortFast on the port.

```
Console> (enable) set spantree portfast mod/port enable
Console> (enable) set spantree bpdu-guard mod/port {enable | disable | default}
```

This example shows how to enable spanning tree BPDU guard on module 1, port 2:

```
Console> (enable) set spantree portfast 1/2 enable
Warning: Connecting layer 2 devices to a fast-start port can cause temporary spanning tree
loops. Use with caution.
Spantree port 1/2 fast start enabled.
Console> (enable)
Console> (enable) set spantree portfast bpdu-guard 1/2 enable
Spantree port 1/2 bpdu guard enabled.
Console> (enable)
```

For more information on the **set spantree bpdu-guard** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1243604

For more information on the **set spantree bpdu-guard** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_q_s.htm#wp1072868

BPDU guard can be globally enabled on system running Catalyst OS by using the **set spantree global-default bpdu-guard** command. When enabled globally, BPDU guard applies to all interfaces that are in an operational PortFast state.

```
Console> (enable) set spantree global-default bpdu-guard {enable | disable}
```

This example shows how to enable the global BPDU guard state on the switch:

```
Console> (enable) set spantree global-default bpdu-guard enable
Spantree global-default bpdu-guard enabled on this switch.
Console> (enable)
```

For more information on the **set spantree global-default bpdu-guard** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1169655

For more information on the **set spantree global-default bpdu-guard** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_q_s.htm#wp1049757

To enable BPDU guard on an interface of a system running Cisco IOS, use the **spanning-tree bpduguard** command. You must first enable PortFast on the port.

```
Switch(config)# interface type slot/port
Switch(config-if)# spanning-tree portfast
Switch(config-if)# spanning-tree bpduguard {enable | disable}
```

This example shows how to enable spanning tree BPDU guard on an interface:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# spanning-tree portfast
Switch(config-if)# spanning-tree bpduguard enable
```

For more information on the **spanning-tree bpduguard** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1084894

For more information on the **spanning-tree bpduguard** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/cmdref/snmp_vtp.htm#wp1065041

BPDU guard can be enabled globally on systems running Cisco IOS by using the **spanning-tree portfast bpduguard default** command. When enabled globally, BPDU guard applies to all interfaces that are in an operational PortFast state.

```
Switch(config)# spanning-tree portfast bpduguard default
```

This example shows how to enable BPDU guard globally:

```
Switch(config)# spanning-tree portfast bpduguard
```

For more information on the **spanning-tree portfast bpduguard default** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1087959

For more information on the **spanning-tree portfast bpduguard default** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/cmdref/snmp_vtp.htm#wp1076298

STP Root Guard

STP Root Guard is an available feature on Catalyst 6500 and 4500 Series switches running Catalyst OS and Cisco IOS software that enforces the placement of the root bridge. STP root guard is a feature that is enabled on selected ports to prevent surrounding switches from becoming the root switch. The root guard feature forces a port to become a designated port so that no switch on the other end of the link can become a root switch. If a port configured for root guard receives a superior BPDU, the port immediately goes into a root-inconsistent (blocked) state. In this way, STP root guard blocks other devices trying to become the root bridge by sending superior BPDUs.

In a typical environment, you can identify ports that will never connect to a root bridge. For example, ports connecting to workstations or servers. STP root guard should be enabled on such ports to ensure that a root bridge will never be negotiated on those ports.

Note Do not enable loop guard and root guard on a port at the same time. Root guard forces a port to always be designated as the root port. Loop guard is effective only if the port is a root port or an alternate port.

To enable STP Root Guard on a port of a system running Catalyst OS, use the **set spantree guard root** command.

```
Console> (enable) set spantree guard {none | root | loop} mod/port
```

This example shows how to enable STP Root Guard:

```
Console> (enable) set spantree guard root 5/1
Rootguard on port 5/1 is enabled.
Warning!! Enabling rootguard may result in a topolopy change.
Console> (enable)
```

For more information on the **set spantree guard root** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1199243

For more information on the **set spantree guard root** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/command/set_q_s.htm#wp1046977

To enable STP Root Guard on an interface of a system running Cisco IOS, use the **spanning-tree guard root** command. You must first enable PortFast on the port.

```
Switch(config)# interface type slot/port
Switch(config-if)# spanning-tree guard {loop | root | none}
```

This example shows how to enable STP Root Guard on an interface:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# spanning-tree guard root
```

For more information on the **spanning-tree guard root** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/s1.htm#wp1031770

For more information on the **spanning-tree guard root** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/cmdref/snmp_vtp.htm#wp1031770

Routing Protocol Security

Routing is one of the most important parts of an infrastructure to keep a network running and, as such, it is absolutely critical to take the necessary measures to secure it. There are different ways routing can be compromised, from the injection of illegitimate updates to DoS attacks that are specifically designed to disrupt routing.

Fortunately, Catalyst 6500 and 4500 Series switches support a set of features for BGP, IS-IS, OSPF, EIGRP and RIPv2, that help secure the routing infrastructure. The following are the recommended best practices:

- Neighbor Authentication, page 24
- Route Filtering, page 25
- TTL Security Check, page 26

Note Cisco IOS software provides other routing security features that are not directly related to infrastructure protection, but that help secure the network. To learn more about these other security features, refer to the *Cisco IOS IP Protocols Configuration Guide* at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_book09186a0080087fa9.html

Neighbor Authentication

Neighbor authentication is a feature that is available on most routing protocols, and which ensures that a router receives only reliable routing information from trusted neighbors. That is achieved by certifying the authenticity of each neighbor and the integrity of its routing updates. Technically, each router is initially configured with a shared secret key that is used to validate each routing update. Before sending a routing update, each router is required to sign it with the predefined secret key and include the resulting signature as part of the update message. Finally, the update is verified by the receiving neighbor to prove its authenticity and integrity.

Most routing protocols support two types of neighbor authentication, plain text and Message Digest Algorithm Version 5 (MD5) authentication. Plain text authentication consists of sending the secret key in the clear inside each routing update message, which does not provide much security because keys can be intercepted while in transit. MD5 authentication works by processing each routing update with an MD5 hash function and by including the resulting signature (digest) as part of the routing update message. MD5 authentication is more secure because the actual shared secret key is never sent over the network.

MD5 neighbor authentication is available for the following routing protocols:

- Border Gateway Protocol (BGP)
- IP Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (IS-IS)

- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP) version 2

The configuration commands and steps to enable neighbor authentication vary depending on the routing protocol. To find complete configuration information for specific routing protocols, refer to the *Cisco IOS IP Protocols Configuration Guide* at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_book09186a0080087fa9.html

The following example shows the configuration of OSPF MD5 neighbor authentication on an IOS router:

```
! OSPF MD5 authentication
interface Ethernet1
  ip address 198.121.115.1 255.255.255.0
  ip ospf message-digest-key 10 md5 oursharedsecret
!
router ospf 20
network 198.121.115.0 0.0.0.255 area 0
area 0 authentication message-digest
```

For more information about neighbor authentication in IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fothersf/scfroutr.htm

Route Filtering

Route filtering is another import tool for securing the routing infrastructure. Most routing protocols allow the configuration of route filters that prevent specific routes from being propagated throughout the network. In terms of security, these filters are useful because they help to ensure that only legitimate networks are advertised, and that networks that are not supposed to be propagated are never advertised, (networks falling within the private address space (RFC 1918)).

The configuration commands and procedures used for route filtering vary, depending on the routing protocol. To find complete configuration information for specific routing protocols, refer to the *Cisco IOS IP Protocols Configuration Guide* at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_book09186a0080087fa9.html

This example shows the configuration of a route filter on BGP. This sample configuration enables Router 100 to deny an update for network 10.10.10.0/24 and permit the updates of networks 192.168.10.0/24 and 10.10.0.0/19 in its BGP table:

```
hostname Router 100
!
router bgp 100
neighbor 172.16.1.2 remote-as 200
neighbor 172.16.1.2 distribute-list 1 in
!
access-list 1 deny 10.10.10.0 0.0.0.255
access-list 1 permit any
```

For more information about route filtering, refer to the following URL:

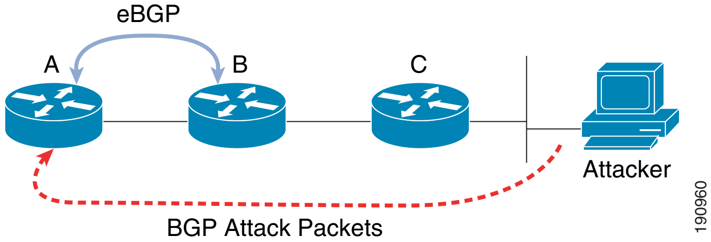
http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008020b51d.shtml

TTL Security Check

Based on the Generalized TTL Security Mechanism (GTSM, RFC 3682), the TTL security check is a security feature that protects BGP peers from multi-hop attacks. This feature allows the configuration of a minimum acceptable TTL value for the packets exchanged between two eBGP peers. When enabled, both peering routers transmit all their traffic to each other with a TTL of 255. In addition, routers establish a peering session only if the other eBGP peer sends packets with a TTL equal to or greater than the TTL value configured for the peering session. All packets received with TTL values less than the predefined value are silently discarded. In this way, the TTL security check prevents all possible attacks from attackers not connected directly to the same physical network connecting the two routers.

For example, when TTL security check is enabled between two eBGP peers, both routers transmit all their traffic to each other with a TTL of 255. If the routers are one hop away, the security check will accept only incoming packets with a TTL equal to or greater than 254. This ensures that traffic from all devices that are not directly connected will not be accepted because all traffic from devices not directly connected will arrive with a TTL of less than 254, as shown in Figure 3.

Figure 3 TTL Security Check



In the example shown in Figure 3, Router A will accept only those packets with a TTL of 254 or greater. Regardless of the TTL value the attacker sets, all of their packets will reach Router A with a TTL of less than 254.

Note The TTL security check feature is currently available for BGP only. Work is currently in progress to implement this feature for other routing protocols, such as OSPF and EIGRP.

In Cisco IOS software, the TTL security check can be enabled per peer using the **neighbor ttl-security** command:

```
Router(config)# router bgp as-number
Router(config-router)# switchport mode access
Router(config-router)# neighbor ip-address ttl-security hops hop-count
```

In this example, TTL security check is enabled for the 10.1.1.1 eBGP neighbor, which resides two hops away:

```
Router(config)# router bgp 1
Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
```

For more information about TTL Security Check, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080455621.html

Catalyst Integrated Security

The first sections of this document presented a collection of basic tools and techniques for infrastructure protection. This section introduces a set of advanced security features that are designed to take advantage of the unique Catalyst 6500 and 4500 hardware architectures, making these switching platforms more resilient to attacks, and thereby providing enhanced protection for the infrastructure.

The following advanced security features are covered in this section:

- Port Security, page 27
- MAC Address Monitoring, page 30
- Traffic Storm Control, page 32
- Unicast and Multicast Flood Blocking, page 37
- DHCP Snooping, page 39
- IP Source Guard, page 43
- Dynamic ARP Inspection (DAI), page 46

Note Catalyst 6500 and 4500 Series switches provide other security services that are not directly related to infrastructure protection, but that help secure the network. Refer to Other Security Services, page 109 to learn more about these other security services.

Port Security

Port security is an available feature on Catalyst 6500 and 4500 Series switches running Catalyst OS and Cisco IOS software that can be configured in a port to restrict the MAC addresses that are allowed to send traffic into that port. Port security helps mitigate MAC flooding and other Layer 2 Content Addressable Memory (CAM) overflow attacks. With port security, a list of allowed MAC addresses can be dynamically learned or statically configured. After a list of secure (trusted) MAC addresses is defined for a port, only packets with source addresses in that list get forwarded throughout that port.

The list of secure MAC addresses can be statically configured by manually declaring each trusted MAC address, or they can be dynamically learned by defining a maximum number of MAC addresses to be learned as traffic is received from the port. If the number of secure MAC addresses is set to one and only one secure MAC address is assigned, the workstation attached to that port has the full bandwidth of the port.

When a port configured with port security receives a packet with a source MAC address that is not found in the trusted list, and if the maximum number of secure MAC addresses has been reached, a security violation occurs.

In Catalyst OS, a port can be set to the following two modes to handle a security violation:

- Shutdown—Shuts down the port permanently or for a specified time. Permanent shutdown is the default mode.
- Restrict—Drops all packets from insecure hosts, but remains enabled.

In Cisco IOS, there are three ways a port can react when a security violation takes place:

- Protect—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value.

- Restrict—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value and causes the SecurityViolation counter to increment.
- Shutdown—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

Note Ports connecting to IP phones need to be configured to allow at least three MAC addresses: one for the workstation, one for the phone on the voice VLAN, and one for the phone on the native VLAN for CDP. In addition, the violation action should be set to restrict so the port is not entirely taken down when a violation occurs.

Configuring Port Security in Catalyst OS

To configure port security on a switch running Catalyst OS, perform the following steps:

Step 1 Enable port security on the desired ports using the **set port security enable** command. Optionally, a secure MAC address can be specified. To enable port security on a trunk port, specify the VLANs on which a secure MAC address is allowed:

```
Console> (enable) set port security mod/port enable [mac_addr] [vlan_list]
```

Step 2 Configuring the Port Security violation mode on the port using the **set port security violation** command:

```
Console> (enable) set port security mod/port violation {shutdown | restrict}
```

Step 3 Add the MAC addresses to the list of secure addresses using the **set port security** command:

```
Console> (enable) set port security mod/port mac_addr
```

Step 4 Set the Maximum Number of Secure MAC Addresses **set port security maximum** command:

```
Console> (enable) set port security mod/port maximum num_of_mac
```

Step 5 Enable Dynamically Learned MAC Addresses port security globally using the **set port security auto-configure enable** command. This feature applies globally to all secure ports on the system.

```
Console> (enable) set port security auto-configure enable | disable
```

In this example, port 2/1 is configured as a secure port, a static secure MAC address entry is defined for 00-90-2b-03-34-08, and the port is configured to accept up to five dynamically learned MAC addresses.

```
Console> (enable) set port security 2/1 enable
Port 2/1 security enabled.
Console> (enable) set port security 2/1 enable 00-90-2b-03-34-08
Port 2/1 port security enabled with 00-90-2b-03-34-08 as the secure mac address
Trunking disabled for Port 2/1 due to Security Mode
Console> (enable)
Console> (enable) set port security 2/1 maximum 5
Maximum number of secure addresses set to 5 for port 2/1.
Console> (enable)
Console> (enable) set port security auto-configure enable
Automatic configuration of secure learnt addresses enabled.
Console> (enable)
```

For more information on how to configure Port Security on Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/sec_port.htm

For more information on how to configure Port Security on Catalyst 4500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/configur/sec_port.htm

Configuring Port Security in Cisco IOS

To configure Port Security on Cisco IOS, perform the following steps:

Step 1 Enable port security on the desired ports using the **switchport port-security interface** command:

```
Switch(config)# interface type slot/port
Switch(config-if)# switchport port-security
```

Step 2 Configuring the Port Security violation mode on the port using the **switchport port-security violation interface** command. By default the violation mode is set to shutdown:

```
Switch(config-if)# switchport port-security violation {protect | restrict | shutdown}
```

Step 3 Add the MAC addresses to the list of secure addresses using the **switchport port-security mac-address interface** command:

```
Switch(config-if)# switchport port-security mac-address [sticky] mac_address [vlan
vlan_ID]
```

Step 4 Set the maximum number of secure MAC addresses using the **switchport port-security maximum interface** command. By default the number is set to 1:

```
Switch(config-if)# switchport port-security maximum number_of_addresses vlan {vlan_ID |
vlan_range}
```

In this example interface GigabitEthernet2/24 is configured as a secure port, a static secure MAC address entry is defined for 0090.2b03.3408, and the port is configured to accept up to five dynamically learned MAC addresses. The port is configured in the protect security violation mode.

```
Switch(config)# interface GigabitEthernet2/24
Switch(config-if)# switchport
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security violation protect
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# switchport port-security mac-address 0090.2b03.3408
Switch(config-if)# end
```

For more information on how to configure Port Security on Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/port_sec.htm

For more information on how to configure Port Security on Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/port_sec.htm

MAC Address Monitoring

MAC address monitoring is a feature present on Catalyst 6500 Series switches running Catalyst OS and Cisco IOS software. This feature helps mitigate MAC flooding and other CAM overflow attacks by limiting the total number of MAC addresses learned by the switch on per-port or per-VLAN basis.

With MAC Address Monitoring, a maximum threshold for the total number of MAC addresses can be configured and enforced on a per-port and/or per-VLAN basis. The system can be configured to notify or disable the port or VLAN every time the number of learned MAC addresses exceeds the predefined threshold.

At a high level, Cisco IOS and CatOS implement MAC address monitoring in a similar manner. However, there are some implementation differences that should be noted, and which are discussed in the following sections.

Configuring MAC Address Monitoring in Catalyst OS

In Catalyst OS, MAC address monitoring allows the definition of a low threshold and a high threshold. Compared to a single threshold the combination of a low and high threshold provides greater flexibility. In addition, the number of MAC addresses learned can be monitored not only on a per-port or per-VLAN basis, but also on a per-port-per-VLAN basis.

Note Before performing the following steps, MAC address monitoring first needs to be enabled globally, which is the default configuration (in case it has been disabled).

To configure MAC address monitoring on a Catalyst 6500 switch running Catalyst OS, perform the following steps:

Step 1 Enable MAC address monitoring globally using the **set cam monitor enable** command:

```
Console> (enable) set cam monitor enable
```

Step 2 Enable the monitoring of MAC addresses that are learned and stored in the CAM table on a per-port basis, per-VLAN basis, or on a per-port- per-VLAN basis using the **set cam monitor** command. Note that MAC-address monitoring is disabled by default on an interface (port, VLAN, or port/VLAN basis):

```
Console> (enable) set cam monitor {disable | enable} [mod/port | {mod/port vlan} | vlan]
```

Step 3 Specify the lower threshold for MAC-address monitoring and the action to be taken when the system exceeds this threshold. Use the **set cam monitor low-threshold** command. The valid range for the lower threshold is 5-32000. Note that if you specify the **no-learn** keyword, and the configuration is a port/VLAN configuration, the violation action stops learning the MAC addresses on the port from all the VLANs. If you specify the **warning** keyword, the system displays a system message when the low threshold is exceeded:

```
Console> (enable) set cam monitor low-threshold value [action {no-learn | warning}]
{mod/port | {mod/port vlan} | vlan}
```

Step 4 Specify the upper threshold or MAC-address monitoring and the action to be taken when the system exceeds this threshold. Use the **set cam monitor high-threshold** command. The valid range for the high threshold is 5-32000. Note that if you specify the **no-learn** keyword, and the configuration is a port/VLAN combination, the violation action stops learning the MAC addresses on the port from all the

VLANs. If you specify the **shutdown** keyword, and the configuration is a port/VLAN combination, the violation action error disables the port. If you specify the **warning** keyword, the system displays a system message when the high threshold is exceeded:

```
Console> (enable) set cam monitor high-threshold value [action {no-learn | shutdown |
warning}] {mod/port | {mod/port vlan} | vlan}
```

Step 5 Optionally, specify the polling interval for monitoring using the **set cam monitor interval** command. MAC address monitoring is supported in software. If there are a large number of MAC addresses in the CAM table and a large number of configured interfaces (ports, VLANs, or port-VLANs), the CPU usage might go up. The load on the CPU can be reduced by adjusting the software polling interval using the **set cam monitor interval** command. By default the polling interval is set to 5 seconds. It can be changed to any value between 5 and 30 seconds:

```
Console> (enable) set cam monitor interval time_s
```

This example illustrates how to enable MAC address monitoring globally and how to activate the monitoring of MAC addresses on a specific port (4/3). A low threshold is configured for the first 10,000 MAC addresses learned in the CAM table, and the system is set to generate a system message when the low threshold is exceeded. A high threshold is configured to 20,000 MAC addresses, and the system is set to shutdown the port when this threshold is exceeded.

```
Console> (enable) set cam monitor enable
Cam monitor enabled
Console> (enable) set cam monitor enable 4/3
Successfully enabled cam monitor on 4/3
Console> (enable) set cam monitor low-threshold 10000 action warning 4/3
Successfully configured cam monitor on 4/3
Console> (enable) set cam monitor high-threshold 20000 action shutdown 4/3
Successfully configured cam monitor on 4/3
Console> (enable)
```

For more information on how to configure MAC address monitoring on Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/sec_port.htm#wp1033648

Configuring MAC Address Monitoring in Cisco IOS

MAC address monitoring in Cisco IOS allows the definition of a single upper (maximum) threshold. In addition, the number of MAC addresses learned can be only monitored on a per-port or per-VLAN basis, and not per-port-per-VLAN.

By default, MAC address monitoring is disabled in Cisco IOS. However, the maximum threshold for all ports and VLANs is configured to 500 MAC address entries, and when the threshold is exceeded the system is set to generate a system message along with a syslog trap. These default values take effect only when MAC address monitoring is enabled.

Note MAC address monitor is supported on Catalyst 6500 Series switches with Supervisor Engine 2 and Supervisor Engine 720. At this moment this feature is not supported on Catalyst 6500 Series switches equipped with Supervisor Engine 32.

To configure MAC address monitoring on a Catalyst 6500 switch running Cisco IOS, perform the following steps:

Step 1 The first step in the configuration is to enable MAC address monitoring globally using the **mac-address-table limit** command. To change the default global configuration, use the following command options. The **maximum** keyword specifies the maximum number of MAC entries per VLAN per EARL allowed, valid values are from 5 to 32000 MAC-address entries. Use the **action** keyword to specify an action to be taken when the maximum threshold is exceeded (warning, limit or shutdown). Setting the action to **warning** means that one syslog message will be sent and no further action will be taken. Setting the action to **limit** means that the one syslog message will be sent and/or a corresponding trap will be generated. Finally, setting the action to **shutdown** means that the one syslog message will be sent and/or the VLAN is moved to the blocked state. The notification mechanism can also be set by using the **notification** keyword. Use **trap** for traps, **syslog** for syslog messages, or **both** for both trap and syslog.

```
Router(config)# mac-address-table limit [maximum num] [action {warning | limit |
shutdown}] [notification {syslog | trap | both}]
```

Step 2 Optionally, enable the monitoring of MAC addresses on a per-port basis or per-VLAN basis. Use the **mac-address-table limit** command. When using the **maximum** and **action** keywords, follow the same guidelines given in the previous step. Use the **flood** keyword to enable unknown unicast flooding on a VLAN (this is enabled by default).

```
Router(config)# mac-address-table limit [{vlan vlan} | {interface type mod/port}] [maximum
num] [action {warning | limit | shutdown}] [flood]
```

This example shows how to enable MAC address monitoring globally, and on a per-VLAN basis. MAC address monitoring is enabled on VLAN 10, and for which a maximum threshold of 500 MAC addresses is configured. The system is set to shutdown VLAN 10 when the maximum threshold is exceeded.

```
Router(config)# mac-address-table limit
Router(config)#
Router(config)# mac-address-table limit vlan 10 maximum 500 action shutdown
Router(config)#
```

For more information on how to configure MAC address monitoring on Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/i1.htm#wp1643725

Traffic Storm Control

Traffic Storm Control is a feature that is available on the Catalyst 6500 and 4500 Series switches to help mitigate DoS and other attacks that generate large volumes of packets, resulting in traffic storms. When a traffic storm occurs, the network is flooded with packets, creating excessive traffic and degrading network performance. The traffic storm control feature (also called traffic suppression) prevents LAN ports from being disrupted by a traffic storm on physical interfaces.

Traffic storm control monitors incoming traffic levels over a 1-second traffic storm control interval and, during the interval, compares the traffic level with the traffic storm control level that you configure. The traffic storm control level is a percentage of the total available bandwidth of the port. Each port has a single traffic storm control level that is used for all types of traffic (broadcast, multicast, and unicast).

Conceptually, Catalyst 6500 and 4500 Series switches implement the traffic storm feature in a similar manner. However, there are some implementation differences. Traffic storms can consist of unicast, multicast, or broadcast packets. Traffic storm control in Catalyst 6500 Series switches can monitor and limit unicast, multicast, and broadcast packet storms while in Catalyst 4500, storm control can monitor and limit multicast and broadcast packet storms only. There are other implementation differences between the two switching platforms, as explained in the following sections.

Catalyst 6500 Broadcast Suppression (Catalyst OS)

The traffic storm control feature on Catalyst 6500 Series switches is supported for both, Catalyst OS and Cisco IOS. In Catalyst OS, traffic storm control is implemented under the name of broadcast suppression. In this switching platform, broadcast traffic storm control is supported on all LAN ports, while multicast and unicast traffic storm control is supported only on Gigabit and 10 Gigabit Ethernet ports.

Broadcast suppression on the Catalyst 6500 Series switches is implemented in the hardware. The suppression circuitry monitors the packets passing from a port to the switching bus. Using the Individual/Group bit in the packet destination address, the broadcast suppression circuitry determines if the packet is unicast or broadcast, keeps track of the current count of broadcasts within the 1-second time interval, and when a threshold is reached, filters out the subsequent broadcast packets.

Note Broadcast and multicast suppression is not supported on the WS-X6148A-GE-TX, WS-X6148A-GE-45A, and WS-X6548-GE-TX modules.

Because hardware broadcast suppression uses a bandwidth-based method to measure the broadcast activity, the most significant implementation factor is setting the percentage of the total available bandwidth that can be used by the broadcast traffic. A threshold value of 100 percent means that no limit is placed on the broadcast traffic. By entering the **set port broadcast** command, you can set up the broadcast suppression threshold value.

Because the packets do not arrive at uniform intervals, the 1-second time interval during which the broadcast activity is measured can affect the behavior of broadcast suppression.

On the Gigabit Ethernet ports, you can use broadcast suppression to filter multicast and unicast traffic. You can suppress the multicast or unicast traffic separately on a port, both require that you configure broadcast suppression. When you specify a percentage of the total bandwidth to be used for the multicast or unicast traffic, the same limit applies to the broadcast traffic.

