Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T Module Overview

The Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T are the next generation Form-factor LAN/WAN NIM modules that provide enhanced security, reliability, and performance. The Cisco C-NIM-4X module provides Small Form-Factor Pluggable Plus (SFP+), 10G and 1 Gigabit connectivity. The Cisco C-NIM-8T module provides 1 Gigabit RJ45 connectivity supporting 100 Mbps. Similarly, the Cisco C-NIM-8M module provides 2.5 Gbps mGig connectivity and supports UPoE+. Also, Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T support Layer 2 and Layer 3 configurable Ethernet network. The Cisco C-NIM-4X, Cisco C-NIM-8M, ard C-NIM-8T modules are supported only in a NIM slot. These modules are not supported in the SM-NIM Carrier Adapter and C-SM-NIM-ADPT. You can install a Cisco C-NIM-8M or C-NIM-8T Network Interface module on the Cisco Catalyst 8200 and 8300 Series Edge Platforms. However, the Cisco C-NIM-4X is supported only on the Cisco Catalyst 8300 Series Edge Platforms.

This chapter includes the following topics:

Prerequisites for the Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T Network Interface Module

Cisco IOS XE Dublin 17.11.1a or a later release is required to install the Cisco C-NIM-4X and C-NIM-8T Network Interface Modules. To install the C-NIM-8M module, you require the Cisco IOS XE Dublin 17.12.2 release or a later release.

To determine the version of Cisco IOS software that is running on your router, log in to the router and enter the **show version** command:

```
Router> show version
Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Experimental Version


Technical Support: http://www.cisco.com/techsupport
```

```
Copyright (c) 1986-2022 by Cisco Systems, Inc.

Compiled Sun 09-Jan-23 04:16 by mcpre
```

Limitation for the Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T Network Interface Module

This section describes limitations for the Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T modules.

- Jumbo MTU support limits is set to 2048 on Cisco Catalyst 8300-2N2S-4T2X/6T including Layer 2, Layer 3, and SVI interface.
- Set the same negotiation setting as peer site (both auto or non-auto) for 1000Base-X based SFP on C-NIM-4X module. If you do not have the same negotiation setting, the link will be down.
- LAN switch-to-host MACsec is not supported. However, switch-to-switch LAN/WAN MACsec is supported.
- Supports 10G SFP+ WAN MACsec scaling limits to a maximum of 16 peers on port basis.
- The 1G WAN port supports maximum of 8 peers on port basis. However, the 2.5G WAN port supports maximum of 16 peers on port basis.

Configuring C-NIM-4X, C-NIM-8M, and C-NIM-8T Network Interface Modules

This section describes how to configure the Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T NIM features and some important concepts about the Cisco C-NIM-4X, C-NIM-4M, and C-NIM-8T modules:

Software Features

The following are the software features supported on the Cisco C-NIM-4X, C-NIM-8M, and C-NIM-8T NIM modules:

Configuring Flex Support on Layer 2 and Layer 3 Ports

About Flex Ports

The flex ports which are the two highest numbered ports that provide more Layer 3 WAN ports flexibility on the device. The flex ports can be configured as either a Layer 2 port or a Layer 3 port as per the requirement.

How to Configure Flex Ports

The flex ports are set to Layer 2 interface by default. They can be configured to the Layer 3 port using **no switchport** command and can be returned to the Layer 2 port using **switchport** command. After the interface is converted to Layer 2 or Layer 3, the corresponding Layer 2 or Layer 3 CLIs will be available on that interface.

Configuring Flex Port to Layer 2 Port

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **switchport mode** {**access** | **dynamic** | **trunk trunk**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
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-if)# interface GigabitEthernet0/1/6`	Enters the configuration mode for the specified interface on the router.
Step 4	**switchport** **Example:** `Device(config-if)# switchport`	Converts the port from the Layer 3 interface to the Layer 2 interface and makes it a switch interface rather than a router interface.
Step 5	**switchport mode** {**access** | **dynamic** | **trunk trunk** **Example:** `Device(config-if)# switchport mode access`	Configures the operational mode on a Layer 2 interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits configuration mode for the specified interface and returns to global configuration mode.

Configuring Flex Port to Layer 3 Port

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **no switchport**
5. **ip address** *address mask*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
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-if)# interface GigabitEthernet0/1/6`	Enters the configuration mode for the specified interface on the router.
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Converts the port from the Layer 2 interface to the Layer 3 interface and makes it a router interface rather than a switch port.
Step 5	**ip address** *address mask* **Example:** `Device(config-if)# ip address 10.10.0.1 255.255.255.0`	Sets the IP address and subnet mask for the specified interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits configuration mode for the specified interface and returns to global configuration mode.

Power over Ethernet

The Power over Ethernet (PoE) feature allows you to manage power. The Cisco C-NIM-8M module supports PoE (802.3af), PoE+ 802.3at) on ports 4 to 7 and provides up to 90 Watts of power. By using PoE, you do not need to supply connected PoE-enabled devices with wall power.

Note The PoE feature is not supported on Cisco C-NIM-4X and C-NIM-8T modules. The Cisco C-NIM-8M supports PoE only on the 4 ports, from port 4 to 7 of the module.

Configuring Power over Ethernet on the Cisco C-NIM-8M Module

By defult the inline power on PoE port is enable and it will be in PoE mode. Optionally, you can specify the maximum wattage that is allowed on the interface. The default value is 90000.

To configure the PoE, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **power inline** {**auto**[**max** *max-wattage*] | **never**}
5. **power inline port poe-ha**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
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** *interface-id* **Example:** `Device(config)# interface twoGigabitEthernet 0/1/7`	Specifies the physical port to be configured, and enters interface configuration mode.

Step 4	**power inline** {**auto**[**max** *max-wattage*] | **never**}
	Example:
	`Device(config-if)# power inline auto`
		Configures the PoE mode on the port. The keywords have these meanings:
		• **auto**- Enables powered—device detection. If enough power is available, automatically allocates power to the PoE port after device detection. This is the default setting.
		• **max** max-wattage — Limits the power allowed on the port. If no value is specified, the maximum is allowed.
		• **never** —Disables device detection, and disable power to the port.

