Layer 2 Access Control Lists on EVCs

The ability to filter packets in a modular and scalable way is important for both network security and network management. Access Control Lists (ACLs) provide the capability to filter packets at a fine granularity. In Metro Ethernet networks, ACLs are directly applied on Ethernet virtual circuits (EVCs).

Layer 2 Access Control Lists on EVCs is a security feature that allows packet filtering based on MAC addresses. This module describes how to implement ACLs on EVCs.

Prerequisites for Layer 2 Access Control Lists on EVCs

- Knowledge of how service instances must be configured.
- Knowledge of extended MAC ACLs and how they must be configured.

Restrictions for Layer 2 Access Control Lists on EVCs

- You can enable a packet capture on the host, based on Layer 2 packet header per EFP (for example, **dst-mac**, **src-mac** and CoS field). Create a **pcap** of the captured packet on host machine.
- A maximum of 512 access control entries (ACEs) are allowed for a given ACL, with the limitation that it does not exceed the maximum tcam entries.
- L2 ACL is supported over port channel with Normal EFPs.
- Egress L2 ACL on EVC is *not* supported.
- L2 ACLs are *not* supported on Trunk EFP.
- L2 ACL counters are *not* supported.
- Layer2 ACL can be applied on layer 2 frame without IPv4 or IPv6 header as layer 2 ACL does not support filter on IPv4 or IPv6 traffic.

- Layer 2 ACLs function inbound only. The Layer 2 ACLs are *not* supported at physical interface level.

Information About Layer 2 Access Control Lists on EVCs

EVCs

An Ethernet virtual circuit (EVC) as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. An EVC contains the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a specified port.

Service instances are configured under a port channel. The traffic carried by the service instance is load balanced across member links. Service instances under a port channel are grouped and each group is associated with one member link. Ingress traffic for a single EVC can arrive on any member of the bundle. All egress traffic for a service instance uses only one of the member links. Load balancing is achieved by grouping service instances and assigning them to a member link.

Ethernet virtual connection services (EVCS) uses the EVCs and service instances to provide Layer 2 switched Ethernet services. EVC status can be used by a customer edge (CE) device either to find an alternative path to the service provider network or in some cases, to fall back to a backup path over Ethernet or over another alternative service such as ATM.

For information about the Metro Ethernet Forum standards, see the Standards table in the “Additional References” section.

Relationship Between ACLs and Ethernet Infrastructure

The following points capture the relationship between ACLs and Ethernet Infrastructure (EI):

- ACLs can be directly applied on an EVC using the command-line interface (CLI). An ACL is applied to a service instance, which is the instantiation of an EVC on a given port.
- One ACL can be applied to more than one service instance at any time.
- One service instance can have one ACL at most applied to it at any time. If a Layer 2 ACL is applied to a service instance that already has a Layer 2 ACL, the new one replaces the old one.
- Only named ACLs can be applied to service instances. The command syntax ACLs is retained; the **mac access-list extended** command is used to create an ACL.
- The **show ethernet service instance id** *id* **interface** *type* *number* detail command can be used to provide details about ACLs on service instances.

Information About Layer 2 Access Control Lists on EVCs

Creating a Layer 2 ACL

Perform this task to create a Layer 2 ACL with a single ACE.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mac access-list extended** *name*

Example:

```
Device(config)# mac access-list extended test-12-acl
```

Defines an extended MAC ACL and enters mac access list control configuration mode.

Step 4 **permit** {{*src-mac mask* | **any**} {*dest-mac mask* | **any**} [*protocol* [**vlan** *vlan*] [*cos value*]]}

Example:

```
Device(config-ext-macl)# permit 00aa.00bb.00cc 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. Creates an ACE for the ACL.

Applying a Layer 2 ACL to a Service Instance

Perform this task to apply a Layer 2 ACL to a service instance. Note that packet filtering takes place only after the ACL has been created and applied to the service instance.

Before you begin

Before applying an ACL to a service instance, you must create it using the **mac access-list extended command. See the "Creating a Layer 2 ACL" section.**

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 1/0/0
```

Specifies the type and location of the interface to configure, where:

- *type* --Specifies the type of the interface.
- *number* --Specifies the location of the interface.