Note Multicast suppression does not drop bridge protocol data unit (BPDU) packets unless it is enabled in one of the following modules: WS-X6724-SFP, WS-X6748-GE-TX, WS-X6748-SFP, WS-X6704-10GE, WS-SUP32-GE-3B, and WS-SUP32-10GE-3B. Enabling multicast suppression on these modules can cause BPDUs to be suppressed when the multicast suppression threshold is exceeded. We strongly advise that you do not use multicast suppression on ports that need to receive BPDUs because potential side effects can be root port loss or spanning tree loops when the suppression threshold is exceeded. As an alternative, consider using the Layer 2 hardware-based rate limiter on Supervisor 32 and 720 if you need to control the volume of BPDUs or other Layer 2 PDUs. Note that the hardware-based rate limiters must be used cautiously, as they control traffic globally and can also have potential side effects on the Layer 2 connectivity. Refer to Configuring Hardware-Based Rate Limiters in Catalyst OS, page 82 for more information.

Broadcast suppression is disabled by default. To configure broadcast suppression on the Catalyst 6500 running Catalyst OS, use the **set port broadcast** command. Enable the broadcast suppression threshold for one or more ports as a percentage of the total bandwidth:

```
Console> (enable) set port broadcast mod/port threshold% [violation {drop-packets |
errdisable}] [multicast {enable | disable}][unicast {enable | disable}]
```

When broadcast, multicast, or unicast suppression occurs, you can configure the ports to either drop the packets or go into the errdisable state. The errdisable state feature can be enabled or disabled on a per-port basis, and is disabled by default (the **drop-packets** option is enabled by default).

When a port is put into errdisable state, it can be re-enabled after a specific timeout interval has expired. Enter the **set errdisable-timeout interval** command to specify the timeout interval.

This example shows how to limit the broadcast traffic to 90 percent and error disable the port when broadcast suppression occurs:

```
Console> (enable) set port broadcast 4/6 90% violation errdisable
Port 4/6 broadcast traffic limited to 90.00%.
On broadcast suppression port 4/6 is configured to move to errdisabled state.
Console> (enable)
```

For more information on Broadcast Suppression on the Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/bcastsup.htm

Catalyst 6500 Traffic Storm Control (Cisco IOS)

The traffic storm control feature on Catalyst 6500 Series switches is supported for both Catalyst OS and Cisco IOS. In this switching platform, broadcast traffic storm control is supported on all LAN ports, while multicast and unicast traffic storm control is supported only on Gigabit and 10 Gigabit Ethernet ports.

Traffic storm control on the Catalyst 6500 Series switches is implemented in hardware. The traffic storm control circuitry monitors packets passing from a LAN interface to the switching bus. Using the Individual/Group bit in the packet destination address, the traffic storm control circuitry determines if the packet is unicast or broadcast, keeps track of the current count of packets within the 1-second interval, and when a threshold is reached, filters out subsequent packets.

Note The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support traffic storm control.

Because hardware traffic storm control uses a bandwidth-based method to measure traffic, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by controlled traffic. Because packets do not arrive at uniform intervals, the 1-second interval during which controlled traffic activity is measured can affect the behavior of traffic storm control.

Note Multicast suppression does not drop bridge protocol data unit (BPDU) packets unless is enabled in one of the following modules: WS-X6724-SFP, WS-X6748-GE-TX, WS-X6748-SFP, WS-X6704-10GE, WS-SUP32-GE-3B, and WS-SUP32-10GE-3B. Enabling multicast suppression on these modules can cause BPDUs to be suppressed when the multicast suppression threshold is exceeded. We strongly advise that you do not use multicast suppression on ports that need to receive BPDUs because potential side effects can be root port loss or spanning tree loops when the suppression threshold is exceeded. As an alternative, consider using the Layer 2 hardware-based rate limiter present on Supervisor 32 and 720 if you need to control the volume of BPDUs or other Layer 2 PDUs. Note that the hardware-based rate limiters must be used cautiously, they control traffic globally and can also have potential side effects on the Layer 2 connectivity. Refer to Hardware-Based Rate Limiters on Supervisors 32 and 720, page 72 for more information.

Traffic storm control is disabled by default.

To configure traffic storm control on a Catalyst 6500 Series switch running Cisco IOS, perform the following steps:

Step 1 At the interface level enable broadcast traffic storm control, configure the traffic storm control level and apply the traffic storm control level to all traffic storm control modes enabled on the interface. Use the **storm-control broadcast level interface** command:

```
Router(config)# interface {{type slot/port | {port-channel number}}
Router(config-if)# storm-control broadcast level level[.level]
```

Step 2 Enable multicast traffic storm control on the interface, configure the traffic storm control level, and apply the traffic storm control level to all traffic storm control modes enabled on the interface. Use the **storm-control multicast level interface** command:

```
Router(config-if)# storm-control multicast level level[.level]
```

Step 3 Enable unicast traffic storm control on the interface, configure the traffic storm control level, and apply the traffic storm control level to all traffic storm control modes enabled on the interface. Use the **storm-control unicast level interface** command:

```
Router(config-if)# storm-control unicast level level[.level]
```

This example shows how to enable multicast traffic storm control on Gigabit Ethernet interface 3/16 and how to configure the traffic storm control level at 70.5 percent. This configuration applies the traffic storm control level to all traffic storm control modes enabled on Gigabit Ethernet interface 3/16:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control multicast level 70.5
Router(config-if)# end
```

For more information on Broadcast Suppression on the Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/storm.htm

Catalyst 4500 Port-Based Traffic Control (Cisco IOS)

On Catalyst 4500 Series switches, the traffic storm control feature is called port-based traffic control. This feature is available only on Cisco IOS software for Catalyst 4500 Series switches, and it primarily supports broadcast suppression. Multicast suppression is available on WS-X4516 supervisor engines and newer hardware.

Broadcast suppression on the Catalyst 4500 Series switches is implemented in hardware. The suppression circuitry monitors packets passing from a LAN interface to the switching bus. If the packet destination address is broadcast, then the broadcast suppression circuitry tracks the current count of broadcasts within the one-second interval, and when a threshold is reached, it filters out subsequent broadcast packets.

Note Storm control is supported in hardware on all ports on the WS-X4516 supervisor engine. In contrast, the supervisor engines WS-X4515, WS-X4014, and WS-X4013+ support storm control in hardware on non-blocking gigabit ports and in software on all other ports, implying that the counters for these interfaces are approximate and computed. Multicast storm control is only supported on the WS-X4516 supervisor engine.

Because hardware broadcast suppression uses a bandwidth-based method to measure broadcast activity, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by broadcast traffic. Because packets do not arrive at uniform intervals, the one-second interval during which broadcast activity is measured can affect the behavior of broadcast suppression.

To configure traffic storm control on the Catalyst 4500 running Cisco IOS, perform the following steps:

Step 1 At the interface level, configure the upper threshold levels for broadcast traffic. The storm control action occurs when traffic utilization reaches this level. Optionally specify the falling threshold level. Use the **storm-control broadcast level interface** command:

```
Switch(config)# interface type slot/port
Switch(config-if)# storm-control broadcast level [high level] [lower level]
```

Step 2 Specify the action to be taken when a storm is detected using the **storm-control action interface** command. The default is to filter out the broadcast traffic and to not send out traps. The **shutdown** keyword sets the port to error-disable state during a storm. If the recover interval is not set, the port remains in a shutdown state. The **trap** keyword generates an SNMP trap when a storm is detected.

```
Switch(config-if)# storm-control action {shutdown | trap}
```

Step 3 Optionally, enable multicast suppression using the **storm-control broadcast include multicast** global configuration command:

```
Switch(config)# storm-control broadcast include multicast
```

The following example shows how to enable storm control on interface:

```
Switch(config)# interface fastethernet 3/1
Switch(config-if)# storm-control broadcast level 50
Switch(config-if)# end
```

The following example shows how to enable multicast suppression on ports that already have broadcast suppression enabled:

```
Switch# configuration terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# storm-control broadcast include multicast
Switch(config)# end
```

For more information on port-based traffic control, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/bcastsup.htm

Unicast and Multicast Flood Blocking

As part of the normal operation of a switch, each time a packet with an unknown destination MAC address is received, the packet is forwarded to all ports on the switch, with the exception of the port from which the packet was received. There are several possible reasons why a MAC address could be unknown (the address is not found in the bridge table), either the switch has not yet received a packet sourced with that MAC address, the bridge table entry for that MAC address has timed out, or the bridge table is already full. In all cases, the indiscriminate flooding of packets consumes bandwidth and poses a security threat. To prevent the flooding of packets with unknown MAC addresses, the Catalyst 6500 and Catalyst 4500 Series switches implement the unicast flood blocking feature.

Unicast flood blocking is available on both Cisco IOS software and Catalyst OS software. This feature is configured at the switch port level and, when enabled, it prevents any unknown unicast traffic from being sent to the port. Prior to enabling unicast flood blocking on a port, you must have a static entry (CAM entry) in the bridge table for each of the systems connected to that port. Enabling this feature without the necessary static entries will break connectivity on the port.

Note By default, all of the Ethernet ports on a switch are configured to allow unicast flooding. Unicast flood blocking allows you to drop the unicast flood packets before they reach the port.

Flood blocking for unicast traffic is supported on both Catalyst 6500 and Catalyst 4500 Series switches. The Catalyst 4500 Series switches have an extended version of this feature called Port Unicast and Multicast Flood Blocking. As the name indicates, this feature can block multicast traffic in addition to unicast traffic.

Note Unicast flood blocking is supported on Catalyst 6500 and Catalyst 4500 Series switches running both Cisco IOS and Catalyst OS software. Multicast flood blocking is supported only on Catalyst 4500 Series switches running Cisco IOS software.

Catalyst 6500 and Catalyst 4500 Unicast Flood Blocking (Catalyst OS)

Unicast flood blocking is supported on both Catalyst 6500 and Catalyst 4500 Series switches running Catalyst OS.

By default, all the Ethernet ports on the switch are configured to allow unicast flooding. To block unicast flooding on the desired Ethernet ports use the **set port unicast-flood** command:

```
Console> (enable) set port unicast-flood mod/port enable
```

This example shows how to enable unicast flood blocking on a port:

```
Console> (enable) set port unicast-flood 4/1 enable
Unicast Flooding is successfully enabled on the port 4/1.
Console> (enable)
```

For more information on unicast flood blocking on the Catalyst 6500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/uniflood.htm

For more information on unicast flood blocking on the Catalyst 4500 running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/configur/uniflood.htm

Catalyst 6500 Unknown Unicast Flood Blocking (Cisco IOS)

In Catalyst 6500 Series switches running Cisco IOS software, the unicast flood blocking feature is called Unknown Unicast Flood Blocking (UUFB).

By default, all the Layer 2 ports on the switch are configured to allow unicast flooding.

To block unicast flooding on Layer 2 ports, perform the following steps:

Step 1 Enter global configuration mode and select the interface to configure.

```
Router# configure terminal
Router(config)# interface {{type slot/port} | {port-channel number}}
```

Step 2 Enable Unknown unicast flood blocking on the port using the **switchport block unicast** command.

```
Router(config-if)# switchport block unicast
```

This example shows how to configure UUFB on Fast Ethernet port 5/12:

```
Router# configure terminal
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport block unicast
```

For more information on unknown unicast flood blocking on the Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/blocking.htm

Catalyst 4500 Port Unicast and Multicast Flood Blocking (Cisco IOS)

Catalyst 4500 Series switches running Cisco IOS software support port unicast and multicast flood blocking. As the name indicates, this feature is capable of blocking multicast floods in addition to unicast floods. By default, all the ports on the switch are configured to allow unicast and multicast flooding. Unicast flood blocking and multicast flood blocking can be enabled independently.

To block unicast or multicast flooding on switch ports, perform the following steps:

Step 1 Enter global configuration mode and select the interface to configure.

```
Switch# configure terminal
Switch(config)# interface interface-id
```

Step 2 Enable unicast flood blocking on the port using the **switchport block unicast** command.

```
Switch(config-if)# switchport block unicast
```

Step 3 Enable multicast flood blocking on the port using the **switchport block multicast** command.

```
Switch(config-if)# switchport block multicast
```

This example shows how to block unicast and multicast flooding on a GigabitEthernet interface1/1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# switchport block multicast
Switch(config-if)# switchport block unicast
Switch(config-if)# end
```

For more information on port unicast and multicast flood blocking on the Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/uniflood.htm

DHCP Snooping

DHCP snooping is a security feature capable of intercepting DHCP messages crossing a switch and blocking bogus DHCP offers. DHCP snooping is available on Catalyst 6500 Series switches running Cisco IOS and Catalyst OS software, and on Catalyst 4500 Series switches running Cisco IOS software. DHCP snooping is required by other security features such as IP Source Guard and Dynamic ARP Inspection.

DHCP snooping uses the concept of trusted and untrusted ports. Typically, the trusted ports are used to reach DHCP servers or relay agents, while untrusted ports connect to clients. All DHCP messages are allowed on trusted ports, while only DHCP client messages are accepted on untrusted ports. Because neither servers nor relay agents are supposed to connect to untrusted ports, server messages like DHCPOFFER, DHCPACK, and DHCPNAK are dropped on untrusted ports. In addition, DHCP snooping builds and maintains a MAC-to-IP binding table that is used to validate DHCP packets received from untrusted ports. DHCP snooping discards all untrusted DHCP packets not consistent with the information in the binding table.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces.

The DHCP-snooping binding table contains the MAC address, IP address, lease time in seconds, and VLAN port information for the DHCP clients on the untrusted ports of a switch. The information that is contained in a DHCP-snooping binding table is removed from the binding table when its lease expires or DHCP snooping is disabled in the VLAN.

The switch drops DHCP packets when any of these situations occur:

- The switch receives a packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet from an untrusted port.
- The switch receives a packet on an untrusted port, and the source MAC address and the DHCP client hardware address do not match (only on Catalyst 6500 with MAC address verification enabled).
- The switch receives a DHCPRELEASE or DHCPDECLINE message that contains a MAC address in the DHCP snooping binding table, but the port information in the binding table does not match the port on which the message was received.
- The switch receives a DHCP packet that includes a relay agent IP address that is not 0.0.0.0.
- The switch receives a packet that includes Option-82 information on an untrusted port.

Conceptually, DHCP snooping on Catalyst 6500 and Catalyst 4500 Series switches work the same way. However, there are some differences that should be noted:

Catalyst 6500 supports MAC address verification. When this feature is enabled, every time the switch receives a packet on an untrusted port, and if the port belongs to a VLAN in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If addresses match, the switch forwards the packet. If the addresses do not match, the switch drops the packet. MAC address verification is available on Catalyst 6500 Series switches and is enabled by default in both, Cisco IOS and Catalyst OS. In Catalyst OS this feature is called the MAC-Address Matching option. This feature is not supported on Catalyst 4500.

Catalyst 6500 Series switches have the capacity to insert Option-82 information as DHCP packets are received. Catalyst 4500 Series switches do not insert Option-82 information, but they can be configured to accept DHCP requests with Option-82 information inserted by downstream switches.

Other functional differences between Catalyst 6500 and Catalyst 4500 Series switches are described in the following sections.

Catalyst 6500 DHCP Snooping (Catalyst OS)

DHCP snooping is disabled by default, and in switches running Catalyst OS software it is configured per VLAN. The configuration of DHCP snooping on a VLAN requires the use of a new or an existing VLAN ACL (VACL), to which a special DHCP-snooping access control entry (ACE) is appended. This special DHCP-snooping ACE tells the switch to turn DHCP snooping on the VLAN to which the VACL is mapped. By default, all ports on the affected VLAN are treated as untrusted. You need to statically configure the ports connecting to DHCP servers or relay agents as trusted.

It is important to note that the VACL used for DHCP-snooping is a regular VACL, which affects all traffic entering the associated VLAN. For this reason, you must analyze where to position the DHCP snooping ACE in the VACL. For example, if you want to deny the DHCP packets that come from a certain host and perform DHCP snooping for the other DHCP packets, then you must place a deny ACE before the DHCP-snooping ACE.

In addition, DHCP Snooping on Catalyst 6500 implements a hardware-based rate limiting function that controls the amount of DHCP packets to be processed by the supervisor engine. This rate limiting function is set by default to 1000 pps, which is shared with ARP inspection and 802.1X-DHCP, and which can be changed by configuration. DHCP rate limiting is supported on PFC2 and later versions. Refer to The ACL Feature (ARP Inspection, DHCP Snooping, 802.1x), page 82 for more information.

Note In Catalyst 6500 Series switches, DHCP snooping is supported on all supervisors.

To enable DHCP snooping on a Catalyst 6500 Series switch running Catalyst OS, perform the following steps:

Step 1 Add DHCP snooping to the VACL using the **set security acl ip permit dhcp-snooping** command:

```
Console> (enable) set security acl ip acl_name permit dhcp-snooping
```

Step 2 Configure the VACL to allow DHCP snooping from all hosts. Use the **set security acl ip** command:

```
Console> (enable) set security acl ip acl_name permit ip any any
```

Step 3 Save the VACL by executing the **commit security acl** command:

```
Console> (enable) commit security acl acl_name
```

Step 4 Apply the VACL to a VLAN using the **set security acl map** command:

```
Console> (enable) set security acl map acl_name vlan_id
```

Step 5 Set ports connecting to DHCP servers and relay agents as trusted using the **set port dhcp-snooping trust enable** command:

```
Console> (enable) set port dhcp-snooping mod/ports {trust | source-guard} {enable |
disable}
```

Step 6 Optionally, enable the MAC-Address Matching option (in case it has been disabled). Use the **set dhcp-snooping match-mac enable** command:

```
Console> (enable) set dhcp-snooping match-mac enable
```

Step 7 Optionally, enable DHCP snooping host-tracking information Option-82 feature when the DHCP clients and servers do not reside in the same subnet or network, and the switch seats between them. By enabling host-tracking information Option-82, every time the switch receives a DHCP request, it adds the Option-82 information in the packet. The Option-82 information contains the switch MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption). Use the **set dhcp-snooping information host-tracking enable** command to enable the host-tracking option:

```
Console> (enable) set dhcp-snooping information host-tracking enable
```

Step 8 Optionally, adjust the rate limit value for DHCP snooping using the **set security acl feature ratelimit** command:

```
Console> (enable) set security acl feature ratelimit rate
```

Note 802.1X-DHCP and DHCP snooping are mutually exclusive. You should not configure a VLAN for both 802.1X-DHCP and DHCP snooping.

This example shows how to enable DHCP snooping for VLAN 10 with a DHCP server on port 1/2:

```
Console> (enable) set security acl ip dhcpsnoop permit dhcp-snooping
Successfully configured DHCP Snooping for ACL dhcpsnoop. Use the 'commit' command to save
changes.
Console> (enable) set security acl ip dhcpsnoop permit ip any any
dhcpsnoop editbuffer modified. Use the 'commit' command to apply changes.
Console> (enable) commit security acl dhcpsnoop
ACL commit in progress.
ACL 'dhcpsnoop' successfully committed.
Console> (enable) set security acl map dhcpsnoop 10
Mapping in progress.
ACL dhcpsnoop successfully mapped to VLAN 10.
Console> (enable) set port dhcp-snooping 1/2 trust enable
Port(s)  1/2 state set to trusted for DHCP Snooping.
```

For more information on DHCP snooping on Catalyst 6500 Series switches running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/dhcp.htm

Catalyst 6500 and Catalyst 4500 DHCP Snooping (Cisco IOS)

In Cisco IOS, DHCP snooping is disabled by default. The configuration of DHCP snooping requires first enabling the DHCP snooping feature globally, and then on the necessary VLANs. Activating DHCP snooping on VLANs does not require the use of VACLs, as is the case with Catalyst OS.

Note In Catalyst 6500 Series switches, DHCP snooping requires Supervisor 2, Supervisor 32, or Supervisor 720.

Additionally in Cisco IOS, DHCP Snooping implements a software-based rate limiting function that controls the number of DHCP packets a port can receive. This rate limiting function is disabled by default but can be enabled by configuration. DHCP snooping puts ports where the rate limit is exceeded into the error-disabled state. Cisco recommends not configuring the rate limit to more than 100 packets per second on an untrusted port. The recommended rate limit for each untrusted client is 15 packets per second.

Note Normally, the rate limit applies to untrusted ports. If you want to set up rate limiting for trusted ports, keep in mind that trusted ports aggregate all DHCP traffic in the switch, and you will need to adjust the rate limit to a higher value. You should fine tune this threshold depending on the network configuration. The CPU should not receive DHCP packets at a sustained rate of more than 1,000 packets per second.

The configuration of DHCP snooping in Cisco IOS for the Catalyst 6500 and 4500 Series switches is identical, with the exception of DHCP Option-82 and MAC address verification.

Catalyst 6500 Series switches have the capacity to insert Option-82 information as DHCP packets are received. This is especially useful when the switch is deployed at the access-level network, such as a wiring closet. Catalyst 4500 Series switches are typically used as aggregation switches. With that in mind, Catalyst 4500 Series switches do not insert Option-82 information, but they can be configured to accept DHCP requests with Option-82 information inserted by downstream switches.

MAC address verification is an available feature on Catalyst 6500 and not on Catalyst 4500. When enabled, DHCP snooping verifies that the source MAC address in DHCP packets that are received on untrusted ports match the client hardware address in the packet.

Note We recommend enabling DHCP snooping during a maintenance window. When DHCP snooping is enabled globally, DHCP requests are dropped until the ports are configured.

To enable DHCP snooping on a Catalyst 6500 or 4500 Series switch running Cisco IOS, perform the following steps:

Step 1 Enable DHCP snooping globally using the **ip dhcp snooping** command:

```
Router(config)# ip dhcp snooping
```

Step 2 Enabling DHCP Snooping on the necessary VLANs using the **ip dhcp snooping vlan** command:

```
Router(config)# ip dhcp snooping vlan {{vlan_ID [vlan_ID]} | {vlan_range}
```

Step 3 Set the interfaces connecting to DHCP servers and relay agents as trusted using the **ip dhcp snooping trust interface** command:

```
Router(config)# interface {type1  slot/port | port-channel number}
Router(config-if)# ip dhcp snooping trust
```

Step 4 Optionally, configure DHCP snooping rate limiting on a Layer 2 LAN interface using the **ip dhcp snooping trust interface** command:

```
Router(config)# interface {type1  slot/port | port-channel number}
Router(config-if)# ip dhcp snooping limit rate
```

Step 5 Optionally, for Catalyst 6500, enable DHCP Snooping MAC Address Verification. Use the **ip dhcp snooping verify mac-address** command:

```
Router(config)# ip dhcp snooping verify mac-address
```

Step 6 Optionally, for Catalyst 6500, enable DHCP Option-82 data insertion when the DHCP clients and servers do not reside in the same subnet or network, and when the switch seats between them. Use the **ip dhcp snooping information option** command:

```
Router(config)# ip dhcp snooping information option
```

Step 7 Optionally, for Catalyst 4500, when used as an aggregation switch, configure the switch to accept DHCP requests with Option-82 information from any snooping untrusted port. Use the **ip dhcp snooping information option** command:

```
Switch(config)# ip dhcp snooping information option allow-untrusted
```

This example shows how to enable DHCP snooping for VLANs 10 through 12 with a DHCP server on interface FastEthernet 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping
Router(config)# ip dhcp snooping vlan 10 12
Router(config)# interface FastEthernet 5/12
Router(config-if)# ip dhcp snooping trust
```

For more information on DHCP snooping on Catalyst 6500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/snoodhcp.htm

For more information on DHCP snooping on Catalyst 4500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/dhcp.htm

IP Source Guard

IP Source Guard is a security feature available on Catalyst 6500 Series switches running Catalyst OS, and on Catalyst 4500 Series switches running Cisco IOS software. IP source guard prevents IP spoofing by allowing only the IP addresses that are obtained through DHCP snooping on a particular port. Initially, all IP traffic on the port is blocked except for the DHCP packets that are captured by DHCP snooping. When a client receives a valid IP address from the DHCP server, a port access control list (PACL) is installed on the port that permits the traffic from the IP address. This process restricts the client IP traffic to those source IP addresses that are obtained from the DHCP server; any IP traffic with a source IP address other than that in the PACLs permit list is filtered out. This filtering limits the ability of a host to attack the network by claiming a neighbor host's IP address.

IP Source Guard in Catalyst OS (Catalyst 6500)

There are some considerations that need to be taken into account before enabling IP Source Guard on Catalyst 6500 running Catalyst OS:

- IP Source Guard requires DHCP snooping to be configured
- IP source guard is supported on PFC 3 and later versions.
- A maximum of 10 IP addresses are allowed per port.
- IP source guard is not recommended on trunk ports.

- IP source guard cannot coexist with PACLs.
- IP source guard is not supported on EtherChannel-enabled ports, and EtherChannel is not supported on IP source guard-enabled ports.
- VLAN-based ACL features, such as static ARP inspection, are disabled when you enable IP source guard.

Note We recommend that you enable high availability when using dynamic ARP inspection (DAI), DHCP snooping, and IP source guard. If high availability is not enabled, clients have to renew their IP addresses for these features to work after a switchover.

Note In Catalyst 6500 Series switches, IP Source Guard is supported on Supervisor 32, and Supervisor 720.

To enable IP Source Guard on Catalyst 6500 Series switches running Catalyst OS, perform the following steps:

Step 1 Configure the port as port based using the **set port security-acl port-based** command:

```
Console> (enable) set port security-acl mod/ports port-based
```

Step 2 Enable IP source guard. Use the **set port dhcp-snooping source-guard enable** command:

```
Console> (enable) set port dhcp-snooping mod/ports source-guard enable
```

Step 3 Enable DHCP snooping using the **set security acl ip permit dhcp-snooping** command:

```
Console> (enable) set security acl ip acl_name permit dhcp-snooping
```

Step 4 Allow the port to forward other traffic. Use the **set security acl ip** command:

```
Console> (enable) set security acl ip acl_name permit ip any any
```

Step 5 Save the ACL configuration by executing the **commit security acl** command:

```
Console> (enable) commit security acl acl_name
```

Step 6 Enable the ACL on the VLAN. Use the **set security acl map** command:

```
Console> (enable) set security acl map acl_name vlan_id
```

Step 7 Enable DHCP-snooping trust on a port, using the **set port dhcp-snooping** command:

```
Console> (enable) set port dhcp-snooping mod/ports trust enable
```

Note Before you can enable IP source guard, you must enable DHCP snooping on the VLAN to which the port belongs. You must configure the port as either port based or in merge mode for security ACLs. You should only enable IP source guard on DHCP-snooping untrusted ports.

This example shows how to enable IP source guard:

```
Console> (enable) set port security-acl 3/1 port-based
Warning:Vlan-based ACL features will be disabled on port 3/1.
ACL interface is set to port-based mode for port(s) 3/1.
Console> (enable) set port dhcp-snooping 3/1 source-guard enable
```

```
IP Source Guard enabled on port(s)  3/1.
Console> (enable) set port dhcp-snooping 1/2 trust enable
Port(s)  1/2 state set to trusted for DHCP Snooping.
Console> (enable) set security acl ip dhcpsnoop permit dhcp-snooping
Successfully configured DHCP Snooping for ACL dhcpsnoop. Use the 'commit' command to save
changes.
Console> (enable) set security acl ip dhcpsnoop permit ip any any
dhcpsnoop editbuffer modified. Use the 'commit' command to apply changes.
Console> (enable) commit security acl dhcpsnoop
ACL commit in progress.
ACL 'dhcpsnoop' successfully committed.
Console> (enable) set security acl map dhcpsnoop 10
Mapping in progress.
ACL dhcpsnoop successfully mapped to VLAN 10.
Console>
```

For more information on IP Source guard on Catalyst 6500 Series switches running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/dhcp.htm

IP Source Guard in Cisco IOS (Catalyst 4500)

IP Source Guard in Catalyst 4500 Series switches running Cisco IOS allows the configuration of static IP source bindings, providing support to systems that have fixed IP addresses and which do not use DHCP (typically servers).

IP Source Guard in Catalyst 4500 Series switches supports Layer 2 ports only, including both access and trunk. For each untrusted Layer 2 port, there are two levels of IP traffic security filtering:

- Source IP address filter—IP traffic is filtered based on its source IP address. Only IP traffic with a source IP address that matches the IP source binding entry is permitted. An IP source address filter is changed when a new IP source entry binding is created or deleted on the port. The per-port VACL (PVACL) will be recalculated and reapplied in the hardware to reflect the IP source binding change. By default, if the IP filter is enabled without any IP source binding on the port, a default PVACL that denies all IP traffic is installed on the port. Similarly, when the IP filter is disabled, any IP source filter PVACL will be removed from the interface.
- Source IP and MAC address filter—IP traffic is filtered based on its source IP address and its MAC address; only IP traffic with source IP and MAC addresses matching the IP source binding entry are permitted.

To enable IP Source Guard on Catalyst 4500 Series switches running Cisco IOS, perform the following steps:

Step 1 Enable DHCP snooping globally using the **ip dhcp snooping** command:

```
Switch(config)# ip dhcp snooping
```

Step 2 Enable DHCP snooping on the necessary VLANs. Use the **ip dhcp snooping vlan** command:

```
Switch(config)# ip dhcp snooping vlan number [number]
```

Step 3 Configure the interface as trusted or untrusted using the **ip dhcp snooping vlan** command:

```
Switch(config-if)# [no] ip dhcp snooping trust
```

Step 4 Enable IP source guard, source IP, and source MAC address filtering on the port. Use the **ip source binding** command:

```
Switch(config-if)# ip verify source vlan dhcp-snooping port-security
```

Step 5 Optionally, configure a static IP binding on the port. Use the **ip verify source vlan dhcp-snooping port-security** command:

```
Switch(config)# ip source binding mac-address vlan vlan-id ip-address interface
interface-name
```

This example shows how to enable per-Layer 2-port IP source guard on VLANs 10 through 20:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10 20
Switch(config)# interface fa6/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan 10
Switch(config-if)# switchport trunk allowed vlan 11-20
Switch(config-if)# no ip dhcp snooping trust
Switch(config-if)# ip verify source vlan dhcp-snooping
Switch(config)# end
```

For more information on IP Source Guard on Catalyst 4500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/dhcp.htm#wp1083306

Dynamic ARP Inspection (DAI)

Dynamic ARP Inspection (DAI) is a security feature that is available on Catalyst 6500 Series switches running Cisco IOS software or Catalyst OS, and on Catalyst 4500 Series switches running Cisco IOS software. Dynamic ARP inspection helps prevent ARP poisoning and other ARP-based attacks by intercepting all ARP requests and responses, and by verifying their authenticity before updating the switch's local ARP cache or forwarding the packets to the intended destinations.

Note In Catalyst 6500 Series switches, Dynamic ARP requires Supervisor 2, Supervisor 32, or Supervisor 720.

The DAI verification consists primarily of intercepting each ARP packet and comparing its MAC address and IP address information against the MAC-IP bindings contained in a trusted binding table. DAI discards any ARP packets that are inconsistent with the information contained in the binding table. The trusted binding table is dynamically populated by DHCP snooping when this feature is enabled. In addition, DAI allows the configuration of static ARP ACLs to support systems that use statically configured IP addresses and that do not rely on DHCP.

DAI can also be configured to drop ARP packets with invalid IP addresses, such as 0.0.0.0 or 255.255.255.255, and ARP packets containing MAC addresses in their payloads that do not match the addresses specified the Ethernet headers.

Another important feature of DAI is that it implements a configurable rate-limit function that controls the number of incoming ARP packets. This function is particularly important because all validation checks are performed by the CPU, and without a rate-limiter, there could be a DoS condition.

Similarly to DHCP snooping, DAI associates a trust state with each interface on the system. Packets arriving on trusted interfaces bypass all DAI validation checks, while those arriving on untrusted interfaces go through the DAI validation process. In a typical network configuration for DAI, all ports connected to host ports are configured as untrusted, while all ports connected to switches are configured as trusted. With this configuration, all ARP packets entering the network from a given switch will have passed the security check. By default, DAI is disabled on all VLANs, and all ports are configured as untrusted.

As previously mentioned, DAI populates its database of valid MAC address to IP address bindings through DHCP snooping. It also validates ARP packets against statically configured ARP ACLs. It is important to note that ARP ACLs have precedence over entries in the DHCP snooping database. ARP packets are first compared to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, then the packet will be denied even if a valid binding exists in the database populated by DHCP snooping.

Cisco IOS and Catalyst OS implement DAI in a similar manner. However, there are some functional differences that should be noted, and which are covered in the following sections.

Dynamic ARP Inspection (DAI) in Catalyst OS (Catalyst 6500)

DAI provides an additional validation function capable of identifying and discarding ARP packets containing MAC addresses in their bodies that are not consistent with the Ethernet headers. In Catalyst OS this validation is based on the source MAC addresses. That is, DIA drops ARP packets whose source Ethernet MAC addresses (in the Ethernet headers) are not the same as the source MAC addresses in the ARP headers. In Cisco IOS this validation can be based on both source and destination MAC addresses.

DAI implements a rate-limit function that controls the number of incoming ARP packets. In Catalyst OS, this function can be configured globally and/or on a per port basis:

- Global ARP inspection rate limit—By default, the global ARP inspection rate limit is set to 500 pps. Packets exceeding the limit are discarded. The limit can be configured with a minimum value of 1, and a maximum value of 1000 pps. For Supervisor Engine 720, the minimum value that is enforced by the hardware is 10 pps (values between 1- 9 are set to 10). The global ARP inspection rate limit can be disabled by setting the limit value to 0.
- Per-port ARP inspection rate limit—The pert-port rate limit uses a drop threshold and a shutdown threshold. If the rate exceeds the drop threshold, the excess packets are dropped (and counted toward the shutdown threshold limit). If the rate exceeds the shutdown threshold, the port that is specified by mod/port is shut down. By default, both threshold values are 0 (no per-port rate limiting is applied). The maximum value for both thresholds is 1000 pps.

To configure Dynamic ARP Inspection on switches running Catalyst OS, perform the following steps:

Step 1 Enable dynamic ARP inspection on a per-VLAN basis using the **set security acl arp-inspection dynamic** command:

```
Console> (enable) set security acl arp-inspection dynamic enable {vlan-list}
```

By default, dynamic ARP inspection is disabled on all VLANs.

Step 2 Configure the port trust state using the **set port arp-inspection trust enable** command:

```
Console> (enable) set port arp-inspection portlist trust {enable | disable}
```

To make the interfaces untrusted, use the **set port arp-inspection trust disable** command. By default, all interfaces are untrusted.

In this example, dynamic ARP inspection is configured on VLAN 100, while port 2/2 is set as trust:

```
Console> (enable) set security acl arp-inspection dynamic enable 100
Dynamic ARP Inspection is enabled for vlan(s) 100.
Console> (enable) set port arp-inspection 2/2 trust enable
Port(s)  2/2 state set to trusted for ARP Inspection.
```

To configure ARP Inspection for non-DHCP systems, perform the following steps:

Step 1 Configure the ARP traffic-inspection ACL with the valid static MAC-IP binding entries. By default, no ARP traffic-inspection access lists are defined. Define the ARP traffic-inspection ACL using the **set security acl ip acl_name arp-inspection host** command:

```
Console> (enable) set security acl ip acl_name {permit | deny} arp-inspection host
ip_address mac_address
```

Step 2 Commit the previously defined ARP traffic-inspection ACL using the **commit** command:

```
Console> (enable) commit security acl {acl_name | all | adjacency}
```

Step 3 Apply the ARP traffic-inspection ACL to the VLAN using the **set security acl map** command:

```
Console> (enable) set security acl map acl_name vlan-list
```

By default, no defined ARP traffic-inspection ACLs are applied to any VLAN.

Step 4 Configure the port trust state using the **set port arp-inspection trust enable** command:

```
Console> (enable) set port arp-inspection portlist trust {enable | disable}
```

Step 5 To make the interfaces untrusted, use the **set port arp-inspection trust disable** command. By default, all interfaces are untrusted.

This example shows how to configure an ARP traffic-inspection ACL called TrustedHosts, to permit ARP packets from a host with IP address 172.20.52.54 and MAC address 00-01-64-61-39-c2, to apply the ACL to VLAN 100, and to configure port 2/2 on the switch as trusted:

```
Console> (enable) set security acl ip TrustedHosts permit arp-inspection host 172.20.52.54
00-01-64-61-39-c2
Operation successful.
Console> (enable) commit security acl TrustedHosts
Console> (enable) ACL commit in progress.
ACL ' TrustedHosts' successfully committed.
Console> (enable) set security acl map TrustedHosts 100
ACL TrustedHosts mapped to vlan 10
Console> (enable) set port arp-inspection 2/2 trust enable
Port(s)  2/2 state set to trusted for ARP Inspection.
```

To optionally configure DAI additional checks on source MAC addresses, use the **set security acl arp-inspection match-mac** command:

```
Console> (enable) set security acl arp-inspection match-mac {enable drop [log]}
Console> (enable) commit security acl {acl_name | all | adjacency}
```

This example shows how to drop the packets where the source Ethernet MAC address is not the same as the source MAC address in the ARP header:

```
Console> (enable) set security acl arp-inspection match-mac enable drop
ARP Inspection match-mac feature enabled with drop option.
Console> (enable) commit security acl TrustedHosts
Console> (enable) ACL commit in progress.
ACL ' TrustedHosts' successfully committed.
```

To optionally configure DAI to drop the packets with invalid MAC or IP addresses use the **set security acl arp-inspection address-validation** command. The following MAC addresses are considered invalid 00-00-00-00-00-00, multicast MAC addresses (the 48th bit is set), and ff-ff-ff-ff-ff-ff (this is a special-case multicast MAC address). The following IP addresses are considered invalid 0.0.0.0, 255.255.255.255, and class D (multicast) IP addresses.

```
Console> (enable) set security acl arp-inspection address-validation {enable drop [log]}
Console> (enable) commit security acl {acl_name | all | adjacency}
```

This example shows how to drop the packets with the invalid MAC or IP addresses:

```
Console> (enable) set security acl arp-inspection address-validation enable drop
ARP Inspection address-validation feature enabled with drop option.
Console> (enable) commit security acl TrustedHosts
Console> (enable) ACL commit in progress.
ACL ' TrustedHosts' successfully committed.
```

To change the default global ARP packet rate limiting configuration, use the set security acl feature ratelimit command. By default, the ARP traffic-inspection traffic is rate limited to 500 packets per second. The minimum value is 1, and the maximum value is 1000 packets per second. For Supervisor Engine 720, the minimum value that is enforced by the hardware is 10 packets per second (values between 1- 9 are set to 10). To disable rate limiting, set the value to 0:

```
Console> (enable) set security acl feature ratelimit rate
```

This example shows how to rate limit the number of ARP traffic-inspection packets that are sent to the CPU to 1000:

```
Console> (enable) set security acl feature ratelimit 1000
Dot1x DHCP and ARP Inspection global rate limit set to 1000 pps.
Console> (enable)
```

To enable a per-port ARP packet rate limit, perform the following steps:

Step 1 Use the **set port arp-inspection** command to define a drop-threshold and a shutdown-threshold. If the rate exceeds the drop-threshold, the excess packets are dropped (and counted toward the shutdown-threshold limit). If the rate exceeds the shutdown-threshold, the port that is specified by mod/port is shut down. The maximum value for both thresholds is 1000 packets-per second (pps):

```
Console> (enable) set port arp-inspection mod/port drop-threshold pps shutdown-threshold
pps
```

Step 2 Optionally, enable error recovery from the dynamic ARP inspection error-disable state. By default, every time the rate of incoming ARP packets exceeds the shutdown threshold, the switch places the port in the error-disabled state. Enabling error-disable recovery allows ports to automatically emerge from this state after a specified timeout period. Use the **set errdisable-timeout** command. Valid values for the timeout interval are from 30 to 86400 seconds (30 seconds to 24 hours):

```
Console> (enable) set errdisable-timeout enable arp-inspection
Console> (enable) set errdisable-timeout interval {interval}
```

This example shows how to set an errdisable timeout of 450 seconds for ARP inspection events:

```
Console> (enable) set errdisable-timeout enable arp-inspection
Successfully enabled errdisable-timeout for arp-inspection.
Console> (enable) set errdisable-timeout interval 450
Successfully set errdisable timeout to 450 seconds.
Console> (enable)
```

For more information on Dynamic ARP Inspection on Catalyst 6500 Series switches running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/acc_list.htm

Dynamic ARP Inspection (DAI) in Cisco IOS (Catalyst 6500 and Catalyst 4500)

As with Catalyst OS, in Cisco IOS, DAI can be configured to drop ARP packets containing MAC addresses in their bodies that do not match the addresses specified in the Ethernet headers. The difference is that in Cisco IOS the MAC address validation can be done based on source and destination MAC addresses:

- Source MAC addresses: DAI checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body. This check is performed on both ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.
- Destination MAC addresses: DAI checks the destination MAC address in the Ethernet header against the target MAC address in ARP body. This check is performed for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

Similarly to Catalyst OS, in Cisco IOS, DAI can be configured to drop ARP packets with invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

Another difference between Catalyst OS and Cisco IOS is that in the later ARP inspection rate limiting is implemented per interface only, and not globally. The default threshold is also different, in Cisco IOS the rate for untrusted interfaces is by default set to 15 pps.

To configure Dynamic ARP Inspection on switches running Cisco IOS, perform the following steps:

Step 1 Enable dynamic ARP inspection on a per-VLAN basis. By default, dynamic ARP inspection is disabled on all VLANs. Use the **ip arp inspection vlan** command:

```
Switch(config)# ip arp inspection vlan vlan-range
```

Step 2 Use the **ip arp inspection trust** command to configure the port trust state. By default, all interfaces are untrusted. To set the interfaces to untrusted, use the **no ip arp inspection trust** command:

```
Switch(config)# interface interface-id
Switch(config-if)# [no] ip arp inspection trust
```

In this example, dynamic ARP inspection is configured in VLAN 100, while interface G3/48 is set as trust:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip arp inspection vlan 100
Switch(config)# interface g3/48
Switch(config-if)# ip arp inspection trust
Switch(config-if)# end
```

To configure ARP Inspection for non-DHCP systems, perform the following steps:

Step 1 Configure the ARP ACL with the valid static MAC-IP binding entries. By default, no ARP access lists are defined. Define the ARP ACL using the **arp access-list** command:

```
Switch(config)# arp access-list acl-name
Switch(config-arp)# permit ip host sender-ip mac host sender-mac [log]
Switch(config-arp)# exit
```

Step 2 Apply the ARP ACL to the VLAN. By default, no defined ARP ACLs are applied to any VLAN.Use the **ip arp inspection filter** command:

```
Switch(config)# ip arp inspection filter arp-acl-name vlan vlan-range [static]
```

Step 3 Configure the port trust state. By default, all interfaces are untrusted. Use the **ip arp inspection trust** command. To make the interfaces untrusted, use the **no ip arp inspection trust** command:

```
Switch(config)# interface interface-id
Switch(config-if)# [no] ip arp inspection trust
```

This example shows how to configure an ARP ACL called TrustedHosts, to permit ARP packets from a host with IP address 170.1.1.2 and MAC address 2.2.2, to apply the ACL to VLAN 100, and to configure interface fastethernet3/48 on the switch as untrusted:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# arp access-list TrustedHosts
Switch(config-arp-nacl)# permit ip host 170.1.1.2 mac host 2.2.2
Switch(config-arp-nacl)# exit
Switch(config)# ip arp inspection filter TrustedHosts vlan 100 static
Switch(config)# interface fa3/48
Switch(config-if)# no ip arp inspection trust
Switch(config-if)# end
```

To optionally configure DAI additional checks on destination MAC addresses, sender and target IP addresses, or source MAC addresses, use the **ip arp inspection validate** global configuration command:

```
Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}
```

This example shows how to configure source mac validation. Packets are dropped and an error message might be generated when the source address in the Ethernet header does not match the sender hardware address in the ARP body:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip arp inspection validate src-mac
Switch(config)# exit
```

To change the default configuration of ARP packet rate limiting, perform the following steps:

Step 1 Use the **ip arp inspection limit interface** command to modify the default rate of 15 pps.

- Use the **rate pps** option to specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps.
- With the **burst interval seconds** option, you can specify the consecutive interval in seconds, over which the interface is monitored for a high rate of ARP packets. The range is 1 to 15, and by default the burst interval is set to 1 second.
- Use **rate none** to specify no upper limit for the rate of incoming ARP packets that can be processed. This is equivalent to disabling ARP packet rate limiting:

```
Switch(config)# interface interface-id
Switch(config-if)# [no] ip arp inspection limit {rate pps [burst interval seconds] |
none}
Switch(config-if)# exit
```

Step 2 Optionally, enable error recovery from the dynamic ARP inspection error-disable state. By default, every time the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state.

Enabling error-disable recovery allows ports to automatically emerge from this state after a specified timeout period. Use the **errdisable recovery** global configuration command:

```
Switch(config)# errdisable recovery {cause arp-inspection | interval interval}
```

This example shows how to set an upper limit for the number of incoming packets (100 pps) and to specify a burst interval (1 second):

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface g3/31
Switch(config-if)# ip arp inspection limit rate 100 burst interval 1
Switch(config-if)# exit
Switch(config)# errdisable recovery cause arp-inspection
Switch(config)# exit
```

For more information on Dynamic ARP Inspection on Catalyst 4500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/dynarp.htm

For more information on Dynamic ARP Inspection on Catalyst 6500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/dynarp.htm

Control Plane Policing

Control Plane Policing (CoPP) is a security infrastructure feature available on Catalyst 6500 and 4500 Series switches running Cisco IOS that allows the configuration of QoS policies that rate limit the traffic handled by the main CPU of the switch. This protects the control plane of the switch from direct DoS attacks and reconnaissance activity. This section provides implementation information for CoPP on Supervisors 720 and 32 (Catalyst 6500), and Catalyst 4500 Series switches and provides additional deployment instructions and guidelines. It includes the following topics:

CoPP Technology Overview

Control Plane Policing (CoPP) protects Catalyst 6500 and 4500 switches by allowing the definition and enforcement of QoS policies that regulate the traffic processed by the main switch CPU (route or switch processor). With CoPP, these QoS policies are configured to permit, block, or rate limit the packets handled by the main CPU.

Packets handled by the main CPU, referred to as *control plane traffic*, typically include the following:

- Routing protocols
- Packets destined to the local IP address of the router
- Packets from network management protocols, such as SNMP
- Interactive access protocols, such as SSH, and telnet
- Other protocols, such as ICMP, or IP options, might also require handling by the switch CPU
- Layer 2 packets such as BPDU, CDP, DOT1X, and so on

CoPP leverages the modular QoS command-line interface (MQC) for its QoS policy configuration. MQC allows the classification of traffic into classes, and lets you define and apply distinct QoS policies to separately rate limit the traffic in each class. MQC lets you divide the traffic destined to the CPU into multiple classes based on different criteria. For example, four traffic classes could be defined based on relative importance: critical, normal, undesirable and default. After the traffic classes are defined, a QoS policy can be defined and enforced for each class according to importance. The QoS policies in each class can be configured to permit all packets, drop all packets, or drop only those packets exceeding a specific rate limit.

Note The actual number of classes should be chosen based on local network requirements, security policies, and a thorough analysis of the baseline traffic. Refer to Recommended CoPP Deployment Methodology, page 62 for more information.

Functionally, Catalyst 6500 and 4500 Series switches implement CoPP in a similar fashion. CoPP comes into play right after the switching or the routing decision, and before traffic is forwarded to the control plane. When CoPP is enabled, at a high level the sequence of events is as follows:

1. A packet enters the switch configured with CoPP on the ingress port.
2. The port performs any applicable input port and QoS services.
3. The packet gets forwarded to the switch CPU.
4. The switch CPU makes a routing or a switching decision, determining whether or not the packet is destined to the control plane.
5. Packets destined for the control plane are processed by CoPP, and are dropped or delivered to the control plane according to each traffic class policy. Packets that have other destinations are forwarded normally.

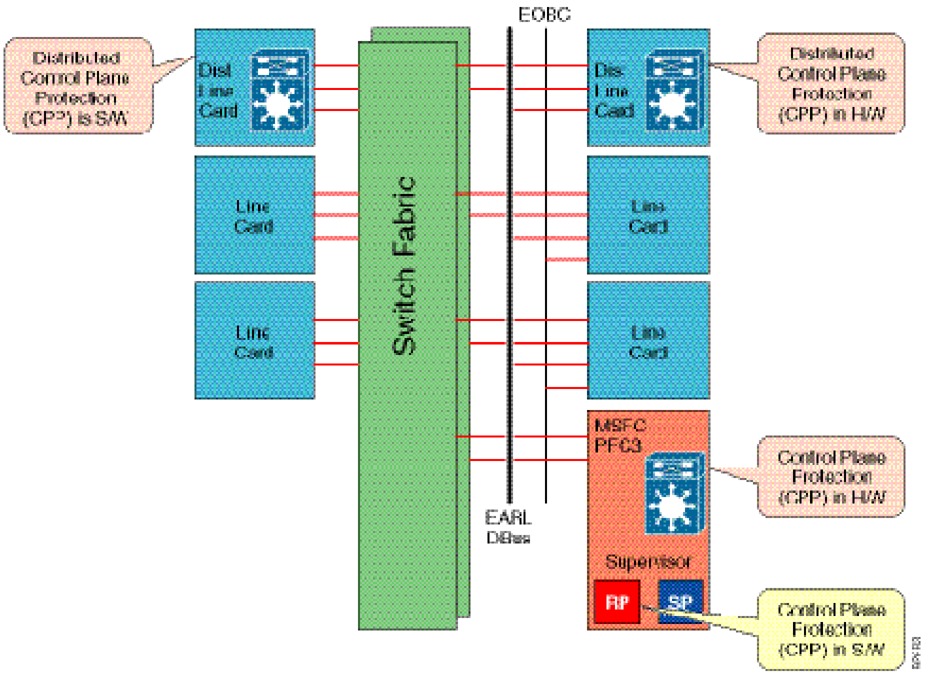
As previously stated, the Catalyst 6500 and Catalyst 4500 Series switches implement CoPP similarly. However, CoPP has been enhanced on both platforms to leverage the benefits of their hardware architectures, and as a result each platform provides unique features. The following sections provide information on CoPP implementations on Catalyst 6500 and Catalyst 4500 Series switches.

CoPP on Supervisors 720 and 32 (Catalyst 6500)

CoPP is supported on Catalyst 6500 Series switches and Cisco 7600 Series Routers with Supervisors 720 and 32 running Cisco IOS software. This section describes the implementation details of CoPP on Supervisors 720 and 32.

In the Catalyst 6500 Series switches, CoPP takes advantage of the processing power present on line-cards by implementing a distributed CoPP model. In this platform, the class QoS policies are centrally configured under the control-plane configuration mode. When configured, these policies are first applied at the route processor (MSFC) level, and then they get automatically pushed to the Policy Feature Card (PFC) and each Distributed Forwarding Card (DFC). This CoPP model is illustrated in Figure 4.

Figure 4 Supervisor 32/720 CoPP



Distributed Control Plane Protection (CPP) is S/W
Dist Line Card
Line Card
Line Card
Switch Fabric
EOBC
Dis Line Card
Distributed Control Plane Protection (CPP) in H/W
Line Card
Line Card
MSFC PFC3
Control Plane Protection (CPP) in H/W
EARL DBus
Supervisor
RP
SP
Control Plane Protection (CPP) in S/W

CoPP at the RP is performed in software while on the PFC and DFCs it is processed in hardware, with no performance degradation or increased latency. In this way, CoPP on Supervisors 32 and 720 provides two layers of protection: first, at wire speed on the PFC and DFCs, and second, at the RP level. This helps to ensure that only the amount of traffic specified by the user actually reaches the control plane.