Step 5	**power inline port poe-ha** **Example:** `Device(config-if)# power inline port poe-ha`	Configures POE High Availability.

	Command or Action	Purpose

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Verifying Power over Ethernet

To verify the PoE status, use the **show power linline** command.

```
router#show power inline

Available:250.0(w)  Used:202.1(w)  Remaining:47.9(w)
Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Tw0/1/4   auto   on         90.0    Ieee PD             5     90.0
Tw0/1/5   auto   on         11.8    IP Phone 9971       4     90.0
Tw0/1/6   auto   on         90.0    Ieee PD             5     90.0
Tw0/1/7   auto   on         10.3    IP Phone 7970       3     90.0
--------- ------ ---------- ---------- ---------- ------ -----
Totals:          4    on    202.1
```

Layer 2 Mode Features

This section includes the following Layer 2 features:

SVI Supported Features

The following table provided the supported features on the SVI.

Table 1: SVI Supported Features

Techolongy	Feature	Use Case
Routing	Routing Protocol	Interconnects Layer 3 networks using protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF) Protocol, and Enhanced Interior Gateway Routing Protocol (EIGRP) configured under SVI. For more informaton on routing protocol, see the IP Routing: Protocol-Independent Configuration Guide.
	Hot Standby Router Protocol (HSRP)	Supports redundancy and high availability with a secondary device connected to the LAN with SVI, using HSRP. For more informaton on HSRP, see the *First Hop Redundancy Protocols Configuration Guide.*.
	DHCP	Cisco devices running Cisco software include Dynamic Host Configuration Protocol (DHCP) server and the relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the device to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the Domain Name System (DNS) server and the default device. For more informaton on HSRP, see the, IP Addressing: DHCP Configuration Guide.
	Multicast (IPv4)	Provides multicast support for clients connected to the switch ports. For more informaton on IP Multicast, see the, IP Multicast: PIM Configuration Guide.

Techolongy	Feature	Use Case
	VRF	Associates a VRF instance with an SVI to map VLANs to different logical or physical VPN WAN connections. For more informaton on VRF protocol, see the IP Routing: Protocol-Independent Configuration Guide.
Security	ACL	Provides packet filtering to control network traffic and restrict the access of users and devices to the network For more informaton on ACL protocol, see the Security Configuration Guide: Access Control Lists.
	NAT	Provides NAT under SVI. For more information on NAT, see the IP Addressing: NAT Configuration Guide.
Qos	Classification with standard and extended access list	Provides QoS classification with standard and extended access lists. For more informtion on QoS, see the Security Configuration Guide: Access Control Lists.
	Class-based marking	Provides QoS marking based on user-defined traffic class with DSCP and IP precedence values. For more information on QoS Marking, see the QoS: Classification Configuration Guide.
	Policing	Limits the input or output transmission rate on SVI and specifies traffic handling policies when the traffic either conforms to or exceeds the specified rate limits. For more informtion on Policing, see the QoS: Policing and Shaping Configuration Guide

Techolongy	Feature	Use Case
Bridging	EVC under SVI	Supports a default encapsulation EFP under SVI, to have VLAN/BD integrated.
	EVC with MAC ACL under SVI	For more information on EVC, see the https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cether/configuration/xe-3s/asr903/16-11-1/b-ce-layer2-xe-xe-16-11-asr900/b-ce-layer2-xe-xe-16-11-asr900_chapter_011.html

IEEE 802.1x Protocol

The IEEE 802.1x standard defines a client/server-based access control and authentication protocol that prevents clients from connecting to a LAN through publicly accessible ports unless they are authenticated. The authentication server authenticates each client connected to a port before making available any services offered by the router or the LAN.

Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic can pass through the port. For more information on IEEE 802.1x port-based authentication, see the Configuring IEEE 802.1x Port-Based Authentication chapter of the *Security Configuration Guide, Cisco IOS XE Gibraltar 16.10.x*.

VLANs

A VLAN is a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs. However, you can group end-stations even if they are not physically located on the same LAN segment. Any device port can belong to a VLAN, unicast, broadcast, and multicast packets are forwarded and flooded only to end-stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router or a device supporting fallback bridging. In a device stack, VLANs can be formed with ports across the stack. Because a VLAN is considered a separate logical network, it contains its own bridge Management Information Base (MIB) information and can support its own implementation of spanning tree.

VLANs are often associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Interface VLAN membership on the device is assigned manually on an interface-by-interface basis. When you assign device interfaces to VLANs by using this method, it is known as interface-based, or static, VLAN membership.

The device can route traffic between VLANs by using device virtual interfaces (SVIs). An SVI must be explicitly configured and assigned an IP address to route traffic between VLANs.

Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet IEEE 802.1Q tagged), the packet is dropped, and the source address is not learned.

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. These trunk port types are supported:

- An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

Creating a VLAN

Before you begin

With VTP version 1 and 2, if the device is in VTP transparent mode, you can assign VLAN IDs greater than 1006, but they are not added to the VLAN database.

To configure the Vlan, perform these steps. You can configure the Vlan in access or trunk mode. The procedure is same for the both the modes.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **name** *vlan-name*
4. **exit**
5. **interface interface-id**
6. **switchport mode access**
7. **switchport access vlan** *vlan id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**vlan** *vlan-id* **Example:** `(config)# vlan 20`	Enters a VLAN ID, and enters VLAN configuration mode. Enter a new VLAN ID to create a VLAN, or enter an existing VLAN ID to modify that VLAN. **Note** The available VLAN ID range for this command is 1 to 4094.
Step 3	**name** *vlan-name* **Example:** `(config-vlan)# name test20`	(Optional) Enters a name for the VLAN. If no name is entered for the VLAN, the default is to append the *vlan-id* value with leading zeros to the word VLAN. For example, VLAN0004 is a default VLAN name for VLAN 4.
Step 4	**exit** **Example:** `(config-vlan)# exit`	Returns to configuration mode.
Step 5	**interface interface-id** **Example:** `router(config)# interface gigabitethernet 0/1/1`	Specifies the physical port to be configured, and enter interface configuration mode.
Step 6	**switchport mode access** **Example:** ` router(config-if)# switchport mode access`	Configures the interface as a VLAN access port.
Step 7	**switchport access vlan** *vlan id* **Example:** `router(config-if)# switchport access vlan 20`	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic..

