Implementing of Layer 2 Access Lists

An Ethernet services access control list (ACL) consists of one or more access control entries (ACE) that collectively define the Layer 2 network traffic profile. This profile can then be referenced by Cisco IOS XR software features. Each Ethernet services ACL includes an action element (permit or deny) based on criteria such as source and destination address, Class of Service (CoS), or VLAN ID.

This module describes tasks required to implement Ethernet services access lists on your Cisco ASR 9000 Series Aggregation Services Router.

Note For a complete description of the Ethernet services access list commands listed in this module, refer to the *Ethernet Services (Layer 2) Access List Commands on Cisco ASR 9000 Series Routers* module in the *Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference* publication.

Feature History for Implementing Ethernet Services Access Lists on Cisco ASR 9000 Series Routers

Release	Modification
Release 3.7.2	This feature was introduced on Cisco ASR 9000 Series Routers.

- Prerequisites for Implementing Layer 2 Access Lists, on page 1
- Information About Implementing Layer 2 Access Lists, on page 2
- How to Implement Layer 2 Access Lists, on page 4
- Configuration Examples for Implementing Layer 2 Access Lists, on page 9

Prerequisites for Implementing Layer 2 Access Lists

This prerequisite applies to implement access lists and prefix lists:

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Implementing Layer 2 Access Lists

Ethernet Services Access Lists Feature Highlights

Ethernet services access lists have these feature highlights:

- The ability to clear counters for an access list using a specific sequence number.
- The ability to copy the contents of an existing access list to another access list.
- Allows users to apply sequence numbers to permit or deny statements and to resequence, add, or remove such statements from a named access list.
- Provides packet filtering on interfaces to forward packets.
- Ethernet services ACLs can be applied on interfaces, VLAN subinterfaces, bundle-Ethernet interfaces, EFPs, and EFPs over bundle-Ethernet interfaces. Atomic replacement of Ethernet services ACLs is supported on these physical interfaces.

Purpose of Ethernet Services Access Lists

Using ACL-based forwarding (ABF), Ethernet services access lists perform packet filtering to control which packets move through the network and where. Such controls help to limit incoming and outgoing network traffic and restrict the access of users and devices to the network at the port level.

How an Ethernet Services Access List Works

An Ethernet services access list is a sequential list consisting of permit and deny statements that apply to Layer 2 configurations. The access list has a name by which it is referenced.

An access list can be configured and named, but it is not in effect until the access list is referenced by a command that accepts an access list. Multiple commands can reference the same access list. An access list can control Layer 2 traffic arriving at the router or leaving the router, but not traffic originating at the router.

Ethernet Services Access List Process and Rules

Use this process and rules when configuring an Ethernet services access list:

- The software tests the source or destination address of each packet being filtered against the conditions in the access list, one condition (permit or deny statement) at a time.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the remaining statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet.

- If no conditions match, the software drops the packet because each access list ends with an unwritten or implicit deny statement. That is, if the packet has not been permitted or denied by the time it was tested against each statement, it is denied.
- The access list should contain at least one permit statement or else all packets are denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same permit or deny statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- Inbound access lists process packets arriving at the router. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, permit means continue to process the packet after receiving it on an inbound interface; deny means discard the packet.
- Outbound access lists process packets before they leave the router. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, permit means send it to the output buffer; deny means discard the packet.
- An access list can not be removed if that access list is being applied by an access group in use. To remove an access list, remove the access group that is referencing the access list and then remove the access list.
- An access list must exist before you can use the **ethernet-services access-group** command.

Helpful Hints for Creating Ethernet Services Access Lists

Consider these when creating an Ethernet services access list:

- Create the access list before applying it to an interface.
- Organize your access list so that more specific references appear before more general ones.

Source and Destination Addresses

Source MAC address and destination MAC address are two of the most typical fields on which to base an access list. Specify source MAC addresses to control packets from certain networking devices or hosts. Specify destination MAC addresses to control packets being sent to certain networking devices or hosts.

Ethernet Services Access List Entry Sequence Numbering

The ability to apply sequence numbers to Ethernet services access-list entries simplifies access list changes. The access list entry sequence numbering feature allows you to add sequence numbers to access-list entries and resequence them. When you add a new entry, you choose the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Sequence Numbering Behavior

These details the sequence numbering behavior:

- If entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483646. If the generated sequence number exceeds this maximum number, this message is displayed:

```
Exceeded maximum sequence number.
```

- If you provide an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- ACL entries can be added without affecting traffic flow and hardware performance.
- Distributed support is provided so that the sequence numbers of entries in the route-switch processor (RSP) and interface card are synchronized at all times.

