



Security Configuration Guide, Cisco Catalyst IE9300 Rugged Series Switches

First Published: 2022-05-20

Last Modified: 2024-11-30

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Communications, services, and additional information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit [Cisco DevNet](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

Cisco Bug Search Tool

[Cisco Bug Search Tool](#) (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Bias Free Language

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.



CONTENTS

Full Cisco Trademarks with Software License iii

Communications, services, and additional information iv

Cisco Bug Search Tool iv

Documentation feedback iv

Bias Free Language v

CHAPTER 1

Configuring RADIUS 1

Prerequisites for Configuring RADIUS 1

Restrictions for Configuring RADIUS 2

Information about RADIUS 2

RADIUS and Switch Access 2

RADIUS Overview 3

RADIUS Operation 3

RADIUS Change of Authorization 4

Change-of-Authorization Requests 6

CoA Request Response Code 7

CoA Request Commands 8

Default RADIUS Configuration 10

RADIUS Server Host 10

RADIUS Login Authentication 11

AAA Server Groups 11

AAA Authorization 12

RADIUS Accounting	12
Vendor-Specific RADIUS Attributes	12
Vendor-Proprietary RADIUS Server Communication	23
DSCP marking for RADIUS packets	23
Configuring RADIUS	24
Identifying the RADIUS Server Host	24
Configuring RADIUS Login Authentication	26
Defining AAA Server Groups	28
Configuring RADIUS Authorization for User Privileged Access and Network Services	29
Starting RADIUS Accounting	30
Configuring Settings for All RADIUS Servers	31
Configuring the Device to Use Vendor-Specific RADIUS Attributes	32
Configuring the Device for Vendor-Proprietary RADIUS Server Communication	33
Configuring DSCP Marking on a RADIUS Server	34
Configuring the Source Interface and DSCP Marking on RADIUS Server Group	36
Configuring CoA on the Device	37
Monitoring CoA Functionality	39

CHAPTER 2

MACsec Encryption	41
MACsec Encryption	41
MACsec Key Agreement	42
MKA Policies	42
Definition of Policy-Map Actions	43
Virtual Ports	43
MKA Statistics	43
Key Lifetime and Hitless Key Rollover	44
Replay Protection Window Size	44
MACsec, MKA and 802.1x Host Modes	44
Single-Host Mode	44
Multiple-Host Mode	45
Multiple-Domain Mode	45
MACsec MKA using Certificate-based MACsec	45
Prerequisites for MACsec MKA Using Certificate-based MACsec	46
Switch-to-Switch MKA MACsec Must Secure Policy	46

MKA/MACsec for Port Channel	46
MACsec Cipher Announcement	47
Limitations for MACsec Cipher Announcement	47
How to Configure MACsec Encryption	47
Prerequisites for MACsec Encryption	47
Restrictions for MACsec Encryption	48
Recommendations for MACsec Encryption	48
MKA and MACsec Configuration	49
Configure an MKA Policy	49
Configure Switch-to-host MACsec Encryption	50
Configure MACsec MKA using PSK	53
Configure MACsec MKA on an Interface using PSK	55
Configuring MACsec MKA Using Certificate-based MACsec	56
Generate Key Pairs	56
Configure Enrollment using SCEP	57
Configure Enrollment Manually	59
Enable 802.1x Authentication and Configure AAA	62
Apply the 802.1x MKA MACsec Configuration on the Interfaces	64
Configure MKA/MACsec for Port Channel using PSK	66
Configure Port Channel Logical Interfaces for Layer 2 EtherChannels	68
Configure Port Channel Logical Interfaces for Layer 3 EtherChannels	69
Configuring MACsec Cipher Announcement	70
Configure an MKA Policy for Secure Announcement	70
Configure Secure Announcement Globally	71
Configure EAPoL Announcements on an Interface	71
Configuration Examples for MACsec Encryption	72
Example: Configuring MKA and MACsec	72
Examples: Configuring MACsec MKA Using PSK	73
Examples: Configuring MACsec MKA using Certificate-based MACsec	74
Examples: Configuring MACsec MKA for Port Channel using PSK	74
Examples: Configuring MACsec Cipher Announcement	81
Examples: Displaying MKA Information	84
Additional References for MACsec Encryption	90
Feature History for MACsec Encryption	91

CHAPTER 3**Network Edge Access Topology 93**

802.1x Supplicant and Authenticator Switches with Network Edge Access Topology 93

Guidelines and Limitations 95

Configure an Authenticator Switch with NEAT 95

Configure a Supplicant Switch with NEAT 97

Verifying Configuration 100

Feature History 101

CHAPTER 4**Layer 2 Network Address Translation 103**

Layer 2 Network Address Translation 103

Guidelines and Limitations 106

NAT Performance and Scalability 108

Configure Layer 2 NAT 108

Verify the Configuration 109

Basic Inside-to-Outside Communications: Example 111

Basic Inside-to-Outside Communications: Configuration 112

Duplicate IP Addresses Example 113

Duplicate IP Addresses Configuration: Switch A 114

Duplicate IP Addresses Configuration: Switch B 116



CHAPTER 1

Configuring RADIUS

- [Prerequisites for Configuring RADIUS, on page 1](#)
- [Restrictions for Configuring RADIUS, on page 2](#)
- [Information about RADIUS, on page 2](#)
- [Configuring RADIUS, on page 24](#)
- [Monitoring CoA Functionality, on page 39](#)

Prerequisites for Configuring RADIUS

This section lists the prerequisites for controlling device access with RADIUS.

General:

- RADIUS and Authentication, Authorization, and Accounting (AAA) must be enabled to use any of the configuration commands in this chapter.
- RADIUS is facilitated through AAA and can be enabled only through AAA commands.
- Use the **aaa new-model** global configuration command to enable AAA.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication.
- Use **line** and **interface** commands to enable the defined method lists to be used.
- At a minimum, you must identify the host or hosts that run the RADIUS server software and define the method lists for RADIUS authentication. You can optionally define method lists for RADIUS authorization and accounting.
- You should have access to and should configure a RADIUS server before configuring RADIUS features on your device.
- The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (Cisco ISE), Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.
- To use the Change-of-Authorization (CoA) interface, a session must already exist on the switch. CoA can be used to identify a session and enforce a disconnect request. The update affects only the specified session.

RADIUS operation:

- Users must first successfully complete RADIUS authentication before proceeding to RADIUS authorization, if it is enabled.
- For RADIUS over IPv6 configurations, users must enable IPv6 unicast routing by enabling the **ipv6 unicast-routing** command.

Restrictions for Configuring RADIUS

General:

- To prevent a lapse in security, you cannot configure RADIUS through a network management application.
- Radius and AAA servers can be configured to run only on the standard default ports:
 - 1812 and 1813
 - 1645 and 1646

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support AppleTalk Remote Access (ARA), NetBIOS Frame Control Protocol (NBFCP), NetWare Asynchronous Services Interface (NASI), or X.25 PAD connections.
- Switch-to-switch or router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

DSCP marking support for RADIUS packets:

- DSCP marking for authentication and accounting is not supported for private servers, fully qualified domain name (FQDN) servers and radsec servers.
- In the case of wired IEEE 802.1x authentication, when source port extension is not enabled, the default ports are in use. The DSCP marking is set to the default ports and all the requests will be marked with the same DSCP value.
- DSCP marking is not supported in the case of wireless IEEE 802.1x authentication, where the source port extension is enabled by default.

Information about RADIUS

RADIUS and Switch Access

This section describes how to enable and configure RADIUS. RADIUS provides detailed accounting information and flexible administrative control over the authentication and authorization processes.

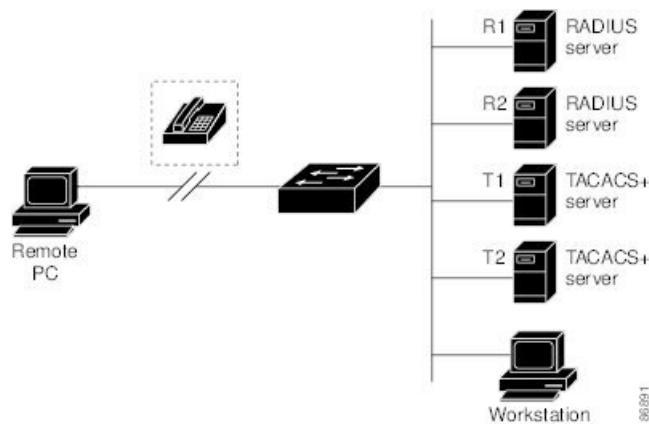
RADIUS Overview

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco devices. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

Use RADIUS in these network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a *smart card* access control system.
- Networks already using RADIUS. You can add a Cisco device containing a RADIUS client to the network. This might be the first step when you make a transition to a TACACS+ server. See the illustration: Transitioning from RADIUS to TACACS+ Services below.

Figure 1: Transitioning from RADIUS to TACACS+ Services



- Network in which the user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to the network through a protocol such as IEEE 802.1x. For more information about this protocol, see the chapter *Configuring IEEE 802.1x Port-Based Authentication*.
- Networks that require resource accounting. You can use RADIUS accounting independently of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, showing the amount of resources (such as time, packets, bytes, and so forth) used during the session. An Internet service provider might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

RADIUS Operation

When a user attempts to log in and authenticate to a device that is access controlled by a RADIUS server, these events occur:

1. The user is prompted to enter a username and password.

2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
 - ACCEPT—The user is authenticated.
 - REJECT—The user is either not authenticated and is prompted to re-enter the username and password, or access is denied.
 - CHALLENGE—A challenge requires additional data from the user.
 - CHALLENGE PASSWORD—A response requests the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for privileged EXEC or network authorization. The additional data included with the ACCEPT or REJECT packets includes these items:

- Telnet, SSH, rlogin, or privileged EXEC services
- Connection parameters, including the host or client IP address, access list, and user timeouts

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. 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 Secure Access Control Server (ACS) 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.

- Change-of-Authorization Requests
- CoA Request Response Code
- CoA Request Commands
- Session Reauthentication
- Stacking Guidelines for Session Termination

A standard RADIUS interface is typically used in a pulled model where the request originates from a network attached device and the response come from the queried servers. Cisco devices support the RADIUS CoA extensions defined in RFC 5176 that are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external AAA or policy servers.

Cisco devices supports these per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce

This feature is integrated with Cisco Secure Access Control Server (ACS) 5.1.

The RADIUS interface is enabled by default on Cisco devices. However, some basic configuration is required for the following attributes:

- Security and Password—refer to the “Preventing Unauthorized Access to Your Switch” section in this guide.
- Accounting—refer to the “Starting RADIUS Accounting” section in the Configuring Switch-Based Authentication chapter in this guide.

Cisco IOS XE software supports the RADIUS CoA extensions defined in RFC 5176 that are typically used in a push model to allow the dynamic reconfiguring of sessions from external AAA or policy servers. 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]

The request is initiated from a CoA client (typically a AAA or policy server) and directed to the device that acts as a listener.

The table below shows the RADIUS CoA commands and vendor-specific attributes (VSAs) supported by Identity-Based Networking Services. All CoA commands must include the session identifier between the device and the CoA client.

Table 1: RADIUS CoA Commands Supported by Identity-Based Networking Services

CoA Command	Cisco VSA
Activate service	Cisco:Avpair="subscriber:command=activate-service" Cisco:Avpair="subscriber:service-name=<service-name>" Cisco:Avpair="subscriber:precedence=<precedence-number>" Cisco:Avpair="subscriber:activation-mode=replace-all"
Deactivate service	Cisco:Avpair="subscriber:command=deactivate-service" Cisco:Avpair="subscriber:service-name=<service-name>"
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Session query	Cisco:Avpair="subscriber:command=session-query"
Session reauthenticate	Cisco:Avpair="subscriber:command=reauthenticate" Cisco:Avpair="subscriber:reauthenticate-type=last" or Cisco:Avpair="subscriber:reauthenticate-type=rerun"
Session terminate	This is a standard disconnect request and does not require a VSA.
Interface template	Cisco:AVpair="interface-template-name=<interfacetemplate>"

Change-of-Authorization Requests

Change of Authorization (CoA) requests, as described in RFC 5176, are used in a push model to allow for session identification, host reauthentication, and session termination. The model is comprised of one request (CoA-Request) and two possible response codes:

- CoA acknowledgment (ACK) [CoA-ACK]
- CoA non-acknowledgment (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the switch that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the switch for session termination.

This table shows the IETF attributes are supported for this feature.

Table 2: Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

This table shows the possible values for the Error-Cause attribute.

Table 3: Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension

Value	Explanation
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request response code can be used to convey a command to the switch.

The packet format for a CoA Request Response code as defined in RFC 5176 consists of the following fields: Code, Identifier, Length, Authenticator, and Attributes in the Type:Length:Value (TLV) format. The Attributes field is used to carry Cisco vendor-specific attributes (VSAs).

Session Identification

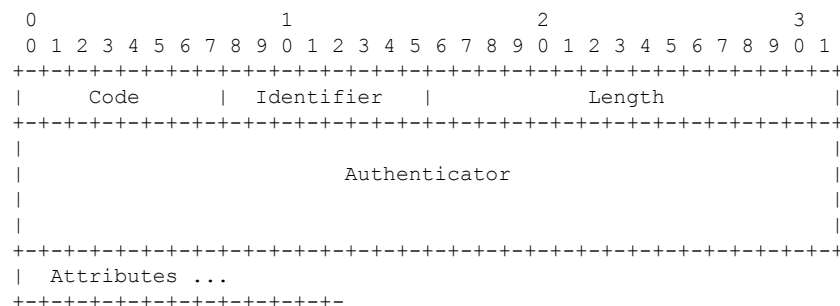
For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)
- Audit-Session-Id (Cisco VSA)
- Calling-Station-Id (IETF attribute #31 which contains the host MAC address)
- IPv6 Attributes, which can be one of the following:
 - Framed-IPv6-Prefix (IETF attribute #97) and Framed-Interface-Id (IETF attribute #96), which together create a full IPv6 address per RFC 3162
 - Framed-IPv6-Address
- Plain IP Address (IETF attribute #8)

Unless all session identification attributes included in the CoA message match the session, the switch returns a Disconnect-NAK or CoA-NAK with the “Invalid Attribute Value” error-code attribute.

If more than one session identification attribute is included in the message, all the attributes must match the session or the switch returns a Disconnect- negative acknowledgment (NAK) or CoA-NAK with the error code “Invalid Attribute Value.”

The packet format for a CoA Request code as defined in RFC 5176 consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format.



The attributes field is used to carry Cisco vendor-specific attributes (VSAs).

For CoA requests targeted at a particular enforcement policy, the device returns a CoA-NAK with the error code “Invalid Attribute Value” if any of the above session identification attributes are included in the message.

CoA ACK Response Code

If the authorization state is changed successfully, a positive acknowledgment (ACK) is sent. The attributes returned within CoA ACK will vary based on the CoA Request and are discussed in individual CoA Commands.

CoA NAK Response Code

A negative acknowledgment (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure. Use **show** commands to verify a successful CoA.

CoA Request Commands

Table 4: Supported CoA Commands

Command	Cisco VSA
1	
Reauthenticate host	Cisco:Avpair=“subscriber:command=reauthenticate”
Terminate session	This is a standard disconnect request that does not require a VSA.
Bounce host port	Cisco:Avpair=“subscriber:command=bounce-host-port”
Disable host port	Cisco:Avpair=“subscriber:command=disable-host-port”

¹ All CoA commands must include the session identifier between the device and the CoA client.

Session Reauthentication

The AAA server typically generates a session reauthentication request when a host with an unknown identity or posture joins the network and is associated with a restricted access authorization profile (such as a guest VLAN). A reauthentication request allows the host to be placed in the appropriate authorization group when its credentials are known.

To initiate session authentication, the AAA server sends a standard CoA-Request message which contains a Cisco VSA in this form: *Cisco:Avpair=“subscriber:command=reauthenticate”* and one or more session identification attributes.

The current session state determines the switch response to the message. If the session is currently authenticated by IEEE 802.1x, the switch responds by sending an EAPoL (Extensible Authentication Protocol over Lan) -RequestId message to the server.

If the session is currently authenticated by MAC authentication bypass (MAB), the switch sends an access-request to the server, passing the same identity attributes used for the initial successful authentication.

If session authentication is in progress when the switch receives the command, the switch terminates the process, and restarts the authentication sequence, starting with the method configured to be attempted first.

If the session is not yet authorized, or is authorized via guest VLAN, or critical VLAN, or similar policies, the reauthentication message restarts the access control methods, beginning with the method configured to be attempted first. The current authorization of the session is maintained until the reauthentication leads to a different authorization result.

Session Termination

There are three types of CoA requests that can trigger session termination. A CoA Disconnect-Request terminates the session, without disabling the host port. This command causes re-initialization of the authenticator state machine for the specified host, but does not restrict that host access to the network.

To restrict a host's access to the network, use a CoA Request with the `Cisco:Avpair="subscriber:command=disable-host-port"` VSA. This command is useful when a host is known to be causing problems on the network, and you need to immediately block network access for the host. When you want to restore network access on the port, re-enable it using a non-RADIUS mechanism.

When a device with no supplicant, such as a printer, needs to acquire a new IP address (for example, after a VLAN change), terminate the session on the host port with port-bounce (temporarily disable and then re-enable the port).

CoA Disconnect-Request

This command is a standard Disconnect-Request. If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK.

If the device fails-over to a standby device before returning a Disconnect-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the session is not found following re-sending, a Disconnect-ACK is sent with the "Session Context Not Found" error-code attribute.

CoA Request: Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. To restore network access on the port, reenable it using a non-RADIUS mechanism. This command is carried in a standard CoA-Request message that has this new vendor-specific attribute (VSA):

```
Cisco:Avpair="subscriber:command=disable-host-port"
```

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section. If the session cannot be located, the device returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device disables the hosting port and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is restarted on the new active device.



Note A Disconnect-Request failure following command re-sending could be the result of either a successful session termination before change-over (if the Disconnect-ACK was not sent) or a session termination by other means (for example, a link failure) that occurred after the original command was issued and before the standby device became active.

CoA Request: Bounce-Port

A RADIUS server CoA bounce port sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer) that does not have a mechanism to detect a change on this authentication port. The CoA bounce port is carried in a standard CoA-Request message that contains the following VSA:

```
Cisco:Avpair="subscriber:command=bounce-host-port"
```

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes. If the session cannot be located, the device returns a CoA-NAK message with the “Session Context Not Found” error-code attribute. If the session is located, the device disables the hosting port for a period of 10 seconds, re-enables it (port-bounce), and returns a CoA-ACK.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is re-started on the new active device.

Default RADIUS Configuration

RADIUS and AAA are disabled by default.

To prevent a lapse in security, you cannot configure RADIUS through a network management application. When enabled, RADIUS can authenticate users accessing the device through the CLI.

RADIUS Server Host

Device-to-RADIUS-server communication involves several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Key string
- Timeout period
- Retransmission value

You identify RADIUS security servers by their hostname or IP address, hostname and specific UDP port numbers, or their IP address and specific UDP port numbers. The combination of the IP address and the UDP

port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address.

If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the %RADIUS-4-RADIUS_DEAD message appears, and then the device tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order that they are configured.)

A RADIUS server and the device use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the device.

The timeout, retransmission, and encryption key values can be configured globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings.

RADIUS Login Authentication

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication to be performed and the sequence in which they are performed; it must be applied to a specific port before any of the defined authentication methods are performed. The only exception is the default method list. The default method list is automatically applied to all ports except those that have a named method list explicitly defined.

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

AAA Server Groups

You can configure the device to use AAA server groups to group existing server hosts for authentication. You select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list, which lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server if each entry has a unique identifier (the combination of the IP address and UDP port number), allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If you configure two different host entries on the same RADIUS server for the same service, (for example, accounting), the second configured host entry acts as a fail-over backup to the first one. If the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order in which they are configured.)

AAA Authorization

AAA authorization limits the services available to a user. When AAA authorization is enabled, the device uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

RADIUS Accounting

The AAA accounting feature tracks the services that users are using and the amount of network resources that they are consuming. When you enable AAA accounting, the device reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. You can then analyze the data for network management, client billing, or auditing.

Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the device and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option by using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named *cisco-avpair*. The value is a string with this format:

```
protocol : attribute sep value *
```

Protocol is a value of the Cisco protocol attribute for a particular type of authorization. *Attribute* and *value* are an appropriate attributevalue (AV) pair defined in the Cisco TACACS+ specification, and *sep* is = for mandatory attributes and is * for optional attributes. The full set of features available for TACACS+ authorization can then be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named IP address pools" feature to be activated during IP authorization (during PPP's Internet Protocol Control Protocol (IPCP) address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional:

```
cisco-avpair= "ip:addr-pool*first"
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, see RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

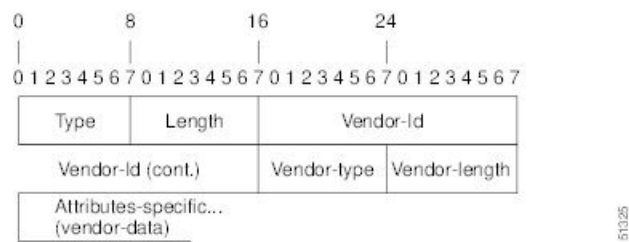
Attribute 26 contains the following three elements:

- Type

- Length
- String (also known as data)
 - Vendor-ID
 - Vendor-Type
 - Vendor-Length
 - Vendor-Data

The figure below shows the packet format for a VSA encapsulated “behind” attribute 26.

Figure 2: VSA Encapsulated Behind Attribute 26



Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

The table below describes significant fields listed in the Vendor-Specific RADIUS IETF Attributes table (second table below), which lists supported vendor-specific RADIUS attributes (IETF attribute 26).

Table 5: Vendor-Specific Attributes Table Field Descriptions

Field	Description
Number	All attributes listed in the following table are extensions of IETF attribute 26.
Vendor-Specific Command Codes	A defined code used to identify a particular vendor. Code 9 defines Cisco VSAs, 311 defines Microsoft VSAs, and 529 defines Ascend VSAs.
Sub-Type Number	The attribute ID number. This number is much like the ID numbers of IETF attributes, except it is a “second layer” ID number encapsulated behind attribute 26.
Attribute	The ASCII string name of the attribute.
Description	Description of the attribute.

Table 6: Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548)
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.
26	9	1	tunnel-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no.
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the mmoip aaa receive-id or the mmoip aaa send-id commands.
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.
26	9	8	Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.
26	9	9	Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.
26	9	10	Fax-Process-Abort-Flag	Indicates that the fax session was terminated or successful. True means that the session was terminated; false means that the session was successful.
26	9	11	Fax-Dsn-Address	Indicates the address to which DSNs will be sent.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	12	Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.
26	9	13	Fax-Mdn-Address	Indicates the address to which MDNs will be sent.
26	9	14	Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.
26	9	15	Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.
26	9	16	Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.
26	9	17	Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.
26	9	18	Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name.
26	9	19	Call-Type	Describes the type of fax activity: fax receive or fax send.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.
26	9	21	Abort-Cause	If the fax session terminates, indicates the system component that signaled the termination. Examples of system components that could trigger an termination are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.
H323 Attributes				
26	9	23	Remote-Gateway-ID (h323-remote-address)	Indicates the IP address of the remote gateway.
26	9	24	Connection-ID (h323-conf-id)	Identifies the conference ID.
26	9	25	Setup-Time (h323-setup-time)	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	Call-Origin (h323-call-origin)	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	Call-Type (h323-call-type)	Indicates call leg type. Possible values are telephony and VoIP .
26	9	28	Connect-Time (h323-connect-time)	Indicates the connection time for this call leg in UTC.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	29	Disconnect-Time (h323-disconnect-time)	Indicates the time this call leg was disconnected in UTC.
26	9	30	Disconnect-Cause (h323-disconnect-cause)	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	Voice-Quality (h323-voice-quality)	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	Gateway-ID (h323-gw-id)	Indicates the name of the underlying gateway.
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-name	<p>PPP name authentication. To apply for PAP, do not configure the ppp pap sent-name password command on the interface. For PAP, “preauth:send-name” and “preauth:send-secret” will be used as the PAP username and PAP password for outbound authentication. For CHAP, “preauth:send-name” will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in “preauth:send-name” in the challenge packet to the caller box.</p> <p>Note The send-name attribute has changed over time: Initially, it performed the functions now provided by both the send-name and remote-name attributes. Because the remote-name attribute has been added, the send-name attribute is restricted to its current behavior.</p>

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-secret	PPP password authentication. The vendor-specific attributes (VSAs) “preauth:send-name” and “preauth:send-secret” will be used as the PAP username and PAP password for outbound authentication. For a CHAP outbound case, both “preauth:send-name” and “preauth:send-secret” will be used in the response packet.
26	9	1	remote-name	Provides the name of the remote host for use in large-scale dial-out. Dialer checks that the large-scale dial-out remote name matches the authenticated name, to protect against accidental user RADIUS misconfiguration. (For example, dialing a valid phone number but connecting to the wrong device.)
Miscellaneous Attributes				

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	2	Cisco-NAS-Port	<p>Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting. To specify additional NAS-Port information in the form an Attribute-Value Pair (AVPair) string, use the radius-server vsa send global configuration command.</p> <p>Note This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.</p>
26	9	1	min-links	Sets the minimum number of links for MLP.
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the ip mobile secure host <addr> configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

Vendor-Proprietary RADIUS Server Communication

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the device and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS XE software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the device. You specify the RADIUS host and secret text string by using the **radius server** global configuration commands.

DSCP marking for RADIUS packets

Differentiated Services (DiffServ) is a Quality of Service (QoS) model that classifies and manages traffic for preferential handling over other traffic classes. DiffServ uses the 6-bit differentiated services code point (DSCP) setting in IP packets to mark traffic classes with relative priorities. Cisco IOS XE Software supports DSCP marking for RADIUS packets to allow faster authentication and accounting of RADIUS packets.

You can configure DSCP marking on the RADIUS server, RADIUS server group, and in global configuration mode. When DSCP marking is configured for the RADIUS server, server group, and in global configuration mode, the DSCP marking values that are entered on the RADIUS server take precedence.

- If there is no DSCP marking configuration on the RADIUS server, the DSCP marking values that are configured for the server group are applied to the RADIUS packets.
- If there is no DSCP marking configuration for the RADIUS server or RADIUS server group, the DSCP marking values that are configured in global configuration mode are applied to the RADIUS packets.

Configuring RADIUS

Identifying the RADIUS Server Host

To apply these settings globally to all RADIUS servers communicating with the device, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **key string**.

You can configure the device to use AAA server groups to group existing server hosts for authentication.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device.

Follow these steps to configure per-server RADIUS server communication.

Before you begin

If you configure both global and per-server functions (timeout, retransmission, and key commands) on the device, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.



Note Radius and AAA servers can be configured to run only on the standard default ports:

- 1812 and 1813
- 1645 and 1646

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server name*
4. **address** {**ipv4** | **ipv6**} *ip address* { **auth-port** *port number* | **acct-port** *port number*}
5. **key string**
6. **retransmit** *value*
7. **timeout** *seconds*
8. **end**

DETAILED STEPS

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	radius server <i>server name</i> Example: Device(config)# radius server rsim	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	address { ipv4 ipv6 } <i>ip address</i> { auth-port <i>port number</i> acct-port <i>port number</i> } Example: Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612	(Optional) Specifies the RADIUS server parameters. For auth-port <i>port-number</i> , specify the UDP destination port for authentication requests. The default is 1645. The range is 0 to 65536. For acct-port <i>port-number</i> , specify the UDP destination port for accounting requests. The default is 1646.
Step 5	key <i>string</i> Example: Device(config-radius-server)# key rad123	(Optional) For key <i>string</i> , specify the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server. Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius server command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 6	retransmit <i>value</i> Example: Device(config-radius-server)# retransmit 10	(Optional) Specifies the number of times a RADIUS request is resent when the server is not responding or responding slowly. The range is 1 to 100. This setting overrides the radius-server retransmit global configuration command setting.
Step 7	timeout <i>seconds</i> Example: Device(config-radius-server)# timeout 60	(Optional) Specifies the time interval that the device waits for the RADIUS server to reply before sending a request again. The range is 1 to 1000. This setting overrides the radius-server timeout global configuration command setting.
Step 8	end Example: Device(config-radius-server)# end	Exits RADIUS server configuration mode and enters privileged EXEC mode.

Configuring RADIUS Login Authentication

Follow these steps to configure RADIUS login authentication:

Before you begin

To secure the device for HTTP access by using AAA methods, you must configure the **ip http authentication aaa** global configuration command. By default, configuring AAA authentication does not secure the device for HTTP access by using AAA methods.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login {default | list-name} method1 [method2...]**
5. **line [console | tty | vty] line-number [ending-line-number]**
6. **login authentication {default | list-name}**
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	aaa new-model Example: Device(config)# aaa new-model	Enables AAA.
Step 4	aaa authentication login {default list-name} method1 [method2...] Example: Device(config)# aaa authentication login default local	Creates a login authentication method list. <ul style="list-style-type: none"> • To create a default list that is used when a named list is <i>not</i> specified in the login authentication command, use the default keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports. • For <i>list-name</i>, specify a character string to name the list you are creating.

	Command or Action	Purpose
		<ul style="list-style-type: none"> For <i>method1</i>..., specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. <p>Select one of these methods:</p> <ul style="list-style-type: none"> <i>enable</i>—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the enable password global configuration command. <i>group radius</i>—Use RADIUS authentication. Before you can use this authentication method, you must configure the RADIUS server. <i>line</i>—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the password password line configuration command. <i>local</i>—Use the local username database for authentication. You must enter username information in the database. Use the username name password global configuration command. <i>local-case</i>—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the username password global configuration command. <i>none</i>—Do not use any authentication for login.
Step 5	line [console tty vty] <i>line-number</i> [<i>ending-line-number</i>] Example: <pre>Device(config)# line 1 4</pre>	Enters line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 6	login authentication {default <i>list-name</i> } Example: <pre>Device(config-line)# login authentication default</pre>	<p>Applies the authentication list to a line or set of lines.</p> <ul style="list-style-type: none"> If you specify default, use the default list created with the aaa authentication login command. For <i>list-name</i>, specify the list created with the aaa authentication login command.
Step 7	end Example: <pre>Device(config-line)# end</pre>	Exits line configuration mode and enters privileged EXEC mode.

Defining AAA Server Groups

You use the **server** group server configuration command to associate a particular server with a defined group server. You can either identify the server by its IP address or identify multiple host instances or entries by using the optional **auth-port** and **acct-port** keywords.

Follow these steps to define AAA server groups:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *name*
4. **address** {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number*
5. **key** *string*
6. **end**

DETAILED STEPS

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	radius server <i>name</i> Example: Device(config)# radius server ISE	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode. The device also supports RADIUS for IPv6.
Step 4	address { ipv4 ipv6 } { <i>ip-address</i> <i>hostname</i> } auth-port <i>port-number</i> acct-port <i>port-number</i> Example: Device(config-radius-server)# address ipv4 10.1.1.1 auth-port 1645 acct-port 1646	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 5	key <i>string</i> Example: Device(config-radius-server)# key cisco123	Specifies the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.

	Command or Action	Purpose
Step 6	end Example: <pre>Device(config-radius-server)# end</pre>	Exits RADIUS server configuration mode and returns to privileged EXEC mode.

Configuring RADIUS Authorization for User Privileged Access and Network Services



Note Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

Follow these steps to configure RADIUS authorization for user privileged access and network services:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization network *authorization-list* radius**
4. **aaa authorization exec *authorization-list* radius**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	aaa authorization network <i>authorization-list</i> radius Example: <pre>Device(config)# aaa authorization network list1 radius</pre>	Configures the device for user RADIUS authorization for all network-related service requests.

	Command or Action	Purpose
Step 4	aaa authorization exec <i>authorization-list</i> radius Example: <pre>Device(config)# aaa authorization exec list1 radius</pre>	Configures the device for user RADIUS authorization if the user has privileged EXEC access. The exec keyword might return user profile information (such as autocommand information).
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

You can use the **aaa authorization** global configuration command with the **radius** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec radius local** command sets these authorization parameters:

- Use RADIUS for privileged EXEC access authorization if authentication was performed by using RADIUS.
- Use the local database if authentication was not performed by using RADIUS.

Starting RADIUS Accounting

Follow these steps to start RADIUS accounting:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network *accounting-list* start-stop radius**
4. **aaa accounting exec *accounting-list* start-stop radius**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
Step 3	aaa accounting network <i>accounting-list</i> start-stop radius Example: <pre>Device(config)# aaa accounting network accounting-list start-stop radius</pre>	Enables RADIUS accounting for all network-related service requests.
Step 4	aaa accounting exec <i>accounting-list</i> start-stop radius Example: <pre>Device(config)# aaa accounting exec acc-list start-stop radius</pre>	Enables RADIUS accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 5	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Settings for All RADIUS Servers

Beginning in privileged EXEC mode, follow these steps to configure settings for all RADIUS servers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server *server name***
4. **key *string***
5. **retransmit *retries***
6. **timeout *seconds***
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	radius server <i>server name</i> Example: Device(config) # radius server rsim	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	key <i>string</i> Example: Device(config-radius-server) # key your_server_key	Specifies the shared secret text string used between the switch and all RADIUS servers. Note The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 5	retransmit <i>retries</i> Example: Device(config-radius-server) # retransmit 5	Specifies the number of times the switch sends each RADIUS request to the server before giving up. The default is 3; the range is 1 to 1000.
Step 6	timeout <i>seconds</i> Example: Device(config-radius-server) # timeout 3	Specifies the number of seconds a switch waits for a reply to a RADIUS request before resending the request. The default is 5 seconds; the range is 1 to 1000.
Step 7	end Example: Device(config-radius-server) # end	Exits RADIUS server configuration mode and enters privileged EXEC mode.

Configuring the Device to Use Vendor-Specific RADIUS Attributes

Follow these steps to configure vendor-specific RADIUS attributes:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send [accounting | authentication]**
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	radius-server vsa send [accounting authentication] Example: Device(config)# radius-server vsa send accounting	Enables the device to recognize and use VSAs as defined by RADIUS IETF attribute 26. <ul style="list-style-type: none"> • (Optional) Use the accounting keyword to limit the set of recognized vendor-specific attributes to only accounting attributes. • (Optional) Use the authentication keyword to limit the set of recognized vendor-specific attributes to only authentication attributes. If you enter this command without keywords, both accounting and authentication vendor-specific attributes are used.
Step 4	end Example: Device(config)# end	Exits global configuration mode and enters privileged EXEC mode.

Configuring the Device for Vendor-Proprietary RADIUS Server Communication

Follow these steps to configure vendor-proprietary RADIUS server communication:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server name*
4. **address { ipv4 | ipv6 }** *ip address*
5. **non-standard**
6. **key** *string*
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	radius server <i>server name</i> Example: Device(config)# radius server rsim	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	address { ipv4 ipv6 } <i>ip address</i> Example: Device(config-radius-server)# address ipv4 172.24.25.10	(Optional) Specifies the IP address of the RADIUS server.
Step 5	non-standard Example: Device(config-radius-server)# non-standard	Identifies that the RADIUS server using a vendor-proprietary implementation of RADIUS.
Step 6	key <i>string</i> Example: Device(config-radius-server)# key rad123	Specifies the shared secret text string used between the device and the vendor-proprietary RADIUS server. The device and the RADIUS server use this text string to encrypt passwords and exchange responses.
Step 7	end Example: Device(config-radius-server)# end	Exits RADIUS server mode and enters privileged EXEC mode.

Configuring DSCP Marking on a RADIUS Server

Follow these steps to configure DSCP marking for authentication and accounting on a radius server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server_name*
4. **address** { **ipv4** | **ipv6** } *ip address* [**auth-port** *auth_port_number* **acct-port** *acct_port_number*]
5. **dscp** { **acct** *dscp_acct_value* | **auth** *dscp_auth_value* }
6. **key** *string*
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	radius server <i>server_name</i> Example: Device(config)# radius server <i>rsim</i>	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	address { ipv4 ipv6 } <i>ip address</i> [auth-port <i>auth_port_number</i> acct-port <i>acct_port_number</i>] Example: Device(config-radius-server)# address ipv4 10.1.1.1 auth-port 1645 acct-port 1646	(Optional) Specifies the IP address of the RADIUS server. <ul style="list-style-type: none"> • auth-port configures the port value for radius authentication server. The default value is 1812. • acct-port configures the port value for radius accounting server. The default value is 1813.
Step 5	dscp { acct <i>dscp_acct_value</i> auth <i>dscp_auth_value</i> } Example: Device(config-radius-server)# dscp auth 10 acct 20	Configures DSCP marking for authentication and accounting on the radius server. <ul style="list-style-type: none"> • acct configures radius DSCP marking value for accounting. The valid range is from 1 to 63. The default value is 0. • auth configures radius DSCP marking value for authentication. The valid range is from 1 to 63. The default value is 0.

	Command or Action	Purpose
Step 6	key <i>string</i> Example: <pre>Device(config-radius-server) # key rad123</pre>	Specifies the shared secret text string used between the device and the vendor-proprietary RADIUS server. The device and the RADIUS server use this text string to encrypt passwords and exchange responses.
Step 7	end Example: <pre>Device(config-radius-server) # end</pre>	Exits RADIUS server mode and enters privileged EXEC mode.

Configuring the Source Interface and DSCP Marking on RADIUS Server Group

Follow these steps to configure the source interface and DSCP marking for authentication and accounting on radius server groups:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group_name*
4. **server name** *name*
5. **{ip | ipv6} radius source-interface** *type number*
6. **dscp** { *acct dscp_acct_value* | *auth dscp_auth_value* }
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	aaa group server radius <i>group_name</i> Example: <pre>Device(config) # aaa group server radius abc</pre>	Defines the RADIUS server group configuration and enters RADIUS server group configuration mode.

	Command or Action	Purpose
Step 4	server name <i>name</i> Example: <pre>Device(config-sg-radius)# server name serv1</pre>	Associates the RADIUS server to the server group.
Step 5	{ip ipv6} radius source-interface <i>type number</i> Example: <pre>Device(config-sg-radius)# ipv6 radius source-interface ethernet 0/0</pre>	Specifies an interface to use for the source address in RADIUS server.
Step 6	dscp { acct <i>dscp_acct_value</i> auth <i>dscp_auth_value</i> } Example: <pre>Device(config-sg-radius)# dscp auth 10 acct 20</pre>	Configures DSCP marking for authentication and accounting on the radius server group. <ul style="list-style-type: none"> • acct configures radius DSCP marking value for accounting. The valid range is from 1 to 63. The default value is 0. • auth configures radius DSCP marking value for authentication. The valid range is from 1 to 63. The default value is 0.
Step 7	end Example: <pre>Device(config-radius-server)# end</pre>	Exits RADIUS server mode and enters privileged EXEC mode.

Configuring CoA on the Device

Follow these steps to configure CoA on a device. This procedure is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*]
6. **server-key** [0 | 7] *string*
7. **port** *port-number*
8. **auth-type** {**any** | **all** | **session-key**}
9. **ignore server-key**
10. **exit**
11. **authentication command bounce-port ignore**

12. authentication command disable-port ignore
13. end

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	aaa new-model Example: Device(config)# aaa new-model	Enables AAA.
Step 4	aaa server radius dynamic-author Example: Device(config)# aaa server radius dynamic-author	Configures the device as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server, and enters dynamic authorization local server configuration mode.
Step 5	client {ip-address name} [vrf vrfname] [server-key string] Example: Device(config-locsvr-da-radius)# client client1 vrf vrf1	Specifies a RADIUS client from which a device will accept CoA and disconnect requests.
Step 6	server-key [0 7] string Example: Device(config-locsvr-da-radius)# server-key your_server_key	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 7	port port-number Example: Device(config-locsvr-da-radius)# port 25	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients.

	Command or Action	Purpose
Step 8	auth-type {any all session-key} Example: <pre>Device(config-locsvr-da-radius)# auth-type any</pre>	<p>Specifies the type of authorization the device uses for RADIUS clients.</p> <p>The client must match all the configured attributes for authorization.</p>
Step 9	ignore server-key Example: <pre>Device(config-locsvr-da-radius)# ignore server-key</pre>	(Optional) Configures the device to ignore the server-key.
Step 10	exit Example: <pre>Device(config-locsvr-da-radius)# exit</pre>	Exits dynamic authorization local server configuration mode and returns to global configuration mode.
Step 11	authentication command bounce-port ignore Example: <pre>Device(config)# authentication command bounce-port ignore</pre>	(Optional) Configures the device to ignore a CoA request to temporarily disable the port hosting a session. The purpose of temporarily disabling the port is to trigger a DHCP renegotiation from the host when a VLAN change occurs and there is no supplicant on the endpoint to detect the change.
Step 12	authentication command disable-port ignore Example: <pre>Device(config)# authentication command disable-port ignore</pre>	<p>(Optional) Configures the device to ignore a nonstandard command requesting that the port hosting a session be administratively shut down. Shutting down the port results in termination of the session.</p> <p>Use standard CLI or SNMP commands to re-enable the port.</p>
Step 13	end Example: <pre>Device(config)# end</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring CoA Functionality

Table 7: Privileged EXEC show Commands

Command	Purpose
show aaa attributes protocol radius	Displays AAA attributes of RADIUS commands.

Table 8: Global Troubleshooting Commands

Command	Purpose
debug radius	Displays information for troubleshooting RADIUS.
debug aaa coa	Displays information for troubleshooting CoA processing.
debug aaa pod	Displays information for troubleshooting POD packets.
debug aaa subsys	Displays information for troubleshooting POD packets.



CHAPTER 2

MACsec Encryption

- [MACsec Encryption, on page 41](#)
- [MACsec Key Agreement, on page 42](#)
- [MACsec MKA using Certificate-based MACsec, on page 45](#)
- [Switch-to-Switch MKA MACsec Must Secure Policy, on page 46](#)
- [MKA/MACsec for Port Channel, on page 46](#)
- [MACsec Cipher Announcement, on page 47](#)
- [How to Configure MACsec Encryption, on page 47](#)
- [Additional References for MACsec Encryption, on page 90](#)
- [Feature History for MACsec Encryption, on page 91](#)

MACsec Encryption

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Catalyst switches support 802.1AE encryption with MACsec Key Agreement (MKA) on switch-to-host links for encryption between the switch and host device. The switch also supports MACsec encryption for switch-to-switch (inter-network device) security using MKA-based key exchange protocol.



Note When switch-to-switch MACSec is enabled, all traffic is encrypted, except the EAP-over-LAN (EAPOL) packets.

Link layer security can include both packet authentication between switches and MACsec encryption between switches (encryption is optional).

Table 9: MACsec Support on Switch Ports

Connections	MACsec support
Switch-to-host	MACsec MKA encryption
Switch-to-switch	MACsec MKA encryption

Cisco TrustSec is meant only for switch-to-switch links and is not supported on switch ports connected to end hosts, such as PCs or IP phones. MKA is supported on switch-to-host facing links as well as switch-to-switch links. Host-facing links typically use flexible authentication ordering for handling

heterogeneous devices with or without IEEE 802.1x, and can optionally use MKA-based MACsec encryption. Network Edge Access Topology (NEAT) is used for compact switches to extend security outside the wiring closet.

MACsec Key Agreement

MACsec, defined in 802.1AE, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. MKA and MACsec are implemented after successful authentication using certificate-based MACsec or Pre Shared Key (PSK) framework.

A switch using MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the MKA peer. MACsec frames are encrypted and protected with an integrity check value (ICV). When the switch receives frames from the MKA peer, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The switch compares that ICV to the ICV within the frame. If they are not identical, the frame is dropped. The switch also encrypts and adds an ICV to any frames sent over the secured port (the access point used to provide the secure MAC service to a MKA peer) using the current session key.

The MKA Protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1x-REV. The MKA Protocol extends 802.1x to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers.

The EAP framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). The switch acts as the authenticator for both uplink and downlink; and acts as the key server for downlink. It generates a random secure association key (SAK), which is sent to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the switch sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes with no MKPDU received from a participant. For example, if a MKA peer disconnects, the participant on the switch continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the MKA peer.



Note Integrity check value (ICV) indicator in MKPDU is optional. ICV is not optional when the traffic is encrypted.

EAPoL Announcements indicate the use of the type of keying material. The announcements can be used to announce the capability of the supplicant as well as the authenticator. Based on the capability of each side, the largest common denominator of the keying material could be used.

MKA Policies

To enable MKA on an interface, a defined MKA policy should be applied to the interface. You can configure these options:

- Policy name, not to exceed 16 ASCII characters.

- Confidentiality (encryption) offset of 0, 30, or 50 bytes for each physical interface

Definition of Policy-Map Actions

This section describes the policy-map actions and its definition:

- **Activate:** Applies a service template to the session.
- **Authenticate:** Starts authentication of the session.
- **Authorize:** Explicitly authorizes a session.
- **Set-domain:** Explicitly sets the domain of a client.
- **Terminate:** Terminates the method that is running, and deletes all the method details associated with the session.
- **Deactivate:** Removes the service-template applied to the session. If not applied, no action is taken.
- **Set-timer:** Starts a timer and gets associated with the session. When the timer expires, any action that needs to be started can be processed.
- **Authentication-restart:** Restarts authentication.
- **Clear-session:** Deletes a session.
- **Pause:** Pauses authentication.

Rest of the actions as self-explanatory and are associated with authentication.

Virtual Ports

Use virtual ports for multiple secured connectivity associations on a single physical port. Each connectivity association (pair) represents a virtual port. In uplink, you can have only one virtual port per physical port. You cannot simultaneously host secured and unsecured sessions in the same VLAN on the same port. Because of this limitation, 802.1x multiple authentication mode is not supported.

The exception to this limitation is in multiple-host mode when the first MACsec supplicant is successfully authenticated and connected to a hub that is connected to the switch. A non-MACsec host connected to the hub can send traffic without authentication because it is in multiple-host mode. We do not recommend using multi-host mode because after the first successful client, authentication is not required for other clients.

Virtual ports represent an arbitrary identifier for a connectivity association and have no meaning outside the MKA Protocol. A virtual port corresponds to a separate logical port ID. Valid port IDs for a virtual port are 0x0002 to 0xFFFF. Each virtual port receives a unique secure channel identifier (SCI) based on the MAC address of the physical interface concatenated with a 16-bit port ID.

MKA Statistics

Some MKA counters are aggregated globally, while others are updated both globally and per session. You can also obtain information about the status of MKA sessions. See [Displaying MKA Statistics](#) for further information.

Key Lifetime and Hitless Key Rollover

A MACsec key chain can have multiple pre-shared keys (PSK) each configured with a key id and an optional lifetime. A key lifetime specifies at which time the key expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the key chain after the lifetime is expired. Time zone of the key can be local or UTC. Default time zone is UTC.

You can Key rolls over to the next key within the same key chain by configuring a second key in the key chain and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless, that is, key rolls over without traffic interruption.

On all participating devices, the MACsec key chain must be synchronised by using Network Time Protocol (NTP) and the same time zone must be used. If all the participating devices are not synchronized, the connectivity association key (CAK) rekey will not be initiated on all the devices at the same time.



Note The lifetime of the keys need to be overlapped in order to achieve hitless key rollover.

Replay Protection Window Size

Replay protection is a feature provided by MACsec to counter replay attacks. Each encrypted packet is assigned a unique sequence number and the sequence is verified at the remote end. Frames transmitted through a Metro Ethernet service provider network are highly susceptible to reordering due to prioritization and load balancing mechanisms used within the network.

A replay window is necessary to support the use of MACsec over provider networks that reorder frames. Frames within the window can be received out of order, but are not replay protected. The default window size is 0, which enforces strict reception ordering. The replay window size can be configured in the range of 0 to $2^{32} - 1$.

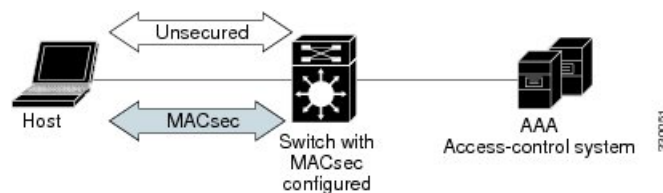
MACsec, MKA and 802.1x Host Modes

You can use MACsec and the MKA Protocol with 802.1x single-host mode, multi-host mode, or Multi Domain Authentication (MDA) mode. Multiple authentication mode is not supported.

Single-Host Mode

The figure shows how a single EAP authenticated session is secured by MACsec by using MKA

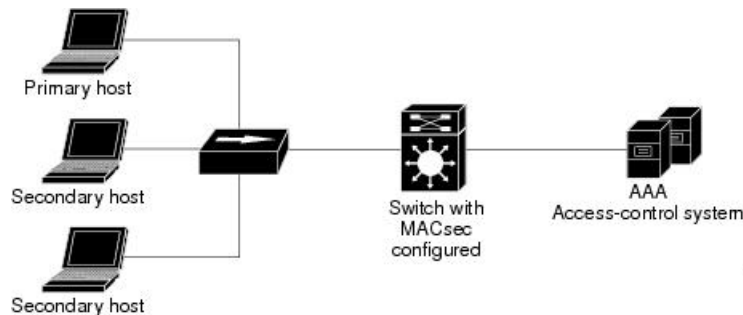
Figure 3: MACsec in Single-Host Mode with a Secured Data Session



Multiple-Host Mode

In standard (not 802.1x REV) 802.1x multiple-host mode, a port is open or closed based on a single authentication. If one user, the primary secured client services client host, is authenticated, the same level of network access is provided to any host connected to the same port. If a secondary host is a MACsec supplicant, it cannot be authenticated and traffic would not flow. A secondary host that is a non-MACsec host can send traffic to the network without authentication because it is in multiple-host mode. The figure shows MACsec in Standard Multiple-Host Unsecure Mode.

Figure 4: MACsec in Multiple-Host Mode - Unsecured



Note Multi-host mode is not recommended because after the first successful client authentication, authentication is not required for other clients, which is not secure.

In standard (not 802.1x REV) 802.1x multiple-domain mode, a port is open or closed based on a single authentication. If the primary user, a PC on data domain, is authenticated, the same level of network access is provided to any domain connected to the same port. If a secondary user is a MACsec supplicant, it cannot be authenticated and traffic would not flow. A secondary user, an IP phone on voice domain, that is a non-MACsec host, can send traffic to the network without authentication because it is in multiple-domain mode.

Multiple-Domain Mode

In standard (not 802.1x REV) 802.1x multiple-domain mode, a port is open or closed based on a single authentication. If the primary user, a PC on data domain, is authenticated, the same level of network access is provided to any domain connected to the same port. If a secondary user is a MACsec supplicant, it cannot be authenticated and traffic would not flow. A secondary user, an IP phone on voice domain, that is a non-MACsec host, can send traffic to the network without authentication because it is in multiple-domain mode.

MACsec MKA using Certificate-based MACsec

MACsec MKA is supported on switch-to-switch links. Using certificate-based MACsec, you can configure MACsec MKA between device uplink ports. Certificate-based MACsec allows mutual authentication and obtains an MSK (master session key) from which the connectivity association key (CAK) is derived for MKA operations. Device certificates are carried, using certificate-based MACsec, for authentication to the AAA server.



Note Certificate-based MACsec is supported for Cisco Catalyst ESS9300 Embedded Series Switch beginning with the Cisco IOS XE 17.13.1 release.

Prerequisites for MACsec MKA Using Certificate-based MACsec

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.
- Ensure that 802.1x authentication and AAA are configured on your device.

Switch-to-Switch MKA MACsec Must Secure Policy

Must-secure support is enabled on both the ingress and the egress. Must-secure is supported for MKA. With must-secure enabled, only EAPoL traffic will not be encrypted. The rest of the traffic will be encrypted. Unencrypted packets are dropped.



Note Must-secure mode is enabled by default.

MKA/MACsec for Port Channel

MKA/MACsec can be configured on the port members of a port channel. MKA/MACsec is agnostic to the port channel since the MKA session is established between the port members of a port channel.



Note Port channel is supported for PSK-based MACsec but not for certificate-based MACsec.



Note EtherChannel links that are formed as part of the port channel can either be congruent or disparate. That is, the links can either be MACsec-secured or non-MACsec-secured. MKA session between the port members is established even if a port member on one side of the port channel is not configured with MACsec.

We recommend that you enable MKA/MACsec on all the member ports for better security of the port channel.

MACsec Cipher Announcement

Cipher Announcement allows the supplicant and the authenticator to announce their respective MACsec Cipher Suite capabilities to each other. Both the supplicant and the authenticator calculate the largest common supported MACsec Cipher Suite and use the same as the keying material for the MKA session.



Note Only the MACsec Cipher Suite capabilities which are configured in the MKA policy are announced from the authenticator to the supplicant.

There are two types of EAPoL Announcements:

- Unsecured Announcements (EAPoL PDUs) : Unsecured announcements are EAPoL announcements carrying MACsec Cipher Suite capabilities in an unsecured manner. These announcements are used to decide the width of the key used for MKA session prior to authentication.
- Secure Announcements (MKPDUs) : Secure announcements revalidate the MACsec Cipher Suite capabilities which were shared previously through unsecure announcements.

Once the session is authenticated, peer capabilities which were received through EAPoL announcements are revalidated with the secure announcements. If there is a mismatch in the capabilities, the MKA session tears down.

Limitations for MACsec Cipher Announcement

- MACsec Cipher Announcement is supported only on the switch-to-host links.
- The MKA session between the supplicant and the authenticator does not tear down even if the MACsec Cipher Suite capabilities configured on both do not result in a common cipher suite.
- Host to Switch MKA MACsec 256-bit Cipher is not supported on both Network Advantage and Network Essentials.

How to Configure MACsec Encryption

Prerequisites for MACsec Encryption

Prerequisites for MACsec Encryption

- Enable the **ssci-based-on-sci** command while configuring MACsec encryption on the device to allow interoperability with non-Cisco and non-IOS XE devices.
- Ensure that 802.1x authentication and AAA are configured on your device.

Prerequisites for Certificate-Based MACsec

- Ensure that you have a Certificate Authority (CA) server configured for your network.

- Generate a CA certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE) Release 2.0.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP). If time is not synchronized on all your devices, certificates will not be validated.

Restrictions for MACsec Encryption

- MACsec Key Agreement (MKA) is not supported with both stateful and stateless high availability.
- MACsec with MKA is supported only on point-to-point links.
- MACsec configuration is not supported on EtherChannel ports. Instead, MACsec configuration can be applied on the individual member ports of an EtherChannel. To remove MACsec configuration, you must first unbundle the member ports from the EtherChannel, and then remove it from the individual member ports.
- Cisco Catalyst IE9300 Rugged Series Switches support 128-bit MACsec encryption with a Network Essentials license and 256-bit MACsec encryption with a Network Advantage license.
- Certificate-based MACsec is supported only if the access-session is configured as closed or in multiple-host mode. None of the other configuration modes are supported.
- Packet number exhaustion rekey is not supported.
- If the **dot1q tag vlan native** command is configured globally, the dot1x reauthentication will fail on trunk ports.
- MACsec with Precision Time Protocol (PTP) is not supported.
- The **should-secure** access mode is supported on switch-to-switch ports only using PSK authentication.
- PSK fallback key chain is not supported for point-to-multipoint cases.
- PSK fallback key chain is not supported on a high availability setup.
- PSK fallback key chain supports infinite lifetime with one key only.
- The connectivity association key name (CKN) ID used in the fallback key chain must not match any of the CKN IDs used in the primary key chain.
- The following limitations apply only to certificate-based MACsec.
 - The port should be in access mode or trunk mode.
 - MKA is not supported on port channels.
 - Ports with no switch port are not supported.

Recommendations for MACsec Encryption

This section lists the recommendations for configuring MACsec encryption:

- Use the confidentiality (encryption) offset as 0 in switch-to-host connections.

- Execute the **shutdown** command, and then the **no shutdown** command on a port, after changing any MKA policy or MACsec configuration for active sessions, so that the changes are applied to active sessions.
- Set the connectivity association key (CAK) rekey overlap timer to 30 seconds or more.

MKA and MACsec Configuration

MACsec is disabled by default. No MKA policies are configured.

Configure an MKA Policy

Beginning in privileged EXEC mode, follow these steps to create an MKA Protocol policy. Note that MKA also requires that you enable 802.1x.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **key-server** *priority*
5. **include-icv-indicator**
6. **macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}
7. **confidentiality-offset** *offset-value*
8. **ssci-based-on-sci**
9. **end**
10. **show mka policy**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	mka policy <i>policy-name</i> Example: Device(config)# mka policy <i>mka_policy</i>	Identifies an MKA policy, and enters MKA policy configuration mode. The maximum policy name length is 16 characters. Note The default MACsec cipher suite in the MKA policy will always be "GCM-AES-128". If the device supports both

	Command or Action	Purpose
		"GCM-AES-128" and "GCM-AES-256" ciphers, it is highly recommended to define and use a user defined MKA policy to include both 128 and 256 bits ciphers or only 256 bits cipher, as may be required.
Step 4	key-server priority Example: Device(config-mka-policy) # key-server priority 200	Configures MKA key server options and set priority (between 0-255). Note When value of key server priority is set to 255, the peer can not become the key server. The key server priority value is valid only for MKA PSK; and not for MKA EAPTLS.
Step 5	include-icv-indicator Example: Device(config-mka-policy) # include-icv-indicator	Enables the ICV indicator in MKPDU. Use the no form of this command to disable the ICV indicator.
Step 6	macsec-cipher-suite {gcm-aes-128 gcm-aes-256} Example: Device(config-mka-policy) # macsec-cipher-suite gcm-aes-128	Configures a cipher suite for deriving SAK with 128-bit or 256-bit encryption.
Step 7	confidentiality-offset offset-value Example: Device(config-mka-policy) # confidentiality-offset 0	Set the confidentiality (encryption) offset for each physical interface. Note Offset Value can be 0, 30 or 50. If you are using Anyconnect on the client, it is recommended to use Offset 0.
Step 8	ssci-based-on-sci Example: Device(config-mka-policy) # ssci-based-on-sci	(Optional) Computes Short Secure Channel Identifier (SSCI) value based on Secure Channel Identifier (SCI) value. The higher the SCI value, the lower is the SSCI value.
Step 9	end Example: Device(config-mka-policy) # end	Exit enters MKA policy configuration mode and returns to privileged EXEC mode.
Step 10	show mka policy Example: Device# show mka policy	Displays MKA policy configuration information.

Configure Switch-to-host MACsec Encryption

Follow these steps to configure MACsec on an interface with one MACsec session for voice and one for data:

SUMMARY STEPS

1. **enable**
2. **configureterminal**
3. **interface** *type number*
4. **switchport access vlan***vlan-id*
5. **switchport mode access**
6. **macsec**
7. **authentication event linksec fail action authorize vlan** *vlan-id*
8. **authentication host-mode multi-domain**
9. **authentication linksec policy must-secure**
10. **authentication port-control auto**
11. **authentication periodic**
12. **authentication timer reauthenticate**
13. **authentication violation protect**
14. **mka policy** *policy-name*
15. **dot1x pae authenticator**
16. **spanning-tree portfast**
17. **end**
18. **show authentication session interface** *interface-id*
19. **show mka sessions**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter the password if prompted.
Step 2	configureterminal Example: Device> configure terminal	Enters the global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface GigabitEthernet 1/0/1	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	switchport access vlan <i>vlan-id</i> Example: Device(config-if)# switchport access vlan 1	Configures the access VLAN for the port.
Step 5	switchport mode access Example:	Configures the interface as an access port.

	Command or Action	Purpose
	Device(config-if)# switchport mode access	
Step 6	macsec Example: Device(config-if)# macsec	Enables 802.1ae MACsec on the interface. The macsec command enables MKA MACsec on switch-to-host links only.
Step 7	authentication event linksec fail action authorize vlan <i>vlan-id</i> Example: Device(config-if)# authentication event linksec fail action authorize vlan 1	(Optional) Specifies that the switch processes authentication link-security failures resulting from unrecognized user credentials by authorizing a restricted VLAN on the port after a failed authentication attempt.
Step 8	authentication host-mode multi-domain Example: Device(config-if)# authentication host-mode multi-domain	Configures authentication manager mode on the port to allow both a host and a voice device to be authenticated on the 802.1x-authorized port. If not configured, the default host mode is single.
Step 9	authentication linksec policy must-secure Example: Device(config-if)# authentication linksec policy must-secure	Sets the LinkSec security policy to secure the session with MACsec if the peer is available. If not set, the default is <i>should secure</i> .
Step 10	authentication port-control auto Example: Device(config-if)# authentication port-control auto	Enables 802.1x authentication on the port. The port changes to the authorized or unauthorized state based on the authentication exchange between the switch and the client.
Step 11	authentication periodic Example: Device(config-if)# authentication periodic	(Optional) Enables or disables re-authentication for this port.
Step 12	authentication timer reauthenticate Example: Device(config-if)# authentication timer reauthenticate	(Optional) Enters a value between 1 and 65535 (in seconds). Obtains re-authentication timeout value from the server. Default re-authentication time is 3600 seconds.
Step 13	authentication violation protect Example: Device(config-if)# configure terminal	Configures the port to drop unexpected incoming MAC addresses when a new device connects to a port or when a device connects to a port after the maximum number of devices are connected to that port. If not configured, the default is to shut down the port.
Step 14	mka policy <i>policy-name</i> Example: Device(config-if)# mka policy mka_policy	Applies an existing MKA protocol policy to the interface, and enable MKA on the interface. If no MKA policy was configured (by entering the mka policy global configuration command).

	Command or Action	Purpose
Step 15	dot1x pae authenticator Example: Device(config-if) # dot1x pae authenticator	Configures the port as an 802.1x port access entity (PAE) authenticator.
Step 16	spanning-tree portfast Example: Device(config-if) # spanning-tree portfast	Enables spanning tree Port Fast on the interface in all its associated VLANs. When the Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes
Step 17	end Example: Device(config) # end	Exits interface configuration mode and returns to privileged EXEC mode.
Step 18	show authentication session interface <i>interface-id</i> Example: Device# show authentication session interface GigabitEthernet 1/0/1	Verifies the authorized session security status.
Step 19	show mka sessions Example: Device# show mka sessions	Verifies the established MKA sessions.

Configure MACsec MKA using PSK

Beginning in privileged EXEC mode, follow these steps to configure MACsec MKA policies using a Pre Shared Key (PSK).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain *key-chain-name* macsec**
4. **key *hex-string***
5. **cryptographic-algorithm {*aes-128-cmac* | *aes-256-cmac*}**
6. **key-string { [*0/6/7*] *pwd-string* | *pwd-string*}**
7. **lifetime local [*start timestamp* {*hh::mm::ss* | *day* | *month* | *year*}] [*duration seconds* | *end timestamp* {*hh::mm::ss* | *day* | *month* | *year*}]**
8. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	key chain key-chain-name macsec Example: Device(config)# key chain keychain1 macsec	Configures a key chain and enters the key chain configuration mode.
Step 4	key hex-string Example: Device(config-key-chain)# key 1000	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. Note For 128-bit encryption, use any value between 1 and 32 hex digit key-string. For 256-bit encryption, use 64 hex digit key-string.
Step 5	cryptographic-algorithm {aes-128-cmac aes-256-cmac} Example: Device(config-key-chain)# cryptographic-algorithm aes-128-cmac	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 6	key-string { [0/6/7] pwd-string pwd-string } Example: Device(config-key-chain)# key-string 12345678901234567890123456789012	Sets the password for a key string. Only hex characters must be entered.
Step 7	lifetime local [start timestamp {hh::mm::ss day month year}] [duration seconds end timestamp {hh::mm::ss day month year}] Example: Device(config-key-chain)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016	Sets the lifetime of the pre shared key.
Step 8	end Example: Device(config-key-chain)# end	Exits key chain configuration mode and returns to privileged EXEC mode.

Configure MACsec MKA on an Interface using PSK

Beginning in privileged EXEC mode, follow these steps to configure MACsec MKA policies on an interface using a Pre Shared Key (PSK).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **macsec network-link**
5. **mka policy** *policy-name*
6. **mka pre-shared-key key-chain** *key-chain name* [**fallback key-chain** *key-chain name*]
7. **macsec replay-protection window-size** *frame number*
8. **end**

DETAILED STEPS

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 <i>interface-id</i> Example: Device(config-if) # interface GigabitEthernet 0/0/0	Enters interface configuration mode.
Step 4	macsec network-link Example: Device(config-if) # macsec network-link	Enables MACsec on the interface.
Step 5	mka policy <i>policy-name</i> Example: Device(config-if) # mka policy <i>mka_policy</i>	Configures an MKA policy.
Step 6	mka pre-shared-key key-chain <i>key-chain name</i> [fallback key-chain <i>key-chain name</i>] Example: Device(config-if) # mka pre-shared-key key-chain <i>key-chain-name</i>	Configures an MKA pre-shared-key key-chain name.

	Command or Action	Purpose
Step 7	macsec replay-protection window-size <i>frame number</i> 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	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

It is not recommended to change the MKA policy on an interface with MKA PSK configured when the session is running. However, if a change is required, you must reconfigure the policy as follows:

1. Disable the existing session by removing **macsec network-link** configuration on each of the participating node using the **no macsec network-link** command
2. Configure the MKA policy on the interface on each of the participating node using the **mka policy policy-name** command.
3. Enable the new session on each of the participating node by using the **macsec network-link** command.

Configuring MACsec MKA Using Certificate-based MACsec

To configure MACsec with MKA on point-to-point links, perform these tasks:

- Configure Certificate Enrollment
 - Generate Key Pairs
 - Configure SCEP Enrollment
 - Configure Certificates Manually
- Configure an Authentication Policy
- Configure certificate-based MACsec Profiles and IEEE 802.1x Credentials
- Configure MKA MACsec using certificate-based MACsec on Interfaces

Generate Key Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa label** *label-name* **general-keys modulus** *size*
4. **end**
5. **show authentication session interface** *interface-id*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto key generate rsa label <i>label-name</i> general-keys modulus <i>size</i> Example: Device(config)# crypto key generate rsa label general-keys modulus 2048	Generates a RSA key pair for signing and encryption. You can also assign a label to each key pair using the label keyword. The label is referenced by the trustpoint that uses the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>. If you do not use additional keywords this command generates one general purpose RSA key pair. If the modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the modulus keyword.
Step 4	end Example: Device(config)# end	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	show authentication session interface <i>interface-id</i> Example: Device# show authentication session interface gigabitethernet 0/1/1	Verifies the authorized session security status.

Configure Enrollment using SCEP

Simple Certificate Enrollment Protocol (SCEP) is a Cisco-developed enrollment protocol that uses HTTP to communicate with the certificate authority (CA) or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint *server name***
4. **enrollment url *url name pem***
5. **rsakeypair *label***
6. **serial-number none**

7. **ip-address none**
8. **revocation-check crl**
9. **auto-enroll *percent* regenerate**
10. **exit**
11. **crypto pki authenticate *name***
12. **end**
13. **show crypto pki certificate *trustpoint name***

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint <i>server name</i> Example: Device(config)# crypto pki trustpoint ka	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	enrollment url <i>url name pem</i> Example: Device(ca-trustpoint)# enrollment url http://url:80	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	rsakeypair <i>label</i> Example: Device(ca-trustpoint)# rsakeypair exampleCAkeys	Specifies which key pair to associate with the certificate. Note The rsakeypair name must match the trust-point name.
Step 6	serial-number none Example: Device(ca-trustpoint)# serial-number none	The none keyword specifies that a serial number will not be included in the certificate request.
Step 7	ip-address none Example: Device(ca-trustpoint)# ip-address none	The none keyword specifies that no IP address should be included in the certificate request.

	Command or Action	Purpose
Step 8	revocation-check <i>crl</i> Example: Device (ca-trustpoint) # revocation-check <i>crl</i>	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	auto-enroll <i>percent regenerate</i> Example: Device (ca-trustpoint) # auto-enroll <i>90 regenerate</i>	<p>Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA.</p> <p>If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration.</p> <p>By default, only the Domain Name System (DNS) name of the device is included in the certificate.</p> <p>Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached.</p> <p>Use the regenerate keyword to generate a new key for the certificate even if a named key already exists.</p> <p>If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: “! RSA key pair associated with trustpoint is exportable.”</p> <p>It is recommended that a new key pair be generated for security reasons.</p>
Step 10	exit Example: Device (ca-trustpoint) # exit	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 11	crypto pki authenticate <i>name</i> Example: Device (config) # crypto pki authenticate <i>myca</i>	Retrieves the CA certificate and authenticates it.
Step 12	end Example: Device (config) # end	Exits global configuration mode and returns to privileged EXEC mode.
Step 13	show crypto pki certificate <i>trustpoint name</i> Example: Device # show crypto pki certificate <i>ka</i>	Displays information about the certificate for the trust point.

Configure Enrollment Manually

If your CA does not support SCEP or if a network connection between the router and CA is not possible, perform the following task to set up manual certificate enrollment:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *server name*
4. **enrollment url** *url name pem*
5. **rsakeypair** *label*
6. **serial-number** *none*
7. **ip-address** *none*
8. **revocation-check** *crl*
9. **exit**
10. **crypto pki authenticate** *name*
11. **crypto pki enroll** *name*
12. **crypto pki import** *name certificate*
13. **end**
14. **show crypto pki certificate** *trustpoint name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint <i>server name</i> Example: Device# crypto pki trustpoint ka	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	enrollment url <i>url name pem</i> Example: Device(ca-trustpoint)# enrollment url http://url:80	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80 . The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	rsakeypair <i>label</i> Example: Device(ca-trustpoint)# rsakeypair exampleCAkeys	Specifies which key pair to associate with the certificate.

	Command or Action	Purpose
Step 6	serial-number none Example: Device(ca-trustpoint)# serial-number none	The none keyword specifies that a serial number will not be included in the certificate request.
Step 7	ip-address none Example: Device(ca-trustpoint)# ip-address none	The none keyword specifies that no IP address should be included in the certificate request.
Step 8	revocation-check crl Example: Device(ca-trustpoint)# revocation-check crl	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.
Step 9	exit Example: Device(ca-trustpoint)# exit	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 10	crypto pki authenticate name Example: Device(config)# crypto pki authenticate myca	Retrieves the CA certificate and authenticates it.
Step 11	crypto pki enroll name Example: Device(config)# crypto pki enroll myca	<p>Generates certificate request and displays the request for copying and pasting into the certificate server.</p> <p>Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request.</p> <p>You are also given the choice about displaying the certificate request to the console terminal.</p> <p>The base-64 encoded certificate with or without PEM headers as requested is displayed.</p>
Step 12	crypto pki import name certificate Example: Device(config)# crypto pki import myca certificate	<p>Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate.</p> <p>The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from “.req” to “.crt”. For usage key certificates, the extensions “-sign.crt” and “-encr.crt” are used.</p> <p>The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch.</p> <p>Note Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information</p>

	Command or Action	Purpose
		in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 13	end Example: Device(config)# end	Exits global configuration mode and returns to privileged EXEC mode.
Step 14	show crypto pki certificate <i>trustpoint name</i> Example: Device# show crypto pki certificate ka	Displays information about the certificate for the trust point.

Enable 802.1x Authentication and Configure AAA

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **dot1x system-auth-control**
5. **radius server** *name*
6. **address** *ip_address* **auth-port** *port_number* **acct-port** *port_number*
7. **automate-tester username** *username*
8. **key** *string*
9. **radius-server deadtime** *minutes*
10. **exit**
11. **aaa group server radius** *group_name*
12. **server** *name*
13. **exit**
14. **aaa authentication dot1x default group** *group_name*
15. **aaa authorization network default group** *group_name*

DETAILED STEPS

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:	Enters global configuration mode.

	Command or Action	Purpose
	<code>device# configure terminal</code>	
Step 3	aaa new-model Example: <code>device(config)# aaa new-model</code>	Enables AAA.
Step 4	dot1x system-auth-control Example: <code>device(config)# dot1x system-auth-control</code>	Enables 802.1X on your device.
Step 5	radius server name Example: <code>device(config)# radius server ISE</code>	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	address ip_address auth-port port_number acct-port port_number Example: <code>device(config-radius-server)# address ipv4 10.64.72.90 auth-port 1645 acct-port 1646</code>	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 7	automate-tester username username Example: <code>device(config-radius-server)# automate-tester username dummy</code>	<p>Enables the automated testing feature for the RADIUS server.</p> <p>With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks for a Radius response from the server. A success message is not necessary; a failed authentication suffices, because it shows that the server is alive.</p>
Step 8	key string Example: <code>device(config-radius-server)# key dummy123</code>	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
Step 9	radius-server deadtime minutes Example: <code>device(config-radius-server)# radius-server deadtime 2</code>	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
Step 10	exit	Returns to global configuration mode.
Step 11	aaa group server radius group_name Example: <code>device(config)# aaa group server radius ISEGRP</code>	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
Step 12	server name Example: <code>device(config)# server ise</code>	

	Command or Action	Purpose
Step 13	exit Example: <code>device(config-radius-server) # exit</code>	Returns to global configuration mode.
Step 14	aaa authentication dot1x default group <i>group_name</i> Example: <code>device(config) # aaa authentication dot1x default group ISEGRP</code>	Sets the default authentication server group for IEEE 802.1x.
Step 15	aaa authorization network default group <i>group_name</i> Example: <code>device(config) # aaa authorization network default group ISEGRP</code>	Sets the network authorization default group.

Apply the 802.1x MKA MACsec Configuration on the Interfaces

To apply MKA MACsec using EAP-TLS to interfaces, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *interface_id***
4. **macsec network-link**
5. **authentication periodic**
6. **authentication timer reauthenticate interval**
7. **access-session host-mode multi-domain**
8. **access-session closed**
9. **access-session port-control auto**
10. **dot1x pae both**
11. **dot1x credentials profile**
12. **dot1x supplicant eap profile *profile_name***
13. **dot1x authenticator eap profile *profile_name***
14. **service-policy type control subscriber *control_policy_name***
15. **exit**
16. **show macsec interface**
17. **copy running-config startup-config**

DETAILED STEPS

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 interface_id Example: device(config)# interface te0/1/2	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 4	macsec network-link Example: device(config)# macsec network-link	Enables MACsec on the interface.
Step 5	authentication periodic Example: device(config)# authentication periodic	Enables reauthentication for this port.
Step 6	authentication timer reauthenticate interval Example: device(config)# authentication timer reauthenticate interval	Sets the reauthentication interval.
Step 7	access-session host-mode multi-domain Example: device(config)# access-session host-mode multi-domain	Allows hosts to gain access to the interface.
Step 8	access-session closed Example: device(config)# access-session closed	Prevents preauthentication access on the interface.
Step 9	access-session port-control auto Example: device(config)# access-session port-control auto	Sets the authorization state of a port.
Step 10	dot1x pae both Example: device(config)# dot1x pae both	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.

	Command or Action	Purpose
Step 11	dot1x credentials profile Example: device(config)# dot1x credentials profile	Assigns a 802.1x credentials profile to the interface.
Step 12	dot1x supplicant eap profile <i>profile_name</i> Example: device(config)# dot1x supplicant eap profile eap1	Assigns the EAP-TLS profile to the interface.
Step 13	dot1x authenticator eap profile <i>profile_name</i> Example: device(config)# dot1x authenticator eap profile eap1	Assigns the EAP-TLS profile to use during 802.1x authentication.
Step 14	service-policy type control subscriber <i>control_policy_name</i> Example: device(config)# service-policy type control subscriber controlPolicy2	Applies a subscriber control policy to the interface.
Step 15	exit Example: device(config)# exit	Returns to privileged EXEC mode.
Step 16	show macsec interface Example: device# show macsec interface	Displays MACsec details for the interface.
Step 17	copy running-config startup-config Example: device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configure MKA/MACsec for Port Channel using PSK

Beginning in privileged EXEC mode, complete the following steps to configure MKA policies on an interface using a pre-shared key (PSK):

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config-if)# interface gigabitethernet 1/0/3	Enters interface configuration mode.
Step 4	macsec network-link Example: Device(config-if)# macsec network-link	Enables MACsec on the interface. Supports layer 2 and layer 3 port channels.
Step 5	mka policy policy-name Example: Device(config-if)# mka policy mka_policy	Configures an MKA policy.
Step 6	mka pre-shared-key key-chain key-chain name [fallback 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 name. Note The MKA pre-shared key can be configured on either physical interface or subinterfaces and not on both.
Step 7	macsec replay-protection window-size frame number Example: Device(config-if)# macsec replay-protection window-size 0	Sets the MACsec window size for replay protection.
Step 8	channel-group channel-group-number mode {auto desirable} {active passive} {on} Example: Device(config-if)# channel-group 3 mode auto active on	Configures the port in a channel group and sets the mode. Note You cannot configure ports in a channel group without configuring MACsec on the interface. You must configure the commands in Step 3, 4, 5 and 6 before this step. The channel-number range is from 1 to 4096. The port channel that is associated with this channel group is automatically created if the port channel does not already exist. For mode, select one of the following keywords: <ul style="list-style-type: none"> • auto: Enables PAgP only if a PAgP device is detected. This places the port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation. Note The auto keyword is not supported when EtherChannel members are from different switches in the switch stack.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • desirable: Unconditionally enables PAgP. This places the port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets. <p>Note The desirable keyword is not supported when EtherChannel members are from different switches in the switch stack.</p> <ul style="list-style-type: none"> • on: Forces the port to channel without PAgP or LACP. In the on mode, an EtherChannel exists only when a port group in the on mode is connected to another port group in the on mode. • active: Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets. • passive: Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.
Step 9	end Example: Device(config-if) # end	Exits interface configuration mode and returns to privileged EXEC mode.

Configure Port Channel Logical Interfaces for Layer 2 EtherChannels

To create a port channel interface for a Layer 2 EtherChannel, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface port-channel <i>channel-group-number</i> Example:	Creates the port channel interface. Note

	Command or Action	Purpose
	Device(config)# interface port-channel 1	Use the no form of this command to delete the port channel interface.
Step 4	switchport Example: Device(config-if)# switchport	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 5	switchport mode {access trunk} Example: Device(config-if)# switchport mode access	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks.
Step 6	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configure Port Channel Logical Interfaces for Layer 3 EtherChannels

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• 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 gigabitethernet 1/0/2	Enters interface configuration mode.
Step 4	no switchport Example: Device(config-if)# no switchport	Switches an interface that is in Layer 2 mode into Layer 3 mode for Layer 3 configuration.
Step 5	ip address ip-address subnet_mask Example: Device(config-if)# ip address 10.2.2.3 255.255.255.254	Assigns an IP address and subnet mask to the EtherChannel.

	Command or Action	Purpose
Step 6	end Example: Device(config-if) # end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring MACsec Cipher Announcement

Configure an MKA Policy for Secure Announcement

Beginning in privileged EXEC mode, follow these steps to create an MKA Protocol policy to enable secure announcement in MKPDUs. By default, secure announcements are disabled.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	mka policy <i>policy-name</i> Example: Device(config) # mka policy mka_policy	Identifies an MKA policy and enters MKA policy configuration mode. The maximum policy name length is 16 characters. Note The default MACsec cipher suite in the MKA policy is GCM-AES-128. If the device supports both "GCM-AES-128" and "GCM-AES-256" ciphers, it is highly recommended to define and use a user defined MKA policy to include both 128 and 256 bits ciphers or only 256 bits cipher, as may be required.
Step 4	key-server <i>priority</i> Example: Device(config-mka-policy) # key-server priority 200	Configures MKA key server options and sets priority between 0-255. Note When value of key server priority is set to 255, the peer cannot become the key server. The key server priority value is valid only for MKA PSK. This does not apply to MKA EAP-TLS.
Step 5	send-secure-announcements Example:	Enables sending of secure announcements. Use the no form of the command to disable sending of secure

	Command or Action	Purpose
	Device(config-mka-policy) # send-secure-announcements	announcements. By default, secure announcements are disabled.
Step 6	macsec-cipher-suite { <i>gcm-aes-128</i> <i>gcm-aes-256</i> } Example: Device(config-mka-policy) # macsec-cipher-suite gcm-aes-128	Configures cipher suite for deriving SAK with 128-bit or 256-bit encryption.
Step 7	end Example: Device(config-mka-policy) # end	Exits MKA policy configuration mode and returns to privileged EXEC mode.
Step 8	show mka policy Example: Device# show mka policy	Displays MKA policies.

Configure Secure Announcement Globally

Beginning in privileged EXEC mode, follow these steps to enable secure announcement globally across all the MKA Policies.

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	mka defaults policy send-secure-announcements Example: Device(config)# mka defaults policy send-secure-announcements	Enables sending of secure announcements in MKPDUs across MKA policies. By default, secure announcements are disabled.
Step 4	end Example: Device(config)# end	Exits global configuration mode and returns to privileged EXEC mode.

Configure EAPoL Announcements on an Interface

Beginning in privileged EXEC mode, follow these steps to configure EAPoL Announcement on an interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • 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 gigabitethernet 1/0/1	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	eapol announcement Example: Device(config-if) # eapol announcement	Enables EAPoL announcements. Use the no form of the command to disable EAPoL announcements. By default, EAPoL announcements are disabled.
Step 5	end Example: Device(config-if) # configure terminal	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for MACsec Encryption

Example: Configuring MKA and MACsec

This example shows how to create an MKA policy:

```
Device> enable
Device# configure terminal
Device(config)# mka policy mka_policy
Device(config-mka-policy)# key-server priority 200
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 30
Device(config-mka-policy)# ssci-based-on-sci
Device(config-mka-policy)#end
```

This example shows how to configure MACsec on an interface:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# switchport access vlan 1
Device(config-if)# switchport mode access
Device(config-if)# macsec
Device(config-if)#access-session event linksec fail action authorize vlan 1
Device(config-if)# access-session host-mode multi-domain
```

```

Device(config-if)# access-session linksec policy must-secure
Device(config-if)# access-session port-control auto
Device(config-if)# authentication periodic
Device(config-if)# authentication timer reauthenticate
Device(config-if)# authentication violation protect
Device(config-if)# mka policy mka_policy
Device(config-if)# dot1x pae authenticator
Device(config-if)# spanning-tree portfast
Device(config-if)# end

```

Examples: Configuring MACsec MKA Using PSK

This example shows how to configure MACsec MKA using PSK.

```

Device> enable
Device# configure terminal
Device(config)# key chain keychain1 macsec
Device(config-keychain)# key 1000
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789012
Device(config-keychain-key)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016
Device(config-keychain-key)# end

```

This example shows how to configure MACsec MKA on an interface using PSK.

```

Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# mka policy mka_policy
Device(config-if)# mka pre-shared-key key-chain key-chain-name
Device(config-if)# macsec replay-protection window-size 10
Device(config-if)# end

```

MKA-PSK: CKN Behavior Change

Starting Cisco IOS XE Fuji 16.8.1 release, for MKA PSK sessions, the CKN uses exactly the same string as the CKN which is configured as the hex-string for the key, instead of the fixed 32 bytes.

```

Device> enable
Device# configure terminal
Device(config)# key chain abc macsec
Device(config-keychain)# key 11
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789013
Device(config-keychain-key)# lifetime local 12:21:00 Sep 9 2015 infinite
Device(config-keychain-key)# end

```

The following is sample output of the **show mka session** command for the above configuration:

```
Device# show mka session
```

```

Total MKA Sessions..... 1
Secured Sessions... 1
Pending Sessions... 0

```

Interface	Local-TxSCI	Policy-Name	Inherited	Key-Server
Port-ID	Peer-RxSCI	MACsec-Peers	Status	CKN
Et0/0	aabb.cc00.6600/0002	icv	NO	NO

In case of interoperability between two images, where one having the CKN behavior change, and one without the CKN behavior change, the hex-string for the key must be a 64-character hex-string with zero padded for it to work on a device that has an image with the CKN behavior change. See the examples below:

```
Device# configure terminal
Device(config)# key chain abc macsec
Device(config-keychain)# key 11
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789013
Device(config-keychain-key)# lifetime local 12:21:00 Sep 9 2015 infinite
Device(config-keychain-key)# end
```

[illegible]

This example shows how to configure MACsec MKA using certificate-based MACsec:

Etherchannel Mode — Static/On

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
```

```

Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode on
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode on
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end

```

Layer 2 EtherChannel Configuration

Device 1

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

Device 2

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

The following is sample output from the `show etherchannel summary` command:

```

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby  (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG

```

```

Number of channel-groups in use: 1
Number of aggregators:           1

```

Group	Port-channel	Protocol	Ports
-----+-----+-----+-----			
2	Po2 (RU)	-	Te1/0/1 (P) Te1/0/2 (P)

Layer 3 EtherChannel Configuration

Device 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.3 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

Device 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.4 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

The following is sample output from the **show etherchannel summary** command:

```
Flags: D - down          P - bundled in port-channel
       I - stand-alone s - suspended
       H - Hot-standby (LACP only)
       R - Layer3        S - Layer2
       U - in use        f - failed to allocate aggregator

       M - not in use, minimum links not met
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port

       A - formed by Auto LAG
```

```
Number of channel-groups in use: 1
Number of aggregators:           1
```

Group	Port-channel	Protocol	Ports
-----+-----+-----+-----			
2	Po2 (RU)	-	Te1/0/1 (P) Te1/0/2 (P)

Etherchannel Mode — LACP

The following is sample configuration on Device 1 and Device 2 with EtherChannel Mode as LACP.

```

Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end

```

Layer 2 EtherChannel Configuration

Device 1

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

Device 2

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end

```

The following is sample output from the **show etherchannel summary** command:

```

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG

```

```

Number of channel-groups in use: 1
Number of aggregators:          1

```

```

-----+-----+-----+-----+
2       Po2 (SU)          LACP      Tel1/1/1 (P)  Tel1/1/2 (P)

```

Layer 3 EtherChannel Configuration

Device 1

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.3 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end

```

Device 2

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.4 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end

```

The following is sample output from the **show etherchannel summary** command:

```

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby  (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG

```

```

Number of channel-groups in use: 1
Number of aggregators:          1

```

```

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----+

```

```

2       Po2 (RU)          LACP      Tel1/1/1 (P)  Tel1/1/2 (P)

```


Etherchannel Mode — PAgP

The following is sample configuration on Device 1 and Device 2 with EtherChannel Mode as PAgP:

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end
```

Layer 2 EtherChannel Configuration

Device 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

Device 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

The following shows a sample output from the **show etherchannel summary** command.

```
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
```

A - formed by Auto LAG

2	Po2 (SU)	PAGP	Te1/1/1 (P)	Te1/1/2 (P)
---	----------	------	-------------	-------------

Device 1

Device 2

The following is sample output from the **show etherchannel summary** command:

A - formed by Auto LAG

```
2          Po2 (RU)          PAgP          Te1/1/1 (P)   Te1/1/2 (P)
```

Displaying Active MKA Sessions

The following shows all the active MKA sessions.

```
Device# show mka sessions interface Te1/0/1
```

Interface	Local-TxSCI	Policy-Name	Inherited	
Key-Server	Peer-RxSCI	MACsec-Peers	Status	CKN
Te1/0/1	00a3.d144.3364/0025	POLICY	NO	NO
37	701f.539b.b0c6/0032	1	Secured	
1000				

Examples: Configuring MACsec Cipher Announcement

This example shows how to configure MKA policy for Secure Announcement:

```
Device> enable
Device# configure terminal
Device(config)# mka policy mka_policy
Device(config-mka-policy)# key-server 2
Device(config-mka-policy)# send-secure-announcements
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128confidentiality-offset 0
Device(config-mka-policy)# end
```

This example shows how to configure Secure Announcement globally:

```
Device> enable
Device# configure terminal
Device(config)# mka defaults policy send-secure-announcements
Device(config)# end
```

This example shows how to configure EAPoL Announcements on an interface:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# eapol announcement
Device(config-if)# end
```

The following is sample output from the **show running-config interface** *interface-name* command with EAPoL announcement enabled.

```
Device# show running-config interface GigabitEthernet 1/0/1

switchport mode access
macsec
access-session host-mode multi-host
access-session closed
access-session port-control auto
dot1x pae authenticator
dot1x timeout quiet-period 10
```

```
dot1x timeout tx-period 5
dot1x timeout supp-timeout 10
dot1x supplicant eap profile peap
eapol announcement
spanning-tree portfast
service-policy type control subscriber Dot1X
```

The following is sample output from the **show mka sessions interface *interface-name* detail** command with secure announcement disabled.

```
Device# show mka sessions interface GigabitEthernet 1/0/1 detail
```

MKA Detailed Status for MKA Session

Status: SECURED - Secured MKA Session with MACsec

[illegible]

```
# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 1
```

Live Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
38046BA37D7DA77E06D006A9	89555	c800.8459.e764/002a	10

Potential Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

Dormant Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

The following is sample output from the **show mka sessions details** command with secure announcement disabled.

```
Device# show mka sessions details
```

MKA Detailed Status for MKA Session

=====

Status: SECURED - Secured MKA Session with MACsec

[illegible]

```
Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status..... FIRST-SAK
Old SAK AN..... 0
Old SAK KI (KN)..... FIRST-SAK (0)
```

```
SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time..... 0s (No Old SAK to retire)
```

```
MKA Policy Name..... p2
Key Server Priority..... 2
Delay Protection..... NO
Replay Protection..... YES
Replay Window Size..... 0
```

```

Confidentiality Offset... 0
Algorithm Agility..... 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite..... 0080C20001000001 (GCM-AES-128)
MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired..... YES

# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                               MN           Rx-SCI (Peer)           KS Priority
  -----
  38046BA37D7DA77E06D006A9   89560       c800.8459.e764/002a    10

Potential Peers List:
  MI                               MN           Rx-SCI (Peer)           KS Priority
  -----

Dormant Peers List:
  MI                               MN           Rx-SCI (Peer)           KS Priority
  -----

```

The following is sample output from the **show mka policy policy-name detail** command with secure announcement disabled.

```

Device# show mka policy p2 detail

MKA Policy Configuration ("p2")
=====
MKA Policy Name..... p2
Key Server Priority.... 2
Confidentiality Offset. 0
Send Secure Announcement..DISABLED
Cipher Suite(s)..... GCM-AES-128

Applied Interfaces...
  GigabitEthernet1/0/1

```

Examples: Displaying MKA Information

The following is sample output from the **show mka sessions** command.

```
Device# show mka sessions
```

```

Total MKA Sessions..... 1
  Secured Sessions... 1
  Pending Sessions... 0

```

Interface	Local-TxSCI	Policy-Name	Inherited
Key-Server			
Port-ID	Peer-RxSCI	MACsec-Peers	Status CKN

[illegible]

The following is sample output from the **show mka sessions interface** *interface-name* command.

```
Device# show mka sessions interface GigabitEthernet 1/0/1
```

Summary of All Currently Active MKA Sessions on Interface
GigabitEthernet1/0/1...

[illegible]

The following is sample output from the **show mka sessions interface *interface-name* detail** command.

```
Device# show mka sessions interface GigabitEthernet 1/0/1 detail
```

MKA Detailed Status for MKA Session

Status: SECURED - Secured MKA Session with MACsec

[illegible]

```
Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status..... FIRST-SAK
Old SAK AN..... 0
Old SAK KI (KN)..... FIRST-SAK (0)
```

```
SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time..... 0s (No Old SAK to retire)
```

MKA Policy Name..... p2

```
# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 1
```

MI	MN	Rx-SCI (Peer)	KS Priority
38046BA37D7DA77E06D006A9	89555	c800.8459.e764/002a	10

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

[illegible]


```

Old SAK KI (KN)..... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time..... 0s (No Old SAK to retire)

MKA Policy Name..... p2
Key Server Priority..... 2
Delay Protection..... NO
Replay Protection..... YES
Replay Window Size..... 0
Confidentiality Offset... 0
Algorithm Agility..... 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite..... 0080C20001000001 (GCM-AES-128)
MACsec Capability..... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired..... YES

# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 1

```

Live Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
38046BA37D7DA77E06D006A9	89560	c800.8459.e764/002a	10

Potential Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

Dormant Peers List:

MI	MN	Rx-SCI (Peer)	KS Priority
----	----	---------------	-------------

The following is sample output from the **show mka policy** command:

```
Device# show mka policy
```

MKA Policy Summary...

Policy Interfaces Name Applied	KS Priority	Delay Protect	Replay Protect	Window Size	Conf Offset	Cipher Suite(s)
DEFAULT POLICY	0	FALSE	TRUE	0	0	GCM-AES-128
p1	1	FALSE	TRUE	0	0	GCM-AES-128
p2 Gi1/0/1	2	FALSE	TRUE	0	0	GCM-AES-128

The following is sample output from the **show mka policy policy-name** command:

```
Device# show mka policy p2
```

MKA Policy Summary...

Policy	KS	Delay	Replay	Window	Conf	Cipher
Interfaces						
Name	Priority	Protect	Protect	Size	Offset	Suite(s)
Applied						
p2	2	FALSE	TRUE	0	0	GCM-AES-128
Gil/0/1						

The following is sample output from the **show mka policy *policy-name* detail** command:

```
Device# show mka policy p2 detail
```

MKA Policy Configuration ("p2")

=====

MKA Policy Name..... p2
 Key Server Priority.... 2
 Confidentiality Offset. 0
 Send Secure Announcement..DISABLED
 Cipher Suite(s)..... GCM-AES-128

Applied Interfaces...

GigabitEthernet1/0/1

The following is sample output from the **show mka statistics interface *interface-name*** command:

```
Device# show mka statistics interface GigabitEthernet 1/0/1
```

MKA Statistics for Session

=====

Reauthentication Attempts.. 0

CA Statistics

Pairwise CAKs Derived... 0
 Pairwise CAK Rekeys..... 0
 Group CAKs Generated.... 0
 Group CAKs Received..... 0

SA Statistics

SAKs Generated..... 1
 SAKs Rekeyed..... 0
 SAKs Received..... 0
 SAK Responses Received.. 1

MKPDU Statistics

MKPDUs Validated & Rx... 89585
 "Distributed SAK".. 0
 "Distributed CAK".. 0
 MKPDUs Transmitted..... 89596

```
"Distributed SAK".. 1
"Distributed CAK".. 0
```

The following is sample output from the **show mka summary** command:

```
Device# show mka summary
```

```
Total MKA Sessions..... 1
    Secured Sessions... 1
    Pending Sessions... 0
```

[illegible]

MKA Global Statistics

=====

MKA Session Totals

```
Secured..... 1
Reauthentication Attempts.. 0

Deleted (Secured)..... 0
Keepalive Timeouts..... 0
```

CA Statistics

```
Pairwise CAKs Derived..... 0
Pairwise CAK Rekeys..... 0
Group CAKs Generated..... 0
Group CAKs Received..... 0
```

SA Statistics

```
SAKs Generated..... 1
SAKs Rekeyed..... 0
SAKs Received..... 0
SAK Responses Received.... 1
```

MKPDU Statistics

```

MKPDUs Validated & Rx..... 89589
    "Distributed SAK"..... 0
    "Distributed CAK"..... 0
MKPDUs Transmitted..... 89600
    "Distributed SAK"..... 1
    "Distributed CAK"..... 0

```

MKA Error Counter Totals

```

=====
Session Failures
  Bring-up Failures..... 0
  Reauthentication Failures..... 0
  Duplicate Auth-Mgr Handle..... 0

SAK Failures
  SAK Generation..... 0
  Hash Key Generation..... 0
  SAK Encryption/Wrap..... 0
  SAK Decryption/Unwrap..... 0
  SAK Cipher Mismatch..... 0

CA Failures
  Group CAK Generation..... 0
  Group CAK Encryption/Wrap..... 0
  Group CAK Decryption/Unwrap..... 0
  Pairwise CAK Derivation..... 0
  CKN Derivation..... 0
  ICK Derivation..... 0
  KEK Derivation..... 0
  Invalid Peer MACsec Capability... 0

MACsec Failures
  Rx SC Creation..... 0
  Tx SC Creation..... 0
  Rx SA Installation..... 0
  Tx SA Installation..... 0

MKPDU Failures
  MKPDU Tx..... 0
  MKPDU Rx Validation..... 0
  MKPDU Rx Bad Peer MN..... 0
  MKPDU Rx Non-recent Peerlist MN.. 0

```

Additional References for MACsec Encryption

Standards and RFCs

Standard/RFC	Title
IEEE 802.1AE-2006	<i>Media Access Control (MAC) Security</i>
IEEE 802.1X-2010	<i>Port-Based Network Access Control</i>
IEEE 802.1AEbw-2013	<i>Media Access Control (MAC) Security (Amendment to IEEE 802.1AE-2006)—Extended Packet Numbering (XPN)</i>

Standard/RFC	Title
IEEE 802.1Xbx-2014	<i>Port-Based Network Access Control (Amendment to IEEE 802.1X-2010)</i>
RFC 4493	<i>The AES-CMAC Algorithm</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	Support & Downloads page on Cisco.com

Feature History for MACsec Encryption

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE 17.13.1	Certificate-based MACsec encryption	Support for this feature was introduced for the Cisco Catalyst ESS9300 Embedded Series Switch in this release.
Cisco IOS XE Cupertino 17.8.x	MACsec encryption	<p>MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices.</p> <p>Support for this feature was introduced for Cisco Catalyst IE9300 Rugged Series Switches in this release.</p>

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.



CHAPTER 3

Network Edge Access Topology

- [802.1x Supplicant and Authenticator Switches with Network Edge Access Topology](#), on page 93
- [Guidelines and Limitations](#), on page 95
- [Configure an Authenticator Switch with NEAT](#), on page 95
- [Configure a Supplicant Switch with NEAT](#), on page 97
- [Verifying Configuration](#), on page 100
- [Feature History](#), on page 101

802.1x Supplicant and Authenticator Switches with Network Edge Access Topology

The 802.1x standard defines a client-server-based access control and authentication protocol that prevents unauthorized clients from connecting to a LAN through publicly accessible ports unless they are properly authenticated. The authentication server authenticates each client connected to a switch port before making available any services offered by the switch or the LAN. For more information about 802.1x, including configuration information, see [Configuring IEEE 802.1x Port-Based Authentication](#).

The Network Edge Access Topology (NEAT) feature extends identity to areas outside the wiring closet. This allows any type of device to authenticate on the port. NEAT uses Client Information Signalling Protocol (CISP) to propagate Client MAC and VLAN information between supplicant and Authenticator. CISP and NEAT are supported only on L2 ports, not on L3 ports. You can configure NEAT on Cisco Catalyst IE9300 Rugged Series Switches.

- **802.1x switch supplicant:** You can configure a switch to act as a supplicant to another switch by using the 802.1x supplicant feature. This configuration is helpful in a scenario, where, for example, a switch is outside a wiring closet and is connected to an upstream switch through a trunk port. A switch configured with the 802.1x switch supplicant feature authenticates with the upstream switch for secure connectivity. Once the supplicant switch authenticates successfully the port mode changes from access to trunk in an authenticator switch. In a supplicant switch you must manually configure the trunk when enabling CISP.
- If the access VLAN is configured on the authenticator switch, it becomes the native VLAN for the trunk port after successful authentication.

In the default state, when you connect a supplicant switch to an authenticator switch that has BPDU guard enabled, the authenticator port could be error-disabled if it receives a Spanning Tree Protocol (STP) bridge protocol data unit (BPDU) packets before the supplicant switch has authenticated. You can control traffic exiting the supplicant port during the authentication period. Entering the **dot1x supplicant controlled transient**

global configuration command temporarily blocks the supplicant port during authentication to ensure that the authenticator port does not shut down before authentication completes. If authentication fails, the supplicant port opens. Entering the **no dot1x supplicant controlled transient** global configuration command opens the supplicant port during the authentication period. This is the default behavior.

We strongly recommend using the **dot1x supplicant controlled transient** command on a supplicant switch when BPDU guard is enabled on the authenticator switch port with the **spanning-tree bpduguard enable** interface configuration command.



Note If you globally enable BPDU guard on the authenticator switch by using the **spanning-tree portfast bpduguard default** global configuration command, entering the **dot1x supplicant controlled transient** command on the Supplicant switch does not prevent the BPDU violation.

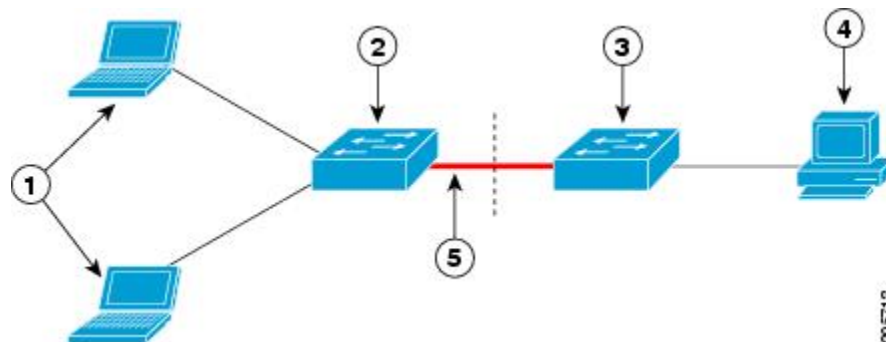
You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. Multihost mode is not supported on the authenticator switch interface.

When you reboot an authenticator switch with single-host mode enabled on the interface, the interface may move to err-disabled state before authentication. To recover from err-disabled state, flap the authenticator port to activate the interface again and initiate authentication.

Use the **dot1x supplicant force-multicast** global configuration command on the supplicant switch for NEAT to work in all host modes.

- **Host Authorization:** Ensures that only traffic from authorized hosts (connecting to the switch with supplicant) is allowed on the network. The switches use CISP to send the MAC addresses connecting to the supplicant switch to the authenticator switch.
- **Auto enablement:** Automatically enables trunk configuration on the authenticator switch, allowing user traffic from multiple VLANs coming from supplicant switches. Configure the `cisco-av-pair as device-traffic-class=switch` at the ISE. (You can configure this under the *group* or the *user* settings.)

Figure 5: Authenticator and Supplicant Switch Using CISP



1	Workstations (clients)
2	Supplicant switch (outside wiring closet)
3	Authenticator switch
4	Cisco ISE

5

Trunk port

**Note**

The **switchport nonegotiate** command is not supported on supplicant and authenticator switches with NEAT. This command should not be configured at the supplicant side of the topology. If configured on the authenticator side, the internal macros will automatically remove this command from the port.

Guidelines and Limitations

The following are guidelines and limitations for configuring and using NEAT.

- A Radius server such as Cisco's Identity Server Engine (ISE) is required.
- CISP and NEAT are supported only on L2 ports, not on L3 ports.
- NEAT and 802.1x are not supported on EtherChannel ports.
- NEAT is not supported on dynamic ports.
- MACsec is supported with NEAT.
- NEAT can operate with PTP.
- MAB and NEAT are mutually exclusive. You cannot enable MAB when NEAT is enabled on an interface, and you should not enable NEAT when MAB is enabled on an interface.

Configure an Authenticator Switch with NEAT

Configuring this feature requires that one switch outside a wiring closet is configured as a supplicant and is connected to an authenticator switch.

**Note**

- The *cisco-av-pairs* must be configured as *device-traffic-class=switch* on the ISE, which sets the interface as a trunk after the supplicant is successfully authenticated.

Beginning in privileged EXEC mode, follow these steps to configure a switch as an authenticator:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **interface *interface-id***
5. **switchport mode access**
6. **authentication port-control auto**
7. **dot1x pae authenticator**

8. spanning-tree portfast
9. end

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	cisp enable Example: <pre>Device(config)# cisp enable</pre>	Enables CISP.
Step 4	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/2</pre>	Specifies the port to be configured, and enters interface configuration mode.
Step 5	switchport mode access Example: <pre>Device(config-if)# switchport mode access</pre>	Sets the port mode to access .
Step 6	authentication port-control auto Example: <pre>Device(config-if)# authentication port-control auto</pre>	Sets the port-authentication mode to auto.
Step 7	dot1x pae authenticator Example:	Configures the interface as a port access entity (PAE) authenticator.

	Command or Action	Purpose
	Device(config-if)# dot1x pae authenticator	
Step 8	spanning-tree portfast Example: Device(config-if)# spanning-tree portfast trunk	Enables the interface to quickly transition to spanning-tree forwarding state for an interface which is a member of multiple VLANs. Use this command only when you are sure that the switch-to-switch connection is not part of a Layer2 loop.
Step 9	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configure a Supplicant Switch with NEAT

Beginning in privileged EXEC mode, follow these steps to configure a switch as a supplicant:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **eap profile** *profile-name*
5. **method** *type*
6. **exit**
7. **dot1x credentials** *profile*
8. **username** *suppswitch*
9. **password** *password*
10. **dot1x supplicant force-multicast**
11. **interface** *interface-id*
12. **switchport trunk encapsulation dot1q**
13. **switchport mode trunk**
14. **dot1x pae supplicant**
15. **dot1x credentials** *profile-name*
16. **dot1x supplicant eap profile** *profile-name*
17. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	cisp enable Example: Device(config)# cisp enable	Enables CISP.
Step 4	eap profile <i>profile-name</i> Example: Device(config)# eap profile CISP	Creates an Extensible Authentication Protocol (EAP) profile and enters EAP profile configuration mode.
Step 5	method <i>type</i> Example: Device(config-eap-profile)# method md5	Specifies the EAP authentication method.
Step 6	exit Example: Device(config-eap-profile)# exit	Exits EAP profile configuration mode.
Step 7	dot1x credentials <i>profile</i> Example: Device(config)# dot1x credentials test	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 8	username <i>suppswitch</i> Example: Device(config)# username suppswitch	Creates a username.

	Command or Action	Purpose
Step 9	password <i>password</i> Example: Device(config)# password myswitch	Creates a password for the new username.
Step 10	dot1x supplicant force-multicast Example: Device(config)# dot1x supplicant force-multicast	Forces the switch to send only multicast EAPOL packets when it receives either unicast or multicast packets. This also allows NEAT to work on the supplicant switch in all host modes.
Step 11	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/1	Specifies the port to be configured, and enters interface configuration mode.
Step 12	switchport trunk encapsulation dot1q Example: Device(config-if)# switchport trunk encapsulation dot1q	Sets the port to trunk mode.
Step 13	switchport mode trunk Example: Device(config-if)# switchport mode trunk	Configures the interface as a VLAN trunk port.
Step 14	dot1x pae supplicant Example: Device(config-if)# dot1x pae supplicant	Configures the interface as a port access entity (PAE) supplicant.
Step 15	dot1x credentials <i>profile-name</i> Example: Device(config-if)# dot1x credentials test	Attaches the 802.1x credentials profile to the interface.
Step 16	dot1x supplicant eap profile <i>profile-name</i> Example: Device(config-if)# dot1x supplicant eap profile cisp	Assigns the EAP-TLS profile to the 802.1X interface.

	Command or Action	Purpose
Step 17	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Configuration

Use the following show commands to verify information about Client Information Signalling Protocol (CISP) and Network Edge Access Topology (NEAT) configuration:

- show cisp interface <interface name>
- show cisp clients
- show cisp summary
- show cisp registrations

Following is example output for **show cisp** commands. GigabitEthernet 1/0/1 is configured as Authenticator, and GigabitEthernet 1/0/2 is configured as Supplicant.

```
Auth# show cisp interface Gi1/0/2
```

```
CISP Status for interface Gi1/0/2
```

```
-----
Version: 1
Mode: Supplicant Peer
Mode: Authenticator
Supp State: Idle
```

```
Auth# show cisp clients
```

```
Authenticator Client Table:
```

```
-----
MAC Address VLAN Interface
```

```
-----
0050.5695.4de8 1 Gi1/0/10
6c03.09e7.3947 1 Gi1/0/10
6c03.09e7.3954 11 Gi1/0/10
6c03.09e7.4485 1 Gi1/0/10
9077.ee4a.8567 1 Gi1/0/10
e41f.7ba1.bbd4 1 Gi1/0/10
```

```
Supplicant Client Table:
```

```
-----
MAC Address VLAN Interface
```

```
-----
9077.ee4a.856b 11 Vl11
9077.ee4a.8572 1 Ap1/1
e41f.7bc7.2f03 1 Gi1/0/9
```

```
Auth# show cisp summary
```

```
CISP is running on the following interface(s):
```

```

Gil/0/2 (Authenticator)

Supp# show cisp summary

CISP is running on the following interface(s):
-----
Gil/0/1 (Supplicant)

Auth# show cisp registrations

Interface(s) with CISP registered user(s):
-----
Gil/0/2
Auth Mgr (Authenticator)

Supp# show cisp registration

Interface(s) with CISP registered user(s):
-----
Gil/0/1
802.1x Sup (Supplicant)
    
```

Use the following debug commands to troubleshoot CISP and NEAT:

- debug access-session errors
- debug access-session event
- debug dot1x errors
- debug dot1x packets
- debug dot1x events

Feature History

Feature Name	Release	Feature Information
Network Edge Access Topology (NEAT)	Cisco IOS XE 17.8.1	Initial support on Cisco Catalyst IE9300 Rugged Series Switches



CHAPTER 4

Layer 2 Network Address Translation

- [Layer 2 Network Address Translation, on page 103](#)
- [Guidelines and Limitations, on page 106](#)
- [NAT Performance and Scalability, on page 108](#)
- [Configure Layer 2 NAT, on page 108](#)
- [Verify the Configuration, on page 109](#)
- [Basic Inside-to-Outside Communications: Example, on page 111](#)
- [Duplicate IP Addresses Example, on page 113](#)

Layer 2 Network Address Translation

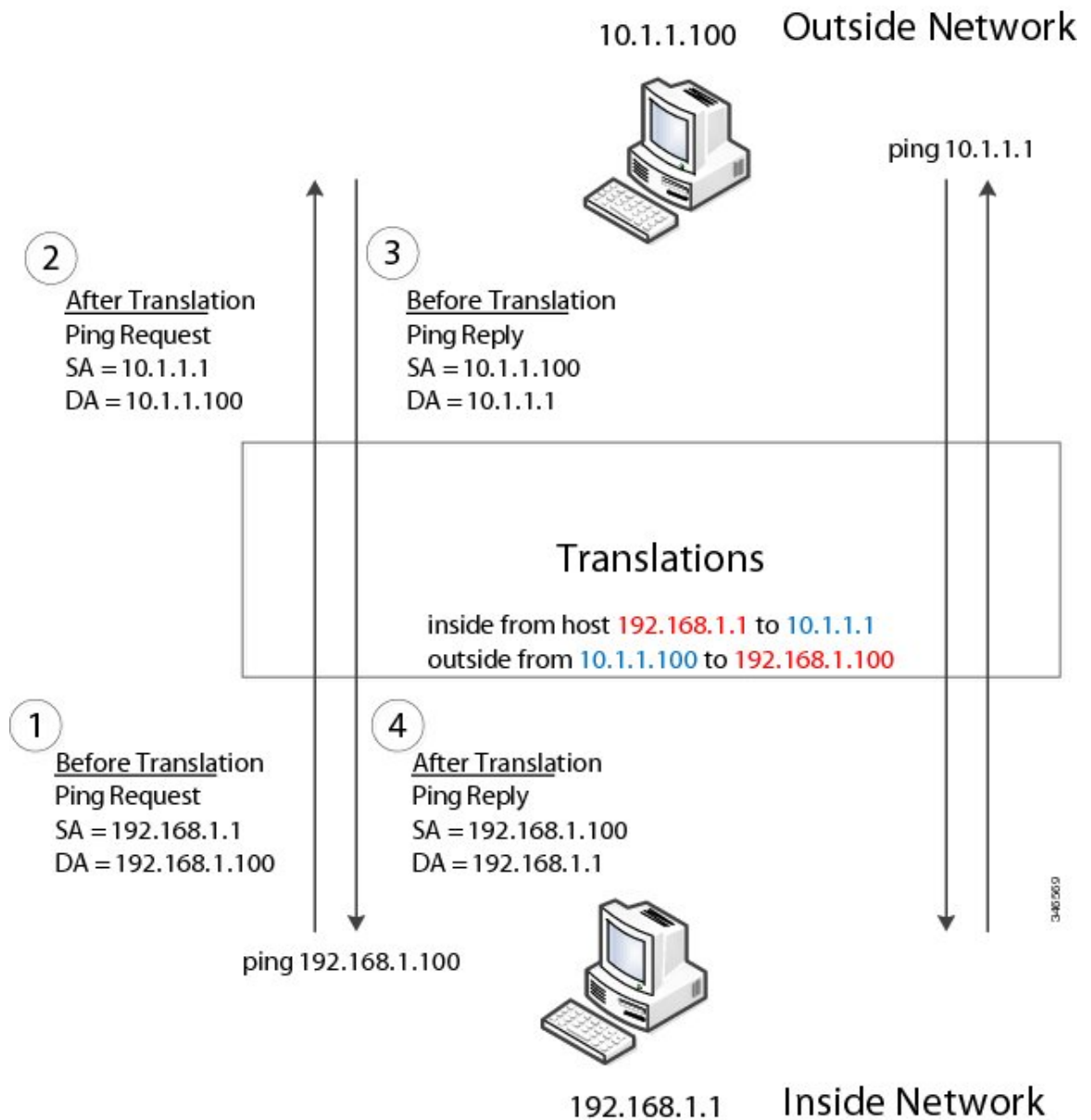
One-to-one Layer 2 NAT (Network Address Translation) is a service that allows the assignment of a unique public IP address to an existing private IP address (end device). The assignment enables the end device to communicate on both the private and public subnets. This service is configured in a NAT-enabled device and is the public “alias” of the IP address that is physically programmed on the end device. This is typically represented by a table in the NAT device.

Layer 2 NAT uses a table to translate IPv4 addresses both public-to-private, and private-to-public at line rate. Layer 2 NAT is a hardware-based implementation that provides the same high level of (bump-on-the-wire) wire-speed performance. This implementation also supports multiple VLANs through the NAT boundary for enhanced network segmentation.

In the following example, Layer 2 NAT translates addresses between sensors on a 192.168.1.x network and a line controller on a 10.1.1.x network.

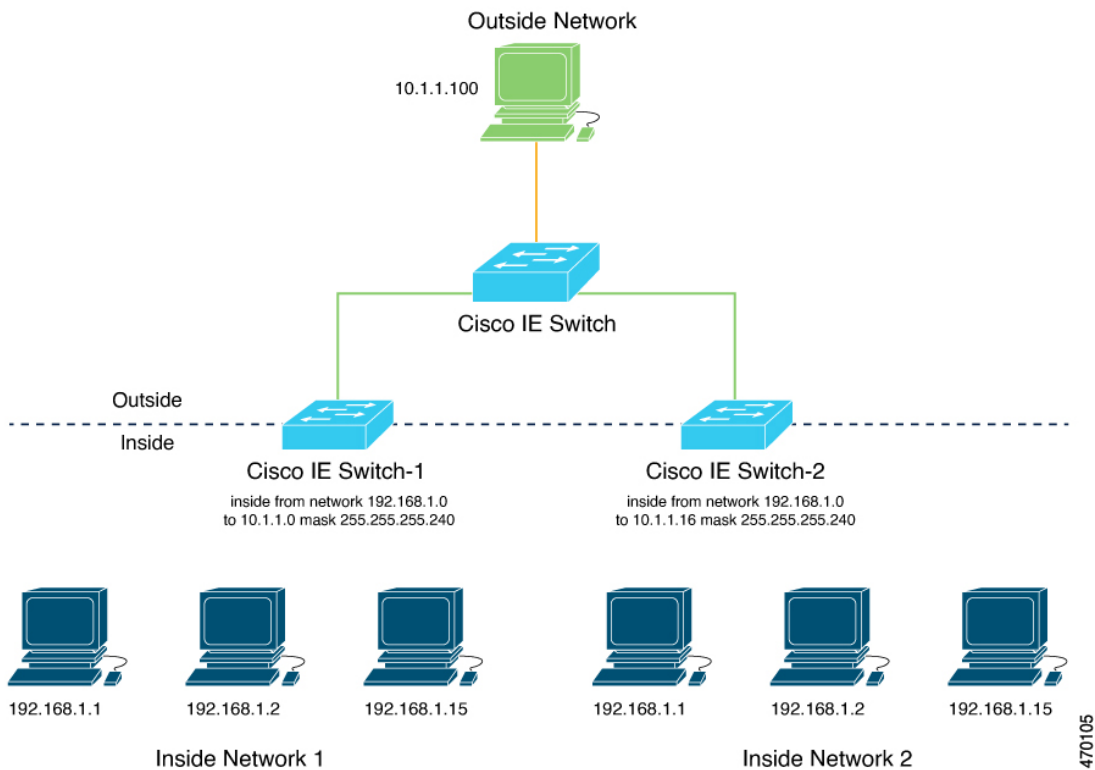
1. The 192.168.1.x network is the inside/internal IP address space and the 10.1.1.x network is the outside or external IP address space.
2. The sensor at 192.168.1.1 sends a ping request to the line controller by using an “inside” address, 192.168.1.100.
3. Before the packet leaves the internal network, Layer 2 NAT translates the source address (SA) to 10.1.1.1 and the destination address (DA) to 10.1.1.100.
4. The line controller sends a ping reply to 10.1.1.1.
5. When the packet is received on the internal network, Layer 2 NAT translates the source address to 192.168.1.100 and the destination address to 192.168.1.1.

Figure 6: Translating Addresses Between Networks



For large numbers of nodes, you can quickly enable translations for all devices in a subnet. In the scenario shown in the following figure, addresses from Inside Network 1 can be translated to outside addresses in the 10.1.1.0/28 subnet, and addresses from Inside Network 2 can be translated to outside addresses in the 10.1.1.16/28 subnet. All addresses in each subnet can be translated with one command. The benefit of using subnet-based translations saves in Layer L2 NAT rules. The switch has limits on the number of Layer 2 NAT rules. A rule with a subnet allows for multiple end devices to be translated with a single rule.

Figure 7: Inside-Outside Address Translation



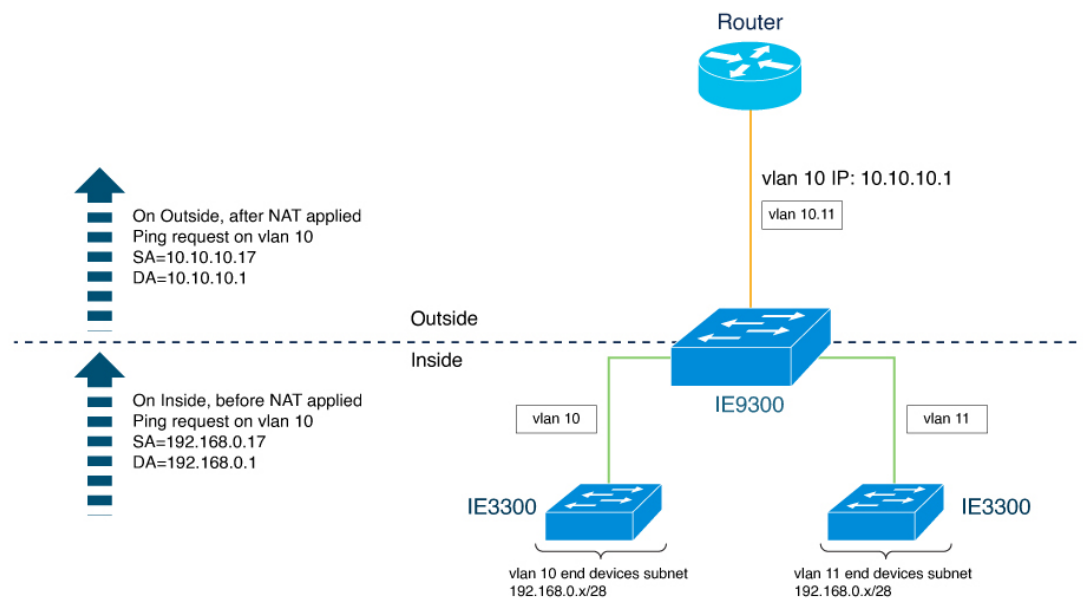
The following figure shows a switch at the aggregation layer forwarding Ethernet packets based on Layer 2 MAC Addresses. In this example, the router is the Layer 3 gateway for all subnets and VLANs.

The L2NAT instance definitions use the **network** command to define a translated row for multiple devices in the same subnet. In this case, it's a /28 subnet with last byte in the IP address starting with 16 and ending with 31. The gateway for the VLAN is the router with last byte of the IP address ending with .1. An outside host translation is provided for the router. The **network** command in the Layer 2NAT definition translates a subnet's worth of host with a single command, saving on Layer 2 NAT translation records.

The Gi1/1 uplink interface has Layer 2NAT translation instances for vlan10 and vlan 11 subnets. Interfaces can support multiple Layer 2 NAT instance definitions.

The downstream switches are examples of access layer switches which do not perform L2NAT and rely on the upstream aggregation layer switch to do it.

Figure 8: NAT on the Switch



470104

The following example shows the NAT configuration for the preceding diagram:

```
!
l2nat instance Subnet10-NAT
instance-id 1
permit all
fixup all
outside from host 10.10.10.1 to 192.168.0.1
inside from network 192.168.0.0 to 10.10.10.16 mask 255.255.255.240
!
l2nat instance Subnet11-NAT
instance-id 1
permit all
fixup all
outside from host 10.10.11.1 to 192.168.0.1
inside from network 192.168.0.0 to 10.10.11.16 mask 255.255.255.240
!
interface GigabitEthernet1/1
switchport mode trunk
l2nat Subnet10-NAT 10
l2nat Subnet11-NAT 11
!
Interface vlan 1
ip address 10.10.1.2
```

Guidelines and Limitations

The following list provides guidelines and limitations for using Layer 2 NAT with the Switch.



Note For scale information, see the section [NAT Performance and Scalability, on page 108](#) in this guide.

- Layer 2 NAT is supported on standalone switches.
- Layer 2 NAT is disabled by default; it becomes enabled when you configure it. See [Configure Layer 2 NAT, on page 108](#) in this guide.
- Layer 2 NAT applies only to unicast traffic. Untranslated unicast traffic, multicast traffic, and IGMP traffic are permitted.
- Layer 2 NAT is supported only on the uplink ports and available in both Network Essentials and Network Advantage licenses.
- Layer 2 NAT supports one-to-one mapping between external and internal IP addresses.
- Layer 2 NAT can be applied to uplink interfaces in access or trunk mode.
- Only IPv4 addresses for Layer 2 traffic can be translated.
- Supported subnet masks on inside network translation are /24, /25, /26, /27, /28, and /32 only.
- Outside translation rule supports only host translations.
- ARP does not work transparently across Layer 2 NAT; however, the switch changes the IP addresses embedded in the payload of IP packets for the protocols to work. Embedded IP addresses are not translated.
- Statistics for debugging include the following statistics: entries for each translation, translated total ingress and egress for each instance, and for each interface. Also included are ARP fixup stats and the number of translations entries allocated in hardware.
- Layer 2 NAT does not support one-to-many and many-to-one IP address mapping.
- Layer 2 NAT cannot save on public IP addresses because public-to-private is a 1:1 translation. It is not 1:N NAT.
- If you configure a translation for a Layer 2 NAT host, do not configure it as a DHCP client.
- When translating an inside address to an outside address using Layer 2 NAT, ensure that the translated IP address is not accessible in the global network.
- The management interface is behind the Layer 2 NAT function. Therefore this interface should not be on the private network VLAN. If it is on the private network VLAN, assign an inside address and configure an inside translation.
- Because Layer 2 NAT is designed to separate outside and inside addresses, we recommend that you do not configure addresses of the same subnet as both outside and inside addresses.
- Layer 2 NAT is only for Layer 2 traffic; do not use it for packets undergoing routing
- Layer 2 NAT does not translate packets destined for CPU and packets coming from CPU. Management traffic should be on a different VLAN from the private network VLAN.
- Layer 2 NAT counters are not based on ports. When the same Layer 2 NAT instance is applied to multiple interfaces, the corresponding Layer 2 NAT counters will be displayed for all those interfaces.

NAT Performance and Scalability

Layer 2 NAT translation and forwarding are performed in the hardware at line rate. The number of Layer 2 NAT rules that are supported depends on the number of hardware entries that can be supported in hardware.

Scale depends on the number of inside/outside combinations. The following list provides scale examples.

- An instance with only inside rules can have a total of 128 translation rules.
- Multiple instances with one inside rule can have a total of 128 such instances applied to 128 different VLANs.
- Multiple instances with one inside rule and one outside rule can have a maximum of 64 instances.
- A single instance with one outside rule can have a maximum of 100 inside rules. The number of inside rules that can be supported reduces with increase in the outside rules.



Note We recommended that you use network translation rules to save on the number of rules.

Configure Layer 2 NAT

You must configure Layer 2 NAT instances that specify the address translations. Attach Layer 2 NAT instances to physical Ethernet interfaces, and configure which VLAN or VLANs the instances will be applied to. Layer 2 NAT instances can be configured from management interfaces (CLI/SNMP). You can view detailed statistics about the packets that are sent and received. See the section [Verify the Configuration, on page 109](#) in this guide.

To configure Layer 2 NAT, follow these steps. Refer to the examples in [Basic Inside-to-Outside Communications: Example, on page 111](#) and [Duplicate IP Addresses Example, on page 113](#) in this guide for more details.

Procedure

-
- Step 1** Enter global configuration mode:
- ```
configure terminal
```
- Step 2** Create a new Layer 2 NAT instance:
- ```
l2nat instance instance_name
```
- After creating an instance, you use this same command to enter the submode for that instance.
- Step 3** Translate an inside address to an outside address:
- ```
inside from [host | range | network] original ip [mask] number | mask
```
- You can translate a single host address, a range of host addresses, or all the addresses in a subnet. Translate the source address for outbound traffic and the destination address for inbound traffic.

**Step 4** Translate an outside address to an inside address:

**outside from** *[host | range | network ] original ip to translated ip [mask ] number | mask*

You can translate a single host address, a range of host addresses, or the addresses in a subnet. Translate the destination address for outbound traffic and the source address for inbound traffic.

**Step 5** Exit config-l2nat mode:

**exit**

**Step 6** Access interface configuration mode for the specified interface (uplink ports only on the IE 3400):

**interface** *interface-id*

**Step 7** Apply the specified Layer 2 NAT instance to a VLAN or VLAN range. If this parameter is missing, the Layer 2 NAT instance applies to the native VLAN.

**l2nat** *instance\_name [vlan | vlan\_range ]*

**Step 8** Exit interface configuration mode:

**end**

## Verify the Configuration

### Procedure

Perform the following commands to verify the Layer 2 NAT configuration.

| Command                                                                                         | Purpose                                                                                        |
|-------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>show l2nat instance</b>                                                                      | Displays the configuration details for a specified Layer 2 NAT instance.                       |
| <b>show l2nat interface</b>                                                                     | Displays the configuration details for Layer 2 NAT instances on one or more interfaces.        |
| <b>show l2nat statistics</b>                                                                    | Displays the Layer 2 NAT statistics for all interfaces.                                        |
| <b>show l2nat statistics interface</b>                                                          | Displays the Layer 2 NAT statistics for a specified interface.                                 |
| <b>debug l2nat</b>                                                                              | Enables showing real-time Layer 2 NAT configuration details when the configuration is applied. |
| <b>show platform hardware fed switch 1 fwd-asic resource tcam table pbr record 0 format 0 -</b> | Displays the hardware entries.                                                                 |

| Command                                                                                      | Purpose                                     |
|----------------------------------------------------------------------------------------------|---------------------------------------------|
| <b>-show platform hardware fed switch active fwd-asic resource tcam utilization   in PBR</b> | Displays the hardware resource utilization. |

The following is an example of output of the **show l2nat instance** and the **show l2nat statistics** commands:

```
switch#show l2nat instance
l2nat instance test
fixup : all
outside from host 10.10.10.200 to 192.168.1.200
inside from host 192.168.1.1 to 10.10.10.1
l2nat instance test2
fixup : all
inside from host 1.1.1.1 to 2.2.2.2
outside from host 2.2.2.200 to 1.1.1.200

Switch#show l2nat interface
FOLLOWING INSTANCE(S) AND VLAN(S) ATTACHED TO ALL INTERFACES
=====
l2nat Gi1/1 test
=====

Switch#show l2nat statistics

STATS FOR INSTANCE: test (IN PACKETS)

TRANSLATED STATS (IN PACKETS)
=====
INTERFACE DIRECTION VLAN TRANSLATED
Gi1/1 EGRESS 50 0
Gi1/1 INGRESS 50 0
=====

PROTOCOL FIXUP STATS (IN PACKETS)
=====
INTERFACE DIRECTION VLAN ARP
Gi1/1 REPLY 50 0
Gi1/1 REQUEST 50 0
=====

PER TRANSLATION STATS (IN PACKETS)
=====
TYPE DIRECTION SA/DA ORIGINAL IP TRANSLATED IP COUNT
OUTSIDE INGRESS SA 10.10.10.200 192.168.1.200 0
OUTSIDE EGRESS DA 192.168.1.200 10.10.10.200 0
INSIDE EGRESS SA 192.168.1.1 10.10.10.1 0
INSIDE INGRESS DA 10.10.10.1 192.168.1.1 0
=====

TOTAL TRANSLATIONS ENTRIES IN HARDWARE: 4
TOTAL INSTANCES ATTACHED : 1
=====
GLOBAL NAT STATISTICS
=====
Total Number of TRANSLATED NAT Packets = 0
Total Number of ARP FIX UP Packets = 0
```

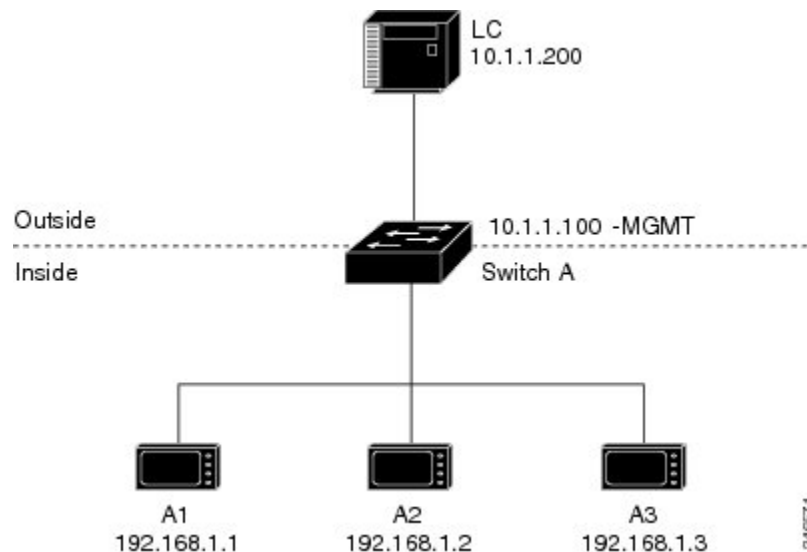


ad

## Basic Inside-to-Outside Communications: Example

In this example, A1 must communicate with a logic controller (LC) that is directly connected to the uplink port. A Layer 2 NAT instance is configured to provide an address for A1 on the outside network (10.1.1.1) and an address for the LC on the inside network (192.168.1.250).

**Figure 9: Basic Inside-to-Outside Communications**



Now this communication can occur:

1. A1 sends an ARP request: SA: 192.168.1.1 DA: 192.168.1.250.
2. Cisco Switch A fixes up the ARP request: SA: 10.1.1.1 DA: 10.1.1.200.
3. LC receives the request and learns the MAC Address of 10.1.1.1.
4. LC sends a response: SA: 10.1.1.200 DA: 10.1.1.1.
5. Cisco Switch A fixes up the ARP response: SA: 192.168.1.250 DA: 192.168.1.1.
6. A1 learns the MAC address for 192.168.1.250, and communication starts.



### Note

- The management interface of the switch must be on a different VLAN from the inside network 192.168.1.x.
- See the section [Basic Inside-to-Outside Communications: Configuration, on page 112](#) for the tasks to configure the example in this section.

## Basic Inside-to-Outside Communications: Configuration

This section contains the steps to configure inside-to-outside communications as described in the preceding section. You create the Layer 2 NAT instance, add two translation entries, and then apply the instance to the interface. ARP fixups are enabled by default.

### Before you begin

Read and understand the content in the section [Basic Inside-to-Outside Communications: Example, on page 111](#).

### Procedure

---

**Step 1** Enter configuration mode.

**Example:**

```
switch# configure
```

**Step 2** Create a new Layer 2 NAT instance called A-LC.

**Example:**

```
switch(config)# l2nat instance A-LC
```

**Step 3** Translate A1's inside address to an outside address.

**Example:**

```
switch(config-l2nat)# inside from host 192.168.1.1 to 10.1.1.1
```

**Step 4** Translate A2's inside address to an outside address.

**Example:**

```
switch(config-l2nat)# inside from host 192.168.1.2 to 10.1.1.2
```

**Step 5** Translate A3's inside address to an outside address.

**Example:**

```
switch(config-l2nat)# inside from host 192.168.1.3 to 10.1.1.3
```

**Step 6** Translate the LC outside address to an inside address.

**Example:**

```
switch(config-l2nat)# outside from host 10.1.1.200 to 192.168.1.250
```

**Step 7** Exit config-l2nat mode.

**Example:**

```
switch(config-l2nat)# exit
```

**Step 8** Access interface configuration mode for the uplink port.

**Example:**

```
switch(config)# interface Gi1/1
```

**Step 9** Apply this Layer 2 NAT instance to the native VLAN on this interface.

**Example:**

```
switch(config-if) # l2nat A-1C
```

**Note**

For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

```
l2nat instance vlan
```

**Step 10** Return to privileged EXEC mode.

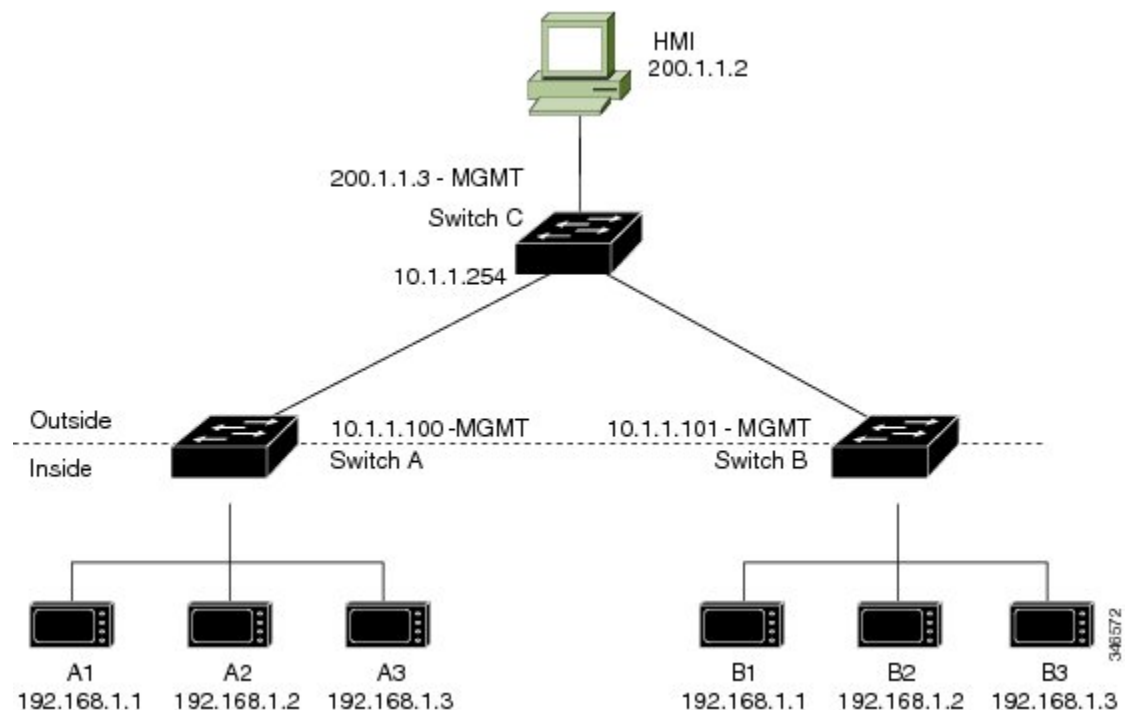
**Example:**

```
switch# end
```

## Duplicate IP Addresses Example

In this scenario, two machine nodes are preconfigured with addresses in the 192.168.1.x space. Layer 2 NAT translates these addresses to unique addresses on separate subnets of the outside network. In addition, for machine-to-machine communications, the Node A machines need unique addresses on the Node B space and the Node B machines need unique addresses in the Node A space.

**Figure 10: Duplicate IP Addresses**



- Switch C needs an address in the 192.168.1.x space. When packets come into Node A or Node B, the 10.1.1.254 address of Switch C is translated to 192.168.1.254. When packets leave Node A or Node B, the 192.168.1.254 address of Switch C is translated to 10.1.1.254.

- Node A and Node B machines need unique addresses in the 10.1.1.x space. For quick configuration and ease of use, the 10.1.1.x space is divided into subnets: 10.1.1.0, 10.1.1.16, 10.1.1.32, and so on. Each subnet can then be used for a different node. In this example, 10.1.1.16 is used for Node A, and 10.1.1.32 is used for Node B.
- Node A and Node B machines need unique addresses to exchange data. The available addresses are divided into subnets. For convenience, the 10.1.1.16 subnet addresses for the Node A machines are translated to 192.168.1.16 subnet addresses on Node B. The 10.1.1.32 subnet addresses for the Node B machines are translated to 192.168.1.32 addresses on Node A.
- Machines have unique addresses on each network:

Table 10: Translated IP Addresses

| Node                           | Address in Node A | Address in Outside Network | Address in Node B |
|--------------------------------|-------------------|----------------------------|-------------------|
| Switch A network address       | 192.168.1.0       | 10.1.1.16                  | 192.168.1.16      |
| A1                             | 192.168.1.1       | 10.1.1.17                  | 192.168.1.17      |
| A2                             | 192.168.1.2       | 10.1.1.18                  | 192.168.1.18      |
| A3                             | 192.168.1.3       | 10.1.1.19                  | 192.168.1.19      |
| Cisco Switch B network address | 192.168.1.32      | 10.1.1.32                  | 192.168.1.0       |
| B1                             | 192.168.1.33      | 10.1.1.33                  | 192.168.1.1       |
| B2                             | 192.168.1.34      | 10.1.1.34                  | 192.168.1.2       |
| B3                             | 192.168.1.35      | 10.1.1.35                  | 192.168.1.3       |
| Switch C                       | 192.168.1.254     | 10.1.1.254                 | 192.168.1.254     |

## Duplicate IP Addresses Configuration: Switch A

This section provides the steps for configuring Layer 2 NAT to translate the duplicated IP address of one machine node in an inside network to a unique address on a subnet of an outside network. This procedure is for Switch A in the section [Duplicate IP Addresses Example, on page 113](#).

### Before you begin

Read and understand the content in the section [Duplicate IP Addresses Example, on page 113](#).

### Procedure

**Step 1** Enter global configuration mode.

#### Example:

```
switch# configure
```

**Step 2** Create a new Layer 2 NAT instance called A-Subnet.

**Example:**

```
switch(config)# l2nat instance A-Subnet
```

**Step 3** Translate the Node A machines' inside addresses to addresses in the 10.1.1.16 255.255.255.240 subnet.

**Example:**

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.16 mask 255.255.255.240
```

**Step 4** Translate the outside address of Switch C to an inside address.

**Example:**

```
switch(config-l2nat)# outside from host 10.1.1.254 to 192.168.1.254
```

**Step 5** Translate the Node B machines' outside addresses to their inside addresses.

**Example:**

```
switch(config-l2nat)# outside from host 10.1.1.32 to 192.168.1.32
outside from host 10.1.1.33 to 192.168.1.33
outside from host 10.1.1.34 to 192.168.1.34
outside from host 10.1.1.35 to 192.168.1.35
```

**Step 6** Exits config-l2nat mode.

**Example:**

```
switch(config-l2nat)# exit
```

**Step 7** Access interface configuration mode for the uplink port.

**Example:**

```
switch(config)# interface Gi1/1
```

**Step 8** Apply this Layer 2 NAT instance to the native VLAN on this interface.

**Example:**

```
switch(config-if)# l2nat A-Subnet
```

**Note**

For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

```
l2nat instance vlan
```

**Step 9** Return to privileged EXEC mode.

**Example:**

```
switch# end
```

---

### What to do next

Configure Layer 2 NAT to translate the duplicated IP address of Switch B in the section [Duplicate IP Addresses Example, on page 113](#). See [Duplicate IP Addresses Configuration: Switch B, on page 116](#).

## Duplicate IP Addresses Configuration: Switch B

This section provides the steps for configuring Layer 2 NAT to translate the duplicated IP address of one machine node in an inside network to a unique address on a subnet of an outside network. This procedure is for Switch B in the section [Duplicate IP Addresses Example, on page 113](#).

### Before you begin

Read and understand the content in the section [Duplicate IP Addresses Example, on page 113](#).

### Procedure

**Step 1** Enter global configuration mode.

**Example:**

```
switch# configure
```

**Step 2** Create a new Layer 2 NAT instance called B-Subnet.

**Example:**

```
switch(config)# l2nat instance B-Subnet
```

**Step 3** Translate the Node B machines' inside addresses to addresses in the 10.1.1.32 255.255.255.240 subnet.

**Example:**

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.32 255.255.255.240
```

**Step 4** Translate the outside address of Switch C to an inside address.

**Example:**

```
switch(config-l2nat)# outside from host 10.1.1.254 to
```

**Step 5** Translate the Node A machines' outside addresses to their inside addresses.

**Example:**

```
switch(config-l2nat)# outside from host 10.1.1.16 to 192.168.1.16
outside from host 10.1.1.17 to 192.168.1.17
outside from host 10.1.1.18 to 192.168.1.18
outside from host 10.1.1.19 to 192.168.1.19
```

**Step 6** Exit config-l2nat mode.

**Example:**

```
switch(config-l2nat)# exit
```

**Step 7** Access interface configuration mode for the uplink port.

**Example:**

```
switch(config)# interface Gi1/1
```

**Step 8** Apply this Layer 2 NAT instance to the native VLAN on this interface.

**Example:**

```
switch(config-if)# l2nat name1
```

**Note**

For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

*l2nat instance vlan*

**Step 9** Show the configuration details for the specified Layer 2 NAT instance.

**Example:**

```
switch# show l2nat instance name1
```

**Step 10** Show Layer 2 NAT statistics.

**Example:**

```
switch# show l2nat statistics
```

**Step 11** Return to privileged EXEC mode.

**Example:**

```
switch# end
```

---

