Port ACLs (PACLs)

- Prerequisites for PACLs, page 75-1
- Restrictions for PACLs, page 75-2
- Information About PACLs, page 75-2
-
- Interface Template ACLs, page 75-7
-

Note

- For complete syntax and usage information for the commands used in this chapter, see these publications:

  http://www.cisco.com/en/US/products/ps11846/prod_command_reference_list.html
- Cisco IOS Release 15.4SY supports only Ethernet interfaces. Cisco IOS Release 15.4SY does not support any WAN features or commands.
- Port ACLs do not support the access-list keywords **log** or **reflexive**. These keywords in the access list are ignored. OAL does not support PACLs.
- PACLs are not supported on private VLANs.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Prerequisites for PACLs

None.

Restrictions for PACLs

- There can be at most one IP access list and one MAC access list applied to the same Layer 2 interface per direction.
- PACLs are not applied to MPLS or ARP messages.
- An IP access list filters only IPv4 and IPv6 packets. For IP access lists, you can define a standard, extended, or named access-list.
- A MAC access list filters ingress packets that are of an unsupported type (not IP, ARP, or MPLS packets) based on the fields of the Ethernet datagram. A MAC access list is not applied to IP, MPLS, or ARP messages. You can define only named MAC access lists.
- The number of ACLs and ACEs that can be configured as part of a PACL are bounded by the hardware resources on the switch. Those hardware resources are shared by various ACL features (such as VACLs) that are configured on the system. If there are insufficient hardware resources to program a PACL in hardware, the PACL is not applied.
- PACL does not support the access-list **log** and **reflect/evaluate** keywords. These keywords are ignored if you add them to the access list for a PACL.
- OAL does not support PACLs.
- The access group mode can change the way PACLs interact with other ACLs. To maintain consistent behavior across Cisco platforms, use the default access group mode (merge mode).
- PACLs cannot filter Physical Link Protocols and Logical Link Protocols, such as CDP, VTP, DTP, PAgP, UDLD, and STP, because the protocols are redirected to the RP before the ACL takes effect. You can apply CoPP or QoS to Physical Link Protocol and Logical Link Protocol traffic.
- Layer 2 ports do not support Layer 3 ACLs.
- An Interface Template ACL can be applied on an interface only when there is no other ACL directly configured on the same interface. An ACL that is configured directly on an interface takes priority over an Interface Template ACL that is applied on the same interface. The Interface Template takes effect once the ACL directly applied on the interface has been removed.

Information About PACLs

- PACL Overview, page 75-2
- EtherChannel and PACL Interactions, page 75-4
- Dynamic ACLs (Applies to Merge Mode Only), page 75-4
- Trunk Ports, page 75-4
- Layer 2 to Layer 3 Port Conversion, page 75-4
- Port-VLAN Association Changes, page 75-4

PACL Overview

PACLs filter incoming traffic on Layer 2 interfaces, using Layer 3 information, Layer 4 header information, or non-IP Layer 2 information.

The PACL feature uses standard or extended IP ACLs or named MAC-extended ACLs that you want to apply to the port.

Port ACLs perform access control on all traffic entering the specified Layer 2 port.

PACLs and VACLs can provide access control based on the Layer 3 addresses (for IP protocols) or Layer 2 MAC addresses (for non-IP protocols).

The port ACL (PACL) feature provides the ability to perform access control on specific Layer 2 ports. A Layer 2 port is a physical LAN or trunk port that belongs to a VLAN. Port ACLs are applied only on the ingress traffic. The port ACL feature is supported only in hardware (port ACLs are not applied to any packets routed in software).

When you create a port ACL, an entry is created in the ACL TCAM. You can use the **show tcam counts** command to see how much TCAM space is available.

The PACL feature does not affect Layer 2 control packets received on the port.

You can use the **access-group mode** command to change the way that PACLs interact with other ACLs.

