Configuring MAC Address Security on Service Instances and EVC Port Channels

The MAC Address Security on Service Instances and EVC Port Channels feature addresses port security with service instances by providing the capability to control and filter MAC address learning behavior at the granularity of a per-service instance. When a violation requires a shutdown, only the customer who is assigned to a given service instance is affected and--not all customers who are using the port. MAC address limiting is a type of MAC security and is also referred to as a MAC security component or element.

Prerequisites for MAC Address Security on Service Instances and EVC Port Channels

- An understanding of service instances and bridge domains.
- An understanding of the concepts of MAC address limiting and how it is used for MAC security.
- An understanding of how port channels and EtherChannels work in a network.

Restrictions for MAC Security on the RSP3 Module

- Maximum MAC scale supported on the RSP3 module is 200000 entries.
- Maximum number of secure EFPs supported is 255.
- Maximum number of secure MAC entries is 16K.
- Maximum number of MAC addresses supported per BD is 64K.

- Maximum number of secure MAC entries per BD is 10K.
- MAC security is supported only on Layer2 EFPs.
- MAC security on local connect and cross-conect is *not* supported.
- MAC security is *not* supported on trunk EFP.

Information About MAC Address Security on Service Instances and EVC Port Channels

Ethernet Virtual Circuits, Service Instances, and Bridge Domains

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 embodies the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a given port.

Support for Ethernet bridging is an important Layer 2 service that is offered on a router as part of an EVC. Ethernet bridging enables the association of a bridge domain with a service instance.

Service instances are configured under a port channel. The traffic carried by service instances 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 service instance 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.

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

EVCs on Port Channels

An EtherChannel bundles individual Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links. The Ethernet Virtual Connection Services (EVCS) EtherChannel feature provides support for EtherChannels on service instances.

Note The MAC Address Security on EVC Port Channel services is supported only on bridge domains over Ethernet and is not supported on xconnect services.

EVCS uses the concepts of EVCs and service instances.

Load balancing is done on an Ethernet flow point (EFP) basis where a number of EFPs exclusively pass traffic through member links.

MAC Security and MAC Addressing

MAC security is enabled on a service instance by configuring the **mac security** command. Various MAC security elements can be configured or removed regardless of whether the **mac security** command is presently configured, but these configurations become operational only when the **mac security** command is applied.

In this document, the term “secured service instance” is used to describe a service instance on which MAC security is configured. The MAC addresses on a service instance on which MAC security is configured are referred to as “secured MAC addresses.” Secured MAC addresses can be either statically configured (as a permit list) or dynamically learned.

MAC Address Permit List

A permit list is a set of MAC addresses that are permitted on a service instance. Permitted addresses permanently configured into the MAC address table of the service instance.

On a service instance that is a member of a bridge domain, the operator is permitted to configure one or more permitted MAC addresses.

For each permitted address, eligibility tests are performed and after the address passes these tests, it is either:

- Programmed into the MAC address table of the bridge domain, if MAC security is enabled on the service instance or,
- Stored in an area of memory referred to as “MAC table cache” if MAC security is not enabled on the service instance. When MAC security is enabled, the addresses from the MAC table cache are added to the MAC address table as secure addresses.

The eligibility tests performed when a user tries to add a MAC address to the permit list on a service instance are as follows:

- If the address is already a denied address on the service instance, the configuration is rejected with an appropriate error message.
- If the acceptance of this address would increase the secure address count on the service instance beyond the maximum number allowed, an attempt is made to make room by removing an existing address from the MAC address table. The only candidate for removal is a dynamically learned address on the service instance. If sufficient room cannot be made, the configuration is rejected. If the acceptance of this address would increase the secure address count on the bridge domain beyond the maximum number allowed, an attempt is made to make room by removing an existing address from the MAC address table. The only candidate for removal is a dynamically learned address on the service instance. If room cannot be made, the configuration is rejected.

Note Default maximum address is '1' for a service instance.

- If the address is already permitted on another service instance in the same bridge domain, one of the following actions occur:
    - If the conflicting service instance has MAC security configured, the configuration is rejected with an appropriate error message.

- If the conflicting service instance does not have MAC security configured, the configuration is accepted silently. (If the operator attempts to enable MAC security on the conflicting service instance, that attempt fails.)

MAC Address Deny List

A deny list is a set of MAC addresses that are not permitted on a service instance. An attempt to learn a denied MAC address will fail. On a service instance that is a member of a bridge domain, the operator is permitted to configure one or more denied MAC addresses. The arrival of a frame with a source MAC address that is part of a deny list will trigger a violation response.

Before a denied address can be configured, the following test is performed:

- If the address is already configured as a permitted address on the specific service instance or if the address has been learned and saved as a sticky address on the service instance, the configuration is rejected with an appropriate error message.

In all other cases, the configuration of the denied address is accepted. Typical cases include:

- The address is configured as a permitted address on another service instance in the same bridge domain, or the address has been learned and saved as a sticky address on another service instance.
- The address is present in the MAC table of the bridge domain as a dynamically learned address on the specific service instance and is deleted from the MAC table before the configuration is accepted.

Violation Response Configuration

A violation response is a response to a MAC security violation or a failed attempt to dynamically learn a MAC address due to an address violation. MAC security violations are of two types:

Type 1 Violation --The address of the ingress frame cannot be dynamically learned due to a deny list, or because doing so would cause the maximum number of secure addresses to be exceeded (see the *MAC Address Limiting and Learning*).

Type 2 Violation --The address of the ingress frame cannot be dynamically learned because it is already "present" on another secured service instance (see the *MAC Move and MAC Locking*) in the same bridge-domain.

There are three possible sets of actions that can be taken in response to a violation:

1. **Shutdown**
    - The ingress frame is dropped.
    - The service instance on which the offending frame arrived is shut down.
    - The violation count is incremented, and the violating address is recorded for later CLI display.
    - The event and the response are logged to SYSLOG.

2. **Restrict**
    - The ingress frame is dropped.
    - The violation count is incremented, and the violating address is recorded for display.

- The event and the response are logged to SYSLOG.

3. **Protect**
    - The ingress frame is dropped.

Note The ingress frame is dropped silently, without sending any violation report to the SYSLOG.

Note The Restrict and Protect modes are applied on EFP level to discard the traffic. Both the modes are not applied on the Erroneous MAC level.

If a violation response is not configured, the default response mode is shutdown. The violation response can be configured to protect or restrict mode. A “no” form of a violation response, sets the violation response to the default mode of shutdown.

You are allowed to configure the desired response for a Type 1 and Type 2 violations on a service instance. For a Type 1 violation on a bridge domain (that is, if the learn attempt conforms to the policy configured on the service instance, but violates the policy configured on the bridge domain), the response is always “Protect.” This is not configurable.

In shutdown mode, the service instance is put into the error disabled state immediate, an SNMP trap notification is transmitted, and a message is sent to the console and SYSLOG as shown below:

```
%ETHER_SERVICE-6-ERR_DISABLED:
Mac security violation - shutdown service instance 100 on interface gig 0/0/0
```

To bring a service instance out of the error-disabled state, use **errdisable recovery cause mac-security** command to set the auto recovery timer, or re-enable it using the EXEC command **clear ethernet service instance id** *id* **interface** *type number* **errdisable**.

In Restrict mode, the violation report is sent to SYSLOG at level LOG_WARNING.

Support for the different types of violation responses depends on the capabilities of the platform. The desired violation response can be configured on the service instance. The configured violation response does not take effect unless and until MAC security is enabled using the **mac security** command.

MAC Address Aging Configuration

A specific time scheduler can be set to age out secured MAC addresses that are dynamically learned or statically configured on both service instances and bridge domains, thus freeing up unused addresses from the MAC address table for other active subscribers.

The set of rules applied to age out secured MAC addresses is called secure aging. By default, the entries in the MAC address table of a secured service instance are never aged out. This includes permitted addresses and dynamically learned addresses.

The **mac security aging time** *aging-time* command sets the aging time of the addresses in the MAC address table to <*n* > minutes. By default, this affects only dynamically learned (not including sticky) addresses--permitted addresses and sticky addresses are not affected by the application of this command.

By default, the aging time <n> configured via the **mac security aging time** *aging-time* command is an absolute time. That is, the age of the MAC address is measured from the instant that it was first encountered on the service instance. This interpretation can be modified by using the **mac security aging time** *aging-time* **inactivity** command, which specifies that the age <n> be measured from the instant that the MAC address was last encountered on the service instance.

The **mac security aging static**and **mac security aging sticky** commands specify that the **mac security aging time**aging-time command must be applicable to permitted and sticky MAC addresses, respectively. In the case of permitted MAC addresses, the absolute aging time is measured from the time the address is entered into the MAC address table (for example, when it is configured or whenever the **mac security** command is entered--whichever is later).

If the **mac security aging time** command is not configured, the **mac security aging static** command has no effect.

Sticky MAC Address Configurations

The ability to make dynamically learned MAC addresses on secured service instances permanent even after interface transitions or device reloads can be set up and configured. A dynamically learned MAC address that is made permanent on a secured service instance is called a “sticky MAC address”. The **mac security sticky** command is used to enable the sticky MAC addressing feature on a service instance.

With the “sticky” feature enabled on a secured service instance, MAC addresses learned dynamically on the service instance are kept persistent across service instance line transitions and device reloads.

The sticky feature has no effect on statically configured MAC addresses. The sticky addresses are saved in the running configuration. Before the device is reloaded, it is the responsibility of the user to save the running configuration to the startup configuration. Doing this will ensure that when the device comes on, all the MAC addresses learned dynamically previously are immediately populated into the MAC address table.

The **mac security sticky address** *mac-address* command can configure a specific MAC address as a sticky MAC address. The use of this command is not recommended for the user because configuring a MAC address as a static address does the same thing. When sticky MAC addressing is enabled by the **mac security sticky** command, the dynamically learned addresses are marked as sticky and a **mac security sticky address** *mac-address* command is automatically generated and saved in the running configuration for each learned MAC address on the service instances.

Aging for Sticky Addresses

MAC addresses learned on a service instance that has the sticky behavior enabled are subject to aging as configured by the **mac security aging time** and **mac security aging sticky** commands. In other words, for the purpose of aging functionality, sticky addresses are treated the same as dynamically learned addresses.

Transitions

This section contains a description of the expected behavior of the different MAC security elements when various triggers are applied; for example, configuration changes or link state transitions.

MAC Security Enabled on a Service Instance

When MAC security is enabled on a service instance, all existing MAC table entries for the service instance are purged. Then, permitted MAC address entries and sticky addresses are added to the MAC table, subject to the prevailing MAC address limiting constraints on the bridge domain.

If MAC address limits are exceeded, any MAC address that fails to get added is reported via an error message to the console, the attempt to enable MAC security on the service instance fails, and the already added permitted entries are backed out or removed.

The aging timer for all entries is updated according to the secure aging rules.

MAC Security Disabled on a Service Instance

The existing MAC address table entries for this service instance are purged.

Service Instance Moved to a New Bridge Domain

This transition sequence applies to all service instances, whether or not they have MAC security configured. All the MAC addresses on this service instance in the MAC address table of the old bridge domain are removed. The count of dynamically learned addresses in the old bridge domain is decremented. Then, all the MAC security commands are permanently erased from the service instance.

Service Instance Removed from a Bridge Domain

All the MAC addresses in the MAC address table that attributable to this service instance are removed, and the count of dynamically learned addresses in the bridge domain is decremented. Since MAC security is applicable only on service instances that are members of a bridge domain, removing a service instance from a bridge domain causes all the MAC security commands to be erased permanently.

Service Instance Shut Down Due to Violation

All dynamically learned MAC addresses in the MAC address table are removed, and all the other MAC security state values are left unchanged. The only change is that no traffic is forwarded, and therefore no learning can take place.

Interface Service Instance Down Linecard OIR Removed

The MAC tables of all the affected bridge domains are cleared of all the entries attributable to the service instances that are down.

Interface Service Instance Re-activated Linecard OIR Inserted

The static and sticky address entries in the MAC tables of the affected bridge domains are re-created to the service instances that are activated.

MAC Address Limit Decreased

When the value of the MAC address limit on the service instance is changed initially, a sanity check is performed to ensure that the new value of <n> is greater than or equal to the number of permitted entries. If not, the command is rejected. The MAC table is scanned for addresses that are attributable to this service instance, and dynamically learned MAC addresses are removed when the new MAC address limit is less than the old MAC address limit.

When the value of <n> on a bridge domain is changed initially, a sanity check is performed to ensure that the new value of <n> is greater than or equal to the sum of the number of permitted entries on all the secured service instances on the bridge domain. If this sanity test fails, the command is rejected. The bridge domain MAC address table (regardless of service instance) is scanned for dynamically learned (or sticky) addresses. All dynamically learned addresses are removed when the new MAC address limit is less than the old MAC address limit.

Sticky Addresses Added or Removed on a Service Instance

Existing dynamically learned MAC addresses remain unchanged. All new addresses learned become “sticky” addresses.

Disabling sticky addresses causes all sticky secure MAC addresses on the service instance to be removed from the MAC address table. All new addresses learned become dynamic addresses on the service instance and are subject to aging.

How to Configure MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

Enabling MAC Security on a Service Instance

Perform this task to enable MAC address security 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 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance on an interface and enters service instance 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 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 8 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Enabling MAC Security on an EVC Port Channel

Before you begin

Note

- Bridge-domain, xconnect, and Ethernet virtual circuits (EVCs) are allowed only over the port channel interface and the main interface.
- If you configure a physical port as part of a channel group, you cannot configure EVCs under that physical port.

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 port-channel** *channel-group*

Example:

```
Device(config)# interface port-channel 2
```

Specifies the port channel group number and enters interface configuration mode.

- Acceptable values are integers from 1 to 64.

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

Example:

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

Creates a service instance on an interface and enters service instance 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 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 8 end

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring a MAC Address Permit List

Perform this task to configure permitted MAC addresses on a service instance that is a member of a bridge domain.

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 gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.

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

Example:

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

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

Step 6 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security address permit** *mac-address*

Example:

```
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
```

Adds the specified MAC address as a permit MAC address for the service instance.

Step 8 **mac security address permit** *mac-address*

Example:

```
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaab
```

Adds the specified MAC address as a permitted MAC address for the service instance.

Step 9 **mac security address permit** *mac-address*

Example:

```
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaac
```

Adds the specified MAC address as a permitted MAC address for the service instance.

Step 10 **mac security address permit** *mac-address*

Example:

```
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaad
```

Adds the specified MAC address as a permitted MAC address for the service instance.

Step 11 **mac security address permit** *mac-address*

Example:

```
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaae
```

Adds the specified MAC address as a permitted MAC address for the service instance.

Step 12 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 13 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring a MAC Address Deny List

Perform this task to configure a list of MAC addresses that are not allowed on a service instance that is a member of a bridge domain.

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 gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance 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 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security address** **deny** *mac-address*

Example:

```
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaaa
```

Adds the specified MAC address as a denied MAC address for the service instance.

Step 8 **mac security address** **deny** *mac-address*

Example:

```
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaab
```

Adds the specified MAC address as a denied MAC address for the service instance.

Step 9 **mac security address** **deny** *mac-address*

Example:

```
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaac
```

Adds the specified MAC address as a denied MAC address for the service instance.

Step 10 **mac security address** **deny** *mac-address*

Example:

```
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaad
```

Adds the specified MAC address as a denied MAC address for the service instance.

Step 11 **mac security address** **deny** *mac-address*

Example:

```
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaae
```

Adds the specified MAC address as a denied MAC address for the service instance.

Step 12 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 13 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring MAC Address Security on a Service Instance

Perform this task to configure an upper limit for the number of secured MAC addresses allowed on a service instance. This number includes addresses added as part of a permit list as well as dynamically learned MAC addresses. If the upper limit is decreased, all learned MAC entries are removed.

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 gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.

Step 5 **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 6 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security maximum addresses** *maximum-addresses*

Example:

```
Device(config-if-srv)# mac security maximum addresses 500
```

Sets the maximum number of secure addresses permitted on the service instance.

Note Default value for a service instance is '1'.

Step 8 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 9 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring a MAC Address Violation

Perform this task to specify the expected behavior of a device when an attempt to dynamically learn a MAC address fails because the configured MAC security policy on the service instance was violated.

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 gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.

Step 5 **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 6 **bridge-domain** *bridge-id*

Example:

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

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 Do one of the following:

- **mac security violation restrict**
- **mac security violation protect**

Example:

```
Device(config-if-srv)# mac security violation restrict
```

Example:

```
Device(config-if-srv)# mac security violation protect
```

Sets the violation mode (for Type 1 and 2 violations) to restrict.

or

Sets the violation mode (for Type 1 and 2 violations) to protect.

- If a MAC security violation response is not specified, by default, the violation mode is shutdown.

Step 8 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 9 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring MAC Address Aging

Perform this task to configure the aging of secured MAC addresses under MAC security. Secured MAC addresses are not subject to the normal aging of MAC table entries. If aging is not configured, secured MAC addresses are never aged out.

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 gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance 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 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security aging time** *aging-time* **[inactivity]**

Example:

```
Device(config-if-srv)# mac security aging time 200 inactivity
```

Sets the aging time for secure addresses, in minutes. The optional **inactivity** keyword specifies that the aging out of addresses is based on inactivity of the sending hosts (as opposed to absolute aging).

Step 8 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 9 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Configuring a Sticky MAC Address

If sticky MAC addressing is configured on a secured service instance, MAC addresses that are learned dynamically on the service instance are retained during a link-down condition. Perform this task to configure sticky MAC addresses 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 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet2/0/1
```

Specifies the interface type and number, and enters interface configuration mode.

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

Example:

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

Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.

Step 5 **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 6 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 200
```

Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

Step 7 **mac security sticky address** *mac-address*

Example:

```
Device(config-if-srv)# mac security sticky address 1111.2222.3333
```

Sets up a MAC address to be declared as a sticky MAC address on the service instance.

Step 8 **mac security**

Example:

```
Device(config-if-srv)# mac security
```

Enables MAC security on the service instance.

Step 9 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to user EXEC mode.

Displaying the MAC Security Status of a Specific Service Instance

Perform this task to display the MAC security status of a service instance.

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* **mac security**

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet1/1 mac security
```

Displays the MAC security status of a specific service instance.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Displaying the Service Instances with MAC Security Enabled

Perform this task to display all the service instances with MAC security enabled.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ethernet service instance mac security**

Example:

```
Device# show ethernet service instance mac security
```

Displays all the service instances with MAC security enabled.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Displaying the Service Instances with MAC Security Enabled on a Specific Bridge Domain

Perform this task to display the service instances on a specific bridge domain that have MAC security enabled.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show bridge-domain** *id* **mac security**

Example:

```
Device# show bridge-domain 100 mac security
```

Displays all the service instances with MAC security enabled on a specific bridge domain.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the MAC Addresses of All Secured Service Instances

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ethernet service instance mac security address**

Example:

```
Device# show ethernet service instance mac security address
```

Displays the secured addresses on all the service instances.

Step 3 **show mac address-table secure**

Example:

```
Device# show mac address-table secure
```

Displays the secure MAC address on the service instances.

Step 4 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the MAC Addresses of a Specific Service Instance

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* **mac security address**

Example:

```
Device# show ethernet service instance id 200 interface GigabitEthernet 1/0 mac security
address
```

Displays the addresses of a specific service instance.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the MAC Addresses of All Service Instances on a Specific Bridge Domain

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show bridge-domain** *id* **mac security address**

Example:

```
Device# show bridge-domain 100 mac security address
```

Displays the secured addresses of all the service instances on a specified bridge domain.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the MAC Security Statistics of a Specific Service Instance

This section describes how to display the MAC security statistics of a specific service instance.

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* **mac security statistics**

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet1/1 mac security
statistics
```

Displays the MAC security statistics of a specific service instance.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the MAC Security Statistics of All Service Instances on a Specific Bridge Domain

Perform this task to display the MAC security statistics of all the service instances on a specific bridge domain.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show bridge-domain** *bridge-id* **mac security statistics**

Example:

```
Device# show bridge-domain 100 mac security statistics
```

Displays the MAC security statistics of all service instances that belong to a specific bridge domain.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Showing the Last Violation Recorded on Each Service Instance on a Specific Bridge Domain

Perform this task to display the last violation recorded on each service instance on a specific bridge domain. Service instances on which there have been no violations are excluded from the output.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show bridge-domain** *bridge-id* **mac security last violation**

Example:

```
Device# show bridge-domain 100 mac security last violation
```

Displays information about the last violation recorded on each of the service instances that belong to the bridge domain.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Clearing All Dynamically Learned Secure MAC Addresses on a Service Instance

Perform this task to clear all dynamically learned Secure MAC addresses on a service instance.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **clear ethernet service instance id** *id* **interface** *type number* **mac table**

Example:

```
Device# clear ethernet service instance id 100 interface gigabitethernet0/0/1 mac table
```

Clears all the dynamically learned Secure MAC addresses on the specified service instance.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Clearing All Dynamically Learned MAC Addresses on a Bridge Domain

Perform this task to clear all dynamically learned MAC addresses on a bridge domain.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **clear bridge-domain** *bridge-id* **mac table**

Example:

```
Device# clear bridge-domain 100 mac table
```

Clears all dynamically learned MAC addresses on the specified bridge domain.

Step 3 **end**

Example:

```
Device# end
```

Returns to user EXEC mode.

Bringing a Specific Service Instance Out of the Error-Disabled State

Perform this task to bring a specific service instance out of the error-disabled state.

Note The **clear ethernet service instance id** *id* **interface** type *number* **errdisable**command can also be used to bring a service instance out of an error disabled state. For more information about this command, see the *Cisco IOS Carrier Ethernet Command Reference*.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Device(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**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 6	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge-domain instance where *bridge-id* is the identifier for the bridge-domain instance.
Step 7	**mac security** **Example:** `Device(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 8	**errdisable recovery cause mac-security** *interval* **Example:** `Device(config-if-srv)# errdisable recovery cause mac-security 50`	Brings a specific service instance out of an error-disabled state and specifies a time interval to recover.

Step 9	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns to user EXEC mode.

Configuration Examples for MAC Address Limiting on Service Instances and Bridge Domains and EVC Port Channels

Example Enabling MAC Security on a Service Instance

The following example shows how to enable MAC security on a service instance:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Enabling MAC Security on an EVC Port Channel

The following example shows how to enable MAC Security on an EVC port channel:

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring a MAC Address Permit List

The following example shows how to configure a MAC address permit list:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaab
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaac
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaad
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaae
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security maximum addresses 5
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaab
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaac
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaad
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaae
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring a MAC Address Deny List

The following example shows how to configure a MAC address deny list:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaaa
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaab
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaac
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaad
Device(config-if-srv)# mac security address deny a2aa.aaaa.aaae
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring a MAC Address Security on a Service Instance

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security maximum addresses 10
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring a MAC Address Violation Response

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
Device(config-if-srv)# mac security violation protect
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring MAC Address Aging

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 4/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security aging time 10
Device(config-if-srv)# mac security
Device(config-if-srv)# end
```

Example Configuring a Sticky MAC Address

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 3/0/1
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1Q 100
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# mac security sticky address 1111.2222.3333
Device(config-if-srv)# mac security
```

Example Displaying the MAC Addresses on a Specific Secure Service Instance

```
Device# show ethernet service instance id 1879665131 interface gigabitethernet 0/2 mac
security address
MAC Address       Type      Rem. Age(min)
0001.0001.0001    static    100
0001.0001.0002    static    100
0001.0001.aaaa    dynamic   100
0001.0001.aaab    dynamic   100

Device# show ethernet service instance id 10 inter gig 0/0/3 mac security
address Bridge-domain 10

MAC Address       Type
0000.00ac.ef02    sticky
0000.00ac.ef03    sticky
0001.0001.aaaa    dynamic
0001.0001.aaab    dynamic
```

Example Displaying the Last Violation on a Specific Service Instance

```
Device# show ethernet service instance id 1879665131 interface gigabitethernet 0/2 mac
security last violation
At: Apr 4 06:57:25.971
Source address: ae4e.b7b5.79ae
Reason: Denied address

Device# show bridge-domain 100 mac security last violation Te0/0/3 ServInst 200
Last violation at: 15:54:25 IST Fri Jun 5 2015
Source MAC address: 0000.1111.1111
```

```
Reason: Re-learn attempt
Total violation count: 321
```

Example Displaying the MAC Security Status of a Specific Service Instance

```
Device# **show ethernet service instance id 1879665131 interface Ethernet0/2 mac security**
MAC Security: enabled

Device# show ethernet service instance id 100 interface te0/0/3 mac security
Bridge-domain 100
MAC Security enabled: yes
```

Example Displaying the MAC Addresses of All Secured Service Instances

```
Device# show ethernet service instance mac security address
Port                   Bridge-domain    MAC Address       Type      Rem. Age(min)
Gi1/0/0 ServInst 1     10               0001.0001.0001    static    82
Gi1/0/0 ServInst 1     10               0001.0001.0002    static    82
Gi1/0/0 ServInst 1     10               0001.0001.aaaa    dynamic   82
Gi1/0/0 ServInst 1     10               0001.0001.aaab    dynamic   82
Gi1/0/0 ServInst 2     10               0002.0002.0002    static    -
Gi1/0/0 ServInst 2     10               0002.0002.0003    static    -
Gi1/0/0 ServInst 2     10               0002.0002.0004    static    -
Gi1/0/0 ServInst 2     10               0002.0002.aaaa    dynamic   -
Gi1/0/0 ServInst 2     10               0002.0002.bbbb    dynamic   -
Gi1/0/0 ServInst 2     10               0002.0002.cccc    dynamic   -
Gi3/0/5 ServInst 10    30               0003.0003.0001    static    200
Gi3/0/5 ServInst 10    30               0003.0003.0002    static    200

Device# show ethernet service instance mac security address
Port                   Bridge-domain    MAC Address       Type
Gi0/0/3 ServInst 10    10  0000.00ac.ef02 sticky
Gi0/0/3 ServInst 10    10   0000.00ac.ef03 sticky
Gi0/0/3 ServInst 10    10   0000.00ac.ef04  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef05  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef06  sticky
Gi0/0/3 ServInst 10    10   0000.00ac.ef07  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef08  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef09  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef0a  dynamic
Gi0/0/3 ServInst 10    10   0000.00ac.ef0b  dynamic
```

Example Displaying the MAC Security Statistics of All Service Instances

In the following example, the numbers of allowed and actual secured addresses recorded on the service instance are displayed.

```
Device# show ethernet service instance mac security statistics
Ethernet0/0 service instance 890597333 (bridge-domain 730)
Secure addresses: 3
Address limit: 7
Ethernet0/0 service instance 1559665780 (bridge-domain 1249)
Secure addresses: 8
Address limit: 8
Ethernet0/0 service instance 1877043343 (bridge-domain 1155)
```

```
Secure addresses: 0
Address limit: 8
Ethernet0/1 service instance 127771402 (bridge-domain 730)
Secure addresses: 12
Address limit: 12
Ethernet0/1 service instance 183598286 (bridge-domain 730)
Secure addresses: 1
Address limit: 1
Ethernet0/1 service instance 433365207 (bridge-domain 1249)
Secure addresses: 0
Address limit: 1
Ethernet0/1 service instance 858688453 (bridge-domain 1328)
Secure addresses: 0
Address limit: 2

Device#  show ethernet serv instance mac security statistics
Te0/0/3 ServInst 100 (bridge-domain 100)
Current secure addresses: 1
Permitted addresses: 10
Te0/0/3 ServInst 200 (bridge-domain 100)
Current secure addresses: 0
Permitted addresses: 1
Te0/0/3 ServInst 300 (bridge-domain 100)
Current secure addresses: 0
Permitted addresses: 1
```

Example: Displaying the MAC Addresses on All Service Instances for a Bridge Domain

```
Router# show bridge-domain 730 mac security address
Port                 MAC Address       Type      Rem. Age(min)
Gi1/0/0 ServInst 1    0001.0001.0001    static    74
Gi1/0/0 ServInst 1    0001.0001.0002    static    74
Gi1/0/0 ServInst 1    0001.0001.aaaa    dynamic   74
Gi1/0/0 ServInst 1    0001.0001.aaab    dynamic   74
Gi1/0/0 ServInst 2    0002.0002.0002    static    -
Gi1/0/0 ServInst 2    0002.0002.0003    static    -
Gi1/0/0 ServInst 2    0002.0002.0004    static    -
Gi1/0/0 ServInst 2    0002.0002.aaaa    dynamic   -
Gi1/0/0 ServInst 2    0002.0002.bbbb    dynamic   -
Gi1/0/0 ServInst 2    0002.0002.cccc    dynamic   -

Router# show bridge-domain 10 mac security address
Port                 MAC Address       Type
Gi0/0/3 ServInst 10   0000.00ac.ef02    sticky
Gi0/0/3 ServInst 10   0000.00ac.ef03    sticky
Gi0/0/3 ServInst 10   0000.00ac.ef04    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef05    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef06    sticky
Gi0/0/3 ServInst 10   0000.00ac.ef07    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef08    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef09    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef0a    dynamic
Gi0/0/3 ServInst 10   0000.00ac.ef0b    dynamic
```

Example Displaying the Secured Service Instances for a Specific Bridge Domain

```
Router# show bridge-domain 730 mac security
Gi1/0/0 ServInst 1
MAC Security enabled: yes
Gi1/0/0 ServInst 2
MAC Security enabled: yes

Router# show bridge-domain 10 mac security
Gi0/0/3 ServInst 10
MAC Security enabled: yes
```