How to Implement Layer 2 Access Lists

Restrictions for Implementing Layer 2 Access Lists

These restrictions apply for implementing Ethernet services access lists:

- Ethernet services access lists are not supported over management interfaces.
- NetIO (software slow path) is not supported for Ethernet services access lists.
- Match on inner VLAN 0 and outer VLAN 0 is not supported on Cisco ASR 9000 High Density 100GE Ethernet Line Card and ASR 9000 Enhanced Ethernet Line Card.

Configuring Ethernet Services Access Lists

This task configures an Ethernet services access list.

SUMMARY STEPS

1. **configure**
2. **ethernet-services access-list** *name*
3. [*sequence-number*] **{ permit | deny }** { *src-mac-address src-mac-mask* | **any** | **host** } [{ *ethertype-number* } | **vlan** *min-vlan-ID* [*max-vlan-ID*]] [**cos** *cos-value*] [**dei**] [**inner-vlan** *min-vlan-ID* [*max-vlan-ID*]] **inner-cos** *cos-value*] [**inner-dei**]
4. Repeat Step 3 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
5. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **configure**

Example:

```
Router# configure
```

Enters the Global Configuration mode.

Step 2 **ethernet-services access-list** *name*

Example:

```
Router(config)# ethernet-services access-list L2ACL2
```

Enters Ethernet services access list configuration mode and configures access list L2ACL2.

Step 3 [*sequence-number*] **{ permit | deny }** { *src-mac-address src-mac-mask* | **any** | **host** } [{ *ethertype-number* } | **vlan** *min-vlan-ID* [*max-vlan-ID*]] [**cos** *cos-value*] [**dei**] [**inner-vlan** *min-vlan-ID* [*max-vlan-ID*]] **inner-cos** *cos-value*] [**inner-dei**]

The following example shows to allow the specified source MAC address and to deny any other source MAC

Example:

```
Router(config-es-al)# 10 permit host 0001.0002.0001 any
```

or

```
Router(config-es-al)# 30 deny any any
```

Specifies one or more conditions allowed or denied, which determines whether the packet is passed or dropped.

Step 4 Repeat Step 3 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.

Allows you to revise an access list.

Step 5 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

What to Do Next

After creating an Ethernet services access list, you must apply it to an interface. See the Applying Ethernet Services Access Lists section for information about how to apply an access list.

Applying Ethernet Services Access Lists

After you create an access list, you must reference the access list to make it work. Access lists can be applied on either outbound or inbound interfaces. This section describes guidelines on how to accomplish this task for both terminal lines and network interfaces.

For inbound access lists, after receiving a packet, Cisco IOS XR software checks the source MAC address of the packet against the access list. If the access list permits the address, the software continues to process the packet. If the access list rejects the address, the software discards the packet.

For outbound access lists, after receiving and routing a packet to a controlled interface, the software checks the source MAC address of the packet against the access list. If the access list permits the address, the software sends the packet. If the access list rejects the address, the software discards the packet.

Note An empty access-list (containing no access control elements) cannot be applied on an interface.

Controlling Access to an Interface

This task applies an access list to an interface to restrict access to that interface. Access lists can be applied on either outbound or inbound interfaces.

SUMMARY STEPS

1. **configure**
2. **interface** *type instance*
3. **ethernet-services access-group** *access-list-name* { **ingress** | **egress** }
4. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

Step 1 **configure**

Example:

```
Router# configure
```

Enters the Global Configuration mode.

Step 2 **interface** *type instance*

Example:

```
Router(config)# interface gigabitethernet 0/2/0/2
```

Configures an interface and enters interface configuration mode.

- The *type* argument specifies an interface type. For more information on interface types, use the question mark (?) online help function.
- The *instance* argument specifies either a physical interface instance or a virtual instance.
  - The naming notation for a physical interface instance is *rack/slot/module/port*. The slash (/) between values is required as part of the notation.

- The number range for a virtual interface instance varies depending on the interface type.

Step 3 **ethernet-services access-group** *access-list-name* { **ingress** | **egress** }

Example:

```
Router(config-if)# ethernet-services access-group p-in-filter ingress
Router(config-if)# ethernet-services access-group p-out-filter egress
```

Controls access to an interface.

- Use the *access-list-name* argument to specify a particular Ethernet services access list.
- Use the *ingress* keyword to filter on inbound packets or the *egress* keyword to filter on outbound packets.

This example applies filters on packets inbound and outbound from GigabitEthernet interface 0/2/0/2.

Step 4 Use the **commit** or **end** command.

commit - Saves the configuration changes and remains within the configuration session.

end - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

Copying Ethernet Services Access Lists

This task copies an Ethernet services access list.

SUMMARY STEPS

1. **copy access-list ethernet-service** *source-acl destination-acl*
2. **show access-lists ethernet-services** [*access-list-name* | **maximum** | **standby** | **summary**]

DETAILED STEPS

Procedure

Step 1 **copy access-list ethernet-service** *source-acl destination-acl*

Example:

```
RP/0/RSP0/CPU0:router# copy access-list ethernet-service list-1 list-2
```

Creates a copy of an existing Ethernet services access list.

- Use the *source-acl* argument to specify the name of the access list to be copied.
- Use the *destination-acl* argument to specify where to copy the contents of the source access list.

- The *destination-acl* argument must be a unique name; if the destination-acl argument name exists for an access list, the access list is not copied.

Step 2 **show access-lists ethernet-services** [*access-list-name* | **maximum** | **standby** | **summary**]

Example:

```
RP/0/RSP0/CPU0:router# show access-lists ethernet-services list-2
```

(Optional) Displays the contents of a named Ethernet services access list. For example, you can verify the output to see that the destination access list list-2 contains all the information from the source access list list-1.

Resequencing Access-List Entries

This task shows how to reassign sequence numbers to entries in a named access list. Resequencing an access list is optional.

SUMMARY STEPS

1. **resequence access-list ethernet-services** *access-list-name* [*starting-sequence-number* [*increment*]]
2. **show access-lists ethernet-services** [*access-list-name* | **maximum** | **standby** | **summary**]

DETAILED STEPS

Procedure

Step 1 **resequence access-list ethernet-services** *access-list-name* [*starting-sequence-number* [*increment*]]

Example:

```
Router# resequence access-list ethernet-services L2ACL2 20 10
```

(Optional) Resequences the specified Ethernet services access list using the desired starting sequence number and the increment of sequence numbers.

- This example resequences an Ethernet services access list named L2ACL2. The starting sequence number is 20 and the increment is 10. If you do not select an increment, the default increment 10 is used.

Note
If during the resequencing process it is determined that the ending number will exceed the maximum sequence number allowed, the configuration will not take effect and will be rejected. The sequence numbers will not be changed.

Step 2 **show access-lists ethernet-services** [*access-list-name* | **maximum** | **standby** | **summary**]

Example:

```
Router# show access-lists ethernet-services L2ACL2
```

(Optional) Displays the contents of a named Ethernet services access list.

- Review the output to see that the access list includes the updated information.

Configuration Examples for Implementing Layer 2 Access Lists

Resequencing Entries in an Access List: Example

This example shows access-list resequencing. The starting value in the resequenced access list is 1, and the increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483646.

When an entry with no sequence number is entered, by default, it has a sequence number of 10 more than the last entry in the access list.

```
ethernet-services access-list test_acl3
 10 permit host 0001.0002.0001 any
 13 permit host 0001.0002.0002 any
 16 permit host 0001.0002.0003 any
 19 deny any any

resequence access-list ethernet-services test_acl3 10 3

show access-lists ethernet-services test_acl3

ethernet service access-list acl_1
10 permit 1.2.3 4.5.6
20 deny 2.3.4 5.4.3
30 permit 3.1.2 5.3.4 cos 5

resequence access-list ethernet service acl_1 10 20

show access-list ethernet-service acl1_1

ipv4 access-list acl_1
  10 permit 1.2.3 4.5.6
  30 deny 2.3.4 5.4.3
  50 permit 3.1.2 5.3.4 cos 5
```

Adding Entries with Sequence Numbers: Example

In this example, a new entry is added to Ethernet services access list test_acl3.

```
 10 permit host 0001.0002.0001 any
 13 permit host 0001.0002.0002 any cos 3
 16 permit host 0001.0002.0003 any vlan 10
 19 deny any any
!

configure
ethernet-services access-list test_acl3
15 permit host 0001.0002.0004 any

ethernet-services access-list test_acl3
 10 permit host 0001.0002.0001 any
 13 permit host 0001.0002.0002 any cos 3
 15 permit host 0001.0002.0004 any
 16 permit host 0001.0002.0003 any vlan 10
```

```
 19 deny any any
!
```