PACLs use the following modes:

- Prefer port mode—If a PACL is configured on a Layer 2 interface, the PACL takes effect and overwrites the effect of other ACLs (Cisco IOS ACL and VACL). If no PACL feature is configured on the Layer 2 interface, other features applicable to the interface are merged and are applied on the interface.
- Merge mode—In this mode, the PACL, VACL, and Cisco IOS ACLs are merged in the ingress direction following the logical serial model shown in Figure 75-2. This is the default access group mode.

You configure the **access-group mode** command on each interface. The default is merge mode.

Note A PACL can be configured on a trunk port only after prefer port mode has been selected. Trunk ports do not support merge mode.

To illustrate access group mode, assume a physical port belongs to VLAN100, and the following ACLs are configured:

- Cisco IOS ACL R1 is applied on routed interface VLAN100.
- VACL (VLAN filter) V1 is applied on VLAN100.
- PACL P1 is applied on the physical port.

In this situation, the following ACL interactions occur:

- In prefer port mode, Cisco IOS ACL R1 and VACL V1 are ignored.
- In merge mode, Cisco IOS ACL R1, VACL V1 and PACL P1 are merged and applied on the port.

Note The CLI syntax for creating a PACL is identical to the syntax for creating a Cisco IOS ACL. An instance of an ACL that is mapped to a Layer 2 port is called a PACL. An instance of an ACL that is mapped to a Layer 3 interface is called a Cisco IOS ACL. The same ACL can be mapped to both a Layer 2 port and a Layer 3 interface.

The PACL feature supports MAC ACLs, IPv4, and IPv6 ACLs. The PACL feature does not support ACLs for ARP or Multiprotocol Label Switching (MPLS) traffic.

EtherChannel and PACL Interactions

This section describes the guidelines for the EtherChannel and PACL interactions:

- PACLs are supported on the main Layer 2 channel interface but not on the port members. A port that has a PACL configured on it may not be configured as an EtherChannel member port. The EtherChannel configuration commands are unavailable on ports that are configured with a PACL.
- Changing the configuration on the logical port affects all the ports in the channel. When an ACL is mapped to the logical port belonging to a channel, it is mapped to all ports in the channel.

Dynamic ACLs (Applies to Merge Mode Only)

Dynamic ACLs are VLAN-based and are used by two features: CBAC and GWIP. The merge mode *does not* support the merging of the dynamic ACLs with the PACLs. In merge mode, the following configurations are not allowed:

- Attempting to apply a PACL on a port where its corresponding VLAN has a dynamic ACL mapped. In this case, the PACL is not applied to traffic on the port.
- Configuring a dynamic ACL on a VLAN where one of its constituent ports has a PACL installed. In this case, the dynamic ACL is not applied.

Trunk Ports

To configure a PACL on a trunk port, you must first configure port prefer mode. The configuration commands to apply a PACL on a trunk or dynamic port will not be available until you configure the port in port prefer mode by entering the **access-group mode prefer port** interface command. Trunk ports do not support merge mode.

Layer 2 to Layer 3 Port Conversion

If you reconfigure a port from Layer 2 to Layer 3, any PACL configured on the port becomes inactive but remains in the configuration. If you subsequently configure the port as Layer 2, any PACL configured on the port becomes active again.

Port-VLAN Association Changes

You can enter port configuration commands that alter the port-VLAN association, which triggers an ACL remerge.

Unmapping and then mapping a PACL, VACL, or Cisco IOS ACL automatically triggers a remerge.

In merge mode, online insertion or removal of a switching module also triggers a remerge, if ports on the module have PACLs configured.

PACL and VACL Interactions

- PACL Interaction with VACLs and Cisco IOS ACLs, page 75-5
- Bridged Packets, page 75-5
- Routed Packets, page 75-6
- Multicast Packets, page 75-6

PACL Interaction with VACLs and Cisco IOS ACLs

This section describes the guidelines for the PACL interaction with the VACLs and Cisco IOS ACLs.