Step 8	**end**
	Example:
	`router(config-if)# end`
		Returns to configuration mode.

Configuring LAN Ports for Layer 2 Switching

This section describes how to configure all three types of ethernet LAN ports for Layer 2 switching on the Cisco Catalyst 8200 and Catalyst 8300 Series routers. The configuration tasks in this section apply to LAN ports on LAN switching modules.

Layer 2 LAN Port Modes

The following table lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 2: Layer 2 LAN Port Modes

Mode	Function
switchport mode access	Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change.
switchport mode dynamic desirable	Makes the LAN port actively attempt to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk**, **desirable**, or **auto** mode. This is the default mode for all LAN ports.
switchport mode dynamic auto	Makes the LAN port willing to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** or **desirable** mode.
switchport mode trunk	Puts the LAN port into permanent trunking mode and negotiates to convert the link into a trunk link. The LAN port becomes a trunk port even if the neighboring port does not agree to the change.
switchport nonegotiate	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link.

Note DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that LAN ports connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the LAN port to become a trunk but not generate DTP frames.

Default Layer 2 LAN Interface Configuration

The following table shows the Layer 2 LAN port default configuration.

Table 3: Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode:	
• Before entering the **switchport** command **Note** You can enter the switchport only for Flex Layer 2 and Layer 3 ports. This is not required for the Layer 2 specific ports.	
• After entering the **switchport** command	**switchport mode dynamic desirable**

Feature	Default
Default access VLAN	VLAN 1
Native VLAN (for 802.1Q trunks)	VLAN 1

Configuring LAN Interfaces for Layer 2 Switching

This section describes how to configure Layer 2 switching on the Cisco Catalyst 8200 and Catalyst 8300 Series routers:

Note Use the **default interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/subslot/port* command to revert an interface to its default configuration.

Configuring VLAN Trunking Protocol

This section describes how to configure the VLAN Trunking Protocol (VTP) on C-NIM-4X and C-NIM-8T of the Cisco Catalyst 8200 and 8300 Series Edge Platforms.

Configuring a VTP Server

When a device is in VTP server mode, you can change the VLAN configuration and have it propagate throughout the network.

SUMMARY STEPS

1. **vtp mode server**
2. **vtp domain** *domain_name*
3. **vtp password** *password_value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vtp mode server** **Example:** `Device(config)#vtp mode server`	Configures the switch as a VTP server.
Step 2	**vtp domain** *domain_name* **Example:** `Device(config)#vtp domain domain1`	Defines the VTP domain name, which can be up to 32 characters long.
Step 3	**vtp password** *password_value* **Example:** `Router(config)#vtp password password1`	(Optional) Sets a password, which can be from 8 to 64 characters long, for the VTP domain.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)#exit`	Exits global configuration mode.

Configuring a VTP Client

When a device is in VTP server mode, you can change the VLAN configuration and have it propagate throughout the network.

SUMMARY STEPS

1. **vtp mode client**
2. **vtp domain** *domain_name*
3. **vtp password** *password_value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vtp mode client** **Example:** `Device(config)#vtp mode client`	Configures the switch as a VTP client.
Step 2	**vtp domain** *domain_name* **Example:** `Device(config)#vtp domain domain1`	Defines the VTP domain name, which can be up to 32 characters long.
Step 3	**vtp password** *password_value* **Example:** `Router(config)#vtp password password1`	(Optional) Sets a password, which can be from 8 to 64 characters long, for the VTP domain.
Step 4	**exit** **Example:** `Device(config)#exit`	Exits global configuration mode.

Configuring MAC Table Manipulation

Port security is implemented by providing the user with the option to make a port secure by allowing only well-known MAC addresses to send in data traffic. Up to 200 secure MAC addresses are supported.

Enabling Known MAC Address Traffic

To enable the MAC address, perform these steps:

SUMMARY STEPS

1. **configure terminal**
2. **mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Devicer#configure terminal`	Enters global configuration mode.
Step 2	**mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id* **Example:** `Device(config)#mac-address-table static 001.002.003 vlan 1 interface g0/1/0`	Sepecifies the MAC address traffic on the port.

Step 3	**end**
	Example:
	`Device(config)#end`
		Exits global configuration mode.

Creating a Static Entry in the MAC Address Table

To create a static entry in the MAC address table, perform these steps:

SUMMARY STEPS

1. **configure terminal**
2. **mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Devicer# configure terminal`	Enters global configuration mode.
Step 2	**mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id* **Example:** `Device(config)#mac-address-table static 001.002.003 vlan 1 interface g0/1/0`	Creates static entry in the MAC address table.
Step 3	**end** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuring the Aging Timer

To configuring the aging time, perform these steps:

SUMMARY STEPS

1. **configure terminal**
2. **mac-address-table aging-time** *aging-timer*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Devicer# configure terminal`	Enters global configuration mode.
Step 2	**mac-address-table aging-time** *aging-timer* **Example:** `Router(config)# mac-address-table aging-time 320`	Configures the MAC address aging timer age, in seconds.

Step 3	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode.

Configuring a Layer 2 Switching Port as a Trunk

These sections describe configuring a Layer 2 switching port as a trunk:

Configuring the Layer 2 Switching Port as 802.1Q Trunk

- When you enter the **switchport** command with no other keywords, the default mode is **switchport mode dynamic desirable** and **switchport trunk encapsulation negotiate**.

To configure the Layer 2 switching port as an ISL or 802.1Q trunk, perform this task:

Command	Purpose
`Router(config-if)#` **`switchport mode trunk`**	(Optional) Configures the Layer 2 switching port mode as 802.1Q trunk.

When configuring the Layer 2 switching port as 802.1Q trunk, note the following information:

- The **switchport mode trunk** command (see the Configuring the Layer 2 Trunk Without the DTP) is not compatible with the **switchport trunk encapsulation negotiate** command.

- To support the **switchport mode trunk** command, you must configure the encapsulation as 802.1Q.

Configuring the Layer 2 Trunk With the DTP

Note Complete the steps in the “Configuring a LAN Port for Layer 2 Switching” section before performing the tasks in this section.

To configure the Layer 2 trunk with the DTP, perform this task:

Command	Purpose
`Router(config-if)# switchport mode dynamic {auto | desirable}`	(Optional) Configures the trunk to use DTP.
`Router(config-if)# no switchport mode`	Reverts to the default trunk trunking mode (**switchport mode dynamic desirable**).

When configuring the Layer 2 trunk to use DTP, note the following information:

- Required only if the interface is a Layer 2 access port or to specify the trunking mode.
- See the Layer 2 LAN Port Modes table for information about trunking modes.

Configuring the 802.1Q Native VLAN

Command	Purpose
`Router(config-if)# switchport access vlan vlan_ID`	(Optional) Configures the access VLAN, which is used if the interface stops trunking. The *vlan_ID* value can be 1 through 4094, except reserved VLANs.
`Router(config-if)# no switchport access vlan`	Reverts to the default value (VLAN 1).

To configure the 802.1Q native VLAN, perform this task:

Command	Purpose
`Router(config-if)# switchport trunk native vlan vlan_ID`	(Optional) Configures the 802.1Q native VLAN.
`Router(config-if)# no switchport trunk native vlan`	Reverts to the default value (VLAN 1).

When configuring the native VLAN, note the following information:

- The *vlan_ID* value can be 1 through 4094, except reserved VLANs.
- The access VLAN is not automatically used as the native VLAN.

IGMP Snooping for IPv4

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content. You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it.

Layer 2 switches can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients. For more information on this feature, see http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-2_1_e/configuration/guide/scg3750x/swigmp.html.

DHCP Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database, also referred to as a DHCP snooping binding table.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces, as untrusted DHCP messages will be forwarded only to trusted interfaces.

An untrusted DHCP message is a message that is received through an untrusted interface. By default, the switch considers all interfaces untrusted. So, the switch must be configured to trust some interfaces to use DHCP Snooping. When you use DHCP snooping in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer’s switch. Messages from unknown devices are untrusted because they can be sources of traffic attacks.

The DHCP snooping binding database has the MAC address, the IP address, the lease time, the binding type, the VLAN number, and the interface information that corresponds to the local untrusted interfaces of a switch. It does not have information regarding hosts interconnected with a trusted interface.

In a service-provider network, an example of an interface you might configure as trusted is one connected to a port on a device in the same network. An example of an untrusted interface is one that is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a packet on an untrusted interface and the interface belongs to a VLAN in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If the addresses match (the default), the switch forwards the packet. If the addresses do not match, the switch drops the packet.

The switch drops a DHCP packet when one of these situations occurs:

- A packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, is received from outside the network or firewall.

- A packet is received on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match.
- The switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that has a MAC address in the DHCP snooping binding database, but the interface information in the binding database does not match the interface on which the message was received.
- A DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0, or the relay agent forwards a packet that includes option-82 information to an untrusted port.

If the switch is an aggregation switch supporting DHCP snooping and is connected to an edge switch that is inserting DHCP option-82 information, the switch drops packets with option-82 information when packets are received on an untrusted interface. If DHCP snooping is enabled and packets are received on a trusted port, the aggregation switch does not learn the DHCP snooping bindings for connected devices and cannot build a complete DHCP snooping binding database.

When an aggregation switch can be connected to an edge switch through an untrusted interface and you enter the **ip dhcp snooping information option allow-untrusted** global configuration command, the aggregation switch accepts packets with option-82 information from the edge switch. The aggregation switch learns the bindings for hosts connected through an untrusted switch interface. The DHCP security features, such as dynamic ARP inspection or IP source guard, can still be enabled on the aggregation switch while the switch receives packets with option-82 information on untrusted input interfaces to which hosts are connected. The port on the edge switch that connects to the aggregation switch must be configured as a trusted interface.

DHCP Snooping Configuration Guidelines

- If a switch port is connected to a DHCP server, configure a port as trusted by entering the **ip dhcp snooping trust interface** configuration command.
- If a switch port is connected to a DHCP client, configure a port as untrusted by entering the **no ip dhcp snooping trust** interface configuration command.
- You can display DHCP snooping statistics by entering the **show ip dhcp snooping statistics** user EXEC command, and you can clear the snooping statistics counters by entering the **clear ip dhcp snooping statistics** privileged EXEC command.

Default DHCP Snooping Configuration

Table 4: Default DHCP Configuration

Feature	Default Setting
DHCP snooping enabled globally	Disabled
DHCP snooping information option	Enabled
DHCP snooping option to accept packets on untrusted input interfaces[1]	Disabled
DHCP snooping limit rate	None configured
DHCP snooping trust	Untrusted
DHCP snooping VLAN	Disabled

Feature	Default Setting
DHCP snooping MAC address verification	Enabled
Cisco IOS DHCP server binding database	Enabled in Cisco IOS software, requires configuration. **Note** The switch gets network addresses and configuration parameters only from a device configured as a DHCP server.
DHCP snooping binding database agent	Enabled in Cisco IOS software, requires configuration. This feature is operational only when a destination is configured.

[1] Use this feature when the switch is an aggregation switch that receives packets with option-82 information from an edge switch.

Enabling DHCP Snooping Globally

To enable DHCP snooping on the device, perform these steps:

SUMMARY STEPS

1. **ip dhcp snooping**
2. **no ip dhcp snooping**
3. **do show ip dhcp snooping**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip dhcp snooping** **Example:** `(config)# ip dhcp snooping`	Enables DHCP snooping globally.
Step 2	**no ip dhcp snooping** **Example:** `(config)# ip dhcp snooping`	Disables DHCP snooping..
Step 3	**do show ip dhcp snooping** **Example:** `(config)# do show ip dhcp snooping`	Verifies the configuration.
Step 4	**exit** **Example:** `(config)# exit`	Exits global configuration mode. This example shows how to enable DHCP snooping globally:

	Command or Action	Purpose
		Device# **configure terminal** Device(config)# **ip dhcp snooping** Device(config)# **do show ip dhcp snooping** Switch DHCP snooping is enabled Device#

Spanning Tree Protocol Overview

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Device might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one device of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The device that has *all* of its ports as the designated role or as the backup role is the root device. The device that has at least *one* of its ports in the designated role is called the designated device.

Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Device send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The device do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending device and its ports, including device and MAC addresses, device priority, port priority, and path cost. Spanning tree uses this information to elect the root device and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a device are part of a loop, the spanning-tree and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

Note By default, the device sends keepalive messages (to ensure the connection is up) only on interfaces that do not have small form-factor pluggable (SFP) modules. You can change the default for an interface by entering the [**no**] **keepalive** interface configuration command with no keywords.

Default STP Configuration

The following table shows the default STP configuration.

Table 5: STP Default Configuration

Feature	Default Value
Disable state	STP disabled for all VLANs
Bridge priority	32768
STP port priority (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	128
STP port cost (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	Gigabit Ethernet: 4
STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	128
STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	Gigabit Ethernet:1000000000
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds
Mode	PVST

Enabling STP

Note STP is disabled by default on all VLANs.

You can enable STP on a per-VLAN basis. The Cisco C-NIM-4X and C-NIM-8T Layer 2 Gigabit EtherSwitch Service Module maintain a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

If you want to enable a mode that is different from the default mode, perform these steps:

SUMMARY STEPS

1. **configure terminal**
2. **spanning-tree mode** {**pvst** | **mst** | **rapid-pvst**}
3. **interface** *interface-id*
4. **spanning-tree link-type point-to-point**
5. **end**
6. **clear spanning-tree detected-protocols**
7. **show spanning-tree vlan** *vlan_id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**spanning-tree mode** {**pvst** | **mst** | **rapid-pvst**}	Configures a spanning-tree mode. All stack members run the same version of spanning tree. • Select **pvst** to enable PVST+. • Select **mst** to enable MSTP. • Select **rapid-pvst** to enable rapid PVST+.
Step 3	**interface** *interface-id*	Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports, VLANs, and port channels. The VLAN ID range is 1 to 4094. The port-channel range is 1 to 48.
Step 4	**spanning-tree link-type point-to-point** **Example:** `Device(config-if)# spanning-tree link-type point-to-point`	Specifies that the link type for this port is point-to-point. If you connect this port (local port) to a remote port through a point-to-point link and the local port becomes a designated port, the device negotiates with the remote port and rapidly changes the local port to the forwarding state.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**clear spanning-tree detected-protocols** **Example:** `Device# clear spanning-tree detected-protocols`	If any port on the device is connected to a port on a legacy IEEE 802.1D device, this command restarts the protocol migration process on the entire device. This step is optional if the designated device detects that this device is running rapid PVST+.
Step 7	**show spanning-tree vlan** *vlan_id*	Verifies that STP is enabled.

What to do next

Caution Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

Caution We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

This example shows how to enable STP on VLAN 200:

```
Device# configure terminal
Device(config)# spanning-tree vlan 200

Device(config)# end

Device#
```

Note STP is disabled by default.

This example shows how to verify the configuration:

```
Device# show spanning-tree vlan 200

G0:VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     00d0.00b8.14c8
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32768
             Address     00d0.00b8.14c8
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg FWD 200000    128.196  P2p
Gi1/5            Back BLK 200000    128.197  P2p
Device#
```

Note You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

Multiple Spanning Tree protocol

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning tree instances required to support a large number of VLANs. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

Per-VLAN Spanning Tree+

Per-VLAN Spanning Tree+ (PVST+) is an extension of the PVST standard. Per-VLAN Spanning Tree+ (PVST+) allows interoperability between CST and PVST in Cisco switches and supports the IEEE 802.1Q standard.

Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. This feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces.

When you configure this feature, it ensures that the rate does not exceed the configured policer rate. When the traffic exceeds the configured rate, packets are dropped to control the traffic.

Enabling Per-Port Storm Control

To enable per-port traffic storm control, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **intterface** *type 0/slot/port*
4. **storm-control** {**unicast** | **broadcast** | **multicast**} **level** {*level_high*}{*level_low*}
5. **storm-control action** { **shutdown** | **trap**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 3	**intterface** *type 0/slot/port* **Example:** `Device#interface gigabitethernet 0/1/2`	Enters interface configuration mode, and enter the port to configure.

Step 4	**storm-control** {**unicast** | **broadcast** | **multicast**} **level** {*level_high*}{*level_low*}
	Example:
	• **Unicast control**
	`Device(config-if)#storm-control unicast level 70.00 50.00`
		Specifies the interface level unicast, broadcast, or multicast storm control suppression level as a percentage of the total bandwidth. Here, the bandwidth is dependent on the operational speed.
		Unicast: Configures the known and unknown unicast storm control.
		Broadcast: Configures broadcast storm control.

	Command or Action	Purpose

	• **Broadcast Control**
	`Device(config-if)#storm-control broadcast level 70.00 50.00`
	• **Multicast Control**
	`Device(config-if)#storm-control multicast level 70.00 50.00`
		Multicast: Configures multicast storm control.
		Level: Specifies the threshold levels for broadcast, multicast, or unicast traffic.

Step 5	**storm-control action** { **shutdown** | **trap**}
	Example:
	`Router(config-if)#storm control action trap`
		Selects the **shutdown** keyword to disable the port during a storm.
		The traffic is blocked when it exceeds the threshold specified by configuration level, irrespective of the shutdown or SNMP trap being enabled or disabled.
		• **shutdown**: The interface enters err-disable state when traffic exceeds the threshold specified by configuration level.
		• **trap**: The interface sends an SNMP trap event when traffic exceeds the threshold specified by configuration level.

Step 6	**end**
	Example:
	`Router(config-if)#end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Disabling Per-Port Storm-Control

To disabe per-port traffic storm control, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **intterface** *type 0/slot/port*
4. **storm-control action** { {**unicast** | **broadcast** | **multicast**} **level** | **shutdown** }
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device#configure terminal`	
Step 3	**intterface** *type 0/slot/port* **Example:** `Device#interface gigabitethernet 0/1/2`	Enters interface configuration mode, and enter the port to configure.

Step 4	**storm-control action** { {**unicast** | **broadcast** | **multicast**} **level** | **shutdown** }
	Example:
	`Router(config-if)#no storm-control action shutdown`
		Disables per-port storm control or the specified storm control action.

Step 5	**end**
	Example:
	`Router(config-if)#end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Change of Authorization

Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated.

Identity-Based Networking Services supports change of authorization (CoA) commands for session query, reauthentication, and termination, port bounce and port shutdown, and service template activation and deactivation.

Change of Authorization-Reauthentication Procedure

Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. The main steps in this procedure are:

- Authentication
- Posture Assessment
- CoA Re-Authentication
- Network Access Authorization

When a policy changes for a user or user group in AAA, administrators can send RADIUS CoA packets from the AAA server, such as a Cisco Identity Secure Engine (ISE) to reinitialize authentication and apply the new policy. This section provides an overview of the RADIUS interface including available primitives and how they are used during a CoA.

The RADIUS CoA provides a mechanism to change the attributes of an AAA session after it is authenticated. When policy changed on user or user group in RADIUS server, administrators can initiate RADIUS CoA process from RADIUS server to re-authenticate or re-authorize new policy

By default, the RADIUS interface is enabled on the device. However, some basic configuration is required for the following attributes:

- Security and Password
- Accounting

After posture assessment is succeessful, full network access is pushed down to the device for specific client through CoA re-authentication command based on its compliance state derived from last assessment. It is optional to enforce downloadable ACLs with Permit-ALL or limited access to certain resources to corresponding clients. Per-session CoA requests are supported for session identification, session termination, host reauthentication, port shutdown, and port bounce. This model comprises one request (CoA-Request) and two possible response codes:

- CoA acknowledgement (ACK) [CoA-ACK]
- CoA nonacknowledgement (NAK) [CoA-NAK]

For more information on Change of Authorization, see the Change of Authorization chapter.

Configuring LAN MACSec Uplink

To configure the LAN MACSec Uplink on the interface, perform these steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **key-server priority** *key-server-priority*
5. **macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**}
6. **confidentiality-offset**{ **0** | **30** | **50**}
7. **key chain** *key-chain-name* **macsec**
8. **key** *hex-string*
9. **cryptographic-algorithm** [**aes-128-cmac** | **aes-256-cmac**]
10. **key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
11. **interface** *type number*
12. **switchport**
13. **switchport mode trunk**
14. **mka policy** *policy-name*
15. **mka pre-shared-key key-chain** *key-chain-name*
16. **macsec network-link**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
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.

	Command or Action	Purpose
Step 3	**mka policy** *policy-name* **Example:** `Device(config)# mka policy MKAPolicy`	Configures an MKA policy.
Step 4	**key-server priority** *key-server-priority* **Example:** `Device(config-mka-policy)# key-server priority 200`	(Optional) Configures MKA key server priority.

Step 5	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**}
	Example:
	`Device(config-mka-policy)# macsec-cipher-suite gcm-aes-256`
		(Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order.

Step 6	**confidentiality-offset**{ **0** | **30** | **50**}
	Example:
	`Device(config-mka-policy)# confidentiality-offset 30`
		(Optional) Configures confidentiality offset for MACsec operation.

Step 7	**key chain** *key-chain-name* **macsec** **Example:** `Device(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode
Step 8	**key** *hex-string* **Example:** `Device(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode. **Note** From Cisco IOS XE Everest Release 16.6.1 onwards, the Connectivity Association Key name (CKN) uses exactly the same string, which is configured as the hex-string for the key. For more information about this behavior change, see the section titled "MKA-PSK: CKN Behavior Change" after this task.
Step 9	**cryptographic-algorithm** [**aes-128-cmac** | **aes-256-cmac**] **Example:** `Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac`	Set cryptographic authentication algorithm.

Step 10	**key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
	Example:
	`Device(config-keychain-key)# key-string 0 pwd`
		Sets the password for a key string.

Step 11	**interface** *type number* **Example:**	Enters the configuration mode for the specified interface on the router.

	Command or Action	Purpose
	`Device(config-if)# interface TenGigabitEthernet0/1/3`	
Step 12	**switchport** **Example:** `Device(config-if)# switchport`	Converts the port from the Layer 2 interface to the Layer 3 interface and makes it a router interface rather than a switch port.
Step 13	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Configures the interface as a trunk port.
Step 14	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy.
Step 15	**mka pre-shared-key key-chain** *key-chain-name* **Example:** `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	Configures an MKA pre-shared-key key-chain keychain1 **Note** The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.
Step 16	**macsec network-link** **Example:** `Device(config-if)# macsec network link`	Sets the IP address and subnet mask for the specified interface.

Step 17	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.
		Note The MKA policy does not process confidentiality offset for XPN ciphers. Therefore when both XPN and non-XPN ciphers are configured in an MKA policy alongwith confidentiality offset, the confidentiality offset is ignored for XPN ciphers. It is therefore strongly recommended to use your discretion while using configuring a MKA policy with XPN or non-XPN ciphers.

Layer 3 Mode Features

This section includes the following Layer 3 features:

PLIM

A physical layer interface module (PLIM) provides the packet interfaces for the routing system. Optics modules on the PLIM contain ports to which fiber-optic cables are connected. User data is received and transmitted through the PLIM ports.

Configuring PLIM

To configure PLIM, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **plim qos input map ipv4v6 qos-value-based**
5. **plim qos input mapipv4v6 qos-value 0 - 63 queue strict-priority**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router#configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config-if)# interface gigabitethernet 0/1/7`	Specifies the PLIM interface on which you want to enable multicast routing, and enters interface configuration mode.
Step 4	**plim qos input map ipv4v6 qos-value-based** **Example:** `Device(config-if)# plim qos input map ipv4v6 qos-value-based`	Attaches the ingress classification class-map template with the specified interface.
Step 5	**plim qos input mapipv4v6 qos-value 0 - 63 queue strict-priority** **Example:** `Device(config-if)# plim qos input map ipv4v6 qos-value 0 - 63 queue strict-priority`	Sets a priority queue on Gigabit Ethernet interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits configuration mode and returns the device to global configuration mode.

WAN MACSec

The WAN MACsec and MKA feature introduces MACsec support on WAN, uplink support, Pre-shared key support for the Macsec Key Agreement protocol (MKA) and Certificate-based MACsec Encryption.

Note The WAN MACSec is supported only on the last two flex ports of all the Cisco Catalyst C-NIM-4X, C-NIM-8M, and C-NIM-8T modules.

Configuring MACsec on the Interface

To configure the MACsec on the interface, perform these steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **key-server priority** *key-server-priority*
5. **macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**}
6. **confidentiality-offset** { **0** | 30 | **50**}
7. **key chain** *key-chain-name* [**macsec**]
8. **key** *hex-string*
9. **cryptographic-algorithm** {**aes-128-cmac** | **aes-256-cmac**}
10. **key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
11. **interface** *type number*
12. **no switchport**
13. **ip address** *address mask*
14. **mka policy** *policy-name*
15. **mka pre-shared-key key-chain** *key-chain-name*
16. **macsec**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
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	**mka policy** *policy-name* **Example:** `Device(config)# mka policy MKAPolicy`	Configures an MKA policy.

	Command or Action	Purpose
Step 4	**key-server priority** *key-server-priority* **Example:** `Device(config-mka-policy)# key-server priority 200`	(Optional) Configures MKA key server priority.

Step 5	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256**}
	Example:
	`Device(config-mka-policy)# macsec-cipher-suite gcm-aes-256`
		(Optional) Configures cipher suite(s) for secure association key (SAK) derivation. Each of the cipher suite options can be repeated only once, but they can be used in any order.

Step 6	**confidentiality-offset** { **0** | 30 | **50**}
	Example:
	`Device(config-mka-policy)# confidentiality-offset 30`
		(Optional) Configures confidentiality offset for MACsec operation.

Step 7	**key chain** *key-chain-name* [**macsec**] **Example:** `Device(config)# Key chain keychain1 macsec`	Configures a key chain and enters keychain configuration mode
Step 8	**key** *hex-string* **Example:** `Device(config-keychain)# key 9ABCD`	Configures a key and enters keychain key configuration mode. **Note** From Cisco IOS XE Everest Release 16.6.1 onwards, the Connectivity Association Key name (CKN) uses exactly the same string, which is configured as the hex-string for the key. For more information about this behavior change, see the section titled "MKA-PSK: CKN Behavior Change" after this task.

Step 9	**cryptographic-algorithm** {**aes-128-cmac** | **aes-256-cmac**}
	Example:
	`Device(config-keychain-key)# cryptographic-algorithm gcm-aes-128`
		Set cryptographic authentication algorithm.

Step 10	**key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*}
	Example:
	`Device(config-keychain-key)# key-string 0 pwd`
		Sets the password for a key string.

Step 11	**interface** *type number* **Example:** `Device(config-if)# interface TenGigabitEthernet 0/1/3`	Enters the configuration mode for the specified interface on the router.

	Command or Action	Purpose
Step 12	**no switchport** **Example:** `Device(config-if)# no switchport`	Converts the port from the Layer 2 interface to the Layer 3 interface and makes it a router interface rather than a switch port.
Step 13	**ip address** *address mask* **Example:** `Device(config-if)# ip address 101.1.1.1 255.255.255.0`	Sets the IP address and subnet mask for the specified interface.
Step 14	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy.
Step 15	**mka pre-shared-key key-chain** *key-chain-name* **Example:** `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	Configures an MKA pre-shared-key key-chain keychain1 **Note** The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.
Step 16	**macsec** **Example:** `Device(config-if)# macsec`	Enables the MACsec under the interface.

Step 17	**end**
	Example:
	`Device(config-mka-policy)# end`
		Returns to privileged EXEC mode.

Configuring MACsec and MKA on Interfaces

To configure MACsec and MKA on an interface, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mka policy** *policy-name*
5. **mka pre-shared-key key-chain** *key-chain-name*
6. **macsec**
7. **macsec replay-protection window-size** *window-size number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
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 0/0/0`	Enters interface configuration mode.
Step 4	**mka policy** *policy-name* **Example:** `Device(config-if)# mka policy MKAPolicy`	Configures an MKA policy
Step 5	**mka pre-shared-key key-chain** *key-chain-name* **Example:** `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	Configures an MKA pre-shared-key key-chain keychain1 **Note** The MKA Pre-shared key can be configured on either physical interface or subinterfaces and not on both physical and subinterfaces.
Step 6	**macsec** **Example:** `Device(config-if)# macsec`	Configures MACsec for the EAPOL frame ethernet type.
Step 7	**macsec replay-protection window-size** *window-size number* **Example:** `Device(config-if)# macsec replay-protection window-size 10`	Sets the MACsec window size for replay protection.

Step 8	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Information About SGT Inline Tagging

Each security group in a Cisco TrustSec domain is assigned a unique 16 bit tag called the Security Group Tag (SGT). The SGT is a single label indicating the privileges of the source within the entire network. It is in turn propagated between network hops allowing any intermediary devices (switches, routers) to enforce polices based on the identity tag.

Cisco TrustSec-capable devices have built-in hardware capabilities than can send and receive packets with SGT embedded in the MAC (L2) layer. This feature is called Layer 2 (L2)-SGT Imposition. It allows ethernet interfaces on the device to be enabled for L2-SGT imposition so that the device can insert an SGT in the packet to be carried to its next hop ethernet neighbor. SGT-over-Ethernet is a method of hop-by-hop propagation of SGT embedded in clear-text (unencrypted) ethernet packets. The inline identity propagation is scalable, provides near line-rate performance and avoids control plane overhead.

The Cisco TrustSec with SGT Exchange Protocol V4 (SXPv4) feature supports Cisco TrustSec metadata-based L2-SGT. When a packet enters a Cisco TrustSec-enabled interface, the IP-SGT mapping database (with dynamic entries built by SXP and/or static entries built by configuration commands) is analyzed to learn the SGT corresponding to the source IP address of the packet, which is then inserted into the packet and carried throughout the network within the Cisco TrustSec header.

As the tag represents the group of the source, the tag is also referred to as the Source Group Tag (SGT). At the egress edge of the network, the group assigned to the packet's destination becomes known. At this point, access control can be applied. With Cisco TrustSec, access control policies are defined between the security groups and are referred to as Security Group Access Control Lists (SGACL). From the view of any given packet, SGACL is simply being sourced from a security group and destined for another security group.

The SGT tag received in a packet from a trusted interface is propagated to the network, and is also be used for Identity firewall classification. When IPsec support is added, the received SGT tag is shared with IPSec for SGT tagging.

A network device at the ingress of Cisco TrustSec cloud needs to determine the SGT of the packet entering the Cisco TrustSec cloud so that it can tag the packet with that SGT when it forwards it into the Cisco TrustSec cloud. The SGT of a packet can be determined with these methods:

- SGT field on Cisco TrustSec header: If a packet is coming from a trusted peer device, it is assumed that the Cisco TrustSec header carries the correct SGT field. This situation applies to a network that is not the first network device in the Cisco TrustSec cloud for the packet.
- SGT lookup based on source IP address: In some cases, the administrator may manually configure a policy to decide the SGT of a packet based upon the source IP address. An IP address to SGT table can also be populated by the SXP protocol.

L2 Inline Tagging is supported for IPv6 multicast traffic with unicast source IPv6 addresses.

SGT Inline Tagging on a NAT Enabled Device

The following scenarios explain how SGT is determined for a packet that flows from a primary device, which has Network Address Translation (NAT) enabled on both ingress and egress ports, to a secondary device:

Note All ports that are used for the flow must have **CTS manual** and trusted configured on both devices.

- If inline tagging is enabled between both devices and SGT tag is not changed with CLI:

  In this case, on the primary device Cisco TrustSec is enforced on the SGT tag corresponding to the packet's source IP. The same SGT tag is tagged to the NAT IP. On the secondary device, Cisco TrustSec is enforced on the SGT tag corresponding to the packet's source IP also.

  For example, a packet is received on the primary device with a source IP 192.0.2.5 and SGT tag 133. Cisco TrustSec is enforced for the SGT tag 133 on the primary device. After NAT translation the packet's IP changes to 198.51.100.10 and tagged to the SGT tag 133. On the secondary device, the packet is received with IP address 198.51.100.10 and SGT tag 133. Cisco TrustSec is enforced with SGT tag 133 on the secondary device.

- If inline tagging is enabled between both devices and SGT tag is changed with CLI:

  In this case, on the primary device Cisco TrustSec is enforced on the SGT tag corresponding to the packet's source IP. The SGT tag is changed by CLI but the SGT tag corresponding to the packets's source IP is tagged to the packet's NAT IP. On the secondary device, Cisco TrustSec is enforced on the SGT tag corresponding to the packet's source IP also.

  For example, a packet is received on the primary device with a source IP 192.0.2.5 and SGT tag 133. Cisco TrustSec is enforced for the SGT tag 133 on the primary device. The SGT tag is changed to 200 with CLI. After NAT translation the packet’s IP changes to 198.51.100.10 but tagged to the SGT tag 133. On the secondary device, the packet is received with IP address 198.51.100.10 and SGT tag 133. Cisco TrustSec is enforced on the SGT tag 133 on the secondary device.

- If inline tagging is disabled (SGT is populated through SXP protocol on the secondary device) and SGT tag is changed with CLI:

  In this case, on the primary device Cisco TrustSec is enforced on the SGT tag corresponding to the packet's source IP. The SGT to Post Nat IP is defined through CLI and is learnt on the primary device. On the secondary device, Cisco TrustSec is enforced on the SGT tag corresponding to the NAT IP, if there is no direct Cisco TrustSec link between primary and secondary device and IP to SGT bindings are learnt through SXP in secondary device.

  For example, a packet is received on the primary device with a source IP 192.0.2.5 and SGT tag 133. After NAT translation the source IP changes to 198.51.100.10, for which the SGT is defined through CLI as 200. Cisco TrustSec is enforced for the SGT tag 133 on the primary device. On the secondary device, IP to SGT binding is received through SXP and Cisco TrustSec is enforced on the SGT tag 200 on the secondary device.

Configuring SGT Inline Tagging

Procedure

	Command or Action	Purpose
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** {**gigabitethernet port** | **vlan number**}
	Example:
	`Device(config)# interface gigabitethernet 0/1/0`
		Configures the interface on which Cisco TrustSec SGT authorization and forwarding is enabled, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**cts manual** **Example:** `Device(config-if)# cts manual`	Enables Cisco TrustSec SGT authorization and forwarding on the interface, and enters Cisco TrustSec manual interface configuration mode.
Step 5	**propagate sgt** **Example:** `Device(config-if-cts-manual)# propagate sgt`	Enables Cisco TrustSec SGT propagation on an interface. **Note** Use this command in situations where the peer device is not capable of receiving SGT over Ethernet packets (that is, when a peer device does not support Cisco Ethertype CMD 0x8909 frame format).
Step 6	**policy static sgt** *tag* [**trusted**] **Example:** `Device(config-if-cts-manual)# policy static sgt 77 trusted`	Configures a static SGT ingress policy on the interface and defines the trustworthiness of an SGT received on the interface. **Note** The **trusted** keyword indicates that the interface is trustworthy for Cisco TrustSec. The SGT value received in the Ethernet packet on this interface is trusted and will be used by the device for any SG-aware policy enforcement or for the purpose of egress-tagging.
Step 7	**exit** **Example:** `Device(config-if-cts-manual)# exit`	Exits Cisco TrustSec manual interface configuration mode and enters interface configuration mode.
Step 8	**dot1x pae authenticator** **Example:** `Device(config-if)# dot1x pae authenticator`	Enables 802.1x authentication on the port.
Step 9	**dot1x authenticator eap profile** *name* **Example:** `Device(config-if)# dot1x authenticator eap profile md5`	Specifies the Extensible Authentication Protocol (EAP) profile.

Step 10	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Example: Configuring SGT Static Inline Tagging

This example shows how to enable an interface on the device for Layer 3 SGT tagging or imposition and defines whether the interface is trusted for Cisco TrustSec

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1/7
Device(config-if)# cts manual
Device(config-if-cts-manual)# propagate sgt
Device(config-if-cts-manual)# policy static sgt 77 trusted
```

Related Documents

Related Topic	Document Title
I Installing the Cisco Catalyst C-NIM-4X or C-NIM-8T Module	Need to add the link.

Conventions

This document uses the following conventions.

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What’s New in Cisco Product Documentation at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html .

Subscribe to What’s New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.