Note The PFC3 and DFC3 provide hardware support for CoPP.

The Cisco Catalyst 6500 supports CoPP on the Supervisor 720 and Supervisor 32 in hardware starting with Cisco IOS release 12.2(18)SXD1. CoPP supports IPv4 in hardware, while multicast and broadcast traffic are only supported in software. Support for IPv6 traffic has been introduced in IOS release 12.2(18)SXE.

Note CoPP is not enforced in hardware unless MLS QoS is globally enabled using the **mls qos** command.

Another important characteristic of CoPP in Supervisors 720 and 32 is that it does not support the definition of non-IP traffic classes, with the exception of the class-default. Class-default is a default class for all remaining traffic destined to the RP that does not match any other class. This default class allows you to specify how to treat traffic that is not explicitly associated with any other user-defined classes. The class-default is the only class in CoPP capable of handling both IP and non-IP traffic. User-defined classes can only handle IP traffic.

Compared to the hardware-base rate limiters present on Supervisors 32 and 720 (refer to Hardware-Based Rate Limiters on Supervisors 32 and 720, page 72), CoPP provides more granularity and control. However, there are certain types of traffic that CoPP does not support in hardware, and for which the hardware-based rate limiters might provide better support. For example, CoPP supports multicast and broadcast traffic in software only, the available hardware-based rate limiters should be used instead. ARP is another good example. CoPP cannot rate limit ARP packets neither in software or hardware, the ARP policing rate limiter should be used instead. Other packet types not supported in hardware include packets with TTL equal to 1, packets that fail the MTU check, packets with IP options, and IP packets with errors.

CoPP helps protect the RP of Catalyst 6500 Series switches in more than one way. From a policing perspective, by filtering traffic sent to the RP, CoPP ensures that only the expected protocols are allowed. This effectively shields the control plane from unwanted and potentially malicious traffic. On the other hand, by rate limiting the traffic sent to the RP, CoPP provides protection against large volumes of packets that might be part of a DoS attack. This helps maintain network stability even during an attack.

For more information about CoPP on the Supervisors 32 and 720, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/dos.htm

Configuring CoPP on Supervisors 720 and 32 (Catalyst 6500)

To configure CoPP on Supervisors 720 and 32 (Catalyst 6500), perform the following steps:

Step 1 To enable CoPP on Supervisors 32 and 720, first enable MLS QOS:

```
Router(config)# mls qos
```

Step 2 Optionally, define the necessary ACLs to be used to match traffic classes:

```
Router((config)# ip access-list extended access-list-name
Router((config-ext-nacl)# {permit | deny} protocol source source-wildcard destination
destination-wildcard [precedence precedence] [tos tos] [established] [fragments]
```

Step 3 Classify the control plane traffic using the **class-map** command. The syntax for this command is as follows:

```
Router((config)# class-map traffic_class_name
```

The **class-map** command defines the class map name and enables a configuration mode for defining the class. Within the traffic class configuration mode, use the **match** command to associate specific traffic with the class. The syntax for this command is as follows:

```
Router((config-cmap)# match {ip precedence} |{ip dscp} | access-group
```

Note Currently only one match clause is supported per class-map in Supervisors 32 and 720.

Step 4 After the traffic is classified, you apply a policy action to each class, indicating whether to permit all packets, to drop all packets, or to drop packets crossing a specified rate limit for that particular class. To apply these policy actions use the **policy-map** command, which has the following syntax:

```
Router((config-pmap)# policy-map service_policy_name
```

The **policy-map** command defines the policy map name and enables a configuration mode for defining the policy. You then use the **class** command to associate one or more traffic classes with the policy. You use the **police** command to define the policy action to apply. The syntax for these commands is as follows:

```
Router((config-pmap)# class traffic_class_name
Router((config-pmap-c)# police cir bits-per-second [bc conform-burst-bytes] [be
excess-burst-bytes] [conform-action action] [exceed-action action]
```

Step 5 Apply the defined CoPP policy to the control plane by using the **service-policy** command from control-plane configuration mode.

```
Router((config)# control-plane
Router((config-cp)# service-policy input service_policy_name
```

Catalyst 6500 Series Switch CoPP Considerations and Restrictions

The following are important considerations and known restrictions that should be taken into account prior to configuring CoPP:

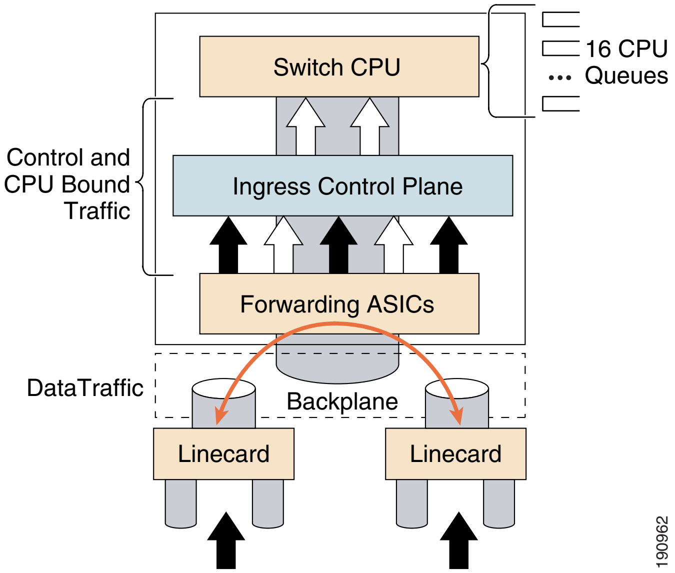
- Because CoPP relies on the QoS implementation, CoPP policies are downloaded to the PFC and DFCs only if QoS is enabled. For this reason, ensure that the **mls qos** command is enabled at the global configuration mode for the PFC and each DFC where CoPP is required.
- CoPP does not support the definition of non-IP traffic classes except for the class-default. ACLs can be used instead of non-IP classes to drop non-IP traffic. At the same time, class-default can be used to limit non-IP traffic that reaches the RP CPU.
- On Supervisors 32 and 720, ARP policing is done with a QoS rate limiter rather than CoPP. Even though there is a match protocol arp for CoPP on these supervisors, this type of traffic is processed in software. Therefore, ARP policing should be configured with the hardware-based QoS rate limiter using the **mls qos protocol arp police bps** command.
- Currently, only one match criteria is supported for each traffic class. To define multiple match rules with a match-any criteria, split the match access-group statements among multiple class maps instead of grouping them together.
- Prior to Cisco IOS software Release 12.2(18)SXE, only one match criteria was allowed for each traffic class. When using one of these earlier releases, to define multiple match rules with a match-any criteria, split the match access-group statements among multiple class maps instead of grouping them together.
- Prior to Cisco IOS software Release 12.2(18)SXE, the MQC class-default was not supported on Supervisor 720. This is a minor limitation because the class-default could be emulated with a normal class configured with an ip permit any rule.
- Omitting the policy parameters in a class causes the class to be handled by software-based CoPP. Use the **police** command and set the policy parameters to ensure the class is handled by hardware-based CoPP.

- Currently, multicast packets are handled only by the software-based CoPP at the RP level. However, there are CPU rate limiters available that can rate limit multicast packets to the CPU in hardware. These CPU rate limiters include the Multicast FIB-miss rate limiter and the Multicast Partial-SC rate limiter. These CPU rate limiters can be used in combination with ACLs and software CoPP to provide protection against multicast and DoS attacks.
- CoPP is not supported in hardware for broadcast packets. The combination of ACLs, traffic storm control, and CoPP software protection provides protection against broadcast DoS attacks.
- With PFC3A, egress QoS and CoPP cannot be configured at the same time. In this situation, CoPP is performed in software, and a warning message is generated.
- In the rare situation where a large QoS configuration is being used, it is possible that the system could run out of TCAM space. When this scenario occurs, CoPP can be performed in software. Use the **show platform hardware capacity** command to monitor TCAM space.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the switches. Filtering this traffic could prevent remote access to the switch, requiring a console connection.
- Supervisor Engines 32 and 720 support built-in special-case rate limiters, which are useful for situations where an ACL cannot be used (for example, TTL, MTU, and IP options). When you enable the special-case rate limiters, you should be aware that the special-case rate limiters will override the CoPP policy for packets matching the rate-limiter criteria.
- CoPP does not support ACEs with the **log** keyword.
- CoPP uses hardware QoS TCAM resources. Use the **show platform hardware capacity** and **show tcam utilization** commands to verify the TCAM use.
- ACE hit counters in hardware are only for ACL logic. You can rely on software ACE hit counters and the show access-list, show policy-map control-plane, and **show mls ip qos** commands to troubleshoot evaluate CPU traffic.

CoPP on Catalyst 4500

The Catalyst 4500 Series switches support CoPP on all supervisor engines compatible with Cisco IOS release 12.2(31)SG. In this platform CoPP is implemented in hardware in a centralized, non-distributed fashion. CoPP policies are centrally configured under the control-plane configuration mode, and then enforced in hardware by the classification TCAM and QoS policers of the supervisor engine. This CoPP model is shown in Figure 5.

Figure 5 Catalyst 4500 CoPP



Contrary to the Catalyst 6500, Catalyst 4500's CoPP supports the definition of non-IP traffic classes in addition to IP traffic classes. With this, instead of using the default class for handling all non-IP traffic, you can define separate policies for non-IP traffic. This results in better and more granular control over non-IP protocols, such as ARP, IPX, BPDUs, CDP, and SSTP.

As in other platforms, CoPP uses the modular QoS command-line interface (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.

One particular characteristic of Catalyst 4500's CoPP is that the CoPP policy must be named system-cpp policy. In fact, system-cpp-policy is the only policy-map that can be attached to the control-plane. To facilitate the configuration of the system-cpp-policy, Catalyst 4500's CoPP provides a global macro function (called system-cpp) that automatically generates and applies CoPP policies to the control-plane. The resulting configuration uses a collection of system defined class-maps for common Layer 3 and Layer 2 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 on any of the system predefined traffic classes. Table 2 lists the predefined system ACLs.

Table 2 Catalyst 4500 System Predefined ACLs

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-garp-range	MacDA = 0180.C200.0020 - 0180.C200.002F
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

Table 2 Catalyst 4500 System Predefined ACLs

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

In addition to the predefined classes, you can configure your own class maps matching other control-plane traffic. In order to take effect, these user-defined class maps need to be added to the system-cpp-policy policy-map.

For more information about CoPP on the Catalyst 4500 Series switches, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/cntl_pln.htm

Configuring CoPP on Catalyst 4500 Series Switches

To enable CoPP on Catalyst 4500 Series switches, perform the following steps:

Step 1 Enable QoS functionality:

```
Switch(config)# qos
```

Step 2 Run the **system-cpp** global macro to create the system-cpp-policy policy-map and attach it to the control-plane.:

```
Switch(config)# macro global apply system-cpp
```

Step 3 Optionally, define the necessary ACLs to be used to match your own traffic classes:

```
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]
```

Step 4 Next, classify the control plane traffic using the **class-map** command. The syntax for this command is as follows:

```
Switch(config)# class-map {traffic-class-name}
```

The **class-map** command defines the class map name and enables a configuration mode for defining the class. Within the traffic class configuration mode, use the **match** command to associate specific traffic with the class. The syntax for this command is as follows:

```
Switch(config-cmap)# match access-group {access-list-number | name {access-list-name}}
```

Step 5 After the traffic is classified, you apply a policy action to each class, indicating whether to permit all packets, to drop all packets, or to drop packets crossing a specified rate limit for that particular class. To apply these policy actions use the **policy-map** command, which has the following syntax:

```
Switch(config)# policy-map system-cpp-policy
```

The **policy-map** command defines the policy map name and enables a configuration mode for defining the policy. Remember that only the system-cpp-policy policy-map can be attached to the control-plane. You then use the **class** command to associate one or more traffic classes with the policy. You use the **police** command to define the policy action to apply. The syntax for these commands is as follows:

```
Switch(config-pmap)# class <class-map-name>
Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop |
transmit}] [{exceed-action {drop | transmit}}]
```

Catalyst 4500 Series Switch CoPP Considerations and Restrictions

The following are important considerations and known restrictions that should be taken into account prior configuring CoPP:

- Only ingress CoPP is supported, so only the **input** keyword is supported in control-plane related CLIs.
- Use the system-defined class maps for policing control plane traffic.
- ARP support is limited to Gratuitous ARPs (destination MAC in the 0180.C200.0020 - 0180.C200.002F range). Broadcast ARPs are not currently supported by CoPP.
- 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. When 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. However, 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.

Defining CoPP Traffic Classes

Developing a CoPP policy starts with the classification of the control plane traffic. To that end, the control plane traffic needs to be first identified and separated into different class-maps. The Catalyst 4500 Series switches provides the system-cpp macro which automatically generates a collection of class-maps for common Layer 3 and Layer 2 control plane traffic. While very useful, these predefined class-maps might not include all the necessary traffic classes reaching the control plane, and as a result they might need to be complemented with other user-defined class-maps. The Catalyst 6500 Series switches do not provide a configuration macro. Therefore, all class-maps need to be defined by the user.

This section presents a classification template that can be used as a model when implementing CoPP on Catalyst 6500 and Catalyst 4500 Series switches. This template presents a realistic classification, where traffic is grouped based on its relative importance and protocol type. The template uses nine different classes, which provide great granularity, and make it suitable for real-world environments. It is important to note that, even though you can use this template as a reference, the actual number and type of classes needed for a given network can differ and should be selected based on local requirements, security policies, and a thorough analysis of baseline traffic.

This template defines the following nine traffic classes:

1. Border Gateway Protocol (BGP)

   This class defines traffic that is crucial to maintaining neighbor relationships for BGP routing protocol, such as BGP keepalives and routing updates. Maintaining BGP routing protocol is crucial to maintaining connectivity within a network or to an ISP. Sites that are not running BGP would not use this class.

2. Interior Gateway Protocol (IGP)

   This class defines traffic that is crucial to maintaining IGP routing protocols such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP) and Routing Information Protocol (RIP). Maintaining IGP routing protocols is crucial to maintaining connectivity within a network.

3. Interactive Management

   This class defines interactive traffic that is required for day-to-day network operations. This class would include light volume traffic used for remote network access and management. For example, telnet, Secure Shell (SSH), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP) and Terminal Access Controller Access Control System (TACACS).

4. File Management

   This class defines high volume traffic used for software image and configuration maintenance. This class would include traffic generated for remote file transfer. For example, Trivial File Transfer Protocol (TFTP), and File Transfer Protocol (FTP).

5. Reporting

   This class defines traffic used for generating network performance statistics for reporting. This class would include traffic required for using Cisco IOS IP Service Level Agreements (SLAs) (feature previously known as Service Assurance Agent) to generate ICMP with different DSCP settings in order to report on response times within different QOS data classes.

6. Monitoring

   This class defines traffic used for monitoring a router. This kind of traffic should be permitted but should never be allowed to pose a risk to the router. With CoPP, this traffic can be permitted but limited to a low rate. Examples would include packets generated by ICMP echo requests (ping) and the **traceroute** command.

7. Critical Applications

   This class defines application traffic that is crucial to a specific network. The protocols that might be included in this class include generic routing encapsulation (GRE), Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Dynamic Host Configuration Protocol (DHCP), IPSec, and multicast traffic.

8. Undesirable

   This explicitly identifies unwanted or malicious traffic that should be dropped and denied access to the RP. For example, this class could contain packets from a well-known worm. This class is particularly useful when specific traffic destined to the router should always be denied rather than be placed into a default category. Explicitly denying traffic allows you to collect rough statistics on this traffic using **show** commands and thereby offers some insight into the rate of denied traffic.

9. Default

   This class defines all remaining traffic destined to the RP that does not match any other class. MQC provides the Default class so you can specify how to treat traffic that is not explicitly associated with any other user-defined classes. It is desirable to give such traffic access to the RP but at a highly reduced rate.

With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined to the control plane. After this traffic is identified, further analysis can be performed to classify it. If needed, the other CoPP policy entries can be updated to account for this traffic.

Note On Supervisors 32 and 720 the default class (class-default) is the only traffic class that matches both IP and non-IP packets.

Recommended CoPP Deployment Methodology

Because CoPP filters traffic, it is critical to gain an adequate level of understanding about the legitimate traffic destined to the RP prior to deployment. CoPP policies built without proper understanding of the protocols, devices, or required traffic rates involved can block critical traffic. This has the potential of creating a DoS condition. Determining the exact traffic profile needed to build the CoPP policies might be difficult in some networks.

The following steps follow a conservative methodology that facilitates the process of designing and deploying CoPP. This methodology uses iterative ACL configurations to help identify and to incrementally filter traffic.

To deploy CoPP, perform the following steps:

Step 1 Determine the classification scheme for your network.

Identify the known protocols that access the RP and divide them into categories using the most useful criteria for your specific network. In the case of the Catalyst 4500 Series switch, you can take advantage of the system predefined classes, and chose to combine them with your own classes. In the case of Catalyst 6500 there are no predefined classes, so you need to define all the classes. As an example of classification, the nine categories template presented earlier in this section (BGP, IGP, Interactive Management, File Management, Reporting, Critical Applications, Undesirable, and Default) uses a combination of relative importance and traffic type. Select a scheme suited to your specific network, which might require a larger or smaller number of classes.

Step 2 Define classification ACLs.

Configure each ACL to permit all known protocols in its class that require access to the RP. At this point, each ACL entry should have both source and destination addresses set to any. In addition, the ACL for the default class should be configured with a single entry: **permit ip any any**. This will match traffic not explicitly permitted by entries in the other ACLs.

After the ACLs have been configured, create a class-map for each class defined in Step 1, including one for the default class. Then assign each ACL to its corresponding class-map.

Note In this step you should create a separate class-map for the default class, rather than using the class-default available in some platforms. Creating a separate class-map and assigning a **permit ip any any** ACL will allow you to identify traffic not yet classified as part of another class.

Each class map should then be associated with a policy-map that permits all traffic, regardless of classification. The policy for each class should be set as conform-action transmit exceed-action transmit.

Step 3 Review the identified traffic and adjust the classification.

Ideally, the classification performed in Step 1 identified all required traffic destined to the router. However, realistically, not all required traffic will be identified prior to deployment and the **permit ip any any** entry in the default class ACL will log a number of packet matches. Some form of analysis will be required to determine the exact nature of the unclassified packets.

Use the **show access-lists** command to see the entries in the ACLs that are in use, and to identify any additional traffic sent to the RP. To analyze the unclassified traffic you can use one of the following techniques:

- General ACL classification as described in *Characterizing and Tracing Packet Floods Using Cisco Routers*, which is available at the following URL:

  http://www.cisco.com/warp/public/707/22.html#topic2

- Packet analyzers

  When traffic has been properly identified, adjust the class configuration accordingly. Remove the ACL entries for those protocols that are not used. Add a **permit any any** entry for each protocol just identified.

Step 4 Restrict a macro range of source addresses.

Refine the classification ACLs, by only allowing the full range of the allocated CIDR block to be permitted as the source address. For example, if the network has been allocated 172.68.0.0/16, then permit source addresses from 172.68.0.0/16 where applicable.

This step provides data points for devices or users from outside the CIDR block that might be accessing the equipment. An axternal BGP (eBGP) peer will require an exception because the permitted source addresses for the session will lie outside the CIDR block. This phase might be left on for a few days to collect data for the next phase of narrowing the ACL entries.

Step 5 Narrow the ACL permit statements to authorized source addresses.

Increasingly limit the source address in the classification ACLs to only permit sources that communicate with the RP. For example, only known network management stations should be permitted to access the SNMP ports on a router.

Step 6 Refine CoPP policies by implementing rate limiting.

Use the **show policy-map control-plane** command to collect data about the actual policies in place. Analyze the packet count and rate information and develop a rate limiting policy accordingly.

At this point, you might decide to remove the class-map and ACL used for the classification of default traffic. If so, you should also replace the previously defined policy for the default class by the class-default policy.

Sample CoPP Configuration

The following example shows how to develop a CoPP policy and how to apply it in order to protect the control plane of a Catalyst 6500 Series switch.

In this example, the control plane traffic is classified based on relative importance and traffic type. Eight classes are defined, each of which is associated with a separate extended ACL:

- BGP (coppacl-bgp): BGP traffic
- IGP (coppacl-igp): OSPF traffic
- Interactive management (coppacl-interactivemanagement): remote access and management traffic such as TACACS, SSH, SNMP, and NTP.
- File management (coppacl-filemanagement): remote file transfer traffic such as TFTP and FTP.
- Monitoring (coppacl-monitoring): ICMP and traceroute traffic
- Critical applications (coppacl-critical-app): HSRP and DHCP traffic
- Undesirable traffic (coppacl-undesirable): explicitly denies unwanted traffic (for example, Slammer worm packets)
- Default (no ACL needed): all IP and non-IP traffic received by the control plane that has not been otherwise identified.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Sample basic ACLs for CoPP classification
!
! In this example, BGP is used and must be classified
ip access-list extended coppacl-bgp
 remark BGP traffic class
! allow BGP from a known peer to this router's BGP TCP port
permit tcp host 192.168.1.1 host 10.1.1.1 eq bgp
! allow BGP from a peer's BGP port to this router
permit tcp host 192.168.1.1 eq bgp host 10.1.1.1
!
! For your IGP class, OSPF is the IGP used in this example
ip access-list extended coppacl-igp
 remark IGP traffic class
! permit OSPF
permit ospf any host 224.0.0.5
permit ospf any host 224.0.0.6
permit ospf any any
!
! The Interactive Management class is for traffic that is required
! for accessing and managing the system, in this example, TACACS,
! ssh, snmp, and ntp is classified in this class
ip access-list extended coppacl-interactivemanagement
 remark CoPP interactive management traffic class
! permit return traffic from TACACS host
permit tcp host 10.2.1.1 host 10.1.1.1 established
! ssh access to the router from a subnet
permit tcp 10.2.1.0 0.0.0.255 host 10.1.1.1 eq 22
! SNMP access from the NMS host to the router
permit udp host 10.2.2.2 host 10.1.1.1 eq snmp
```

```
! Allow the router to receive NTP packets from a known clock source
permit udp host 10.2.2.3 host 10.1.1.1 eq ntp
!
! The File Management class is for file transfer traffic required
! for software and configuration maintenance, in this example, TFTP
! and FTP is classified in this class
ip access-list extended coppacl-filemanagement
 remark CoPP file management traffic class
! Allow router initiated FTP (active and passive)
 permit tcp 10.2.1.0 0.0.0.255 eq 21 host 10.1.1.1 gt 1023 established
 permit tcp 10.2.1.0 0.0.0.255 eq 20 host 10.1.1.1 gt 1023
 permit tcp 10.2.1.0 0.0.0.255 gt 1023 host 10.1.1.1 gt 1023 established
! Allow router initiated TFTP
 permit udp 10.2.1.0 0.0.0.255 gt 1023 host 10.1.1.1 gt 1023
!
! The monitoring class is used for traffic that is required for
! monitoring the system. Monitoring traffic is traffic that we expect
! to see destined to the router and want to track and limit
ip access-list extended coppacl-monitoring
 remark CoPP monitoring traffic class
! permit router originated traceroute
permit icmp any any ttl-exceeded
permit icmp any any port-unreachable
! permit receipt of responses to router originated pings
permit icmp any any echo-reply
! allow pings to router/switch
permit icmp any any echo
!
! The critical-app class is used for traffic that is crucial to
! the particular customer's environment. In this example, HSRP
! and DHCP are used.
ip access-list extended coppacl-critical-app
 remark CoPP critical apps traffic class
! permit HSRP
permit ip any host 224.0.0.2
! permit DHCP requests
permit udp host 0.0.0.0 host 255.255.255.255 eq bootps
! permit DHCP replies from DHCP server
permit udp host 10.2.2.8 eq bootps any eq bootps
!
! This ACL identifies traffic that should always be blocked from
! accessing the Route Processor. Once undesirable traffic flow is
! identified, an ACE entry classifying it can be added and mapped to the
! undesirable traffic class. This can be used as a reaction tool.
ip access-list extended coppacl-undesirable
 remark explicitly defined "undesirable" traffic
! permit, for policing, all traffic destined to UDP 1434
permit udp any any eq 1434
```

When the control plane traffic has been classified, the next step is to define the policy action for each traffic class. In this example, our intention is to deploy a policy that protects the router while limiting the risk of dropping critical traffic. To that end, CoPP policies are configured to permit each traffic class with an appropriate rate limit. Table 3 shows the parameters used in the CoPP policies.

Table 3 Sample CoPP Policy

Traffic Class	Rate (bps)	Conform Action	Exceed Action
BGP	4,000,000	Transmit	Drop
IGP	300,000	Transmit	Drop
Interactive management	500,000	Transmit	Drop

Table 3 Sample CoPP Policy

File management	6,000,000	Transmit	Drop
Monitoring	900,000	Transmit	Drop
Critical applications	900,000	Transmit	Drop
Undesirable	32,000	Drop	Drop
Default	500,000	Transmit	Drop

Note The rates defined in Table 3 were successfully tested on a Cisco Catalyst 6500 Series switch with Supervisor 720. It is important to note that the values presented here are solely for illustration purposes, every environment will have different baselines.

The following is the policy for the configuration shown in Table 3:

```
! Define a class for each "type" of traffic and associate it with an ACL
class-map match-all coppclass-bgp
  match access-group name coppacl-bgp
class-map match-all coppclass-igp
  match access-group name coppacl-igp
class-map match-all coppclass-interactivemanagement
  match access-group name coppacl-interactivemanagement
class-map match-all coppclass-filemanagement
  match access-group name coppacl-filemanagement
class-map match-all coppclass-monitoring
  match access-group name coppacl-monitoring
class-map match-all coppclass-critical-app
  match access-group name coppacl-critical-app
class-map match-all coppclass-undesirable
  match access-group name coppacl-undesirable
!
! This is the actual policy. Depending on class of traffic, rates and associated actions
! are defined
policy-map copp-policy
!
! BGP traffic is limited to a rate of 4,000,000 bps, if traffic exceeds
! that rate it is dropped. NOTE: In this example BGP traffic is rate-limited
! to control attacks based on BGP packets. Once the normal rates are determined,
! and depending on the hardware platform used, it's recommended you consider
! readjusting the rate-limiting parameters.
  class coppclass-bgp
     police cir 4000000 bc 400000 be 400000 conform-action transmit exceed-action drop
!
! IGP traffic is limited to a rate of 300,000 bps, if traffic exceeds
! that rate it is dropped.
  class coppclass-igp
     police cir 300000 bc 3000 be 3000 conform-action transmit exceed-action drop
!
! Interactive Management traffic is limited to a rate of 500,000 bps, if traffic
! exceeds that rate it is dropped
  class coppclass-interactivemanagement
    police cir 500000 bc 5000 be 5000 conform-action transmit exceed-action drop
!
! File Management traffic is limited to a rate of 6,000,000 bps, if traffic exceeds
! that rate it is dropped
  class coppclass-filemanagement
    police cir 6000000 bc 60000 be 60000 conform-action transmit exceed-action drop
```

```
!
! Monitoring traffic is limited to a rate of 900,000 bps, if traffic exceeds
! that rate it is dropped
  class coppclass-monitoring
    police cir 900000 bc 9000 be 9000 conform-action transmit exceed-action drop
!
! Critical-app traffic is limited to a rate of 900,000 bps, if traffic
! exceeds that rate it is dropped
  class coppclass-critical-app
    police cir 900000 bc 9000 be 9000 conform-action transmit exceed-action drop
!
! This policy drops all traffic categorized as undesirable, regardless
! of rate.
  class coppclass-undesirable
    police cir 32000 bc 3000 be 3000 conform-action drop exceed-action drop
!
! The default class applies to all IP and non-IP traffic received by the
! control plane that has not been otherwise identified. In this example,
! all default traffic is limited to 500,000 bps and violations of that
! limit are dropped.
  class class-default
    police cir 500000 bc 5000 be 5000 conform-action transmit exceed-action drop
….
! Applies the defined CoPP policy to the control plane
Router(config)# mls qos
Router(config)# control-plane
Router(config-cp)# service-policy input copp-policy
```

Additional Catalyst 6500 Infrastructure Protection Features

The Catalyst 6500 Series switches offer an additional set of features, not available on Catalyst 4500 Series switches, and that provide unmatched protection against distributed denial of service (DDoS) and other types of attacks affecting the switching infrastructure. This section describes the unique implementation of the following features on Catalyst 6500 Series switches.

- Unicast Reverse Path Forwarding (uRPF), page 67
- Hardware-Based Rate Limiters on Supervisor 2, page 69
- Hardware-Based Rate Limiters on Supervisors 32 and 720, page 72

Unicast Reverse Path Forwarding (uRPF)

Unicast Reverse Path Forwarding (uRPF) is an available feature on Catalyst 6500 Series switches running Cisco IOS software and that helps prevent attacks based on IP address spoofing. uRPF can be used as an alternative to ACLs for implementing BCP38/RFC 2827 ingress traffic filtering.

In Catalyst 6500 Series switches uRPF can be enabled on physical and VLAN interfaces. When enabled on an interface, the switch verifies that all packets received from that interface have a source address that is reachable via that same interface. To that end, uRPF verifies that there is a reverse path route pointing to the same interface where the packet came from. If there is one, the packet gets forwarded normally. Otherwise, the packet is dropped. This ensures that the source addresses of the incoming packets are consistent with the routing information held on the switch, which in term helps prevent packets with forged IP addresses.

Note uRPF requires that Cisco Express Forwarding (CEF) is enabled.

An important characteristic of uRPF is that it enables this functionality with minimal operational overhead and in a scalable, timely manner. In addition, uRPF introduces minimal performance impact to a device. It is thus a highly attractive alternative to traditional ACLs. The Catalyst 6500 Supervisor 2, Supervisor 32 and Supervisor 720 support uRPF in hardware.

There are currently two uRPF modes available: strict mode and loose mode. uRPF strict mode requires that the source IP address of an incoming packet has a reverse path to the same interface as that on which the packet arrived. uRPF loose mode requires that the source IP address of an incoming packet has a reverse path to any interface on the device, except null0. By design, uRPF loose mode does not offer the same degree of source IP address spoofing protection as uRPF strict mode. However, strict mode can be used in scenarios where loose mode can't. uRPF strict mode should only be used in deployments where the reverse path entries match the traffic paths, otherwise there is a risk uRPF could drop valid packets.

The PFC2 (Policy Feature Card) supports Unicast RPF check with hardware processing for packets that have a single return path. The MSFC2 processes traffic in software that has multiple return paths (for example, load sharing).

The PFC3 provides hardware support for RPF check of traffic from multiple interfaces. With strict-method Unicast RPF check, the PFC3 supports two parallel paths for all prefixes in the routing table, and up to four parallel paths for prefixes reached through any of four user-configurable RPF interface groups (each interface group can contain four interfaces). With loose-method Unicast PRF check (also known as exist-only method), the PFC3 supports up to eight reverse-path interfaces (the Cisco IOS software is limited to eight reverse paths in the routing table).

To configure uRPF on a Catalyst 6500 Series switch running Catalyst IOS, use the **ip verify unicast interface** command. Use the **rx** keyword to enable strict check mode, and the **any** keyword to enable loose (exist-only) check mode. The **allow-default** keyword allows the use of the default route for RPF verification:

```
Router(config-if)# ip verify unicast source reachable-via {rx | any} [allow-default]
[list]
```

This example shows how to enable Unicast RPF strict check mode on Gigabit Ethernet port 4/2:

```
Router(config)# interface gigabitethernet 4/2
Router(config-if)# ip verify unicast source reachable-via rx
Router(config-if)# end
Router#
```

To configure the multiple-path Unicast PRF check mode on a PFC3, use the **mls ip cef rpf mpath** command.

When configuring multiple path RPF check, note the following information:

- Punt (default—The PFC3 performs the Unicast PRF check in hardware for up to two interfaces per prefix. Packets arriving on any additional interfaces are redirected (punted) to the MSFC3 for Unicast PRF check in software.
- Pass—The PFC3 performs the Unicast PRF check in hardware for single-path and two-path prefixes. Unicast RPF check is disabled for packets coming from multipath prefixes with three or more reverse-path interfaces (these packets always pass the Unicast RPF check).

- Interface-group—The PFC3 performs the Unicast PRF check in hardware for single-path and two-path prefixes. The PFC3 also performs the Unicast PRF check for up to four additional interfaces per prefix through user-configured multipath Unicast PRF check interface groups. Unicast RPF check is disabled for packets coming from other multipath prefixes that have three or more reverse-path interfaces (these packets always pass the Unicast PRF check).

For more information on uRPF on Catalyst 6500 Series switches, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/secure.htm

Hardware-Based Rate Limiters on Supervisor 2

The Supervisor Engine 2 for the Catalyst 6500 Series switches implements four hardware-based rate limiters that can control the rate with which packets are sent to the MSFC, helping mitigate DoS and other attacks that try to overwhelm the MSFC. These rate limiters use four rate-limiter registers that are configured globally on the switch. The rate-limiter registers are located in the Layer 3 forwarding engine (PFC) and are responsible for containing rate-limiting information for packets destined to the MSFC.

The four rate-limiter registers can be shared along different rate-limiting scenarios. The registers are assigned on a first-come, first-serve basis. If all registers are being used, the only way to configure another rate limiter is to free one register.

The hardware-based rate limiters available on the Supervisor Engine 2 are as follows:

- Ingress-Egress ACL Bridged Packets (Unicast Only), page 69
- FIB (CEF) Receive and FIB Glean Cases (Unicast Only), page 70
- VACL Log (Unicast Only), page 70
- Layer 3 Security Features (Unicast Only), page 71
- Routing Protocol Policing, page 71

Note The rate limiters are a very useful tool to protect the MSFC. However, special care should be taken when deployed. Rate limiters do not discriminate between good frames and bad frames. There is always a chance good frames are discarded under attack conditions.

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the MSFC because of an ingress/egress ACL bridge result. Example of ACL bridged packets include packets hitting the **log** keyword, packets requiring special ACL features and non-supported hardware packet types such as IPX and AppleTalk.

To enable and set the ACL-bridged rate limiter, use the **mls rate-limit unicast acl** command:

```
Router(config)# mls rate-limit unicast acl {input | output } {pps [packets-in-burst]}
```

Ingress and egress values can be defined independently. However, when used together, both the ingress and egress values will be the same as they both share the same rate-limiter register.

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to 50000 packets per second, and 50 packets in burst:

```
Router(config)# mls rate-limit unicast acl input 50000 50
```

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to the same rate (50000 pps and 50 packets in burst) for egress ACL bridge results:

```
Router(config)# mls rate-limit unicast acl output 50000 50
```

Because both ingress and egress limiters share the same rate-limiter register, when one of them is changed, both values change to the last configured value. In the following example, the output rate is changed to 40000 pps:

```
Router(config)# mls rate-limit unicast acl output 40000 50
```

For more information on the **mls rate-limit unicast acl** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1719874

FIB (CEF) Receive and FIB Glean Cases (Unicast Only)

The FIB receive rate limiter provides the capability to rate limit all packets that contain the MSFC IP address as the destination address.

To enable and set the FIB receive rate limiter, use the **mls rate-limit unicast cef receive** command.

```
Router(config)# mls rate-limit unicast cef receive pps [packets-in-burst]
```

This example shows how to rate-limit the traffic to 25000 pps with a burst of 60:

```
Router(config)# mls rate-limit unicast cef receive 25000 60
```

The FIB glean rate limiter does not limit ARP traffic, but provides the capability to rate limit traffic that requires address resolution (ARP) and requires that it be sent to the MSFC. This situation occurs when traffic enters a port and contains the destination of a host on a subnet that is locally connected to the MSFC, but no ARP entry exists for that destination host. In this case, because the MAC address of the destination host will not be answered by any host on the directly connected subnet that is unknown, the glean adjacency is hit and the traffic is sent directly to the MSFC for ARP resolution. This rate limiter limits the possibility of an attacker overloading the CPU with such ARP requests.

To enable and set the FIB glean rate limiter, use the **mls rate-limit unicast cef glean** command:

```
Router(config)# mls rate-limit unicast cef glean pps [packets-in-burst]
```

This example shows how to rate limit the rate at which this traffic is sent to the MSFC to 20000 pps and a burst of 60:

```
Router(config)# mls rate-limit unicast cef glean 20000 60
```

For more information on the **mls rate-limit unicast cef** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500483

VACL Log (Unicast Only)

Packets that are sent to the MSFC because of VLAN-ACL logging can be rate limited to ensure that the CPU is not overwhelmed with logging tasks. VACLs are processed in hardware, but the MSFC does the logging. When VACL logging is configured on the switch, IP packets that are denied in the VACL generate log messages. Use this rate-limiter only when VACL logging is configured.

To enable and set the VACL log rate limiter, use the **mls rate-limit unicast acl vacl-log** command.

```
Router(config)# mls rate-limit unicast acl vacl-log pps [packets-in-burst]
```

This example shows how to rate limit logging requests to 5000 pps (the range for this rate limiter is from 10 to 5000 pps):

```
Router(config)# mls rate-limit unicast acl vacl-log 5000
```

For more information on the **mls rate-limit unicast acl vacl-log** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1719874

Layer 3 Security Features (Unicast Only)

Some security features are processed by first being sent to the MSFC. For these security features, you need to rate limit the number of these packets being sent to the MSFC to reduce any potential overloading. The security features include authentication proxy (auth-proxy), IPSEC, and inspection. Do not enable this rate limiter unless you are planning to use any of the aforementioned features.

Authentication proxy is used to authenticate inbound or outbound users or both. These users are normally blocked by an access list, but with auth-proxy, the users can bring up a browser to go through the firewall and authenticate on a terminal access controller access control system plus (TACACS+) or RADIUS server (based on the IP address). The server passes additional access list entries down to the switch to allow the users through after authentication. These ACLs are stored and processed in software, and if there are many users using auth-proxy, the MSFC might be overwhelmed. Rate limiting would be advantageous in this situation.

IPSec and inspection are also done by the MSFC and they might require rate limiting. When the Layer 3 security feature rate limiter is enabled, all Layer 3 rate limiters for auth-proxy, IPSec and inspection are enabled at the same rate.

To enable and set the Layer 3 security features rate limiter, use the **mls rate-limit unicast ip features** command:

```
Router(config)# mls rate-limit unicast ip features pps [packets-in-burst]
```

This example shows how to rate limit the security features to the MSFC to 100000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit unicast ip features 100000 10
```

For more information on the **mls rate-limit unicast ip features** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

Routing Protocol Policing

The Catalyst 6500 Series switches provide specific hardware-based policing mechanisms that can rate limit routing protocols destined to the switch. These mechanisms help protect the switch from DoS attacks based on BGP, IGRP, LDP, ND, OSPF, and RIP packets.

Note The routing protocol and ARP policers not only police traffic destined to the switch, but also traffic crossing the switch.

This rate limiter is enabled by default with a rate burst of 1000 bits per second. To set the routing protocol and ARP policing, use the **mls qos protocol** global configuration command.

```
Router(config)# mls qos protocol protocol-name {pass-through | {police rate burst} |
{precedence value [police rate burst]}}
```

Note This command does not support ARP, ISIS, or EIGRP on Catalyst 6500 Series switches that are configured with a Supervisor Engine 2.

This example shows how to define the routing-protocol packet policing:

```
Router(config)# mls qos protocol bgp police 32000
```

For more information on the **mls qos protocol** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1502440

Hardware-Based Rate Limiters on Supervisors 32 and 720

The Supervisor Engine 32 and Supervisor Engine 720 for the Catalyst 6500 Series switches implement ten hardware-based rate limiters that can control the rate with which packets are sent to the route or switch processor CPU, helping mitigate DoS and other attacks that try to overwhelm the CPU. Eight of these registers are present in the Layer 3 forwarding engine and two of these registers are present in the Layer 2 forwarding engine.

The ten rate-limiter registers can be shared along different rate-limiting scenarios. The registers are assigned on a first-come, first-serve basis. If all registers are being used, the only way to configure another rate limiter is to free one register.

These hardware-based rate limiters are supported on both Supervisor 720 and Supervisor 32 Catalyst OS and Cisco IOS versions.

Note Rate limiters are a very useful tool to protect the route or switch processor CPU. However, special care should be taken when deployed. Rate limiters do not discriminate between good frames and bad frames. There is always a chance good frames are discarded under attack conditions.

The hardware-based rate limiters available on the Supervisor Engines 32 and 720 are as follows:

- Ingress and egress ACL bridged packets
- uRPF check failure
- ICMP unreachable (no route, ACL drop)
- ICMP redirects
- IP Errors
- FIB receive
- FIB glean
- VACL log
- Layer 3 security features
- TTL failure
- MTU failure
- Layer 2 PDU
- Layer 2 protocol tunneling
- Layer 2 multicast IGMP snooping
- Multicast IPv4
- Multicast IPv6
- Routing protocol and ARP policing

Note The hardware-based rate limiters don't provide the same level of granularity as CoPP but can be used in cases where CoPP cannot classify particular types of traffic (for example, packets that fail the MTU check, and packets with IP options). We recommend that you use CoPP and hardware-based rate limiters together. However, be aware that some hardware-based rate limiters override the CoPP policy.

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the MSFC because of an ingress/egress ACL bridge result. Example of ACL bridged packets include packets hitting the **log** keyword, packets requiring special ACL features and non-supported hardware packet types, such as IPX and AppleTalk.

This rate limiter is disabled by default. To enable and set the ACL-bridged rate limiter, use the **mls rate-limit unicast acl** command.

```
Router(config)# mls rate-limit unicast acl {input | output } pps [packets-in-burst]
```

Burst values regulate how many packets can be allowed in a burst. Each allowed packet consumes a token and a token must be available for a packet to be allowed. One token is generated per millisecond. When packets are not coming in, tokens can be accumulated up to the burst value. For example, if the burst value is set to 50, the switch can accumulate up to 50 tokens and absorb a burst of 50 packets.

Ingress and egress values can be defined independently. However, when used together, both the ingress and egress values will be the same as they both share the same rate-limiter register.

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to 50000 packets per second, and 50 packets in burst:

```
Router(config)# mls rate-limit unicast acl input 50000 50
```

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to the same rate (50000 pps and 50 packets in burst) for egress ACL bridge results:

```
Router(config)# mls rate-limit unicast acl output 50000 50
```

Because both ingress and egress limiters share the same rate-limiter register, when one of them is changed, both values change to the last configured value. In the following example, the output rate is changed to 40000 pps:

```
Router(config)# mls rate-limit unicast acl output 40000 50
```

For more information on the **mls rate-limit unicast acl** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1719874

uRPF Check Failure (Unicast Only)

The uRPF check failure rate limiter allows you to configure a rate for the packets that need to be sent to the MSFC because they failed the uRPF check. The uRPF checks validate that incoming packets on an interface are from a valid source, which minimizes the potential threat of DoS attacks from users using spoofed addresses. When spoofed packets fail the uRPF check, those failures can be sent to the MSFC. The uRPF check rate limiters allow you to rate limit the packets per second that are bridged to the MSFC CPU when a uRPF check failure occurs.

This rate limiter is enabled by default with a limit of 100pps, and burst of 10 packets. To set the uRPF Check Failure rate limiter, use the **mls rate-limit unicast ip rpf-failure** command.

```
Router(config)# mls rate-limit unicast ip rpf-failure pps [packets-in-burst]
```

This example shows how to rate limit the uRPF check failure packets sent to the MSFC to 100000 pps with a burst of 100 packets:

```
Router(config)# mls rate-limit unicast ip rpf-failure 100000 100
```

Note The ICMP unreachable no route, ICMP unreachable ACL drop, IP errors, and IP RPF failure rate-limiters share a single rate-limiter register. If any of these limiters are enabled, all of the limiters in this group will share the same value and sometimes the same state (for example, ON/ON/ON). When verifying the rate limiters, if the members of this register are enabled through another feature, an ON-Sharing status (instead of an ON status) is displayed. The exception is the TTL failure rate limiter: its value shares the same value as the other members in the register if you have manually enabled the feature.

For more information on the **mls rate-limit unicast ip rpf-failure** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

ICMP Unreachable (Unicast Only)

In an ICMP overload unreachable attack, the victim device is flooded with a large number of packets that require the generation of ICMP unreachable packets. By IP standards, a router is required to generate an ICMP unreachable message to the source of a packet it can't deliver because there is no route to the destination, or because the packet has to be blocked by an ACL. The ICMP unreachable rate limiter allows you to rate limit the packets sent to the MSFC that trigger ICMP unreachables.

This rate limiter is enabled by default with a limit of 100pps, and burst of 10 packets. To set the ICMP Unreachable rate limiter, use the **mls rate-limit unicast ip icmp unreachable** command.

```
Router(config)# mls rate-limit unicast ip icmp unreachable {acl-drop pps} | no-route pps}
[packets-in-burst]
```

This example shows how to rate limit the packets that are sent to the MSFC because of an ACL drop to 10000 pps and a burst of 100:

```
Router(config)# mls rate-limit unicast ip icmp unreachable acl-drop 10000 100
```

This example shows how to rate limit the packets that require generation of ICMP-unreachable messages because of a FIB miss to 80000 pps and burst to 70:

```
Router(config)# mls rate-limit unicast ip icmp unreachable no-route 80000 70
```

Note The ICMP unreachable no route, ICMP unreachable ACL drop, IP errors, and IP RPF failure rate-limiters share a single rate-limiter register. If any of these limiters are enabled, all of the limiters in this group will share the same value and sometimes the same state (for example, ON/ON/ON). When verifying the rate limiters, if the members of this register are enabled through another feature, an ON-Sharing status (instead of an ON status) is displayed. The exception is the TTL failure rate limiter: its value shares the same value as the other members in the register if you have manually enabled the feature.

For more information on the **mls rate-limit unicast ip icmp unreachable** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

ICMP Redirects (Unicast Only)

The ICMP-redirect rate limiter allows you to rate limit ICMP traffic. For example, when a host sends packets through a nonoptimal switch, the MSFC sends ICMP-redirect messages to the host to correct its sending path. If this traffic occurs continuously, and is not rate limited, the MSFC will continuously generate ICMP-redirect messages.

This rate limiter is disabled by default. To enable and set the ICMP Unreachable rate limiter, use the **mls rate-limit unicast ip icmp redirect** command.

```
Router(config)# mls rate-limit unicast ip icmp redirect pps [packets-in-burst]
```

This example shows how to rate limit the ICMP redirects to 20000 pps, with a burst of 20 packets:

```
Router(config)# mls rate-limit unicast ip icmp redirect 20000 20
```

For more information on the **mls rate-limit unicast ip icmp redirect** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

IP Errors (Unicast Only)

This rate limiter limits the packets with IP checksum and length errors. When a packet reaches the PFC3 with an IP checksum error or a length inconsistency error, it must be sent to the MSFC for further processing. An attacker might use the malformed packets to carry out a DoS attack, but the network administrator can configure a rate for these types of packets to protect the control path.

This rate limiter is enabled by default with a limit of 100pps, and burst of 10 packets. To set the IP Errors rate limiter, use the **mls rate-limit unicast ip errors** command.

```
Router(config)# mls rate-limit unicast ip errors pps [packets-in-burst]
```

This example shows how to rate limit IP errors sent to the MSFC to 1000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit unicast ip errors 1000 20
```

Note The ICMP unreachable no route, ICMP unreachable ACL drop, IP errors, and IP RPF failure rate-limiters share a single rate-limiter register. If any of these limiters are enabled, all of the limiters in this group will share the same value and sometimes the same state (for example, ON/ON/ON). When verifying the rate limiters, if the members of this register are enabled through another feature, an ON-Sharing status (instead of an ON status) is displayed. The exception is the TTL failure rate limiter, its value shares the same value as the other members in the register if you have manually enabled the feature.

For more information on the **mls rate-limit unicast ip errors** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

FIB (CEF) Receive (Unicast Only)

The FIB receive rate limiter provides the capability to rate-limit all packets that contain the MSFC IP address as the destination address. Always choose CoPP over this rate limiter and do not use both mechanisms at the same time.

This rate limiter is disabled by default. To enable and set the FIB receive rate limiter, use the **mls rate-limit unicast cef receive** command.

```
Router(config)# mls rate-limit unicast cef receive pps [packets-in-burst]
```

This example shows how to rate limit the traffic to 25000 pps with a burst of 60:

```
Router(config)# mls rate-limit unicast cef receive 25000 60
```

Note Do not enable the FIB receive rate limiter if you are using CoPP. The FIB receive rate limiter overrides the CoPP policies.

For more information on the **mls rate-limit unicast cef receive** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500483

FIB (CEF) Glean (Unicast Only)

The FIB glean rate limiter does not limit ARP traffic, but provides the capability to rate limit traffic that requires address resolution (ARP) and requires that it be sent to the MSFC. This situation occurs when traffic enters a port and contains the destination of a host on a subnet that is locally connected to the MSFC, but no ARP entry exists for that destination host. In this case, because the MAC address of the destination host will not be answered by any host on the directly connected subnet that is unknown, the glean adjacency is hit and the traffic is sent directly to the MSFC for ARP resolution. This rate limiter limits the possibility of an attacker overloading the CPU with such ARP requests.

This rate limiter is disabled by default. To enable and set the FIB glean rate limiter, use the **mls rate-limit unicast cef glean** command.

```
Router(config)# mls rate-limit unicast cef glean pps [packets-in-burst]
```

This example shows how to rate limit the rate at which this traffic is sent to the MSFC to 20000 pps and a burst of 60:

```
Router(config)# mls rate-limit unicast glean receive 20000 60
```

For more information on the **mls rate-limit unicast cef glean** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500483

VACL Log (Unicast Only)

Packets that are sent to the MSFC because of VLAN-ACL logging can be rate limited to ensure that the CPU is not overwhelmed with logging tasks. VACLs are processed in hardware, but the MSFC does the logging. When VACL logging is configured on the switch, IP packets that are denied in the VACL generate log messages. Use this rate-limiter only when VACL logging is configured.

This rate limiter is enabled by default with a limit of 2000pps, and burst of one packet. To set the VACL log rate limiter, use the **mls rate-limit unicast acl vacl-log** command.

```
Router(config)# mls rate-limit unicast acl vacl-log pps [packets-in-burst]
```

This example shows how to rate limit logging requests to 5000 pps (the range for this rate limiter is from 10 to 5000 pps):

```
Router(config)# mls rate-limit unicast acl vacl-log 5000
```

For more information on the **mls rate-limit unicast acl vacl-log** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1719874

Layer 3 Security Features (Unicast Only)

Some security features are processed by first being sent to the MSFC. For these security features, you need to rate limit the number of these packets being sent to the MSFC to reduce any potential overloading. The security features include authentication proxy (auth-proxy), IPSec, and inspection. Do not enable this rate limiter unless you are planning to use any of these features.

Authentication proxy is used to authenticate inbound or outbound users or both. These users are normally blocked by an access list, but with auth-proxy, the users can bring up a browser to go through the firewall and authenticate on a terminal access controller access control system plus (TACACS+) or RADIUS server (based on the IP address). The server passes additional access list entries down to the switch to allow users access after authentication. These ACLs are stored and processed in software, and if there are many users using auth-proxy, the MSFC can be overwhelmed. Rate limiting would be advantageous in this situation.

IPSec and inspection are also done by the MSFC and might require rate limiting. When the Layer 3 security feature rate limiter is enabled, all Layer 3 rate limiters for auth-proxy, IPSec and inspection are enabled at the same rate.

This rate limiter is disabled by default. To enable and set the Layer 3 security features rate limiter, use the **mls rate-limit unicast ip features** command.

```
Router(config)# mls rate-limit unicast ip features pps [packets-in-burst]
```

This example shows how to rate limit the security features to the MSFC to 100000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit unicast ip features 100000 10
```

For more information on the **mls rate-limit unicast ip features** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1500566

TTL Failure (Unicast and Multicast)

This rate limiter rate limits packets sent to the MSFC because of a time-to-live (TTL) check failure. As indicated by the **all** keyword in the following example, this rate limiter applies to both multicast and unicast traffic.

This rate limiter is disabled by default. To enable and set the TTL Failure rate limiter, use the **mls rate-limit all ttl-failure** command.

```
Router(config)# mls rate-limit all ttl-failure pps [packets-in-burst]
```

This example shows how to rate limit the TTL failures to 70000 pps with a burst of 150:

```
Router(config)# mls rate-limit all ttl-failure 70000 150
```

Note Do not use this rate limiter in conjunction with Layer 2 multicast in a system with PFC3A.

For more information on the **mls rate-limit all ttl-failure** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1497651

MTU Failure (Unicast and Multicast)

Similar to the TTL failure rate limiter, the rate limiter for MTU failures is supported for both unicast and multicast traffic. Packets that fail an MTU check are sent to the MSFC CPU. This might cause the MSFC to be overwhelmed.

This rate limiter is disabled by default. To enable and set the MTU Failure rate limiter, use the **mls rate-limit all mtu-failure** command.

```
Router(config)# mls rate-limit all mtu-failure pps [packets-in-burst]
```

This example shows how to rate limit packets failing the MTU failures from being sent to the MSFC to 10000 pps with a burst of 10:

```
Router(config)# mls rate-limit all mtu-failure 10000 10
```

Note Do not use this rate limiter in conjunction with Layer 2 multicast in a system with PFC3A.

For more information on the **mls rate-limit all mtu-failure** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1497651

Layer 2 PDU

The Layer 2 protocol data unit (PDU) rate limiter allows you to limit the number of Layer 2 PDU protocol packets (including BPDUs, DTP, PAgP, CDP, STP, and VTP packets) destined for the supervisor engine and not the MSFC CPU. You cannot enable the Layer 2 PDU rate limiter if the Catalyst 6500 Series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This rate limiter is disabled by default. To enable and set the Layer 2 PDU rate limiter, use the **mls rate-limit layer2 pdu** command.

```
Router(config)# mls rate-limit layer2 pdu pps [packets-in-burst]
```

This example shows how to rate limit Layer 2 PDUs to 20000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit layer2 pdu 20000 20
```

For more information on the **mls rate-limit layer2 pdu** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1729104

Layer 2 Protocol Tunneling

This rate limiter limits the Layer 2 protocol tunneling packets, which include control PDUs, CDP, STP, and VTP packets destined for the supervisor engine. These packets are encapsulated in software (rewriting the destination MAC address in the PDU), and then forwarded to a proprietary multicast address (01-00-0c-cd-cd-d0). You cannot enable the Layer 2 Protocol Tunneling rate limiter if the Catalyst 6500 Series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This rate limiter is disabled by default. To enable and set the Layer 2 protocol tunneling rate limiter, use the **mls rate-limit layer2 l2pt** command.

```
Router(config)# mls rate-limit layer2 l2pt pps [packets-in-burst]
```

This example shows how to rate limit Layer 2 protocol tunneling packets to 10000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit layer2 l2pt 10000 10
```

For more information on the **mls rate-limit layer2 l2pt** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1729104

Layer 2 Multicast IGMP Snooping

The IGMP snooping rate limiter limits the number of Layer 2 IGMP packets destined for the supervisor engine. IGMP snooping listens to IGMP messages between the hosts and the supervisor engine. You cannot enable the Layer 2 multicast IGMP snooping rate limiter if the Catalyst 6500 Series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel. This rate limiter should be used only when IGMP snooping is enabled.

This rate limiter is disabled by default. To enable and set the Layer 2 protocol tunneling rate limiter, use the **mls rate-limit multicast ipv4 igmp** command.

```
Router(config)# mls rate-limit multicast ipv4 igmp pps [packets-in-burst]
```

This example shows how to rate limit IGMP-snooping traffic:

```
Router(config)# mls rate-limit multicast ipv4 igmp 20000 40
```

For more information on the **mls rate-limit multicast ipv4 igmp** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1499703

IPv4 Multicast

The IPv4 multicast rate limiter limits IPv4 multicast packets. The rate limiters can rate-limit packets that are sent from the data path in the hardware to the data path in the software. The rate limiters protect the control path in the software from congestion and drop the traffic that exceeds the configured rate. Within the IPv4 multicast rate limiter, there are three rate limiters that you can also configure: the FIB-miss rate limiter, the multicast partially switched flows rate limiter, and the multicast directly connected rate limiter.

The FIB-miss rate limiter allows you to rate limit the multicast traffic that does not match an entry in the mroute table. This rate limiter is enabled by default with a limit of 100000pps, and a burst of 100 packets. To enable and set the FIB-miss rate limiter, use the **mls rate-limit multicast ipv4 fib-miss** command.

```
Router(config)# mls rate-limit multicast ipv4 fib-miss pps [packets-in-burst]
```

The partially switched flow rate limiter allows you to rate limit the flows destined to the MSFC3 for forwarding and replication. For a given multicast traffic flow, if at least one outgoing Layer 3 interface is multilayer switched, and at least one outgoing interface is not multilayer switched (no H-bit set for hardware switching), the particular flow is considered partially switched, or partial-SC (partial shortcut). The outgoing interfaces that have the H-bit flag are switched in hardware and the remaining traffic is switched in software through the MSFC3. For this reason, you might want to rate limit the flow destined to the MSFC3 for forwarding and replication, which might otherwise increase CPU utilization. This rate limiter is enabled by default with a limit of 100000pps, and a burst of 100 packets. To enable and set the multicast partially switched flows rate limiter, use the **mls rate-limit multicast ipv4 partial** command.

```
Router(config)# mls rate-limit multicast ipv4 partial pps [packets-in-burst]
```

The multicast directly connected rate limiter limits the multicast packets from directly connected sources. This rate limiter is disabled by default. To enable and set the multicast directly connected rate limiter, use the **mls rate-limit multicast ipv4 connected** command.

```
Router(config)# mls rate-limit multicast ipv4 connected pps [packets-in-burst]
```

This example shows how to rate limit the multicast packets to 30000 pps with a burst of 30:

```
Router(config)# mls rate-limit multicast ipv4 connected 30000 30
```

Note The **ip-option** keyword and the **ip-option** rate limiter are supported in PFC3B or PFC3BXL mode only.

This example shows how to set the rate limiters for the IPv4 multicast packets failing the uRPF check:

```
Router(config)# mls rate-limit multicast ipv4 non-rpf 100
```

This example shows how to rate limit the multicast FIB miss packets to 10000 pps with a burst of 10:

```
Router(config)# mls rate-limit multicast ipv4 fib-miss 10000 10
```

This example shows how to rate limit the partial shortcut flows to 20000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit multicast ipv4 partial 20000 20
```

This example shows how to rate limit the multicast packets to 30000 pps with a burst of 20:

```
Router(config)# mls rate-limit multicast ipv4 connected 30000 20
```

For more information on the **mls rate-limit multicast ipv4** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1499703

IPv6 Multicast

The IPv6 multicast rate limiter limits IPv6 multicast packets. There are seven IPv6 rate limiters:

- Connected: Directly connected source traffic
- Default-drop: * (*, G/m) SSM, * (*, G/m) SSM non-rpf
- Route-control: * (*, FF02::X/128)
- Secondary-drop: * (*, G/128) SPT threshold is infinity
- SG: * (S, G)RP-RPF post-switchover, * (*, FFx2/16)
- Starg-bridge: * (*, G/128) SM, * SM non-rpf traffic when (*, G) exists
- Starg-M-bridge: * (*, G/m) SM, * (*, FF/8), * SM non-rpf traffic when (*, G) doesn't exist

The IPv6 multicast traffic rate limiters can be configured using one of the following methods:

- Direct association of the rate limiters for a traffic class—Select a rate and associate the rate with a rate limiter. This example shows how to pick a rate of 1000 pps and 20 packets per burst and associate the rate with the default-drop rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 default-drop 1000 20
  ```

- Static sharing of a rate limiter with another pre-configured rate limiter—When there are not enough adjacency-based rate limiters available, you can share a rate limiter with a pre-configured rate limiter (target rate limiter). This example shows how to share the route-cntl rate limiter with the default-drop target rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 route-cntl share default-drop
  ```

  If the target rate limiter is not configured, a message is displayed that indicates that the target rate limiter must be configured for it to be shared with other rate limiters.
- Dynamic sharing of rate limiters—If you are unsure about which rate limiter you should share with, use the **share auto** keywords to enable dynamic sharing. When you enable dynamic sharing, the system selects a pre-configured rate limiter and shares the given rate limiter with the pre-configured rate limiter. This example shows how to choose dynamic sharing for the route-cntrl rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
  ```

This example shows how to set the rate limiters for the IPv6 multicast packets from a directly connected source:

```
Router(config)# mls rate-limit multicast ipv6 connected 1500 20
```

This example shows how to configure a direct association of the rate limiters for a traffic class:

```
Router(config)# mls rate-limit multicast ipv6 default-drop 1000 20
```

This example shows how to configure the static sharing of a rate limiter with another pre-configured rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share default-drop
```

This example shows how to enable dynamic sharing for the route control rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
```

For more information on the **mls rate-limit multicast ipv6** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1719870

Routing Protocol and ARP Policing

The Catalyst 6500 Series switches provide specific hardware-based policing mechanisms that can rate limit routing protocols and ARP packets destined to the switch. These mechanisms help protect the switch from DoS attacks based on ARP, BGP, EIGRP, IGRP, ISIS, LDP, ND, OSPF, and RIP packets.

Note The routing protocol and ARP policers not only police traffic destined for the switch, but also traffic crossing the switch.

This rate limiter is disabled by default. When enabled, the rate burst is automatically set to 1000 bits per second. To set the routing protocol and ARP policing, use the **mls qos protocol** global configuration command.

```
Router(config)# mls qos protocol protocol-name {pass-through | {police rate burst} |
{precedence value [police rate burst]}}
```

This example shows how to define the routing-protocol packet policing:

```
Router(config)# mls qos protocol arp police 43000
```

For more information on the **mls qos protocol** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#wp1502440

Configuring Hardware-Based Rate Limiters in Catalyst OS

The following list shows the hardware-based rate limiters available on Catalyst 6500 Series switches running Catalyst OS:

- ACL Bridge Packets, page 82
- The ACL Feature (ARP Inspection, DHCP Snooping, 802.1x), page 82
- VACL Log, page 83
- Layer 2 Port Security, page 84
- Layer 2 PDU, page 84
- Layer 2 Protocol Tunneling, page 85
- Multicast IGMP, page 85

ACL Bridge Packets

This rate limiter limits the number of packets sent to the route processor CPU for ACL bridge results. Example of ACL bridged packets include packets hitting the **log** keyword, and packets requiring special ACL features.

This rate limiter is disabled by default. To enable and set the ACL-bridged rate limiter, use the **set acllog ratelimit** command.

```
Console> (enable) set acllog ratelimit rate
```

After entering the **set acllog ratelimit** command, you must either reset the route processor or perform a **shut/not shut** on the route processor interfaces that have ACEs with the **log** keyword applied. The **reset** or **shut/no shut** action causes the bridged ACEs to be redirected to the route processor with rate limiting.

If the number of packets per second is greater than the rate that you specify, the packets that exceed the specified rate are dropped. A rate value of 500 is recommended.

This example shows how to enable ACL logging and to specify a rate of 500 for rate limiting:

```
Console> (enable) set acllog ratelimit 500
If the ACLs-LOG were already applied, the rate limit mechanism will be effective on system
restart, or after shut/no shut the interface.
Console> (enable)
```

For more information on the **set acllog ratelimit** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/ses_sete.htm#wp1121288

The ACL Feature (ARP Inspection, DHCP Snooping, 802.1x)

The ACL feature rate limiter controls the rate at which packets are sent to the supervisor engine for processing by the ARP inspection, DHCP snooping, and 802.1X DHCP features.

Note The rate limit is available on the PFC2 or later.

This rate limiter is enabled by default with a rate of 1000 pps. To set the ACL feature rate limiter, use the **set security acl feature ratelimit** command.

```
Console> (enable) set security acl feature ratelimit rate
```

A rate value of 0 disables this rate limiter. We strongly recommend, however, that you do not disable rate limiting because traffic that is redirected by various security features might flood the supervisor engine and diminish system performance.

This example shows how to set the global rate limit to 600:

```
Console> (enable) set security acl feature ratelimit 600
ARP Inspection, DHCP Snooping, and Dot1x DHCP global rate limit set to 600 pps.
Console> (enable)
```

For more information on the **set security acl feature ratelimit** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1393048

To specify the rate limit for the number of ARP inspection packets that are sent to the CPU on a per-port basis, use the **set port arp-inspection** command. The per-port basis rate limiter is disabled by default.

```
Console> (enable) set port arp-inspection mod/port drop-threshold rate shutdown-threshold
rate
```

This example shows how to set the drop-threshold to 500 and the shutdown-threshold to 1000 for port 2/1:

```
Console> (enable) set port arp-inspection 2/1 drop-threshold 500 shutdown-threshold 1000
Drop Threshold=500, Shutdown Threshold=1000 set on port 2/1.
Console> (enable)
```

For more information on the **set port arp-inspection** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_m_pi.htm#wp1138749

VACL Log

The VACL Log rate limiter controls the rate at which packets are sent to the route processor because of VLAN-ACL logging. When VACL logging is configured on the switch, IP packets that are denied in the VACL generate log messages. Use this rate-limiter only when VACL logging is configured.

Note The VACL Log rate limiter is supported on systems configured with Supervisor Engine 2 with Layer 3 Switching Engine II (PFC2) only.

This rate limiter is enabled by default with a rate of 2500 pps. To set the VACL log rate limiter, use the **set security acl log ratelimit** command. A rate value of 0 disables this rate limiter.

```
Console> (enable) set security acl log ratelimit rate
```

This example shows how to set the rate limit:

```
Console> (enable) set security acl log ratelimit 3444
Max logging eligible packet rate set to 3444pps.
Console> (enable)
```

For more information on the **set security acl log ratelimit** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/setsn_su.htm#wp1099527

Layer 2 Port Security

The Layer 2 Port Security rate limiter limits the rate at which packets are processed on ports with port security enabled.

Note Hardware-based rate limiters are supported on Catalyst 6500 Series switches that are configured with a Distributed Forwarding Card 3A (DFC3A) or the Policy Feature Card 3 (PFC3) only. The Catalyst 6500 Series switch cannot be in truncated mode. If you attempt to enable rate limiting while in truncated mode, an error message is displayed.

This rate limiter is disabled by default. To enable or set the Layer 2 Port Security rate limiter, use the **set rate-limit l2port-security** command. A rate value of 0 disables this rate limiter.

```
Console> (enable) set rate-limit l2port-security enable
Console> (enable) set rate-limit l2port-security rate
```

This example shows how to enable Layer 2 rate limiting for port security:

```
Console> (enable) set rate-limit l2port-security enable
2port-security rate limiter enabled.
Console> (enable)
Console> (enable) set rate-limit l2port-security rate 10000
l2port-security rate limiter rate set to 10000 pps.
Console> (enable)
```

For more information on the **set rate-limit l2port-security** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_po_r.htm#wp1597259

Layer 2 PDU

The Layer 2 protocol data unit (PDU) rate limiter allows you to limit the number of Layer 2 PDU protocol packets (including BPDUs, DTP, PAgP, CDP, STP, and VTP packets) destined for the route processor.

Note Hardware-based rate limiters are supported on Catalyst 6500 Series switches that are configured with a Distributed Forwarding Card 3A (DFC3A) or the Policy Feature Card 3 (PFC3) only. The Catalyst 6500 Series switch cannot be in truncated mode. If you attempt to enable rate limiting and you are in truncated mode, a message is displayed.

This rate limiter is disabled by default. To enable or set the Layer 2 PDU rate limiter, use the **set rate-limit l2pdu** command. A rate value of 0 disables this rate limiter.

```
Console> (enable) set rate-limit l2pdu enable
Console> (enable) set rate-limit l2pdu rate
```

This example shows how to enable Layer 2 PDU rate limiting:

```
Console>(enable) set rate-limit l2pdu enable
Layer 2 rate limiter for PDUs enabled on the switch.
Console>(enable)
Console>(enable) set rate-limit l2pdu rate 1000
Layer 2 rate limiter for PDU rate set to 1000.
Console>(enable)
```

For more information on the **set rate-limit l2pdu** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_po_r.htm#wp1597259

Layer 2 Protocol Tunneling

This rate limiter limits Layer 2 protocol tunneling packets, which include control PDUs, CDP, STP, and VTP packets destined for the route processor.

Note Hardware-based rate limiters are supported only on Catalyst 6500 Series switches that are configured with a Distributed Forwarding Card 3A (DFC3A) or the Policy Feature Card 3 (PFC3). The Catalyst 6500 Series switch cannot be in truncated mode. If you attempt to enable rate limiting while in truncated mode, an error message is displayed.

This rate limiter is disabled by default. To enable or set the Layer 2 protocol tunneling rate limiter, use the **set rate-limit l2protocol-tunnel** command. A rate value of 0 disables this rate limiter.

```
Console> (enable) set rate-limit l2protocol-tunnel enable
Console> (enable) set rate-limit l2protocol-tunnel rate
```

This example shows how to enable Layer 2 rate limiting for protocol tunnel-encapsulated PDUs:

```
Console>(enable) set rate-limit l2protocol-tunnel enable
Layer 2 rate limiter for l2protocol-tunnel enabled on the switch.
Console>(enable)
Console>(enable) set rate-limit l2protocol-tunnel rate 2000
Layer 2 rate limiter for l2protocol-tunnel rate set to 2000.
Console>(enable)
```

For more information on the **set rate-limit l2protocol-tunnel** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_po_r.htm#wp1597259

Multicast IGMP

This rate limiter limits the rate at which multicast IGMP packets are sent to route processor.

This rate limiter is disabled by default. To enable or set the multicast IGMP rate limiter, use the **set multicast ratelimit** command. A rate value of 0 disables this rate limiter.

```
Console> (enable) set multicast ratelimit enable
Console> (enable) set multicast ratelimit rate rate
```

This example shows how to enable multicast IGMP rate limiting:

```
Console> (enable) set multicast ratelimit enable
Enabling Multicast Ratelimiting
Set a non-zero threshold rate to operationally enable multicast ratelimiting
Console> (enable)
Console> (enable) set multicast ratelimit rate 300
Multicast ratelimit watermark rate is set to 300 pps
Console> (enable)
```

For more information on the **set multicast ratelimit** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_m_pi.htm#wp1119887

Integrated Deployment Guidelines

The tools and techniques described in this document are very valuable for protecting Cisco Catalyst 6500 and 4500 Series switches from direct attacks and the negative effects of accidental misconfiguration. Even though most of the tools described in this section work independently, they are also complementary and provide even greater value when deployed in an integrated fashion. This section describes the interrelations between these tools and provides general guidelines for deploying these tools as an integrated solution, rather than as isolated elements. This section includes the following topics:

- Deploying Basic Device Hardening Tools and Techniques, page 86
- Spanning Tree Protocol Security, page 87
- Deploying Routing Protocol Security, page 88
- Deploying Catalyst Integrated Security, page 89
- Catalyst 6500 Hardware Rate Limiters and CoPP, page 90

Deploying Basic Device Hardening Tools and Techniques

The section, Basic Tools and Techniques for Device Hardening, page 5 describes a collection of features and techniques that form an essential toolkit for securing Catalyst 6500 and 4500 Series switches. The deployment of these tools on switches is recommended for most environments.

This toolkit includes features that help prevent indiscriminate consumption of the limited resources on a device. In addition, Unneeded Services, page 91 and Access Control, page 98 provide techniques for disabling unneeded services, and control access to the device. These are essential security services that, when combined, provide a security baseline.

Disabling unneeded services is always recommended. Routers and switches often run a collection of services by default, and some services could potentially be used maliciously. Disabling all unneeded global and interface services greatly reduces the risk of security incidents. In cases where a service is still needed, it should be selectively deployed. The service might not be needed globally, but only on specific interfaces.

Directed broadcasts are a good example of a service that, as a general best practice, should never be globally enabled. However, in certain circumstances, it might be needed. For example, messaging middleware, such as TIBCO, might rely on directed broadcasts for system communication. Directed broadcast should not be enabled indiscriminately on all interfaces because this clearly increases the likelihood of security incidents, such as Smurf attacks; directed broadcasts should be enabled only on those interfaces that have systems communicating with TIBCO or other essential services requiring directed broadcasts. In addition, and to properly secure this type of environment, ACLs need to be deployed on all interfaces to ensure that only expected sources send directed broadcasts to the specific interfaces enabled to accept them.

Enforcing appropriate device access is always a recommended practice as well. The following are best practices that should be followed whenever possible:

- Adequate password management
- Controlling console and interactive access

- Deploying banners
- Implementing role-based access
- Securing web-based GUI Access
- Use secure access protocols (SSH) instead of clear text protocols (telnet)
- Controlling SNMP access

These practices are described in Access Control, page 98.

Layer 2 and Layer 3 Access Control Lists (ACLs) are also essential security features because they can help shield the infrastructure from DoS, source address spoofing, and other attacks. Here are some best practices:

- Deploy ACLs at the edge switches to restrict external access to the infrastructure address space, allowing only authorized devices to communicate with infrastructure elements, and providing basic anti-spoofing controls.
- Deploy ACLs on switches that seat between trust boundaries. These ACLs can be configured to provide basic anti-spoofing and access control.
- Deploy IP permit lists and access-classes on every switch to control SSH, Telnet, and SNMP access to the device.

When deploying several types of ACLs, it is important to understand how they interact. For example, when combining PACLs, VACLs and IOS ACLs, a PACL is first applied on an incoming packet on a physical port. If the packet is permitted by the PACL, it is filtered by the VACL that is applied to the corresponding ingress VLAN. If the packet is Layer 3 forwarded and is permitted by the VACL, it is filtered by the Cisco IOS ACL on the same VLAN. The same process happens in reverse in the egress direction.

Finally, by default, Catalyst switches running Catalyst OS come with all Ethernet ports enabled and set to VLAN 1. Leaving all unused ports configured in VLAN 1 opens the chance for unauthorized access. For this reason, we recommend that you disabled all unused ports, and place them in an unused VLAN. In Catalyst switches running Cisco IOS, all interfaces are shut down by default. The interfaces should be enabled only as needed.

Spanning Tree Protocol Security

Spanning Tree (STP) is a widely used protocol that makes it possible to implement redundant topologies in bridged networks while preventing undesirable loops. Unfortunately, STP communications are neither encrypted nor authenticated, leaving STP vulnerable to a variety of attacks, including the injection of bogus BPDUs, man-in-the-middle, and even DoS. Catalyst 6500 and 4500 Series switches provide a set of tools that can be deployed jointly to mitigate, and in some cases even prevent, attacks against STP. The following guidelines pertain to deploying these tools in a systematic manner.

By default, all Ethernet ports on Catalyst switches are set to auto-negotiated trunking mode. Ports configured in this mode automatically negotiate the configuration of trunks. Leaving auto-negotiated trunking mode indiscriminately enabled on all ports could allow anyone connected to one of these ports to establish an illegal trunk. Therefore, auto-negotiated trunking should be disabled on all ports connecting to all non-switching devices, such as workstations and servers. In a more restrictive approach, administrators can opt to disable auto-negotiated trunks on all ports, and allow only for the manual configuration of trunks as needed.

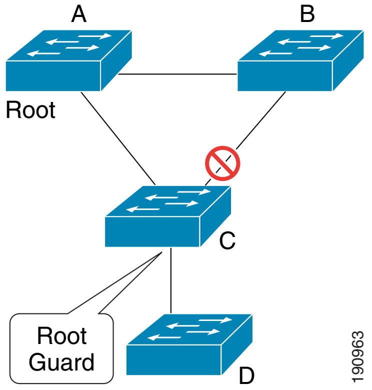
Per-VLAN Spanning Tree (PVST) is another recommended feature that should be enabled on all switches. PVST implements a separate instance of spanning tree for each VLAN configured in the network, making the network more resilient from attacks against spanning tree. With PVST, if a problem occurs in one VLAN, the effects are contained in that VLAN, shielding the rest of the network. The Catalyst switches implement several versions of PVST (such as PVST+ and Rapid-PVST+). It should be noted that some of these versions are Cisco proprietary, and they should not be used in multi-vendor environments.

Another good practice is to enable BPDU guard on all ports connecting to non-switching devices, such as workstations and servers. Non-switching devices are not supposed to participate in STP. Therefore, they should not send any BPDUs. When enabled on a port, BPDU guard shutdowns the port as soon as a BPDU is received in that port. In this way, BPDU guard helps prevent unauthorized access and the illegal injection of forged BPDUs. It should be noted that BPDU guard requires STP Portfast to first be configured on the port.

Because all topology calculations in STP are based on the location of the root bridge, another good security practice is to enforce the placement of the root bridge. To that end, an administrator can set the root bridge priority to 0. But unfortunately, nothing can prevent another switch from being configured with a priority 0 and a lower MAC address. Fortunately, STP root guard provides an effective way to enforce the placement of the root bridge. When enabled on selected ports STP root guard prevents the surrounding switches from becoming the root bridge.