For an incoming packet on a physical port, the PACL is applied first. If the packet is permitted by the PACL, the VACL on the ingress VLAN is applied next. 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. However, there is currently no hardware support for output PACLs.

The PACLs override both the VACLs and Cisco IOS ACLs when the port is configured in prefer port mode. The one exception to this rule is when the packets are forwarded in the software by the route processor (RP). The RP applies the ingress Cisco IOS ACL regardless of the PACL mode. Two examples where the packets are forwarded in the software are as follows:

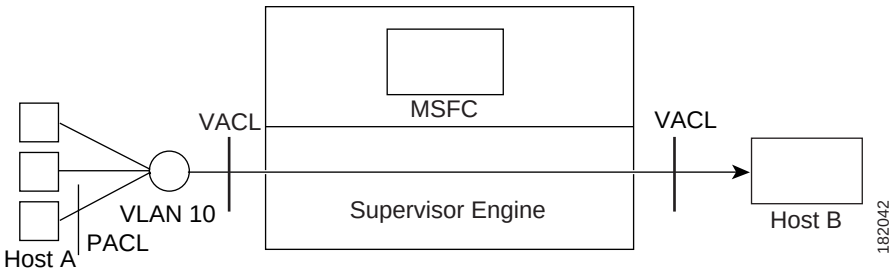
- Packets that are egress bridged (due to logging or features such as NAT)
- Packets with IP options

Bridged Packets

Figure 75-1 shows a PACL and a VACL applied to bridged packets. In merge mode, the ACLs are applied in the following order:

1. PACL for the ingress port
2. VACL for the ingress VLAN
3. VACL for the egress VLAN

Figure 75-1 Applying ACLs on Bridged Packets



In prefer port mode, only the PACL is applied to the ingress packets (the input VACL is not applied).

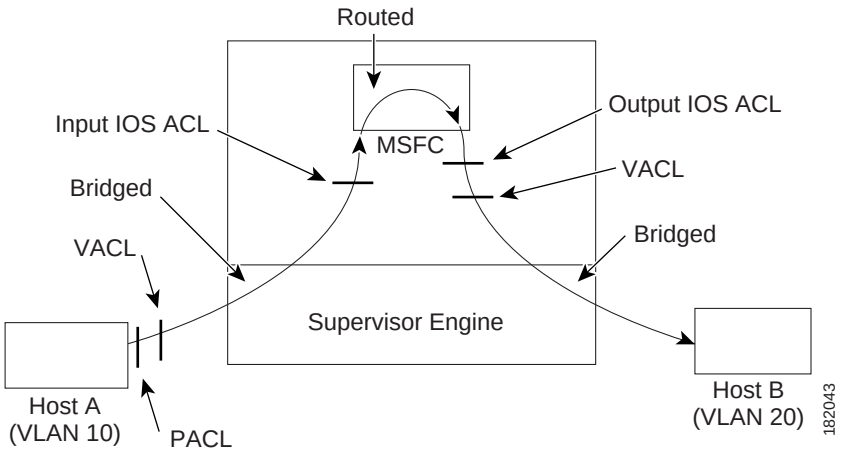
Routed Packets

Figure 75-2 shows how ACLs are applied on routed and Layer 3-switched packets. In merge mode, the ACLs are applied in the following order:

1. PACL for the ingress port
2. VACL for the ingress VLAN
3. Input Cisco IOS ACL
4. Output Cisco IOS ACL
5. VACL for the egress VLAN

In prefer port mode, only the PACL is applied to the ingress packets (the input VACL and Cisco IOS ACL are not applied).