Step 4 **service instance** *id* ethernet

Example:

```
Device(config-if)# service instance 100 ethernet
```

Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

Step 5 **encapsulation dot1q** *vlan-id*

Example:

```
Device(config-if-srv)# encapsulation dot1q 100
```

Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.

Step 6 **mac access-group** *access-list-name* in

Example:

```
Device(config-if-srv)# mac access-group test-12-acl in
```

Applies a MAC ACL to control incoming traffic on the interface.

Step 7 **bridge -domain** *bridge-id* in

Example:

```
Device(config-if-srv)# bridge-domain 100
```

Configure the bridge domain ID.

Configuring a Layer 2 ACL with ACEs on a Service Instance

Perform this task to configure the same ACL with three ACEs and stop all other traffic on a service instance.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mac access-list extended** *name*

Example:

```
Device(config)# mac access list extended test-12-acl
```

Defines an extended MAC ACL and enters mac access control list configuration mode.

Step 4 **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddea 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 5 **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddeb 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 6 **permit** {*src-mac mask* | **any**} {*dest-mac mask*} | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddec 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 7 **deny any any**

Example:

```
Device(config-ext-macl)# deny any any
```

Prevents forwarding of Layer 2 traffic except for the allowed ACEs.

Step 8 **exit**

Example:

```
Device(config-ext-macl)# exit
```

Exits the current command mode and returns to global configuration mode.

Step 9 **interface** *type* *number*

Example:

```
Device(config)# interface gigabitethernet 1/0/0
```

Specifies the interface.

Step 10 **service instance** *id* **ethernet**

Example:

```
Device(config-if)# service instance 200 ethernet
```

Configures an Ethernet service instance on an interface and enters service instance configuration mode.

Step 11 **encapsulation dot1q** *vlan-id*

Example:

```
Device(config-if-srv)# encapsulation dot1q 100
```

Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.

Step 12 **mac access-group** *access-list-name* **in**

Example:

```
Device(config-if-srv)# mac access-group test-12-acl in
```

Applies a MAC ACL to control incoming traffic on the interface.

Verifying the Presence of a Layer 2 ACL on a Service Instance

Perform this task to verify that a Layer 2 ACL is present on an EVC. This verification task can be used after an ACL has been configured to confirm its presence.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ethernet service instance id** *id* **interface** *type* *number* detail

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail
```

Displays detailed information about Ethernet customer service instances.

Configuring a Layer 2 Packet Header Per Ethernet Flow Point (EFP)

Procedure

Step 1 **Example:**

```
Device# show run inte gi0/1/0
Building configuration...

Current configuration : 239 bytes
interface GigabitEthernet0/1/0
no ip address
negotiation auto
service instance 1 ethernet
  encapsulation dot1q 1
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1
!
service instance 2 ethernet
  encapsulation dot1q 2
```

Get the running configuration of the interface to enable capture.

Step 2 **Example:**

```
Device# show platform hardware pp active interface all | begin 0/1/0
Name: GigabitEthernet0/1/0, Asic: 0, hwidx: 26
lpn: 0, ppn: 26, gid: 26, mac: 00af.1f5a.bc09
InLportId: 0, ELportId: 0, dpidx: 15, l3ID: 13 /* Put this dpidx to the below command */
port_flags: 0, port_speed: 1000 Mbps, efp_count: 2, destIndex: 26, intType: 1
etherchnl: 0, efp: 0, bdi: 0, l2PhyIf: 1, l3PhyIf: 0, l3TDM: 0, loopBack: 0
tunnel: 0, tunneltp: 0, icmp_flags: 0, icmp6_flags: 0
bandwidth: 1000000, fcid: 0, cid: 0, mpls_tbid: 65535, protocols: 0
```

```
v4_netsmask: 0, v4_tableid: 0, v6_tableid: 65535, vrf_tbid_dstrm: , snmp_index: 0
bd_id: 0, encap: 1, ip_mtu: 1500, l2_max_tu: 1500, l2_min_tu: 0
vrfid: 0, enctype: 0, admin_state: 1, admin_state_oir: 0, flcd_state: 1
```

Get the dpidx of the interface.

Step 3 **Example:**