STP root guard should be configured on all ports connecting to non-root candidate switches (switches that should never become the root bridge). In a typical environment, you can always identify ports where the root bridge should not appear. For example, as shown in Figure 6, Switch A is the root bridge and Switch D is not expected to become root. Hence, to prevent Switch D from negotiating a root role, STP root guard should be enabled on the Switch C port that connects to Switch D.

Figure 6 STP Root Guard



Note Loop Guard cannot be enabled if root guard is enabled.

Deploying Routing Protocol Security

This section provides the deployment guidelines for the tools and best practices used to protect the dynamic exchange of routing updates, as described in Routing Protocol Security, page 24.

Implementing neighbor authentication is a recommended security practice. Protocols such as BGP, EIGRP, OSPF, and IS-IS support various forms of neighbor authentication, including plain text and MD5 authentication. Plain text authentication sends the authenticating key itself over the wire, which is not secure. Plain text authentication only helps avoid accidental changes to the routing updates. Whenever available, MD5 authentication should be used instead because it does not reveal any key information. MD5 authentication helps prevent the insertion of bogus routers into the routing domain, prevents the injection of forged routing updates, and in addition it ensures the integrity of routing updates.

IS-IS, EIGRP and RIPv2 offer the additional function of *key chains*. A key chain is a series of keys with lifetimes, and which are used in sequence. This decreases the likelihood of keys being compromised. BGP and OSPF support MD5 authentication, but they do not currently support MD5 key chains, although work on this is in progress.

Enabling TTL security check is another valuable practice because it helps mitigate multi-hop attacks. TTP security check is currently available on BGP only. However, the support of this feature will be extended to other routing protocols such as OSPF and EIGRP.

Another important security practice is to avoid using interior gateway protocols with external routing domains, such as extranets, to customers or business partners. It is a better practice to use an exterior gateway protocol such as BGP, or even static routes. BGP is designed to handle the rapid changes involved in dealing with routing information outside the local network administrator's control. In addition, BGP allows the definition of flexible route filtering policies.

Route filtering is another highly recommended security practice because it protects the network from the accidental or intentional injection of invalid routing information. Route filtering should be implemented at the edge routers connecting to the Internet and extranets and, ideally, inside the network at topological boundaries.

Filtering at the extranet edge routers should be aggressive, only allowing the minimum necessary, rather than accepting everything except a few selected networks. These filters should deny all external routes by default, and only permit just those necessary to reach the intended networks. It is also recommended to make sure outside peers do not advertise again your own routes to other peers. While there is no way to guarantee this, using BGP you can tag the routes you advertise to your extranet peers with the NO_EXPORT community, which instructs their routers not to advertise the route to any of their external peers.

Filtering at the Internet edge follows other requirements that differ from the ones governing filtering at the extranet edge. At the Internet edge routers, ingress route filtering should be designed to permit most routes, and deny only a specific set of routes. The routes you typically want to deny are your own networks, private address, and special use networks and bogons.

Route filtering should also be deployed within a network, at topological edges and redistribution points, to prevent false routing information from being injected. Typically, consider filtering routing information coming from remote sites back to a central location (or data center), and filtering routing information coming from any open areas of the network, such as lab networks.

Deploying Catalyst Integrated Security

The section, Catalyst Integrated Security, page 27 introduces the advanced security features available on Catalyst 6500 and 4500 Series switches. As some of these features rely on each other, it is critical to understand their dependencies prior to deployment. This section describes the relationship between some of these advanced features, and provides the recommended deployment guidelines.

Port security is a valuable feature that should be enabled on ports connecting to non-switching devices like workstations and servers. Port Security helps mitigate MAC flooding and other Layer 2 Content Addressable Memory (CAM) overflow attacks by only allowing packets with trusted MAC addresses.

When deploying port security on ports connecting to IP phones, ports should be configured to allow at least three MAC addresses: one for the workstation, one for the phone on the voice VLAN, and one for the phone on the native VLAN for CDP. In addition, the violation action should be set to “restrict” to prevent the port from being taken down entirely when a violation occurs.

Traffic storm control (traffic suppression) is a feature that can be configured on selected ports to control packet storms. We recommend that you configure traffic storm control on ports where traffic storms can enter the network, typically the access ports. When deploying traffic control. it is important to understand the platform limitations. Traffic storm control in Catalyst 6500 Series switches can limit unicast, multicast, and broadcast packet storms, while in Catalyst 4500, storm control can limit multicast and broadcast packet storms only.

DHCP snooping is a highly recommended security feature for DHCP environments, and it is required by IP Source Guard and Dynamic ARP Inspection. DHCP snooping can intercept DHCP messages crossing a switch and can block bogus DHCP offers. When configuring DHCP, ports are set as trusted or untrusted. Typically, the trusted ports are used to reach DHCP servers or relay agents. Links and trunks between switches should also be set as trusted ports, while ports connecting to clients, workstations, and servers should be configured as untrusted.

IP source guard is a feature that relies on DHCP snooping, and that is used to mitigate IP address spoofing. IP source guard mitigates spoofing by allowing only the IP addresses that are obtained through DHCP snooping on a particular port. Typically, IP source guard should be enabled on ports connecting to non-switching devices (such as workstations and servers). IP Source Guard can also be enabled on trunks. However, if it is enabled on a trunk port with a large number of VLANs that have DHCP snooping enabled, you might run out of ACL hardware resources, and some packets might be switched in software instead.

Dynamic ARP Inspection (DAI) is another feature that uses DHCP snooping. DAI is a useful tool that helps prevent ARP poisoning and other ARP-based attacks. DAI prevents these attacks by intercepting all ARP requests and responses, and by verifying their authenticity before updating the switch's local ARP cache or forwarding the packets to the intended destinations. To validate ARP packets, DAI uses the binding table dynamically populated by DHCP snooping. DAI also allows the configuration of static entries to support systems with fixed addresses and which do not use DHCP for their address configuration. In a typical DAI configuration, ports connecting switches should be configured as trusted, while ports connecting to clients and severs should be left as untrusted, which is the default setting.

It should be noted that because DAI relies on the information learned by DHCP snooping, ports should have the same trust configuration for both features. For example, a port configured as trusted with DIA needs to also be configured as trusted for DHCP snooping. Otherwise, all ARP requests and responses on that port will be blocked unless an ARP ACL entry is configured to allow systems to be reachable throughout that port.

Catalyst 6500 Hardware Rate Limiters and CoPP

As described in Additional Catalyst 6500 Infrastructure Protection Features, page 67 the Catalyst 6500 Series switches implement Control Plane Policing (CoPP) and specific hardware-based rate limiters that help protect the switch from direct infrastructure attacks and collateral damage.

Because both CoPP and hardware-based rate limiters help protect the switch itself, and because they operate in a similar fashion, they could be wrongly perceived as overlapping technologies, rather than being complementary technologies. On one hand, CoPP provides a more flexible and granular policy definition that can handle a wide variety of attacks, while each hardware-based rate limiter can cover only a limited set of specific DoS scenarios. There are certain types of traffic that CoPP does not support in hardware, and for which the hardware-based rate limiters provide better support. For example, CoPP processes multicast and broadcast traffic in software, while there are hardware-based rate limiters that handle that sort of traffic. Other packet types that CoPP does not support in hardware include packets with TTL equal to 1, packets that fail the MTU check, packets with IP options, and IP packets with errors. There are also other types of traffic, such as ARP, that CoPP cannot handle in either software or hardware. ARP rate limiting can only be done with hardware-based rate limiters because CoPP cannot process ARP traffic. To rate limit ARP traffic you should use the ARP policing rate limiter.

While deploying hardware-based rate limiters there are some important considerations that should be taken into account:

- CoPP is preferable over the FIB (CEF) Receive rate limiter. Use CoPP rather than this rate limiter and do not use both mechanisms in conjunction.
- Do not use the IP Sec features rate limiter unless you are using authentication proxy, IPSec, or inspection.
- Do not use the VACL log rate limiter unless VACL Log is configured.
- None of the Layer 2 rate-limiters (Layer 2 multicast IGMP, Layer 2 protocol tunneling, Layer 2 PDU) is supported in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed.
- Being too aggressive with the Layer 2 PDU rate limiter could have adverse effects on the Layer 2 network stability.
- The multicast IGMP rate limiter should be used only when IGMP snooping is enabled.

Additional References

This section provides links and references to information on some of the subjects covered in this document:

- Catalyst 6500 Series switches DoS protection:

  http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_guide_chapter09186a0080435872.html
- Control Plane Policing White Paper:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a008052446b.html
- Virtual LAN Security Best Practices:

  http://www.cisco.com/en/US/products/hw/switches/ps708/products_white_paper09186a008013159f.shtml
- Understanding Rapid Spanning Tree Protocol (802.1w):

  http://www.cisco.com/en/US/tech/tk389/tk621/technologies_white_paper09186a0080094cfa.shtml
- SAFE: Best Practices for Securing Routing Protocols:

  http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008020b51d.shtml

Unneeded Services

Cisco Catalyst switches and other infrastructure devices are typically shipped with many services that are considered appropriate for most network environments enabled by default. While default services certainly ease deployment, from a security standpoint there is always a risk that services could potentially present a vulnerability that could be used maliciously to gain unauthorized access, or generate a denial of service. For this reason, it is a good practice to disable all unneeded services.

Note Before disabling a service, first verify that the service is not needed.

This section describes how to disable some services that might not be needed.

Unneeded Services in Cisco IOS and Catalyst OS

The following is a list of services available in both Cisco IOS (native) and Catalyst OS (hybrid):

- Cisco Discovery Protocol (CDP), page 92
- ICMP Redirects, page 93
- ICMP Unreachables, page 93

Cisco Discovery Protocol (CDP)

Cisco Discovery Protocol is a Cisco proprietary Layer 2 protocol designed to facilitate the administration and troubleshooting of network devices by providing information on neighboring equipment. With CDP enabled, network administrators can run CDP commands that provide them with the platform, model, software version, and even the IP addresses of adjacent equipment.

CDP is a useful protocol, but could clearly reveal important information to an attacker. CDP is enabled by default and can be disabled globally or for each interface. The best practice is to disable CDP globally when the service is not used, or per interface when CDP is still required. In cases where CDP is used for troubleshooting, CDP should be left enabled globally, and should be disabled only on those interfaces on which the service could represent a risk, for example, interfaces connecting to the Internet. As a general practice, CDP should not be enabled on interfaces that connect to external networks, such as the Internet.

To disable CDP globally:

- On systems running Catalyst OS use the **set cdp disable** command, as shown in the following example:

  ```
  Console> (enable) set cdp disable
  ```

- On systems running Cisco IOS use the **no cdp run** command from global configuration mode, as shown in the following example:

  ```
  Router(config)# no cdp run
  ```

To disable CDP on one or more interfaces:

- On systems running Catalyst OS, use the **set cdp disable**{*mod/ports*...} command, as shown in the following example:

  ```
  Console> (enable) set cdp disable 2/1
  ```

- On systems running Cisco IOS, use the **no cdp enable** command from interface configuration mode, as shown in the following example:

  ```
  Router(config-if)# no cdp enable
  ```

Note Features such as ODR (on demand routing) depend on CDP, so check for dependencies prior to disabling CDP.

For more information about CDP in Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/ses_sete.htm#wp1026797

For more information about CDP in IOS, refer to the following URL:

http://www.cisco.com/en/US/partner/tech/tk962/technologies_tech_note09186a00801aa000.shtml

ICMP Redirects

By default, Cisco IOS and Catalyst OS software send ICMP redirect messages when the switch is forced to resend a packet through the same interface on which it was received. By sending these redirect messages the switch instructs the host the specific router to use to reach a particular destination. The ICMP redirect messages can also reveal information that can potentially be used by an attacker for discovering the network topology. Therefore, we recommend that you disable this service on all interfaces:

- On systems running Catalyst OS use the set ip redirect disable to globally disable IP redirects, as shown in the following example:

```
Console> (enable) set ip redirect disable
```

- On systems running Cisco IOS, IP redirects can be disabled per interface by using the no ip redirects interface configuration command, as shown in the following example:

```
Router(config-if)# no ip redirects
```

For more information about the Catalyst OS **set ip redirect** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_f_l.htm#wp1026328

For more information about the Cisco IOS **ip redirect** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipras_r/ip1_i2g.htm#wp1081518

ICMP Unreachables

According to Internet standards (RFC 1812), whenever a router needs to drop a packet, it should return an ICMP unreachable message to the source. Routers typically drop incoming packets either because they cannot find a valid route or because the packet should be routed to the Null interface. The latter is typically the case with black hole filtering. In some cases, it is possible to overwhelm a router by sending large amounts of packets that require the creation of ICMP unreachables. For this reason it is highly recommended to control the generation of ICMP unreachables by either rate-limiting or disabling it.

ICMP unreachables are generated by default on switches running Cisco IOS and Catalyst OS. Use one of the following best practices to protect the switches from ICMP unreachable overload:

- Disable ICMP unreachable messages
- Rate limit ICMP unreachable traffic

The first workaround is to prevent the switch from sending ICMP unreachables:

- On systems running Catalyst OS, ICMP unreachables can be globally disabled by using the **set ip unreachable disable** command, as shown in the following example:

```
Console> (enable) set ip unreachable disable
```

- On systems running Cisco IOS, ICMP unreachables can be disabled per interface by using the **no ip unreachables interface configuration** command, as shown in the following example:

```
Router(config-if)# no ip unreachables
```

For more information about the Catalyst OS **set ip unreachable disable** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_f_l.htm#wp1372177

For more information about the **no ip unreachables** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124tcr/tiap_r/apl_i2ht.htm#wp1196977

The first workaround is effective. However, in some cases ICMP unreachables are necessary, so preventing the switch from sending them is not always appropriate. The second workaround is to rate limit the number of ICMP unreachables packets that are sent, which is possible on Catalyst 6500 Series switches.

The Cisco Catalyst 6500 Series Supervisor Engine 32 and Supervisor Engine 720 forwarding engines provide a hardware-based rate limiter that controls the generation of ICMP unreachables. This rate limiter is supported in all available Catalyst OS and Cisco IOS Software releases.

- On systems running Cisco IOS, the ICMP unreachable rate limiter can be configured using the **mls rate-limit unicast ip icmp unreachable** command, as shown in the following example:

```
Router(config)# mls rate-limit unicast ip icmp unreachable
```

For more information about the **ip icmp rate-limit unreachable** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipras_r/ip1_i1g.htm#wp1081902

For more information about the ICMP unreachable rate limiter and other DoS protection controls available on the Supervisor 720, refer to the following URL:

http://www.cisco.com/en/US/partner/products/hw/switches/ps708/products_configuration_guide_chapter09186a0080435872.html

Possible Unneeded Services in Cisco IOS

The following services are available on Cisco IOS:

- Directed broadcast
- Finger protocol
- IP BOOTP server
- IP Source routing
- PAD
- Proxy ARP
- TCP and UDP small servers
- IPv6

Directed Broadcast

An IP directed broadcast packet is an IP packet whose destination address is a valid broadcast address for an IP subnet. When a directed broadcast packet reaches a router that is directly connected to its destination subnet, and if the router is configured to do so, that packet is "exploded" as a broadcast on the destination subnet. By default, earlier releases of Cisco IOS software handle directed broadcasts this way. However, because directed broadcasts have been used for attacks, such as the SMURF attack, the default behavior has been changed to drop directed broadcasts since Cisco IOS software Release 11.2.

In the case the forwarding of directed broadcast has been enabled, or in the case of Cisco IOS software releases prior to Cisco IOS software Release 11.2, it is s recommended that you disable this feature on all interfaces using the **no ip directed-broadcast interface** configuration command, as shown in the following example:

```
Router(config-if)# no ip directed-broadcast
```

For more information about the **ip directed-broadcast** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipras_r/ip1_i1g.htm#wp1081245

Finger Protocol

Finger, as defined in RFC 742, is a protocol that can be used to obtain information about users logged into a remote host or network device. Cisco IOS software incorporates a finger service, which in Cisco IOS software releases prior to 12.1(5) and 12.1(5)T was turned on by default. Although the finger service does not reveal any extremely sensitive information, it can be used by a potential attacker to gather information. Therefore it is recommended that you disable this service.

In older releases of Cisco IOS software where the finger service was enabled by default, it can be disabled using the **no service finger** global configuration command, as shown in the following example:

```
Router(config)#no service finger
```

Starting in Cisco IOS software 12.1(5) and 12.1(5)T, the finger service is disabled by default. If finger has been enabled and the service is not needed, it can be disabled using the **no ip finger** global configuration command, as shown in the following example:

```
Router(config)# no ip finger
```

For more information on the finger service, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g03.htm#wp1033299

IP BOOTP Server

As defined by RFC 951, the Bootstrap protocol allows a diskless workstation to configure itself at boot time by dynamically obtaining an IP address, the IP address of the BOOTP server, and a configuration file. Cisco IOS software implements a bootstrap service that allows a router to act as a BOOTP server providing dynamic configuration services to other Cisco IOS software routers. This service is turned on by default and it is used by features like AutoInstall, which simplifies or automates the configuration of Cisco devices. If not needed, this service should be disabled using the **no ip bootp server** global configuration command, as shown in the following example:

```
Router(config)# no ip bootp server
```

For more information about the BOOTP server service, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g03.htm#wp1031545

For more information about AutoInstall, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca735.html

IP Source Routing

The IP protocol supports source routing options that allow the sender of an IP packet to control the route that the datagram will take toward its ultimate destination, and generally the route that any reply will take. These options are rarely used for legitimate purposes in real networks. Some older IP implementations do not process source-routed packets properly, and it might be possible to crash machines running these implementations by sending them datagrams with source routing options. By default Cisco IOS software forwards IP packets with source routing header options. As a general best practice, IP source routing should be disabled unless strictly necessary. To have the software discard any IP packet containing a source-route option, use the **no ip source-route** global configuration command as shown in the following example:

```
Router(config)# no ip source-route
```

For more information about the **ip source-route** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipras_r/ip1_i2g.htm#wp1081830

PAD

Cisco IOS software provides a PAD (packet assembler/disassembler) service that allows simple devices such as character-mode terminals to connect to legacy X.25 networks. With this service, Cisco IOS software devices and other X.25 network equipment can establish PAD sessions. By default, the PAD service is enabled on Cisco IOS software, but it could be used to gain unauthorized or inappropriate access. Therefore, unless needed, this service should be disabled using the **no service pad** global configuration command, as shown in the following example:

```
Router(config)# no service pad
```

For more information about the PAD service refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/wan_r/wan_s1g.htm#wp1032441

Proxy ARP

Proxy Address Resolution Protocol (ARP), as defined in RFC 1027, is a technique that helps machines on a subnet reach remote subnets without configuring routing or a default gateway. Proxy ARP is typically implemented on routers, and when configured, the router answers all ARP requests on the local subnet on behalf of systems some hops away.

In this model, local hosts send ARP requests for each destination for which they do not have any routing information, and the router replies with its own MAC address as the next hop. By default, Cisco IOS software implements proxy ARP on all interfaces. However, unless it is specifically needed, it should be disabled using the **no ip proxy-arp interface** configuration command, as shown in the following example:

```
Router(config-if)# no ip proxy-arp
```

For more information about the **ip proxy-arp** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipras_r/ip1_i2g.htm#wp1081466

TCP and UDP Small Servers

TCP and UDP small servers are daemons that typically run on Unix systems and that were designed for diagnostic purposes. Cisco IOS software also provides an implementation of UDP and TCP small servers that enables echo, chargen, daytime and discard services. Unless strictly necessary, these services should be disabled because they can be used by a potential attacker to gather information, or to directly attack the Cisco IOS software device.

TCP and UDP small services are enabled by default on Cisco IOS software release 11.2 and earlier. These commands are disabled by default on Cisco IOS software versions 11.3 and later.

These commands can be disabled using the no service tcp-small-servers and no service udp-small-servers global configuration commands, as shown in the following example:

```
Router(config)# no service tcp-small-servers
Router(config)# no service udp-small-servers
```

For more information about TCP and UDP small servers, refer to the following URL:

http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1818/products_tech_note09186a008019d97a.shtml#tcp_udp_servers

IP version 6 (IPv6)

IPv6 is the new Internet Protocol, version 6, designed by the IETF to replace the current IPv4 (IP version 4) and IPv6 is also known as next generation Internet Protocol or IPng. On Cisco IOS software-based devices, IPv6 is disabled by default. We recommend that you keep IPv6 disabled and enable it only when necessary.

In the past, a couple of vulnerabilities found on Cisco IOS affected systems running IPv6. These vulnerabilities could lead to a system crash, or the running of arbitrary code. Only those devices that were explicitly configured to process IPv6 traffic were affected. Disabling IPv6 when it is not needed eliminates the potential exposure to such vulnerabilities.

On Cisco IOS devices where IPv6 is not needed but is enabled, the processing of IPv6 packets can be disabled per interface using the **no ipv6 enable** and **no ipv6 address** commands, as shown in the following example:

```
Router(config)# interface ethernet 0/0
Router(config-if)# no ipv6 enable
Router(config-if)# no ipv6 address
```

For more information about the **ipv6 enable** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_r/ipv6_05g.htm#wp1947766

For more information about the **ipv6 address** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_r/ipv6_04g.htm#wp1875806

Access Control

There are more access mechanisms to a switch than many administrators realize, from console to a variety of remote sessions based on protocols like Telnet, rlogin and SSH. Most of these mechanisms are not enabled by default, but others like console are. In every case it is critical to control who accesses the device. Anyone who gains access to a switch can obtain critical information about the network, reconfigure the device, and even take the device out of service. For this reason, every switch in the network infrastructure should be carefully configured to prevent any unauthorized access.

This section provides best practices to help control access to Cisco Catalyst switches:

- Secure Local Password Management, page 98
- Interactive Access Control, page 102
- Cisco IOS Login Enhancements, page 104
- Warning Banners, page 104
- Web-Based GUI Access, page 105
- Secure Shell (SSH), page 107
- SNMP Access, page 108

Secure Local Password Management

Passwords (and similar secrets, such as SNMP community strings) are the primary defense against unauthorized access to your switch. The best way to handle most passwords is to maintain them on a TACACS+ or RADIUS authentication server. However, almost every router and switch will still have a locally configured password for privileged access, and each might also have other password information in its configuration file. The following paragraphs describe some of the Cisco IOS and Catalyst OS commands available on Catalyst switches to help prevent unauthorized access.

Password Management in Catalyst OS

The following are the Catalyst OS commands that are used to implement best practices:

- **set password**
- **set enablepass**
- **set authentication login**
- **set authentication enable**

The set password Command

CLI access to switches running Catalyst OS is controlled with a local login password, which by default is not configured. Use the **set password** command to configure a login password. Passwords are case sensitive and can be from 0 to 19 characters in length, including spaces. The command prompts you for the old password. If the password you enter is valid, you are prompted to enter a new password and to verify the new password. A zero-length password is allowed by pressing Return.

The following example illustrates the use of this command:

```
Console> (enable) set password
Enter old password: <old_password>
Enter new password: <new_password>
```

```
Retype new password: <new_password>
Password changed.
Console> (enable)
```

For more information about the **set password** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_m_pi.htm#wp1025848

The set enablepass Command

In switches running Catalyst OS, privileged access to the CLI is controlled with a local enable password, which by default is not configured. Use the **set enablepass** command to configure a CLI enable password. Passwords are case sensitive and can be from 0 to 19 characters in length, including spaces. The command prompts you for the old password. If the password you enter is valid, you are prompted to enter a new password and to verify the new password.

This example shows how to set a new enable password:

```
Console> (enable) set enablepass
Enter old password: <old_password>
Enter new password: <new_password>
Retype new password: <new_password>
Password changed.
Console> (enable)
```

For more information about the **set password** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_f_l.htm#wp1061821

The set authentication login Command

In Catalyst OS, by default, access to the switches is controlled with the local login password. The **set authentication login** command can be used to enable TACACS+, RADIUS, or Kerberos as alternative authentication methods for login.

In addition, the **set authentication login** command allows you to limit the number of unsuccessful login attempts. When a user fails to authenticate after the specified number of attempts, the system delays access and logs the user ID and the IP address of the station with a syslog message and a SNMP trap.

The maximum number of login attempts is configurable through the **set authentication login attempt** *count* command. The configurable range is three (default) to ten tries. Setting the login authentication limit to zero (0) disables this function.

The lockout (delay) time can be configured through the **set authentication login lockout** *time* command. The configurable range is 30-43200 seconds. Setting the lockout time to zero (0) disables this function. When a user is locked out at the console, the console does not allow any login attempt during the lockout time. If the user is locked out with a Telnet session, the connection closes when the time limit is reached. The switch closes any subsequent access from that station during the lockout time and provides an appropriate notice.

This example shows how to limit login attempts to 5, set the lockout time for both console and Telnet connections to 50 seconds, and verify the configuration:

```
Console> (enable) set authentication login attempt 5
Login authentication attempts for console and telnet logins set to 5.
Console> (enable) set authentication login lockout 50
Login lockout time for console and telnet logins set to 50.
```

For more information about the **set authentication login attempt** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/ses_sete.htm#wp1052196

The set authentication enable Command

By default, privilege access to switches running Catalyst OS is controlled with the local enable password. The **set authentication enable** command can be used to enable TACACS+, RADIUS, or Kerberos as alternative authentication methods for enable (privilege) access.

In addition, the **set authentication enable** command allows you to limit the number of unsuccessful access attempts to the enable mode. When a user fails to authenticate after the specified number of attempts, the system delays access and logs the user ID and the IP address of the station with a syslog message and a SNMP trap.

The maximum number of login attempts is configurable through the **set authentication enable attempt** *count* command. The configurable range is three (default) to ten tries. Setting the login authentication limit to zero (0) disables this function.

The lockout (delay) time can be configured through the **set authentication enable lockout** *time* command. The configurable range is 30-43200 seconds. Setting the lockout time to zero (0) disables this function.

This example shows how to limit enable mode login attempts to 5, set the lockout time for both console and Telnet connections to 50 seconds, and verify the configuration:

```
Console> (enable) set authentication enable attempt 5
Enable mode authentication attempts for console and telnet logins set to 5.
Console> (enable) set authentication enable lockout 50
Enable mode lockout time for console and telnet logins set to 50.
```

For more information about the **set authentication login attempt** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/ses_sete.htm#wp1052090

Password Management in Cisco IOS

The following are the Cisco IOS commands that can be used to implement the above best practices:

- **service password-encryption**
- **enable secre**t
- **security password min-length**
- **security authentication failure rate**

The service password-encryption Command

By default, some passwords and secrets are shown in clear text in a Cisco IOS software configuration file or listing. The **service password-encryption** global configuration command instructs Cisco IOS software to encrypt the passwords, CHAP secrets, and similar data that are saved in the configuration file. This is shown in the following example:

```
Router(config)# service password-encryption
```

This command is primarily useful for keeping unauthorized individuals from viewing passwords in the configuration file. However, it is important to note that the algorithm used by service password-encryption is a simple Vigenere cipher that can be easily reversed, and for that reason this command should not be used with the intention to protect configuration files against serious attacks.

For more information about the **service password-encryption** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/secur_r/sec_r1g.htm#wp1070450

The enable secret Command

The **enable secret** global configuration command is used to set the password that grants privileged administrative access to the Cisco IOS software system. By default, no enable secret password is enabled, and as a general best practice, one should always be set.

To set an enable secret password, use the **enable secret** global configuration command, as shown in the following example:

```
Router(config)# enable secret Hard2Guess
```

Cisco IOS software also offers the older **enable password** command, but it is not recommended because it uses a weak encryption algorithm. The **enable secret** command provides stronger encryption based on MD5 hashing.

In addition, if no enable secret is set, and a password is configured for the console TTY line, the console password might be used to get privileged access, even from a remote VTY session. This is not a recommended practice, and makes for another good reason to configure an enable secret.

For more information about the **enable secret** command, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_command_reference_chapter09186a008017cf1d.html#wp1081495

The security password min-length Command

Introduced in Cisco IOS software Release 12.3(1), the **security password min-length** global configuration command provides enhanced security access to the router by allowing the user to specify a minimum password length, eliminating common passwords that are prevalent on most networks, such as lab and cisco. For example:

```
Router(config)# security password min-length
```

This command affects user passwords, enable passwords and secrets, and line passwords. After this command is enabled, any password that is less than the specified length will fail. This command was also integrated into Cisco IOS software Release 12.2(18)T.

For more information about the **security password min-length** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/secur_r/sec_r1g.htm#wp1081495

The security authentication failure rate Command

Introduced in Cisco IOS software Release 12.3(1), the **security authentication failure rate** global configuration command provides protection against dictionary attacks. In a dictionary attack, automated software attempts to log in using every word in a dictionary.

The **security authentication failure rate** command allows the user to define a maximum number of consecutive unsuccessful login attempts, after which, device access is locked for a period of 15 seconds. Additionally, this command can be configured to generate a syslog message every time the number of unsuccessful login attempts exceeds the configured threshold rate.

The best practice is to configure a maximum threshold of 3 consecutive unsuccessful login attempts, and to enable the generation of syslog messages, as shown in the following configuration.

```
Router(config)# security authentication failure rate 3 log
```

This configuration causes access to the router to be locked for a period of 15 seconds after three unsuccessful login attempts, disabling the dictionary method of attack. In addition to locking access to the router, this configuration causes a log message to be generated after three unsuccessful login attempts, warning the administrator of the unsuccessful login attempts.

For more information about the **security authentication failure rate** global configuration command, refer to the following URL:

http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps5187/products_feature_guide09186a008017d101.html#wp1048684

Interactive Access Control

Cisco Catalyst switches can be managed through a diverse set of mechanisms, from console to remote access protocols like Telnet and SSH. The best way to protect a switch is to make sure that appropriate controls are applied on all access mechanisms. Because it is difficult to be certain that all possible modes of access have been blocked, administrators should make sure that logins on all lines are controlled using some sort of authentication mechanism, even on machines that are supposed to be inaccessible from untrusted networks.

The following are the recommended security guidelines to control interactive access:

- Restrict access to allow only the needed protocols.
- Enable authentication on all lines. As mentioned previously, the best way to use passwords and authentication is by deploying protocols like TACACS+, RADIUS or Kerberos.
- Implement line ACLs to restrict the IP addresses or subnets from which access will be granted.
- Define timeouts to control idle sessions to prevent idle sessions from remaining up indefinitely.

The next sections describe how these best practices can be implemented in Cisco IOS and Catalyst OS.

Interactive Access in Catalyst OS

Because Catalyst switches can be accessed through a variety of remote access protocols, it is fundamental to restrict access by only allowing connections from the expected protocols. For example, if the switch is supposed to be accessed with SSH, then there is no reason to leave Telnet enabled. The **set authentication login local** command can be used to disable access for the protocols from which connections will not be expected.

Another good practice is the use of the **set ip permit** command to restrict the IP addresses from which the switch will accept connections. By default, after an access protocol is enabled any host can initiate a connection using that protocol. The **set ip permit** command defines a list of hosts or networks from which access will be allowed, which prevents unauthorized access from untrusted sources. This practice also helps mitigate a denial-of-service attack on the virtual lines.

Decreasing the connection timeout is another useful tactic. This can be done using the **set logout** command, which prevents an idle session from staying open indefinitely. By default, a session has a 20-minute timeout. Although the effectiveness of this technique against deliberate attacks is relatively limited, it also provides some protection against sessions accidentally left idle.

The following configuration illustrates the best practices just described. In this configuration, telnet access is disabled. Access is only allowed for SSH connections coming from the 172.16.0.0/16 network, and the idle timeout is set to 3 minutes.

```
Console> (enable) set authentication login local disable telnet
local login authentication set to disable for telnet session.
Console> (enable)
Console> (enable) set ip permit 172.16.0.0 255.255.0.0 ssh
172.16.0.0 with mask 255.255.0.0 added to telnet permit list.
Console> (enable)
Console> (enable) set ip permit enable ssh
SSH permit list enabled.
Console> (enable)
Console> (enable) set logout 3
Sessions will be automatically logged out after 3 minutes of idle time.
Console> (enable)
```

Interactive Access in Cisco IOS

In Cisco IOS software, all interactive access mechanisms involve sessions or lines. The local console uses a standard line, known as CTY, while remote network connections use virtual TTYs, called VTYs. Regardless of their type, both CTYs and VTYs need to be configured according to security best practices.

The first best practice consists in restricting the type of protocols accepted in each line. Prior to Cisco IOS software Release 11.1, all VTY lines were configured using the **transport input all** command by default, allowing all type of connections to the VTY lines. Starting in Cisco IOS software Release 11.1, no connections are permitted to VTY lines, unless an incoming protocol or all the protocols are specified using the **transport input** command.

Every VTYs must be configured to accept connections only from the expected protocols, which can be done by using the **transport input** command. For example, a VTY that is expected to receive only Telnet sessions must be configured with transport input telnet, while a VTY permitting both Telnet and SSH sessions should have transport input telnet ssh.

Another good practice is the use of the **access-class** command to restrict the IP addresses from which the VTY will accept connections. By default, after an access protocol is enabled for a VTY line any host can initiate a connection using that protocol. The **access-class** command defines a list of hosts or networks from which access will be allowed, which prevents unauthorized access from untrusted sources. This practice also helps mitigate a denial-of-service attack on the VTY lines.

Cisco IOS software devices have only a limited number of VTY lines, usually five. When all of the VTYs are in use, no more remote interactive connections can be established and this creates an opportunity for a denial-of-service attack. If an attacker can open remote sessions to all the VTYs on the system, the legitimate administrator might not be able to log in. The attacker does not have to log in to do this, the sessions can simply be left at the login prompt.

One way to protect against this attack is to configure a restrictive access-class configuration on the last VTY in the system. The last VTY, usually VTY 4, can be restricted to accept connections only from a single, specific administrative workstation, whereas the other VTYs might accept connections from any address in a corporate network.

Another useful tactic is to decrease the VTY timeouts using the **exec-timeout** command. This prevents an idle session from consuming a VTY indefinitely. By default, a VTY session has a 10-minute timeout. Although the effectiveness of this technique against deliberate attacks is relatively limited, it also provides some protection against sessions accidentally left idle. Similarly, enabling TCP keepalives on incoming connections (using the **service tcp-keepalives-in** command) can help to guard against malicious attacks and orphan sessions caused by remote system crashes. By default, keepalives are not enabled for incoming connections.

The following configuration illustrates the best practices just described. In this configuration, access for VTY 4 is restricted to only SSH connections coming from the IP address 10.0.0.1. The line timeout is set to 2 minutes and 30 seconds, and tcp keepalives are enabled.

```
service tcp-keepalives-in
access-list 10 permit host 10.0.0.1
line vty 4
     transport input ssh
     access-class 10 in
     exec-timeout 2 30
```

Cisco IOS Login Enhancements

Cisco IOS software Release 12.3(4)T introduced Cisco IOS login enhancements, a feature implemented with several new commands that help protect the router against dictionary and possible DoS attacks.

The login enhancements include the **login delay** global configuration command, which allows the introduction of a delay between login attempts. In the event of a dictionary attack, introducing a delay between login attempts slows down the attack, making it less likely to succeed.

This feature also includes a new global configuration command, **login block-for**, which allows you to limit the frequency of failed login attempts. The frequency is limited by defining a maximum number of failed attempts within a specified time period, after which, the IOS router will not accept any additional connections for a quiet period. It is possible to define an exception ACL for trusted systems and networks from which legitimate connections are expected. This exception ACL can be defined using the **login quiet-mode access-class** global configuration command.

In addition, the login enhancements provide the **login on-success**, and the **login on-failure** commands, which enabled the generation of syslog messages for successful and failed login attempts, respectively.

The following example shows how to configure a router to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds. All login requests will be denied during the quiet period except hosts from the ACL myacl. Also, logging messages will be generated for every 10th failed login and every 15th successful login.

```
Router(config)# login block-for 100 attempts 15 within 100
Router(config)# login quiet-mode access-class myacl
Router(config)# login on-failure log every 10
Router(config)# login on-success log every 15
Router(config)# access-list 10 permit host 10.0.0.1
```

For more information about Cisco IOS login enhancements, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hsec_c/part30/h_login.htm

Warning Banners

In some jurisdictions, civil and/or criminal prosecution of attackers who break into your systems is made much easier if you provide a banner informing unauthorized users that their use is in fact unauthorized. In other jurisdictions, you might be forbidden to monitor the activities of even unauthorized users unless you have taken steps to notify them of your intent to do so. One way of providing this notification is to put it into a banner message configured using the Catalyst OS **set banner telnet** commands, or the Cisco IOS software **banner login** global configuration command.

Legal notification requirements are complex, and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary, and this issue should be discussed with your own legal counsel. In cooperation with counsel, you should consider which of the following information should be put into your banner:

- A notice that the system is to be logged in to or used only by specifically authorized personnel, and perhaps information about who can authorize use.
- A notice that any unauthorized use of the system is unlawful, and might be subject to civil and/or criminal penalties.
- A notice that any use of the system might be logged or monitored without further notice, and that the resulting logs can be used as evidence in court.
- Specific notices required by specific local laws.

From a security, rather than a legal, point of view, your login banner usually should not contain any specific information about your router, its name, its model, what software it is running, or who owns it because this kind of information can be abused by an attacker.

For more information about the **set banner telnet** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/ses_sete.htm#wp1112270

For more information about the **banner login** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g01.htm#wp1029652

Web-Based GUI Access

Cisco IOS and Catalyst OS software provide a web browser user interface that allows the configuration of catalyst switches by using a web browser such as Internet Explorer or Netscape. This user interface relies on a built-in HTTP server service that runs on Cisco IOS and Catalyst OS software, and which is turned off by default in both operating systems.

Because of the nature of HTTP this service does not provide encryption for client connections, which leaves communication between clients and servers vulnerable to interception and attack. Whenever available, Secure HTTP (HTTPS) should be used instead of plain HTTP. Secure HTTP (HTTPS) uses Secure Sockets Layer (SSL) and Transport Layer Security (TLS) to provide device authentication and data encryption, delivering an acceptable level of protection.

Note Currently, HTTPS for Catalyst switches is available only for Cisco IOS, and is not supported in Catalyst OS. The following section provides information on web-based GUI access for Catalyst switches running Catalyst OS.

Web-Based GUI Access in Catalyst OS

Catalyst OS does not support HTTPS, but fortunately, there is set of mechanisms that can be used to secure HTTP access. Follow these steps to implement the recommended security guidelines:

Step 1 Enable user authentication using protocols such as RADIUS or TACACS+.

In Catalyst OS, HTTP authentication can be enabled using the **set authentication login** command. The following example shows a configuration listing for HTTP authentication using TACACS+.

```
Console> (enable) set tacacs server 170.1.2.20 primary
170.1.2.20 added to TACACS server table as primary server.
Console> (enable) set tacacs key MyKey
The tacacs key has been set to MyKey.
Console> (enable) set ip http server enable
HTTP server is enabled.
Console> (enable) set authentication login tacacs enable http primary
tacacs login authentication set to enable for HTTP sessions as primary authentification
method.
Console> (enable)
```

For more information about HTTP authentication using AAA, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/authent.htm

Step 2 Use a non-standard port for HTTP (other than TCP/80). In Catalyst OS this can be done using the **set ip http port** command.

For more information about the **set ip http port** command refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/cmd_ref/set_f_l.htm#wp1026145

Web-Based GUI Access in Cisco IOS

When the web interface is require, the best practice is to enable it with HTTPS instead of HTTP. In Cisco IOS the HTTPS service can be enabled using the **ip http secure-server** global configuration command.

For more information on the **ip http secure-server** command, refer to the following URL:

http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1833/products_feature_guide09186a00800d9eee.html#wp1021949

In cases where HTTPS cannot be used or is not available, perform the following steps to secure the HTTP-based GUI:

Step 1 Enable user authentication using protocols like RADIUS or TACACS+. In Cisco IOS HTTP authentication can be enabled using the **ip http authentication** global configuration command. The following example shows a configuration listing for HTTP authentication using TACACS+.

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
ip http server
ip http authentication aaa
tacacs-server host 171.68.18.10
tacacs-server key Cisco
```

For more information about HTTP authentication refer to the following URL:

http://www.cisco.com/en/US/partner/tech/tk59/technologies_configuration_example09186a0080178a51.shtml

Step 2 Use a non-standard port for HTTP (other than TCP/80). In IOS this can be done using the **ip http port** global configuration command.

For more information about the **ip http port** command refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g04.htm#wp1028992

Step 3 Deploy HTTP ACLs to allow access only from trusted hosts or networks. In IOS this can be done using the **ip http access-class** global configuration command. The following example shows how access to the HTTP server is configured to be allowed from a single host (10.0.0.1) only.

```
Router(config)# ip http access-class 10
Router(config)# access-list 10 permit host 10.0.0.1
```

For more information on the **http access-class** command, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g04.htm#wp1028455

Step 4 Limit the maximum number of connections to the built-in HTTP server. Cisco IOS supports the **ip http max-connections** global configuration command for this purposes. The following example shows how to limit the maximum number of concurrent connections to three:

```
Router(config)# ip http max-connections 3
```

For more information, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fun_r/cfr_1g04.htm#wp1028838

Secure Shell (SSH)

SSH is a remote access protocol that implements strong authentication and encryption, and which for that it is recommended over insecure protocols like telnet. There are two versions of SSH, v1 and v2. SSHv2 performs better, and fixes a series of security issues found in the previous version, for those reasons v2 should be used whenever it is supported. Both versions of SSH are currently supported by Catalyst OS and Cisco IOS.

SSH in Catalyst OS

The following steps are required to enable SSH on a Catalyst switch running Catalyst OS:

Step 1 Generate an RSA key pair.

Step 2 Limit SSH access to trusted hosts/networks (this is not mandatory, but we recommend it).

Step 3 Enable SSH.

Note SSH requires CatOS K9 software images.

The following example shows how SSH can be configured on Catalyst OS:

```
!--- Step 1: Generate an RSA key pair for your switch.
Console> (enable) set crypto key rsa
Generating RSA keys..... [OK]
Console> (enable) ssh_key_process: host/server key size: 1024/768

!--- Step 2: Optionally define an IP filter to control which hosts/network accesses the
switch.
```

```
Console> set ip permit 172.18.124.0 255.255.255.0
172.18.124.0 with mask 255.255.255.0 added to IP permit list.

!--- Step 3: Turn SSH on.
Console> (enable) set ip permit enable ssh
SSH permit list enabled.
```

For more information about SSH configuration on Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/connect.htm#wp1023537

SSH in Cisco IOS

The following steps are required to enable SSH support on a Catalyst switch running Cisco IOS:

Step 1 Configure a hostname and DNS domain for the router.

Step 2 Generate an RSA key pair.

Step 3 Configure time-out and number of authentication retries.

Step 4 Limit VTYs to SSH only (this is not mandatory, but we recommend it).

Note SSH requires an IPSec (DES or 3DES) encryption IOS software image.

The following example shows how SSH can be configured on an IOS router:

```
!--- Step 1: Configure a hostname and domain name.
Router(config)# hostname router
Router(config)# ip domain-name nyc.cisco.com

!--- Step 2: Generate an RSA key pair for your router, which automatically enables SSH.
Router(config)# cry key generate rsa

!--- Step 3: Configure time-out and number of authentication retries.
Router(config)# ip ssh time-out 60
Router(config)# ip ssh authentication-retries 2

!--- Step 4: Configure VTYs to only accept SSH.
Router(config)# line vty 0 4
Router(config-line)# transport input ssh
```

For more information about SSH configuration on IOS routers, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hsec_c/part25/ch10/index.htm

SNMP Access

Simple Network Management Protocol (SNMP) is the most popular network management protocol, and as such is widely supported in the networking industry. There are three versions of SNMP: version 1, the oldest one but still frequently supported, version 2c, the most commonly deployed, and version 3, an IETF standard that provides enhanced security.

SNMP versions 1 and 2c are weak in security. In these earlier versions of SNMP, access to MIB objects is primarily controlled by the use of community strings, but neither version provides authentication or encryption. Without authentication it is possible for unauthorized users to execute SNMP transactions, and even masquerade legitimate users. In addition, the lack of encryption facilitates the interception of SNMP messages, potentially leading to the disclosure of community strings and other sensitive information. SNMP version 3 deals with these issues by incorporating security features such as authentication, identity, and access control.

SNMPv3 supports multiple authentication options including username, Message Digest 5 (MD5), and Secure Hash Algorithm (SHA) authentication. This version also provides privacy with DES encryption, and authorization and access controls based on views. For these enhanced security functions, SNMPv3 should be preferred over SNMPv1 and SNMPv2c wherever it is supported.

In cases where SNMPv3 is not available, security can be improved by taking the following basic security measures:

- Change any default or standard community strings such as private or public.
- Define non-trivial community strings.
- Set SNMP to send a trap on community-name authentication failures.
- Define ACLs to control from which hosts or networks management messages will be accepted.

For more information about SNMP, refer to the following URL:

http://www.cisco.com/en/US/tech/tk648/tk362/tk605/tsd_technology_support_sub-protocol_home.html

Other Security Services

This document is focused on security techniques and features destined to protect the switches from direct attacks. Catalyst 6500 and 4500 Series switches implement other security services that do not provide infrastructure protection, but that still help secure a network:

TCP Intercept

TCP Intercept is a security feature available on Catalyst 6500 Series switches running Cisco IOS software, and that protect servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack.

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/secure.htm#wp1038471

Private VLANs

Private VLANs (PVLANs) is an available feature on Catalyst 6500 and 4500 Series switches running Catalyst OS and Cisco IOS software. PVLANs provide Layer 2 isolation between ports within the same PVLAN. There are three types of PVLAN ports:

- Promiscuous—A promiscuous port can communicate with all interfaces, including the isolated and community ports within a PVLAN.
- Isolated—An isolated port has complete Layer 2 separation from the other ports within the same PVLAN, but not from the promiscuous ports. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic from isolated port is forwarded only to promiscuous ports.

- Community—Community ports communicate among themselves and with their promiscuous ports. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.

For more information about PVLANS on Catalyst 6500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/pvlans.htm

For more information about PVLANS on Catalyst 4500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/pvlans.htm

802.1X Authentication

802.1X Authentication is an available feature on Catalyst 6500 and 4500 Series switches that provides advanced per port access control.

802.1X defines a client-server-based access control and authentication protocol that restricts unauthorized devices from connecting to a LAN through publicly accessible ports. 802.1X controls network access by creating two distinct virtual access points at each port. One access point is an uncontrolled port, the other is a controlled port. All traffic through the single port is available to both access points. 802.1X authenticates each user device that is connected to a switch port and assigns the port to a VLAN before making available any services that are offered by the switch or the LAN. Until the device is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the device is connected. After authentication is successful, normal traffic can pass through the port. You can restrict the traffic in both directions, or you can restrict just the incoming traffic.

For more information about 802.1X Authentication on Catalyst 6500 Series switches running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_5/confg_gd/8021x.htm

For more information about 802.1X Authentication on Catalyst 6500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/dot1x.htm

For more information about 802.1X Authentication on Catalyst 4500 Series switches running Catalyst OS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/8_3/configur/8021x.htm

For more information about 802.1X Authentication on Catalyst 4500 Series switches running Cisco IOS, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_2_31s/conf/dot1x.htm

Catalyst 6500 Security Service Modules

The Catalyst 6500 Series switches support a suite of advanced security modules, such as firewall, IPSec VPNs, intrusion prevention, DoS mitigation, SSL, and gigabit network analysis. These modules are described in this section.

Firewall Services Module (FWSM)

The Firewall Services module (FWSM) is a high-speed, integrated firewall module for Cisco Catalyst 6500 switches and Cisco 7600 Series routers, and provides the fastest firewall data rates in the industry: 5-Gbps throughput, 100,000 CPS, and 1M concurrent connections. Up to four FWSMs can be installed in a single chassis providing scalability to 20 Gbps per chassis. Based on Cisco PIX®Firewall technology, the FWSM provides large enterprises and service providers with unmatched security, reliability, and performance.

IPSec VPN Services Module

The Cisco IPSec VPN Services module is a high-speed module for the Cisco Catalyst 6500 Series switch and the Cisco 7600 Series Internet router that provides infrastructure-integrated IPSec VPN services to meet the need for ubiquitous connectivity and increased bandwidth requirements.

WebVPN Services Module

The Cisco WebVPN Services module is a high-speed, integrated Secure Sockets Layer (SSL) VPN services module for Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers that addresses the scalability, performance, application support, and security required for large-scale, remote-access SSL VPN deployments. Supporting up to 32,000 SSL VPN users and 128,000 connections per chassis, the Cisco WebVPN Services module can cost-effectively meet the capacity requirements of large enterprises. The unique virtualization capabilities integrated into the module simplify policy creation and enforcement for diverse enterprise user communities and make it an ideal solution for managed service providers. Taking advantage of the broad, industry-proven application support and endpoint security provided by Cisco VPN 3000 Series concentrators, the Cisco WebVPN Services module is ideally suited to meet the secure connectivity demands of any organization.

Content Switching Module with SSL (CSM-S)

The Cisco Content Switching module with SSL (CSM-S) is a service module that combines advanced Layer 4 to Layer 7 content switching capabilities with Secure Sockets Layer (SSL) acceleration in a single-slot line card for Catalyst 6500 Series switches and Cisco 7600 Series routers.

Anomaly Guard Services Module

The Cisco Anomaly Guard Services module for Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers delivers a powerful, extensive, and integrated solution for defending online applications and businesses, data centers, and the network infrastructure against increasingly complex and elusive DDoS attacks. This solution can be deployed in the largest enterprises and at service provider locations for managed services delivery.

Traffic Anomaly Detector Services Module

The Cisco Traffic Anomaly Detector Services module is an integrated services module for Cisco Catalyst 6500 Series switches and Cisco 7600 Series routers that protects large organizations against DDoS and other online assaults by quickly detecting attacks and automatically activating the Cisco Anomaly Guard to initiate mitigation services that block attacks before business is adversely affected. Working with Cisco Anomaly Guard services modules deployed in Cisco Catalyst 6500 Series switches or Cisco 7600 Series routers, the Cisco Traffic Anomaly Detector Services module contributes to the industry's most complete DDoS protection system. By constantly monitoring for, and detecting the start of, potential DDoS attacks, the Cisco Traffic Anomaly Detector Services module enables the activation of intelligent mitigation by the Cisco Anomaly Guard. The combined solution provides a scalable, flexible, and cost-effective method to help ensure that business integrity is always preserved, even while under attack.

Network Analysis Module (NAM)

The Network Analysis module (NAM) is a powerful integrated traffic monitoring solution for the high-performance Catalyst 6500 Series switches and Cisco 7600 Series routers that enables network managers to gain application-level visibility into network traffic with the ultimate goal of improving performance, reducing failures, and maximizing returns on network investment. The new generation of NAMs are available in two hardware versions, NAM-1 and NAM-2, to meet diverse network analysis needs in scalable switching and routing environments running at gigabit speeds. The NAMs come with an embedded, web based traffic analyzer, which provides full scale remote monitoring and troubleshooting capabilities that are accessible using a web browser.

Commonly Used Protocols

Table 4 lists the protocols commonly used in the infrastructure.

Table 4 Commonly Used Protocols

Protocol	Protocol Number, TCP/UDP Ports, and Message Type
BGP	TCP/179
OSPF	Prot 89
EIGRP	Prot 88
GRE	Prot 47
AH	Prot 51
ESP	Prot 50
TACACS+	TCP/49
RADIUS	UDP/1812, UDP/1813, in the past UDP/1645 and UDP/1646
SSH	TCP/22
TELNET	TCP/23
SNMP	UDP/161
NTP	UDP/123
ICMP	Prot 1, ttl-exceeded, port-unreachable, echo, echo-reply
DNS	UDP/53