Figure 75-2 Applying ACLs on Routed Packets



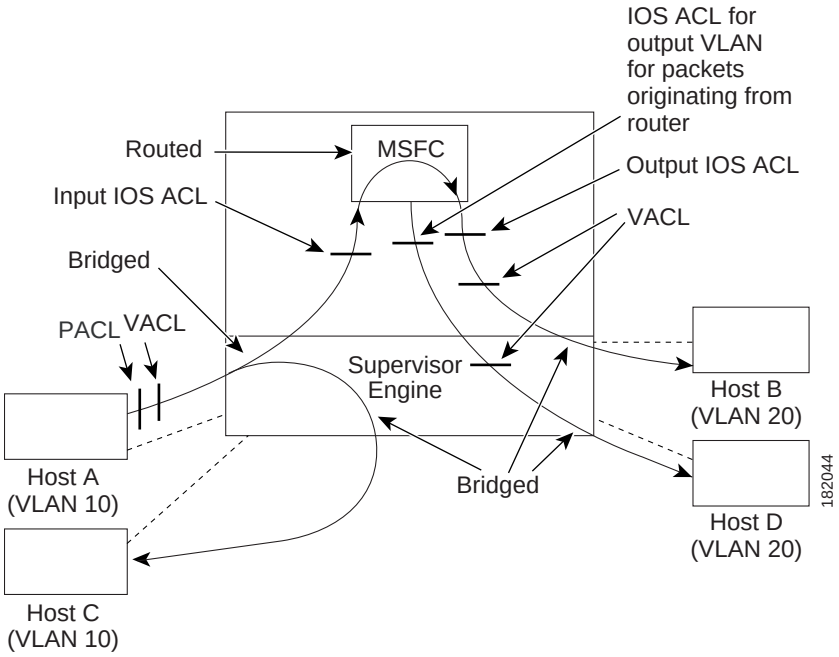
Multicast Packets

Figure 75-3 shows how ACLs are applied on packets that need multicast expansion. For packets that need multicast expansion, the ACLs are applied in the following order:

1. Packets that need multicast expansion:
   a. PACL for the ingress port
   b. VACL for the ingress VLAN
   c. Input Cisco IOS ACL
2. Packets after multicast expansion:
   a. Output Cisco IOS ACL
   b. VACL for the egress VLAN
3. Packets originating from router:
   a. Output Cisco IOS ACL
   b. VACL for the egress VLAN

In prefer port mode, only the PACL is applied to the ingress packets (the input VACL and Cisco IOS ACL are not applied).

Figure 75-3 Applying ACLs on Multicast Packets



Interface Template ACLs

This section describes the Interface Template ACL feature on Catalyst 6500 series Switches. An interface template provides a mechanism to configure multiple commands at the same time and associate it with a target such as an interface. An interface template is a container of configurations or policies that can be applied to specific ports.

Interface Templates provide an efficient way to apply ACLs along with other commands on interfaces. Template ACLs can be applied on an interface by first configuring an ACL inside an interface template, and then applying the template to any number of desired interfaces. A single template having an ACL can be applied to any number of physical or virtual interfaces either statically or dynamically.

The support for configuring Interface Template ACLs was introduced on Catalyst 6500 series Switches starting with the 15.5(1)SY2 release.

How to Configure PACLs

- Configuring IP and MAC ACLs on a Layer 2 Interface, page 75-8
- Configuring Access-group Mode on Layer 2 Interface, page 75-8
- Applying ACLs to a Layer 2 Interface, page 75-9

- Applying ACLs to a Port Channel, page 75-9
- Displaying an ACL Configuration on a Layer 2 Interface, page 75-9

Configuring IP and MAC ACLs on a Layer 2 Interface

IP and MAC ACLs can be applied to Layer 2 physical interfaces. Standard (numbered, named) and Extended (numbered, named) IP ACLs, and Extended Named MAC ACLs are supported.

To apply IP or MAC ACLs on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface` *interface*	Enters interface configuration mode for a Layer 2 port.
Step 3	`Switch(config-if)# {ip | mac} access-group {name | number | in | out}`	Applies a numbered or named ACL to the Layer 2 interface.
Step 4	`Switch(config)# show running-config`	Displays the access list configuration.