```
Device# debug platform hardware pp active infrastructure pi nft captur interface 15
service-instance 1 dst-mac any src-mac any COS 1
```

Enable packet capture at the EFP level.

Step 4 **Example:**

```
Device# show platform hardware pp active infrastructure pi nft statistics | include MIRROR
                          32          MIRROR Q                        0                0
```

Verify if EFP capture is working (counter gets incremented).

Step 5 **Example:**

```
Device# no debug platform hardware pp active infrastructure pi nft captur interface 15
service-instance 1 dst-mac any src-mac any COS 1
```

Disable packet capture.

Step 6 Get the capture file. The capture file can be located at the path */tmp/fp/trace/* by name *cisco-capture-all_q-<timestamp>.pcap*.

Configuration Examples for Layer 2 Access Control Lists on EVCs

Example Applying a Layer 2 ACL to a Service Instance

The following example shows how to apply a Layer 2 ACL called mac-20-acl to a service instance. The ACL has five permitted ACEs and all other traffic is not allowed.

```
enable
configure terminal
 mac access-list extended mac-20-acl


 permit 00aa.bbcc.adec 0.0.0 any


 permit 00aa.bbcc.bdec 0.0.0 any


 permit 00aa.bbcc.cdec 0.0.0 any


 permit 00aa.bbcc.edec 0.0.0 any
 permit 00aa.bbcc.fdec 0.0.0 any

 deny any any
 exit
interface gigabitethernet 10/0/0
 service instance 100 ethernet
 encapsulation dot1q 100
 mac access-group mac-20-acl in
```

Example Applying a Layer 2 ACL to Three Service Instances on the Same Interface

The following example shows how to apply a Layer 2 ACL called mac-07-acl to three service instances on the same interface:

```
enable
configure terminal
mac access-list extended mac-07-acl

permit 00aa.bbcc.adec 0.0.0 any

permit 00aa.bbcc.bdec 0.0.0 any

permit 00aa.bbcc.cdec 0.0.0 any

deny any any
exit
interface gigabitethernet 10/0/0
service instance 100 ethernet
encapsulation dot1q 100
mac access-group mac-07-acl in
service instance 101 ethernet
encapsulation dot1q 101
mac access-group mac-07-acl in
service instance 102 ethernet
encapsulation dot1q 102
mac access-group mac-07-acl in
```

Verifying the Presence of a Layer 2 ACL on a Service Instance

Perform this task to verify that a Layer 2 ACL is present on an EVC. This verification task can be used after an ACL has been configured to confirm its presence.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ethernet service instance id** *id* **interface** *type* *number* detail

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail
```

Displays detailed information about Ethernet customer service instances.

Example Displaying the Details of a Layer 2 ACL on a Service Instance

The following sample output displays the details of a Layer 2 ACL called test-acl on a service instance.

```
Device# show ethernet service instance id 100 interface gig3/0/1 detail
Service Instance ID: 100
L2 ACL (inbound): test-acl
Associated Interface: Gig3/0/1
Associated EVC: test
L2protocol drop
CEVlans:
Interface Dot1q Tunnel Ethertype: 0x8100
State: Up
L2 ACL permit count: 10255
L2 ACL deny count: 53
```

The table below describes the significant fields in the output.

Table 1: show ethernet service instance Field Descriptions

Field	Description
Service Instance ID	Displays the service instance ID.
L2 ACL (inbound):	Displays the ACL name.
Associated Interface:	Displays the interface details of the service instance.
Associated EVC:	Displays the EVC with which the service instance is associated.
CEVlans:	Displays details of the associated VLAN ID.
State:	Displays whether the service instance is in an up or down state.
L2 ACL permit count:	Displays the number of packet frames allowed to pass on the service instance by the ACL.
L2 ACL deny count	Displays the number of packet frames not permitted to pass on the service instance by the ACL.

Example Displaying the Details of Configured Layer 2 ACL

The following sample output displays the details of a configured Layer 2 ACL.

```
Device# show access-lists
Extended IP access list ip-acl
10 permit ip any any
Extended MAC access list mac-acl
permit any any vlan 10
Device#
Device#sh access-lists mac-acl
Extended MAC access list mac-acl
permit any any vlan 10
```