This example shows how to configure the Extended Named IP ACL simple-ip-acl to permit all TCP traffic and implicitly deny all other IP traffic:

```
Switch(config)# ip access-list extended simple-ip-acl
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# end
```

This example shows how to configure the Extended Named MAC ACL simple-mac-acl to permit source host 000.000.011 to any destination host:

```
Switch(config)# mac access-list extended simple-mac-acl
Switch(config-ext-macl)# permit host 000.000.011 any
Switch(config-ext-macl)# end
```

Configuring Access-group Mode on Layer 2 Interface

To configure the access mode on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface` *interface*	Enters interface configuration mode for a Layer 2 port.
Step 3	`Switch(config-if)# [no] access-group mode {prefer port | merge}`	Sets the mode for this Layer 2 interface. The **no** prefix sets the mode to the default value (which is merge).
Step 4	`Switch(config)# show running-config`	Displays the access list configuration.

This example shows how to configure an interface to use prefer port mode:

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# access-group mode prefer port
```

This example shows how to configure an interface to use merge mode:

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# access-group mode merge
```

Applying ACLs to a Layer 2 Interface

To apply IP and MAC ACLs to a Layer 2 interface, perform one of these tasks:

Command	Purpose
`Switch(config-if)# ip access-group ip-acl in`	Applies an IP ACL to the Layer 2 interface.
`Switch(config-if)# mac access-group mac-acl in`	Applies a MAC ACL to the Layer 2 interface.

This example applies the extended named IP ACL simple-ip-acl to interface GigabitEthernet 6/1 ingress traffic:

```
Switch# configure t
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# ip access-group simple-ip-acl in
```

This example applies the extended named MAC ACL simple-mac-acl to interface GigabitEthernet 6/1 ingress traffic:

```
Switch# configure t
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# mac access-group simple-mac-acl in
```

Applying ACLs to a Port Channel

To apply IP and MAC ACLs to a port channel logical interface, perform this task:

Command	Purpose
`Switch(config-if)# interface port-channel number`	Enters configuration mode for the port channel.
`Switch(config-if)# ip access-group ip-acl {in | out}`	Applies an IP ACL to the port channel interface.
`Switch(config-if)# mac access-group mac-acl {in | out}`	Applies a MAC ACL to the port channel interface.

This example applies the extended named IP ACL simple-ip-acl to port channel 3 ingress traffic:

```
Switch# configure t
Switch(config)# interface port-channel 3
Switch(config-if)# ip access-group simple-ip-acl in
```

Displaying an ACL Configuration on a Layer 2 Interface

To display information about an ACL configuration on Layer 2 interfaces, perform one of these tasks:

Command	Purpose
`Switch# show ip access-lists [interface interface-name]`	Shows the IP access group configuration on the interface.
`Switch# show mac access-group [interface interface-name]`	Shows the MAC access group configuration on the interface.
`Switch# show access-group mode [interface interface-name]`	Shows the access group mode configuration on the interface.

This example shows that the IP access group simple-ip-acl is configured on the inbound direction of interface fa6/1:

```
Switch# show ip interface gigabitethernet 6/1
GigabitEthernet6/1 is up, line protocol is up
  Inbound  access list is simple-ip-acl
  Outgoing access list is not set
```

This example shows that MAC access group simple-mac-acl is configured on the inbound direction of interface Gigabit Ethernet 6/1:

```
Switch# show mac access-group interface gigabitethernet 6/1
Interface GigabitEthernet6/1:
   Inbound access-list is simple-mac-acl
   Outbound access-list is not set
```

This example shows that access group merge is configured on interface Gigabit Ethernet 6/1:

```
Switch# show access-group mode interface gigabitethernet 6/1
Interface GigabitEthernet6/1:
   Access group mode is: merge
```

How to Configure Interface Template ACLs

- Configuring an IPv4 ACL by Statically Applying an Interface Template, page 75-10
- Configuring an IPv6 ACL by Statically Applying an Interface Template, page 75-12
- Configuration Examples for Template ACL Configuration, page 75-12

Configuring an IPv4 ACL by Statically Applying an Interface Template

To configure an IPv4 ACL on an interface by statically applying a template, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ip access-list [extended] <name | number>`	Defines an IPv4 access list by name or number.

	Command	Purpose
Step 3	`Switch(config-ext-nacl)# [`*sequence-number*`] permit` *protocol source source-wildcard destination destination-wildcard* `[option` *option-name*`] [precedence` *precedence*`] [tos` *tos*`] [ttl` *operator value*`] [log] [time-range` *time-range-name*`] [fragments]`	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one permit statement. • Use the no sequence-number form of this command to delete an entry.
Step 4	`Switch(config-ext-nacl)# [`*sequence-number*`] deny` *protocol source source-wildcard destination destination-wildcard* `[option` *option-name*`] [precedence` *precedence*`] [tos` *tos*`] [ttl` *operator value*`] [log] [time-range` *time-range-name*`] [fragments]`	(Optional) Specifies a deny statement in named IP access list mode. • Use the no sequence-number form of this command to delete an entry.
Step 5	**`Repeat Step 3 and Step 4 as necessary.`**	Allows you to revise the access list.
Step 6	**`Switch(config-ext-nacl)# exit`**	Exits access list configuration mode and returns to global configuration mode.
Step 7	**`Switch(config)# template`** *template-name*	Creates a user template and enters template configuration mode.
Step 8	**`Switch(config-template)# ip access-group`** {*access-list-name* | *access-list-number*} **`{in |out}`**	Associates the specified access list with the interface template.
Step 9	**`Switch(config-template)# exit`**	Exits template configuration mode and returns to global configuration mode.
Step 10	**`Switch(config)# interface`** *interface-name*	Enters the interface configuration mode for the specified interface.
Step 11	**`Switch(config-if)# source template`** *template-name*	Statically applies an interface template to a target interface.
Step 12	**`Switch(config-if)# end`**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an IPv6 ACL by Statically Applying an Interface Template

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ipv6 access-list` *name*	Defines an IPv6 access list by name and enters the IPv6 access list configuration mode.
Step 3	`Switch(config-ipv6-acl)# permit` *protocol* {*source-ipv6-prefix/ prefix-length* | `any` | `host` *source-ipv6-address*} [`operator` [*port-number*]] {*destination-ipv6-prefix/ prefix-length* | `any` | `host` *destination-ipv6-address*} [`operator` [*port-number*]] [dscp value] [fragments] [log] [log-input] [`sequence` *value*] [`time-range` *name*]	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one permit statement. • Use the no sequence-number form of this command to delete an entry.
Step 4	`Switch(config-ipv6-acl)# deny` *protocol* {*source-ipv6-prefix/ prefix-length* | `any` | `host` *source-ipv6-address*} [`operator` [*port-number*]] {*destination-ipv6-prefix/ prefix-length* | `any` | `host` *destination-ipv6-address*} [`operator` [*port-number*]] [dscp value] [fragments] [log] [log-input] [`sequence` *value*] [`time-range` *name*]	(Optional) Specifies a deny statement in named IP access list mode. • Use the no sequence-number form of this command to delete an entry.
Step 5	`Repeat Step 3 and Step 4 as necessary.`	Allows you to revise the access list.
Step 6	`Switch(config-ipv6-acl)# exit`	Exits access list configuration mode and returns to global configuration mode.
Step 7	`Switch(config)# template` *template-name*	Creates a user template and enters template configuration mode.
Step 8	`Switch(config-template)# ipv6 traffic-filter` *access-list-name* {`in` |`out`}	Associates the specified access list with the interface template.
Step 9	`Switch(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 10	`Switch(config)# interface` *interface-name*	Enters the interface configuration mode for the specified interface.
Step 11	`Switch(config-if)# source template` *template-name*	Statically applies an interface template to a target interface.
Step 12	`Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Template ACL Configuration

- Configuration of IPv4 and IPv6 ACLs by Statically Applying an Interface Template, page 75-13
- Verifying Template Details, page 75-13
- Viewing Access List details, page 75-13
- Verifying ACL Configuration on an Interface with a Statically Applied Template, page 75-14
- Verifying ACL Configuration on an Interface with a Dynamically Applied Template, page 75-15

Configuration of IPv4 and IPv6 ACLs by Statically Applying an Interface Template

The following example shows the configuration of IPv4 and IPv6 ACLs on an interface using an interface template:

```
Switch#configure terminal
Switch(config)#ip access-list extended IPV4_ACL
Switch(config-ext-nacl)#10 permit ip any host 10.1.1.10
Switch(config-ext-nacl)#20 deny ip any host 100.1.1.10
Switch(config-ext-nacl)#30 permit ip any host 12.1.1.1
Switch(config-ext-nacl)#40 deny ip any host 20.1.1.1
Switch(config-ext-nacl)#exit
Switch(config)#ipv6 access-list IPV6_ACL
Switch(config-ipv6-acl)#permit ipv6 any 2001:DB8:0101:0101::/32
Switch(config-ipv6-acl)#deny ipv6 any 2001:DB8:1001:0000::/32
Switch(config-ipv6-acl)#deny ipv6 any 2001:DB8:1001:0123::/32
Switch(config)#exit
Switch(config)#template ACL_test
Switch(config-template)#ip access-group IPV4_ACL in
Switch(config-template)#ipv6 traffic-filter IPV6_ACL out
Switch(config-template)#exit
Switch(config)#interface Te1/1/2
Switch(config-if)#source template ACL_test
Switch(config-if)# end
```

Verifying Template Details

The following example shows the sample output for **show template binding target** *interface-name* command for a statically applied Interface Template:

```
Switch# show template binding target Te1/1/2

Interface Templates
===================
Interface: Te1/1/2

Method              Source            Template-Name
------              ------            -------------
static              User              ACL_test
```

The following example shows the sample output for **show template binding target** *interface-name* command for a dynamically applied Interface Template:

```
Switch# show template binding target Gi160/3/0/13

Interface Templates
===================
Interface: Gi160/3/0/13

Method              Source            Template-Name
------              ------            -------------
dynamic             User              INTERFACE_TEMPLATE_ACL_IPV6
```

Viewing Access List details

The following examples shows the access list details for a given ACL:

```
Switch# show ip access-list IPV4_ACL
Extended IP access list IPV4_ACL
    10 permit ip any host 10.1.1.10
    20 deny ip any host 100.1.1.10
    30 permit ip any host 12.1.1.1
```

```
    40 deny ip any host 20.1.1.1

Switch# show ipv6 access-list IPV6_ACL
IPv6 access list IPV6_ACL
    permit ipv6 any 2001:DB8::/32 sequence 10
    deny ipv6 any 2001:DB8::/32 sequence 20
```

The following examples show the access list details configured inside a given Template:

```
Switch# show template interface source user ACL_test
Building configuration...

Template Name       : ACL_test
Template Definition :
 ip access-group test_ACL in
 ipv6 traffic-filter test_ACL2 out
!
end

Switch# show template interface source user INTERFACE_TEMPLATE_ACL_IPV6
Building configuration...

Template Name       : INTERFACE_TEMPLATE_ACL_IPV6
Template Definition :
 ipv6 traffic-filter Block_ipv6 in
 load-interval 30
 description ipv6 acl temp cca
!
end
```

The following example shows the ACL entries programmed in the hardware using the sample output for **show platform hardware acl entry interface** *interface-name* command

```
show platform hardware acl entry interface TenGigabitEthernet/1/1/2 securityin ip
mls_if_index:20004052 dir:0 feature:0 proto:0
pass#0 features
fno:0
tcam:B, bank:0, prot:0    Aces
Permit                ip host 1.1.1.3 host 4.4.4.4
Permit                ip host 1.1.1.2 host 3.3.3.3
Permit                ip host 1.1.1.1 host 2.2.2.2
L3_Deny               ip any any
```

Verifying ACL Configuration on an Interface with a Statically Applied Template

The following example shows the sample output for **show running-config interface** *interface-name* command:

```
Switch#show running-config int Te1/1/2
Building configuration...

Current configuration : 576 bytes
!
interface TenGigabitEthernet1/1/2
 switchport
 switchport mode access
 switchport trunk allowed vlan 20,30,40,50,60
 switchport access vlan 20
 switchport voice vlan 30
 switchport port-security
 ip access-group directACL in
 authentication periodic
 authentication timer reauthenticate server
```

```
 access-session host-mode multi-domain
 access-session closed
 access-session port-control auto
 dot1x pae authenticator
 source template ACL_test
 spanning-tree portfast edge
 spanning-tree bpdufilter enable
 service-policy type control subscriber POLICY_Gi160/3/0/13_dot1x
end
```

The following example shows the sample output for s**how derived-config interface** *interface-name* command for an interface with a statically applied Template ACL:

```
Switch# show derived-config int Te1/1/2
Building configuration...

Derived configuration : 550 bytes
!
interface TenGigabitEthernet1/1/2
 switchport
 switchport access vlan 20
 switchport trunk allowed vlan 20,30,40,50,60
 switchport mode access
 switchport voice vlan 30
 switchport port-security
 ip access-group directACL in
 authentication periodic
 authentication timer reauthenticate server
 access-session host-mode multi-domain
 access-session closed
 access-session port-control auto
 dot1x pae authenticator
 spanning-tree portfast edge
 spanning-tree bpdufilter enable
 service-policy type control subscriber POLICY_Gi160/3/0/13_dot1x
end
```

Verifying ACL Configuration on an Interface with a Dynamically Applied Template

The following example shows the sample output for **show running-config interface** *interface-name* command:

```
Switch# show running-config int Gi160/3/0/13
Building configuration...

Current configuration : 669 bytes
!
interface GigabitEthernet160/3/0/13
 switchport
 switchport trunk allowed vlan 20,30,40,50,60
 switchport mode access
 switchport access vlan 20
 switchport voice vlan 30
 switchport port-security
 access-group mode prefer port
 ipv6 traffic-filter Block_ipv6 in
 authentication periodic
 authentication timer reauthenticate server
 access-session closed
 access-session port-control auto
 mab
 dot1x pae authenticator
 spanning-tree portfast edge
 spanning-tree bpdufilter enable
```

```
 service-policy type control subscriber POLICY_Gi160/3/0/13_violation
 service-policy input AutoQoS-FEX-INGRESS-QoS-POLICY
 service-policy type lan-queuing output AutoQoS-FEX-EGRESS-QoS-POLICY
end
```

The following example shows the sample output for s**how derived-config interface** *interface-name* command for an interface with a dynamically applied Template ACL:

```
Switch# show derived-config int Gig160/3/0/13
Building configuration...

Derived configuration : 753 bytes
!
interface GigabitEthernet160/3/0/13
 description ipv6 acl temp cca
 switchport
 switchport access vlan 20
 switchport trunk allowed vlan 20,30,40,50,60
 switchport mode access
 switchport voice vlan 30
 switchport port-security
 access-group mode prefer port
 ipv6 traffic-filter Block_ipv6 in
 load-interval 30
 authentication periodic
 authentication timer reauthenticate server
 access-session closed
 access-session port-control auto
 mab
 dot1x pae authenticator
 spanning-tree portfast edge
 spanning-tree bpdufilter enable
 service-policy type control subscriber POLICY_Gi160/3/0/13_violation
 service-policy input AutoQoS-FEX-INGRESS-QoS-POLICY
 service-policy type lan-queuing output AutoQoS-FEX-EGRESS-QoS-POLICY
end
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum