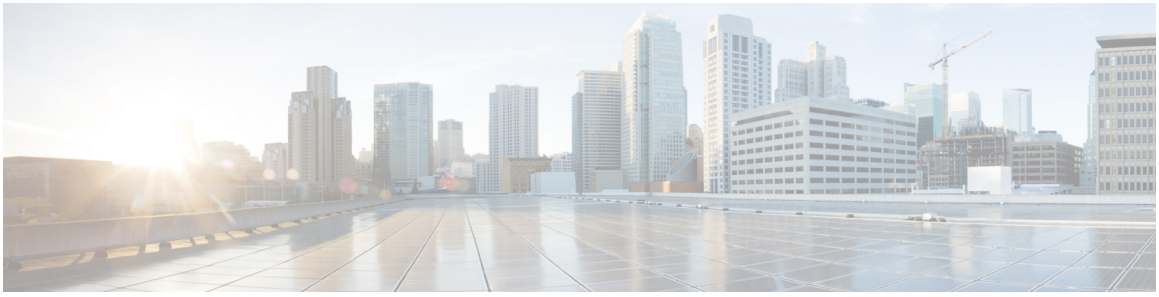
Security Configuration Guide, Cisco Catalyst IE3x00 and IE3100 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

First Published: 2020-08-10

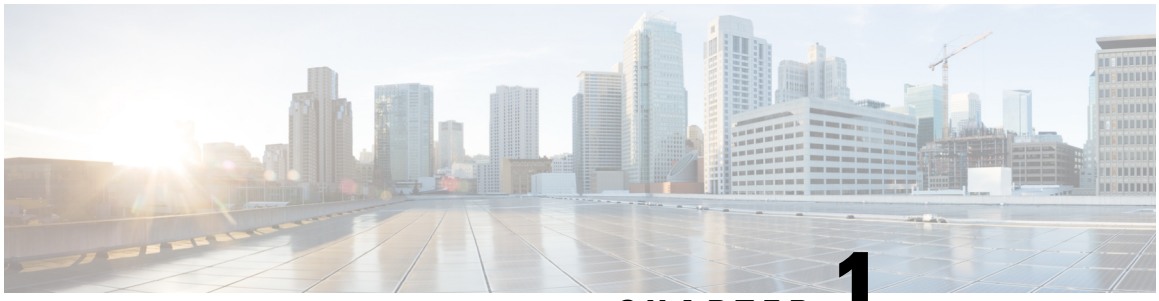
Last Modified: 2024-04-05

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

Note: 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 RFP documentation, or language that is used by a referenced third-party product.



CONTENTS



CHAPTER 1

Configuring RADIUS

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 Secure Access Control Server Version 3.0), 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.

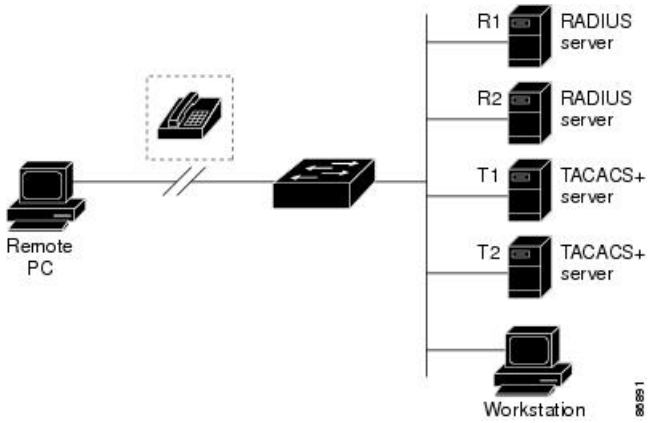
RADIUS Overview

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

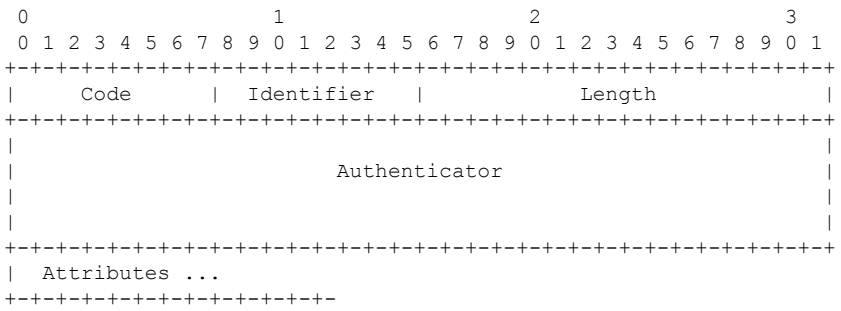
For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- 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 [1]	Cisco VSA
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)."

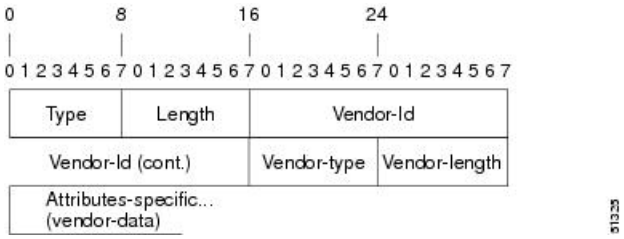
Attribute 26 contains the following three elements:

- 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	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. **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.

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	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. **Note** This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.
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

	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
	`Device# configure terminal`	
Step 3	**radius server** *server name* **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**}*ip address*{ **auth-port** *port number* | **acct-port** *port number*} **Example:** `Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612`	(Optional) Specifies the RADIUS server parameters. For **auth-port** *port-number*, specify the UDP destination port for authentication requests. The default is 1645. The range is 0 to 65536. For **acct-port** *port-number*, specify the UDP destination port for accounting requests. The default is 1646.
Step 5	**key** *string* **Example:** `Device(config-radius-server)# key rad123`	(Optional) For **key** *string*, 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** *value* **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** *seconds* **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 configur the **ip http authentication aaa** global configuration command. 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

	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	**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. • To create a default list that is used when a named list is *not* 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 *list-name*, specify a character string to name the list you are creating. • For *method1...*, specify the actual method the authentication algorithm tries. The additional methods

	Command or Action	Purpose
		of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • *enable*—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. • *group radius*—Use RADIUS authentication. Before you can use this authentication method, you must configure the RADIUS server. • *line*—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. • *local*—Use the local username database for authentication. You must enter username information in the database. Use the **username** *name* **password** global configuration command. • *local-case*—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. • *none*—Do not use any authentication for login.
Step 5	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*] **Example:** `Device(config)# line 1 4`	Enters line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 6	**login authentication** {**default** | *list-name*} **Example:** `Device(config-line)# login authentication default`	Applies the authentication list to a line or set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 7	**end** **Example:** `Device(config-line)# end`	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

	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** *name* **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**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number* **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** *string* **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:** `Device(config-radius-server)# end`	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 priviledged 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

	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	**aaa authorization network** *authorization-list* **radius** **Example:** `Device(config)# aaa authorization network list1 radius`	Configures the device for user RADIUS authorization for all network-related service requests.
Step 4	**aaa authorization exec** *authorization-list* **radius** **Example:**	Configures the device for user RADIUS authorization if the user has privileged EXEC access.

	Command or Action	Purpose
	`Device(config)# aaa authorization exec list1 radius`	The **exec** keyword might return user profile information (such as **autocommand** information).
Step 5	**end** **Example:** `Device(config)# end`	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

	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	**aaa accounting network** *accounting-list***start-stop radius** **Example:** `Device(config)# aaa accounting network start-stop`	Enables RADIUS accounting for all network-related service requests.

	Command or Action	Purpose
	radius	
Step 4	**aaa accounting exec** *accounting-list* **start-stop radius** **Example:** `Device(config)# aaa accounting exec acc-list start-stop radius`	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:** `Device(config)# end`	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

	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** *server name* **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** *string* **Example:**	Specifies the shared secret text string used between the switch and all RADIUS servers.

	Command or Action	Purpose
	`Device(config-radius-server)# key your_server_key`	**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** *retries* **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 1 to 1000.
Step 6	**timeout** *seconds* **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

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

	Command or Action	Purpose
Step 3	**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. • (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

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

	Command or Action	Purpose
Step 3	**radius server** *server name* **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** } *ip address* **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** *string* **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

	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** *server_name* **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** } *ip address* [**auth-port** *auth_port_number* **acct-port** *acct_port_number*] **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. • **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** *dscp_acct_value* | **auth** *dscp_auth_value*} **Example:** `Device(config-radius-server)# dscp auth 10 acct 20`	Configures DSCP marking for authentication and accounting on the radius server. • **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 6	**key** *string* **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 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

	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	**aaa group server radius** *group_name* **Example:** `Device(config)# aaa group server radius abc`	Defines the RADIUS server group configuration and enters RADIUS server group configuration mode.
Step 4	**server name** *name* **Example:** `Device(config-sg-radius)# server name serv1`	Associates the RADIUS server to the server group.
Step 5	{**ip**	**ipv6**} **radius source-interface** *type number* **Example:** `Device(config-sg-radius)# ipv6 radius source-interface ethernet 0/0`	Specifies an interface to use for the source address in RADIUS server.
Step 6	**dscp** {**acct** *dscp_acct_value*	**auth** *dscp_auth_value*} **Example:**	Configures DSCP marking for authentication and accounting on the radius server group.

	Command or Action	Purpose
	`Device(config-sg-radius)# dscp auth 10 acct 20`	• **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:** `Device(config-radius-server)# end`	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

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

	Command or Action	Purpose
Step 3	**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.
Step 8	**auth-type** {**any** | **all** | **session-key**} **Example:** `Device(config-locsvr-da-radius)# auth-type any`	Specifies the type of authorization the device uses for RADIUS clients. The client must match all the configured attributes for authorization.
Step 9	**ignore server-key** **Example:** `Device(config-locsvr-da-radius)# ignore server-key`	(Optional) Configures the device to ignore the server-key.
Step 10	**exit** **Example:**	Exits dynamic authorization local server configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-locsvr-da-radius)# exit`	
Step 11	**authentication command bounce-port ignore** **Example:** `Device(config)# authentication command bounce-port ignore`	(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:** `Device(config)# authentication command disable-port ignore`	(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. Use standard CLI or SNMP commands to re-enable the port.
Step 13	**end** **Example:** `Device(config)# end`	Exits global configuration mode an 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.
debug cmdhd [**detail** | **error** | **events**]	Displays information for troubleshooting command headers.

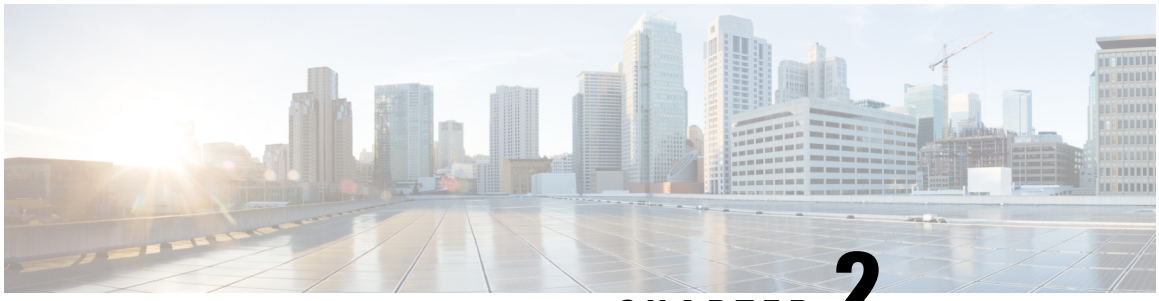
Feature History for RADIUS

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.

Feature	Feature Information	Release
DSCP marking on Radius servers	This feature allows you to configure Differentiated Services Code Point (DSCP) marking on RADIUS servers and RADIUS server groups using **dscp** command. The **radius-server dscp** command is used to configure DSCP marking for authentication and accounting on RADIUS servers in global configuration mode.	IE 3x00 and ESS 3300: Cisco IOS XE Cupertino 17.8.1
RADIUS	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.	IE 3x00: Cisco IOS XE Gibraltar 16.11.1 ESS 3300: Cisco IOS XE Gibraltar 16.9.1

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 2

Delayless IPDT

Information About Delayless IPDT

The Delayless IP Device Tracking (IPDT) feature allows faster processing of ARP packets in a network with IPDT enabled. Delayless IPDT is supported on all IE3x00 switches. Delayless IPDT does not require any configuration other than enabling IPDT, and there are no specific commands to verify Delayless IPDT.

IPDT uses the DHCP snooping and ARP snooping features to build a database of IP-to-MAC binding present in the switch. Without the Delayless IPDT feature, when IPDT is configured, all ARP packets are punted to the CPU for processing and then the packets are forwarded to the final destination from the CPU. This delays ARP delivery, which could cause communication errors between hosts or stop production.

With the Delayless IPDT feature, when IPDT is configured, the original ARP traffic is forwarded through hardware and only a copy of the ARP packets are sent to software for IP-MAC binding creation. This reduces the ARP delivery time. Delayless IPDT does not change IPv6 neighbor discovery behavior.

Delayless IPDT does not work if DAI (Dynamic ARP inspection) is enabled. DAI is a security feature that provides a mechanism to filter ARP requests and responses to prevent layer 2 attacks such as ARP cache poisoning. Filtering is done based on the DHCP snooping binding database or user configured ARP Access Control Lists (ACLs).

The following table summarizes how ARP packets are processed based on the IPDT and DAI configuration.

Configured Feature	ARP Packet Processing
Only IPDT enabled	ARP packets are forwarded through hardware and a copy is punted to CPU (Delayless IPDT). **Note** Copied packets are discarded after processing.
Only DAI enabled	ARP packets are punted to CPU for processing (no Delayless IPDT). With DAI enabled, ARP packets are delayed slightly as the CPU processes.

Configured Feature	ARP Packet Processing
IPDT and DAI enabled	ARP packets are punted to CPU for processing (no Delayless IPDT).

Guidelines and Limitations

- Delayless IPDT takes effect automatically when an IPDT policy is enabled on at least one interface or VLAN.

  This feature is enabled globally irrespective of which interface or VLAN has an IPDT policy attached.
- Delayless IPDT works in both access and trunk modes.
- Delayless IPDT does not work if the switch has DAI enabled on any VLAN.
- Delayless IPDT cannot be enabled or disabled per interface or VLAN.

Example IPDT Configuration

The Delayless IPDT feature does not have any specific CLI to configure it. You only need to have IPDT configured and enabled. The following example shows the basic commands for configuring an IPDT policy and attaching it to an interface or VLAN:

```
configure terminal
device-tracking policy test
 limit address-count <count>
 security-level glean
 tracking enable
exit
interface GigabitEthernet1/5
 device-tracking attach-policy test
 exit
vlan configuration 5
 device-tracking attach-policy test
exit
```

Verifying IPDT

There are no specific commands to verify Delayless IPDT. You can use the following IPDT **show** commands to display details about the IPDT database:

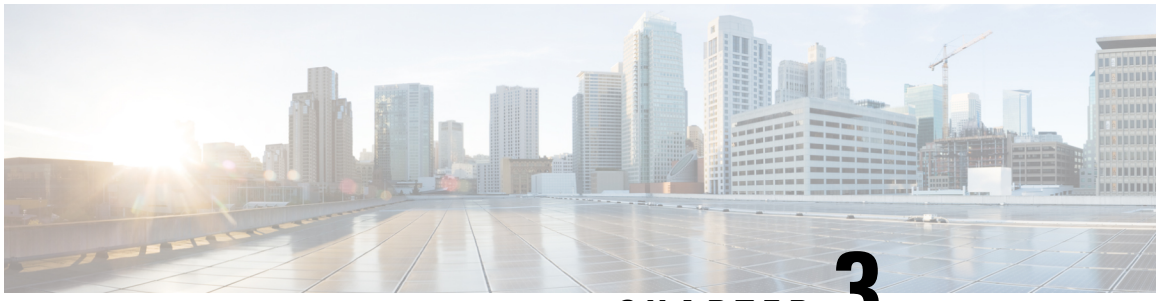
- show device-tracking database
- show device-tracking database interface <interfaceid>
- show device-tracking database details
- show device-tracking database vlanid <vlanid>

The following is an example of output for the **show device-tracking database interface** command:

```
Switch#show device-tracking database interface GigabitEthernet1/5
portDB has 4 entries for interface Gi1/5, 4 dynamic
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                  Link Layer Address     Interface  vlan
prlvl      age        state      Time left
ARP 3.1.1.10                               0000.1100.0004         Gi1/5      30
0005       12s        REACHABLE  297 s
ARP 3.1.1.9                                0000.1100.0003         Gi1/5      30
0005       17s        REACHABLE  289 s
ARP 3.1.1.8                                0000.1100.0002         Gi1/5      30
0005       21s        REACHABLE  292 s
ARP 3.1.1.7                                0000.1100.0001         Gi1/5      30
0005       25s        REACHABLE  276 s
```

Feature History

Feature Name	Release	Feature Information
Delayless IPDT	Cisco IOS XE 17.14.1	Initial support on IE3x00



CHAPTER 3

Configuring IPv6 First Hop Security

Prerequisites for First Hop Security in IPv6

You have configured the necessary IPv6 enabled SDM template.

Restrictions for First Hop Security in IPv6

- The following restrictions apply when applying FHS policies to EtherChannel interfaces (Port Channels):
  - A physical port with an FHS policy attached cannot join an EtherChannel group.
  - An FHS policy cannot be attached to an physical port when it is a member of an EtherChannel group.
- By default, a snooping policy has a security-level of guard. When such a snooping policy is configured on an access switch, external IPv6 Router Advertisement (RA) or Dynamic Host Configuration Protocol for IPv6 (DHCPv6) server packets are blocked, even though the uplink port facing the router or DHCP server/relay is configured as a trusted port. To allow IPv6 RA or DHCPv6 server messages, do the following:

- Apply an IPv6 RA-guard policy (for RA) or IPv6 DHCP-guard policy (for DHCP server messages) on the uplink port.
- Configure a snooping policy with a lower security-level, for example glean or inspect. However; configuring a lower security level is not recommended with such a snooping policy, because benefits of First Hop security features are not effective.

- Host and Guard configuration on the same node is not supported.
- For DHCPv6 guard to work, an SVI must be to configured on the corresponding vlan on the same switch.

Information about First Hop Security in IPv6

First Hop Security in IPv6 (FHS IPv6) is a set of IPv6 security features, the policies of which can be attached to a physical interface, or a VLAN. An IPv6 software policy database service stores and accesses these policies. When a policy is configured or modified, the attributes of the policy are stored or updated in the software policy database, then applied as was specified. The following IPv6 policies are currently supported:

- IPv6 Snooping Policy—IPv6 Snooping Policy acts as a container policy that enables most of the features available with FHS in IPv6.

Note IPv6 Snooping Policy feature is deprecated and the Switch Integrated Security Feature (SISF)-based device tracking feature replaces it. While the IPv6 Snooping Policy commands are still available on the CLI and the existing configuration continues to be supported, the commands will be removed from the CLI in a later release. For more information about the replacement feature, see the *Configuring SISF-Based Device Tracking* chapter in this guide.

- IPv6 FHS Binding Table Content—A database table of IPv6 neighbors connected to the switch is created from information sources such as Neighbor Discovery (ND) protocol snooping. This database, or binding, table is used by various IPv6 guard features (such as IPv6 ND Inspection) to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

Note IPv6 FHS Binding Table Content feature is supported through SISF-based device tracking. For more information, see the *Configuring SISF-Based Device Tracking* chapter in this guide.

- IPv6 Neighbor Discovery Inspection—IPv6 ND inspection learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables. IPv6 ND inspection analyzes neighbor discovery messages in order to build a trusted binding table database and IPv6 neighbor discovery messages that do not conform are dropped. An ND message is considered trustworthy if its IPv6-to-Media Access Control (MAC) mapping is verifiable.

  This feature mitigates some of the inherent vulnerabilities of the ND mechanism, such as attacks on DAD, address resolution, router discovery, and the neighbor cache.

Note Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it. While the IPv6 ND Inspection commands are still available on the CLI and the existing configuration continues to be supported, the commands will be removed from the CLI in a later release. For more information about the replacement feature, see the *Configuring SISF-Based Device Tracking* chapter in this guide.

- IPv6 Router Advertisement Guard—The IPv6 Router Advertisement (RA) guard feature enables the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network switch platform. RAs are used by routers to announce themselves on the link. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized routers. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 device with the information found in the received RA frame. Once the Layer 2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.
- IPv6 DHCP Guard—The IPv6 DHCP Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents. IPv6 DHCP guard can prevent forged messages from being entered in the binding table and block DHCPv6 server messages when they are received on ports that are not explicitly configured as facing a DHCPv6 server or DHCP relay. To use this feature, configure a policy and attach it to an interface or a VLAN. To debug DHCP guard packets, use the **debug ipv6 snooping dhcp-guard** privileged EXEC command.

How to Configure an IPv6 Snooping Policy

The IPv6 Snooping Policy feature has been deprecated. Although the commands are visible on the CLI and you can configure them, we recommend that you use the Switch Integrated Security Feature (SISF)-based Device Tracking feature instead.

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Snooping Policy :

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 snooping policy** *policy-name*
3. {[**default**] | [**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**protocol** {**dhcp** | **ndp**}] | [**security-level** {**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] }
4. **end**
5. **show ipv6 snooping policy** *policy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**ipv6 snooping policy** *policy-name* **Example:** `(config)# ipv6 snooping policy example_policy`	Creates a snooping policy and enters IPv6 Snooping Policy Configuration mode.

Step 3	{[**default**] | [**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**protocol** {**dhcp** | **ndp**}] | [**security-level** {**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] }
	Example:
	`(config-ipv6-snooping)# security-level inspect`
	Example:
	`(config-ipv6-snooping)# trusted-port`
		Enables data address gleaning, validates messages against various criteria, specifies the security level for messages.
		• (Optional) **default**—Sets all to default options.
		• (Optional) **device-role**{**node**] | **switch**}—Specifies the role of the device attached to the port. Default is **node**.
		• (Optional) **limit address-count** *value*—Limits the number of addresses allowed per target.
		• (Optional) **no**—Negates a command or sets it to defaults.
		• (Optional) **protocol**{**dhcp** | **ndp**}—Specifies which protocol should be redirected to the snooping feature for analysis. The default, is **dhcp** and **ndp**. To change the default, use the **no protocol** command.
		• (Optional) **security-level**{**glean**|**guard**|**inspect**}—Specifies the level of security enforced by the feature. Default is **guard.**
		glean—Gleans addresses from messages and populates the binding table without any verification.
		guard—Gleans addresses and inspects messages. In addition, it rejects RA and DHCP server messages. This is the default option.
		inspect—Gleans addresses, validates messages for consistency and conformance, and enforces address ownership.
		• (Optional) **tracking** {**disable** | **enable**}—Overrides the default tracking behavior and specifies a tracking option.
		• (Optional) **trusted-port**—Sets up a trusted port. It disables the guard on applicable targets. Bindings learned through a trusted port have preference over

	Command or Action	Purpose
		bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
Step 4	**end** **Example:** `(config-ipv6-snooping)# exit`	Exits configuration modes to Privileged EXEC mode.
Step 5	**show ipv6 snooping policy** *policy-name* **Example:** `#show ipv6 snooping policy example_policy`	Displays the snooping policy configuration.

What to do next

Attach an IPv6 Snooping policy to interfaces or VLANs.

How to Attach an IPv6 Snooping Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an interface or VLAN:

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
3. **switchport**
4. **ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
5. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface** Interface_type *stack/module/port* **Example:** `(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	**switchport** **Example:**	Enters the Switchport mode.

	Command or Action	Purpose
	`(config-if)# switchport`	**Note** To configure Layer 2 parameters, if the interface is in Layer 3 mode, you must enter the switchport interface configuration command without any parameters to put the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When you put an interface that is in Layer 3 mode into Layer 2 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration. The command prompt displays as (config-if)# in Switchport configuration mode.

Step 4	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if)# ipv6 snooping`
	or
	`(config-if)# ipv6 snooping attach-policy example_policy`
	or
	`(config-if)# ipv6 snooping vlan 111,112`
	or
	`(config-if)# ipv6 snooping attach-policy example_policy vlan 111,112`
		Attaches a custom ipv6 snooping policy to the interface or the specified VLANs on the interface. To attach the default policy to the interface, use the **ipv6 snooping** command without the **attach-policy** keyword. To attach the default policy to VLANs on the interface, use the **ipv6 snooping vlan** command. The default policy is, security-level **guard**, device-role **node**, protocol **ndp** and **dhcp.**

Step 5	**do show running-config** **Example:** `#(config-if)# do show running-config`	Verifies that the policy is attached to the specified interface without exiting the interface configuration mode.

How to Attach an IPv6 Snooping Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface range** *Interface_name* **Example:** `(config)# interface range Po11`	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the **do show interfaces summary** command for quick reference to interface names and types.

Step 3	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if-range)# ipv6 snooping attach-policy example_policy`
	or
	`(config-if-range)# ipv6 snooping attach-policy example_policy vlan 222,223,224`
	or
	`(config-if-range)#ipv6 snooping vlan 222, 223,224`
		Attaches the IPv6 Snooping policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 4	**do show running-config interface***portchannel_interface_name* **Example:** `#(config-if-range)# do show running-config int po11`	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Snooping Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping Policy to VLANs across multiple interfaces:

SUMMARY STEPS

1. **configure terminal**
2. **vlan configuration** *vlan_list*
3. **ipv6 snooping** [**attach-policy** *policy_name*]

4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**vlan configuration** *vlan_list* **Example:** `(config)# vlan configuration 333`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	**ipv6 snooping** [**attach-policy** *policy_name*] **Example:** `(config-vlan-config)#ipv6 snooping attach-policy example_policy`	Attaches the IPv6 Snooping policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, security-level **guard**, device-role **node**, protocol **ndp** and **dhcp.**
Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Verifies that the policy is attached to the specified VLANs without exiting the interface configuration mode.

How to Configure the IPv6 Binding Table Content

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ipv6 neighbor binding** [**vlan** *vlan-id* {*ipv6-address* **interface** interface_type *stack/module/port hw_address* [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**tracking**{ [default | disable] [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**enable** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**retry-interval** {*seconds*| **default** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] }]
3. [**no**] **ipv6 neighbor binding max-entries** *number* [**mac-limit** *number* | **port-limit** *number* [**mac-limit** *number*] | **vlan-limit** *number* [[**mac-limit** *number*] | [**port-limit** *number* [**mac-limit***number*]]]]
4. **ipv6 neighbor binding logging**
5. **exit**
6. **show ipv6 neighbor binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	[**no**] **ipv6 neighbor binding** [**vlan** *vlan-id* {*ipv6-address* **interface** interface_type *stack/module/port hw_address* [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**tracking**{ [default | disable] [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**enable** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**retry-interval** {*seconds*| **default** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] }] **Example:** `(config)# ipv6 neighbor binding`	Adds a static entry to the binding table database.
Step 3	[**no**] **ipv6 neighbor binding max-entries** *number* [**mac-limit** *number* | **port-limit** *number* [**mac-limit** *number*] | **vlan-limit** *number* [[**mac-limit** *number*] | [**port-limit** *number* [**mac-limit***number*]]]] **Example:** `(config)# ipv6 neighbor binding max-entries 30000`	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 4	**ipv6 neighbor binding logging** **Example:** `(config)# ipv6 neighbor binding logging`	Enables the logging of binding table main events.
Step 5	**exit** **Example:** `(config)# exit`	Exits global configuration mode, and places the router in privileged EXEC mode.
Step 6	**show ipv6 neighbor binding** **Example:** `# show ipv6 neighbor binding`	Displays contents of a binding table.

How to Configure an IPv6 Neighbor Discovery Inspection Policy

Starting with 17.1.1, the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it. For the corresponding replacement task, see *Creating a Custom Device Tracking Policy with Custom Settings* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 ND Inspection Policy:

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 nd inspection policy** *policy-name*
3. **device-role** {**host** | **switch**}
4. **limit address-count** *value*
5. **tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]}
6. **trusted-port**
7. **validate source-mac**
8. **no** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**}
9. **default** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**}
10. **do show ipv6 nd inspection policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	[**no**]**ipv6 nd inspection policy** *policy-name* **Example:** `(config)# ipv6 nd inspection policy example_policy`	Specifies the ND inspection policy name and enters ND Inspection Policy configuration mode.
Step 3	**device-role** {**host**	**switch**} **Example:** `(config-nd-inspection)# device-role switch`	Specifies the role of the device attached to the port. The default is **host**.
Step 4	**limit address-count** *value* **Example:** `(config-nd-inspection)# limit address-count 1000`	Enter 1–10,000.
Step 5	**tracking** {**enable** [**reachable-lifetime** {*value*	**infinite**}]	**disable** [**stale-lifetime** {*value*	**infinite**}]} **Example:** `(config-nd-inspection)# tracking disable stale-lifetime infinite`	Overrides the default tracking policy on a port.
Step 6	**trusted-port** **Example:** `(config-nd-inspection)# trusted-port`	Configures a port to become a trusted port.
Step 7	**validate source-mac** **Example:** `(config-nd-inspection)# validate source-mac`	Checks the source media access control (MAC) address against the link-layer address.

	Command or Action	Purpose
Step 8	**no** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**} **Example:** `(config-nd-inspection)# no validate source-mac`	Remove the current configuration of a parameter with the **no** form of the command.
Step 9	**default** {**device-role** | **limit address-count** | **tracking** | **trusted-port** | **validate source-mac**} **Example:** `(config-nd-inspection)# default limit address-count`	Restores configuration to the default values.
Step 10	**do show ipv6 nd inspection policy** *policy_name* **Example:** `(config-nd-inspection)# do show ipv6 nd inspection policy example_policy`	Verifies the ND Inspection Configuration without exiting ND inspection configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to an Interface

Starting with 17.1.1, the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to an interface or VLANs on an interface :

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
3. **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface** Interface_type *stack/module/port* **Example:** `(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.

	Command or Action	Purpose

Step 3	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if)# ipv6 nd inspection attach-policy example_policy`
	or
	`(config-if)# ipv6 nd inspection attach-policy example_policy vlan 222,223,224`
	or
	`(config-if)# ipv6 nd inspection vlan 222, 223,224`
		Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Verifies that the policy is attached to the specified interface without exiting the interface configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to a Layer 2 EtherChannel Interface

Starting with 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Neighbor Discovery Inspection policy on an EtherChannel interface or VLAN:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*
3. **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* \| **add** *vlan_ids* \| **except** *vlan_ids* \| **none** \| **remove** *vlan_ids* \| **all**}] \| **vlan** [{*vlan_ids* \| **add** *vlan_ids* \| **except***vlan_ids* \| **none** \| **remove** *vlan_ids* \| **all**}]
4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**interface range** *Interface_name* **Example:** `(config)# interface Po11`	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the **do show interfaces summary** command for quick reference to interface names and types.

Step 3	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if-range)# ipv6 nd inspection attach-policy example_policy`
	or
	`(config-if-range)# ipv6 nd inspection attach-policy example_policy vlan 222,223,224`
	or
	`(config-if-range)#ipv6 nd inspection vlan 222, 223,224`
		Attaches the ND Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 4	**do show running-config interface***portchannel_interface_name* **Example:** `#(config-if-range)# do show running-config int po11`	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to VLANs Globally

Starting with 17.1.1, the IPv6 ND Inspection feature is deprecated and the SISF- based device tracking feature replaces it. For the corresponding replacement task, see *Attaching a Device Tracking Policy to a VLAN* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to VLANs across multiple interfaces:

SUMMARY STEPS

1. **configure terminal**
2. **vlan configuration** *vlan_list*
3. **ipv6 nd inspection** [**attach-policy** *policy_name*]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**vlan configuration** *vlan_list* **Example:** `(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	**ipv6 nd inspection** [**attach-policy** *policy_name*] **Example:** `(config-vlan-config)#ipv6 nd inspection attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, device-role **host**, no drop-unsecure, limit address-count disabled, sec-level minimum is disabled, tracking is disabled, no trusted-port, no validate source-mac.
Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure an IPv6 Router Advertisement Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 Router Advertisement policy :

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 nd raguard policy** *policy-name*
3. [**no**]**device-role** {**host** | **monitor** | **router** | **switch**}
4. [**no**]**hop-limit** {**maximum** | **minimum**} *value*
5. [**no**]**managed-config-flag** {**off** | **on**}
6. [**no**]**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*}
7. [**no**]**other-config-flag** {**on** | **off**}
8. [**no**]**router-preference maximum** {**high** | **medium** | **low**}
9. [**no**]**trusted-port**
10. **default** {**device-role** | **hop-limit** {**maximum** | **minimum**} | **managed-config-flag** | **match** {**ipv6 access-list** | **ra prefix-list** } | **other-config-flag** | **router-preference maximum**| **trusted-port**}
11. **do show ipv6 nd raguard policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	[**no**]**ipv6 nd raguard policy** *policy-name* **Example:** `(config)# ipv6 nd raguard policy example_policy`	Specifies the RA Guard policy name and enters RA Guard Policy configuration mode.
Step 3	[**no**]**device-role** {**host** | **monitor** | **router** | **switch**} **Example:** `(config-nd-raguard)# device-role switch`	Specifies the role of the device attached to the port. The default is **host**. **Note** For a network with both host-facing ports and router-facing ports, along with a RA guard policy configured with **device-role host** on host-facing ports or vlan, it is mandatory to configure a RA guard policy with **device-role router** on router-facing ports to allow the RA Guard feature to work properly.
Step 4	[**no**]**hop-limit** {**maximum** | **minimum**} *value* **Example:** `(config-nd-raguard)# hop-limit maximum 33`	(1–255) Range for Maximum and Minimum Hop Limit values. Enables filtering of Router Advertisement messages by the Hop Limit value. A rogue RA message may have a low Hop Limit value (equivalent to the IPv4 Time to Live) that when accepted by the host, prevents the host from generating traffic to destinations beyond the rogue RA message generator. An RA message with an unspecified Hop Limit value is blocked. If not configured, this filter is disabled. Configure **minimum** to block RA messages with Hop Limit values lower than the value you specify. Configure **maximum**to block RA messages with Hop Limit values greater than the value you specify.
Step 5	[**no**]**managed-config-flag** {**off** | **on**} **Example:** `(config-nd-raguard)# managed-config-flag on`	Enables filtering of Router Advertisement messages by the Managed Address Configuration, or "M" flag field. A rouge RA message with an M field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. **On**—Accepts and forwards RA messages with an M value of 1, blocks those with 0. **Off**—Accepts and forwards RA messages with an M value of 0, blocks those with 1.

	Command or Action	Purpose
Step 6	[**no**]**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*} **Example:** `(config-nd-raguard)# match ipv6 access-list example_list`	Matches a specified prefix list or access list.
Step 7	[**no**]**other-config-flag** {**on** | **off**} **Example:** `(config-nd-raguard)# other-config-flag on`	Enables filtering of Router Advertisement messages by the Other Configuration, or "O" flag field. A rouge RA message with an O field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. **On**—Accepts and forwards RA messages with an O value of 1, blocks those with 0. **Off**—Accepts and forwards RA messages with an O value of 0, blocks those with 1.
Step 8	[**no**]**router-preference maximum** {**high** | **medium** | **low**} **Example:** `(config-nd-raguard)# router-preference maximum high`	Enables filtering of Router Advertisement messages by the Router Preference flag. If not configured, this filter is disabled. • **high**—Accepts RA messages with the Router Preference set to high, medium, or low. • **medium**—Blocks RA messages with the Router Preference set to high. • **low**—Blocks RA messages with the Router Preference set to medium and high.
Step 9	[**no**]**trusted-port** **Example:** `(config-nd-raguard)# trusted-port`	When configured as a trusted port, all attached devices are trusted, and no further message verification is performed.
Step 10	**default** {**device-role** | **hop-limit** {**maximum** | **minimum**} | **managed-config-flag** | **match** {**ipv6 access-list** | **ra prefix-list** } | **other-config-flag** | **router-preference maximum**| **trusted-port**} **Example:** `(config-nd-raguard)# default hop-limit`	Restores a command to its default value.
Step 11	**do show ipv6 nd raguard policy** *policy_name* **Example:** `(config-nd-raguard)# do show ipv6 nd raguard policy example_policy`	(Optional)—Displays the ND Guard Policy configuration without exiting the RA Guard policy configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to an interface or to VLANs on the interface :

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
3. **ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface** Interface_type *stack/module/port* **Example:** `(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.

Step 3	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if)# ipv6 nd raguard attach-policy example_policy`
	or
	`(config-if)# ipv6 nd raguard attach-policy example_policy vlan 222,223,224`
	or
	`(config-if)# ipv6 nd raguard vlan 222, 223,224`
		Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement Guard Policy on an EtherChannel interface or VLAN:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*
3. **ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface range** *Interface_name* **Example:** `(config)# interface Po11`	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the do show interfaces summary command for quick reference to interface names and types.

Step 3	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if-range)# ipv6 nd raguard attach-policy example_policy`
	`or`
	`(config-if-range)# ipv6 nd raguard attach-policy example_policy vlan 222,223,224`
	`or`
	`(config-if-range)#ipv6 nd raguard vlan 222, 223,224`
		Attaches the RA Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

	Command or Action	Purpose
Step 4	**do show running-config interface***portchannel_interface_name* **Example:** `#(config-if-range)# do show running-config int po11`	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to VLANs regardless of interface:

SUMMARY STEPS

1. **configure terminal**
2. **vlan configuration** *vlan_list*
3. **ipv6 dhcp guard** [**attach-policy** *policy_name*]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 2	**vlan configuration** *vlan_list* **Example:** `(config)# vlan configuration 335`	Specifies the VLANs to which the IPv6 RA Guard policy will be attached ; enters the VLAN interface configuration mode.
Step 3	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `(config-vlan-config)#ipv6 nd raguard attach-policy example_policy`	Attaches the IPv6 RA Guard policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used.
Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure an IPv6 DHCP Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 DHCP (DHCPv6) Guard policy:

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 dhcp guard policy** *policy-name*
3. [**no**]**device-role** {**client** | **server**}
4. [**no**] **match server access-list** *ipv6-access-list-name*
5. [**no**] **match reply prefix-list** *ipv6-prefix-list-name*
6. [**no**]**preference**{ **max** *limit* | **min** *limit* }
7. [**no**] **trusted-port**
8. **default** {**device-role** | **trusted-port**}
9. **do show ipv6 dhcp guard policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	[**no**]**ipv6 dhcp guard policy** *policy-name* **Example:** `(config)# ipv6 dhcp guard policy example_policy`	Specifies the DHCPv6 Guard policy name and enters DHCPv6 Guard Policy configuration mode.
Step 3	[**no**]**device-role** {**client** | **server**} **Example:** `(config-dhcp-guard)# device-role server`	(Optional) Filters out DHCPv6 replies and DHCPv6 advertisements on the port that are not from a device of the specified role. Default is **client**. • **client**—Default value, specifies that the attached device is a client. Server messages are dropped on this port. • **server**—Specifies that the attached device is a DHCPv6 server. Server messages are allowed on this port.

Step 4	[**no**] **match server access-list** *ipv6-access-list-name*
	Example:
	`;;Assume a preconfigured IPv6 Access List as follows:`
	`(config)# ipv6 access-list my_acls`
	`(config-ipv6-acl)# permit host FE80::A8BB:CCFF:FE01:F700 any`
	`;;configure DCHPv6 Guard to match approved access list.`
	`(config-dhcp-guard)# match server access-list my_acls`
		(Optional). Enables verification that the advertised DHCPv6 server or relay address is from an authorized server access list (The destination address in the access list is 'any'). If not configured, this check will be bypassed. An empty access list is treated as a permit all.

Step 5	[**no**] **match reply prefix-list** *ipv6-prefix-list-name* **Example:**	(Optional) Enables verification of the advertised prefixes in DHCPv6 reply messages from the configured authorized

	Command or Action	Purpose
	;;Assume a preconfigured IPv6 prefix list as follows: (config)# **ipv6 prefix-list my_prefix permit 2001:0DB8::/64 le 128** ;; Configure DCHPv6 Guard to match prefix (config-dhcp-guard)# **match reply prefix-list my_prefix**	prefix list. If not configured, this check will be bypassed. An empty prefix list is treated as a permit.
Step 6	[**no**]**preference**{ **max** *limit* | **min** *limit* } **Example:** (config-dhcp-guard)# **preference max 250** (config-dhcp-guard)#**preference min 150**	Configure **max** and **min** when **device-role** is **server**to filter DCHPv6 server advertisements by the server preference value. The defaults permit all advertisements. **max** *limit*—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. Default is 255. If not specified, this check will be bypassed. **min** *limit*—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. Default is 0. If not specified, this check will be bypassed.
Step 7	[**no**] **trusted-port** **Example:** (config-dhcp-guard)# **trusted-port**	(Optional) **trusted-port**—Sets the port to a trusted mode. No further policing takes place on the port. **Note** If you configure a trusted port then the device-role option is not available.
Step 8	**default** {**device-role** | **trusted-port**} **Example:** (config-dhcp-guard)# **default device-role**	(Optional) **default**—Sets a command to its defaults.
Step 9	**do show ipv6 dhcp guard policy** *policy_name* **Example:** (config-dhcp-guard)# **do show ipv6 dhcp guard policy example_policy**	(Optional) Displays the configuration of the IPv6 DHCP guard policy without leaving the configuration submode. Omitting the *policy_name* variable displays all DHCPv6 policies.

Example of DHCPv6 Guard Configuration

```
enable
configure terminal
ipv6 access-list acl1
 permit host FE80::A8BB:CCFF:FE01:F700 any
ipv6 prefix-list abc permit 2001:0DB8::/64 le 128
ipv6 dhcp guard policy pol1
 device-role server
 match server access-list acl1
 match reply prefix-list abc
 preference min 0
 preference max 255
 trusted-port
interface GigabitEthernet 0/2/0
```

```
 switchport
 ipv6 dhcp guard attach-policy pol1 vlan add 1
 vlan 1
  ipv6 dhcp guard attach-policy pol1
show ipv6 dhcp guard policy pol1
```

How to Attach an IPv6 DHCP Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy to VLANs across multiple interfaces:

SUMMARY STEPS

1. **configure terminal**
2. **vlan configuration** *vlan_list*
3. **ipv6 dhcp guard** [**attach-policy** *policy_name*]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**vlan configuration** *vlan_list* **Example:** `(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `(config-vlan-config)#ipv6 dhcp guard attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, device-role **client**, **no** trusted-port.
Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Attach an IPv6 DHCP Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy on an EtherChannel interface or VLAN:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*

3. **ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface range** *Interface_name* **Example:** `(config)# interface Po11`	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the **do show interfaces summary** command for quick reference to interface names and types.

Step 3	**ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`(config-if-range)# ipv6 dhcp guard attach-policy example_policy`
	`or`
	`(config-if-range)# ipv6 dhcp guard attach-policy example_policy vlan 222,223,224`
	`or`
	`(config-if-range)#ipv6 dhcp guard vlan 222, 223,224`
		Attaches the DHCP Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the **attach-policy** option is not used.

Step 4	**do show running-config interface***portchannel_interface_name* **Example:** `#(config-if-range)# do show running-config int po11`	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 DHCP Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy to VLANs across multiple interfaces:

SUMMARY STEPS

1. **configure terminal**
2. **vlan configuration** *vlan_list*
3. **ipv6 dhcp guard** [**attach-policy** *policy_name*]
4. **do show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**vlan configuration** *vlan_list* **Example:** `(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **Example:** `(config-vlan-config)#ipv6 dhcp guard attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the **attach-policy** option is not used. The default policy is, device-role **client**, **no** trusted-port.
Step 4	**do show running-config** **Example:** `#(config-if)# do show running-config`	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure IPv6 Source Guard

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ipv6 source-guard policy** *policy_name*
3. [**deny global-autoconf**] [**permit link-local**] [**default**{. . . }] [**exit**] [**no**{. . . }]
4. **end**
5. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	[**no**] **ipv6 source-guard policy** *policy_name* **Example:** `(config)# ipv6 source-guard policy example_policy`	Specifies the IPv6 Source Guard policy name and enters IPv6 Source Guard policy configuration mode.
Step 3	[**deny global-autoconf**] [**permit link-local**] [**default**{. . . }] [**exit**] [**no**{. . . }] **Example:** `(config-sisf-sourceguard)# deny global-autoconf`	(Optional) Defines the IPv6 Source Guard policy. • **deny global-autoconf**—Denies data traffic from auto-configured global addresses. This is useful when all global addresses on a link are DHCP-assigned and the administrator wants to block hosts with self-configured addresses to send traffic. • **permit link-local**—Allows all data traffic that is sourced by a link-local address. **Note** Trusted option under source guard policy is not supported.
Step 4	**end** **Example:** `(config-sisf-sourceguard)# end`	Exits out of IPv6 Source Guard policy configuration mode.
Step 5	**show ipv6 source-guard policy** *policy_name* **Example:** `# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

What to do next

Apply the IPv6 Source Guard policy to an interface.

How to Attach an IPv6 Source Guard Policy to an Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
3. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
4. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**interface** Interface_type *stack/module/port* **Example:** `(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 4	**show ipv6 source-guard policy** *policy_name* **Example:** `#(config-if)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

How to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *port-channel-number*
3. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
4. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface port-channel** *port-channel-number* **Example:** `(config)# interface Po4`	Specifies an interface type and port number and places the switch in the port channel configuration mode.
Step 3	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `(config-if) # ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 4	**show ipv6 source-guard policy** *policy_name* **Example:** `(config-if) #show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

How to Configure IPv6 Prefix Guard

Note To allow routing protocol control packets sourced by a link-local address when prefix guard is applied, enable the permit link-local command in the source-guard policy configuration mode.

SUMMARY STEPS

1. [**no**] **ipv6 source-guard policy** *source-guard-policy*
2. [**no**] **validate address**
3. **validate prefix**
4. **exit**
5. **show ipv6 source-guard policy** [*source-guard-policy*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	[**no**] **ipv6 source-guard policy** *source-guard-policy* **Example:** `(config)# ipv6 source-guard policy my_snooping_policy`	Defines an IPv6 source-guard policy name and enters switch integrated security features source-guard policy configuration mode.
Step 2	[**no**] **validate address** **Example:** `(config-sisf-sourceguard)# no validate address`	Disables the validate address feature and enables the IPv6 prefix guard feature to be configured.
Step 3	**validate prefix** **Example:** `(config-sisf-sourceguard)# validate prefix`	Enables IPv6 source guard to perform the IPv6 prefix-guard operation.
Step 4	**exit** **Example:** `(config-sisf-sourceguard)# exit`	Exits switch integrated security features source-guard policy configuration mode and returns to privileged EXEC mode.
Step 5	**show ipv6 source-guard policy** [*source-guard-policy*] **Example:** `# show ipv6 source-guard policy policy1`	Displays the IPv6 source-guard policy configuration.

How to Attach an IPv6 Prefix Guard Policy to an Interface

SUMMARY STEPS

1. **configure terminal**

2. **interface** Interface_type *stack/module/port*
3. **ipv6 source-guard attach-policy** *policy_name*
4. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface** Interface_type *stack/module/port* **Example:** `(config)# interface gigabitethernet 1/1/4`	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	**ipv6 source-guard attach-policy** *policy_name* **Example:** `(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 4	**show ipv6 source-guard policy** *policy_name* **Example:** `(config-if)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

How to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *port-channel-number*
3. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
4. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters the global configuration mode.
Step 2	**interface port-channel** *port-channel-number* **Example:** `(config)# interface Po4`	Specifies an interface type and port number and places the switch in the port channel configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **Example:** `(config-if)# ipv6 source-guard attach-policy example_policy`	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the **attach-policy** option is not used.
Step 4	**show ipv6 source-guard policy** *policy_name* **Example:** `(config-if)# show ipv6 source-guard policy example_policy`	Shows the policy configuration and all the interfaces where the policy is applied.

Configuration Examples for IPv6 First Hop Security

Examples: How to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

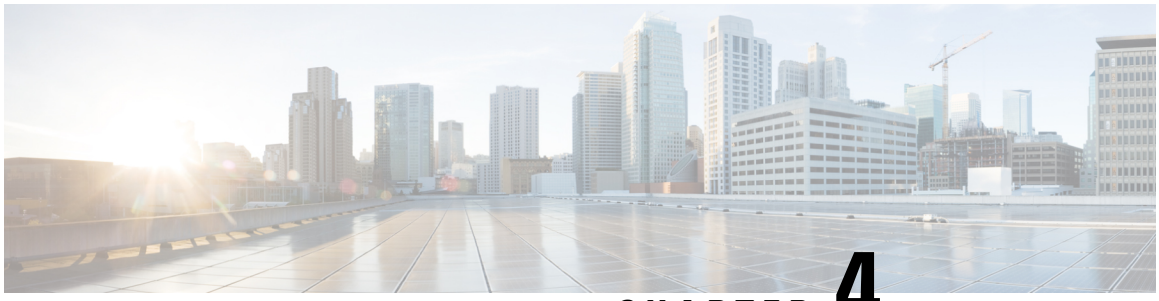
The following example shows how to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface:

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch(config-sisf-sourceguard) # validate address
switch(config-sisf-sourceguard)# exit
Switch(config)# interface Po4
Switch(config)# ipv6 snooping
Switch(config-if)# ipv6 source-guard attach-policy POL
Switch(config-if)# exit
switch(config)#
```

Examples: How to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

The following example shows how to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface:

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch (config-sisf-sourceguard)# no validate address
Switch((config-sisf-sourceguard)# validate prefix
Switch(config)# interface Po4
Switch(config-if)# ipv6 snooping
Switch(config-if)# ipv6 source-guard attach-policy POL
```



CHAPTER 4

Configuring Layer 2 NAT

Layer 2 Network Address Translation

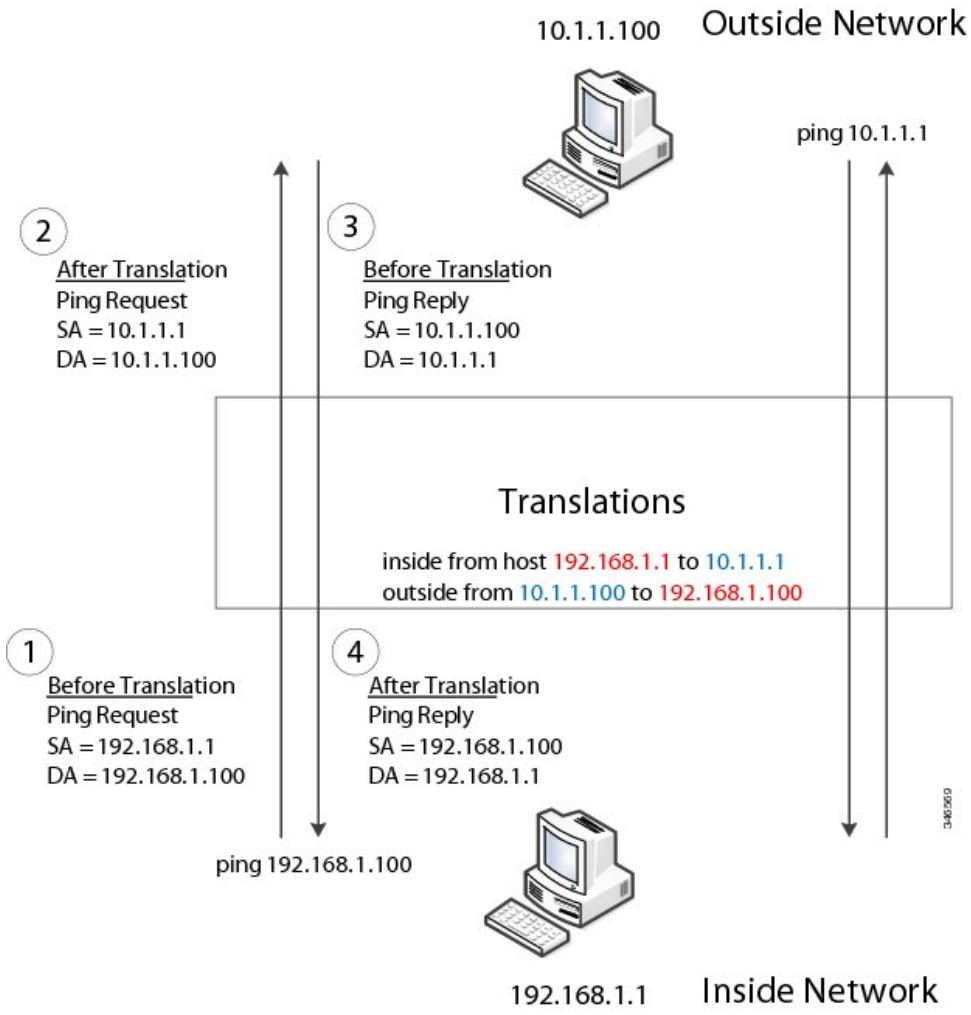
One-to-one (1:1) Layer 2 Network Address Translation (NAT) is a service that allows the assignment of a unique public IP address to an existing private IP address (end device). The assignment enables the end device to communicate on both the private and public subnets. This service is configured in a 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.

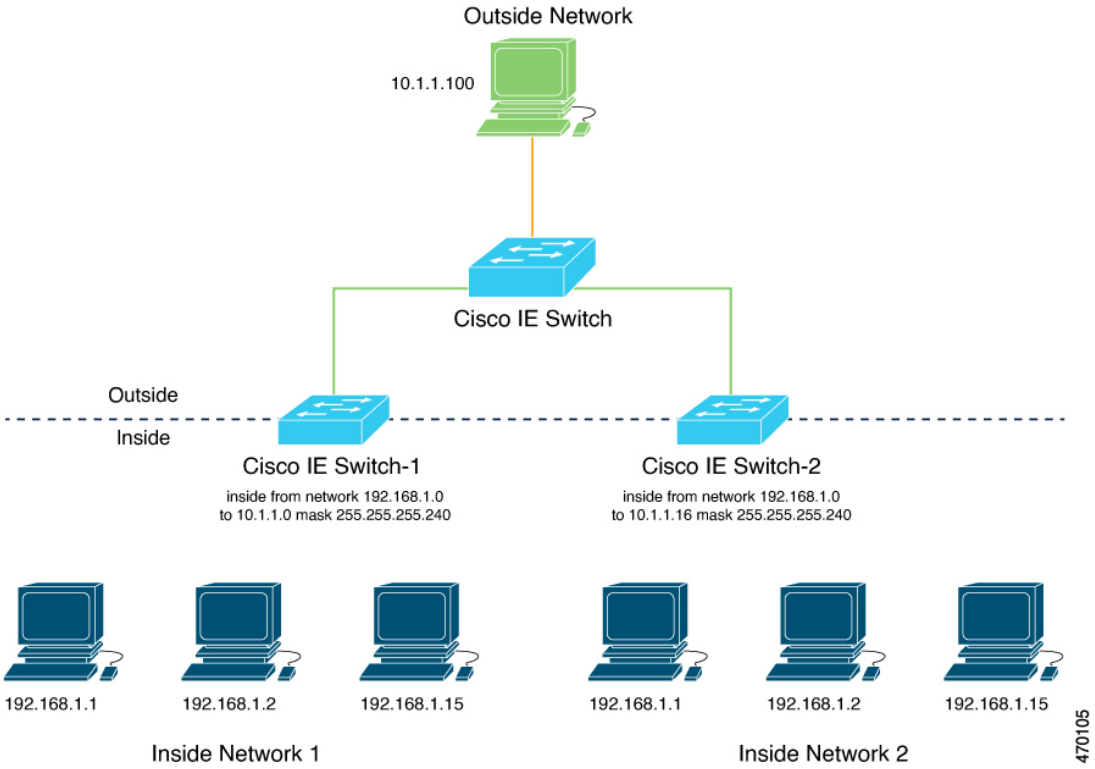
1. The 192.168.1.x network is the inside/internal IP address space and the 10.1.1.x network is the outside or external IP address space.
2. The sensor at 192.168.1.1 sends a ping request to the line controller by using an “inside” address, 192.168.1.100.
3. Before the packet leaves the internal network, 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 3: 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 4: Inside-Outside Address Translation



The following figure shows a Cisco Catalyst IE3400 Rugged Series 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.

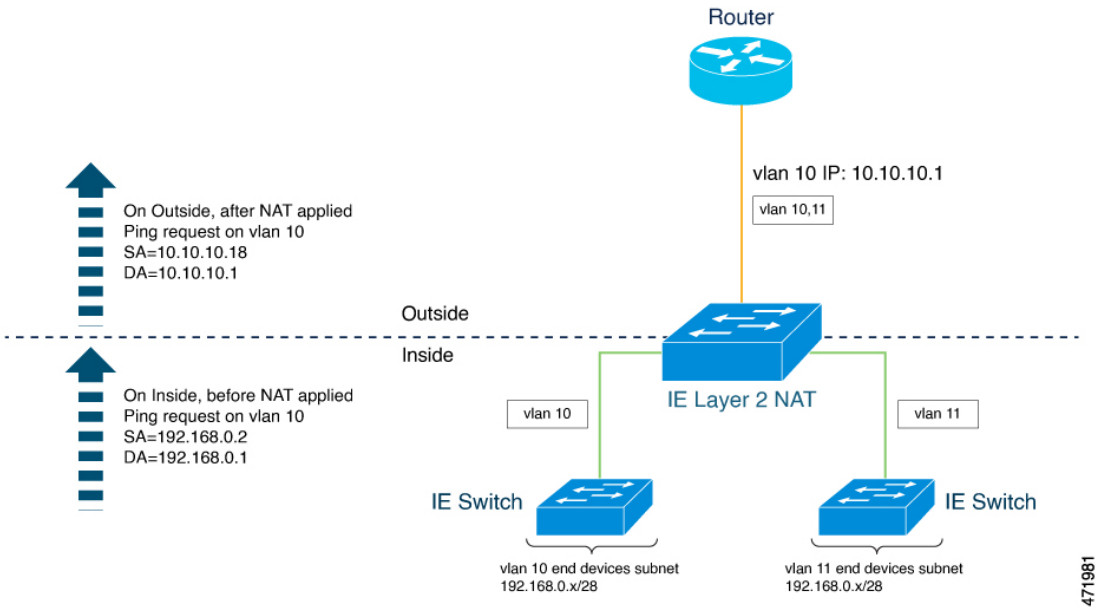
The L2NAT instance definitions use the `network` command to define a translation row for multiple devices in the same subnet. In this the case it's a /28 subnet. With this subnet mask, the last nibble in 4th byte of the inside IP address will not change. The last byte will be in range 16 – 31 because the translated IP address is 10.10.10.16.

The gateway for the VLAN is the router with the last byte of the IP address ending with .1. An outside host translation is provided for the gateway router. The `network` command in the Layer 2 NAT definition translates a subnet's worth of host with a single command, saving on Layer 2 NAT translation rules.

The Gi1/1 uplink interface has two Layer 2 NAT translation instances for vlan 10 and vlan 11 subnets. Interfaces can support multiple Layer 2 NAT instance definitions.

The downstream IE switches are examples of access layer switches that do not perform Layer 2 NAT and rely on the upstream aggregation layer switch to do it.

Figure 5: NAT on the IE3400 Switch



The IE3400 NAT configuration for the diagram shown in the preceding figure is as follows:

```
!
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
```

Layer 2 NAT Switch Support

- IE3105: Layer 2 NAT feature is supported only on uplink ports (Gig 1/1 and Gig 1/2) and available in both (essential and advantage) licenses.
- IE3300: Layer 2 NAT feature is supported only on uplink ports (Gig 1/1 and Gig 1/2) and available in both (essential and advantage) licenses.

- IE3400: Layer 2 NAT feature is supported only on uplink ports (Gig 1/1 and Gig 1/2) and available in both (essential and advantage) licenses.

Guidelines and Limitations

- Only IPv4 addresses can be translated.
- Layer 2 NAT applies only to unicast traffic. You can permit or allow untranslated unicast traffic, multicast traffic, and IGMP traffic.
- Layer 2 NAT does not support one-to-many and many-to-one IP address mapping.
- Layer 2 NAT supports one-to-one mapping between external and internal IP addresses.
- Layer 2 NAT cannot save on public IP addresses.
- If you configure a translation for a Layer 2 NAT host, do not configure it as a DHCP client.
- Certain protocols such as ARP and ICMP do not work transparently across Layer 2 NAT but are "fixed up" by default. "Fixed up" means that changes are made to IP addresses embedded in the payload of IP packets for the protocols to work.
- The downlink port can be VLAN, trunk, or Layer 2 channel.
- You can configure 128 Layer 2 NAT rules on the switch.
- Up to 128 VLANs are allowed to have Layer 2 NAT configuration.
- 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 L2NAT 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.
- The interfaces that support NAT instance configurations are Gig 1/1 and Gig 1/2 (uplinks).

Default Settings

Feature	Default Setting
Permit or drop packets for unmatched traffic and traffic types that are not configured to be translated	Drop all unmatched, multicast, and IGMP packets
Protocol fixups	Fixup is enabled for ARP and ICMP.

Note In the preceding table, *unmatched* refers to any host without an IP address defined for translation as a rule in the Layer 2 NAT instance that is applied to the interface.

Configuring Layer 2 NAT

You need to configure Layer 2 NAT instances that specify the address translations. You then attach these rules to uplink interfaces. For unmatched traffic and traffic types that are not configured to be translated, you can choose to permit or drop the traffic. The IE switch management interface is behind the management interfaces (CLI/SNMP/CIP/WebUI). You can view detailed statistics about the packets sent and received (see Verifying Configuration , on page 81).

To configure Layer 2 NAT, follow these steps. Refer to the examples in Basic Inside-to-Outside Communications Example, on page 81 and Duplicate IP Addresses Example, on page 83 for more details.

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 sub-mode for that instance.

Step 3 Translate an inside address to an outside address:

inside 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 all of the addresses in a subnet. Translate the source address for outbound traffic and the destination address for inbound traffic. Use the `inside from` command when the host or hosts are physically present on the inside and have private IP addresses.

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 all of the addresses in a subnet. Translate the destination address for outbound traffic and the source address for inbound traffic. Use the `outside from` command when the host or hosts are physically present on the outside and have public IP addresses.

Step 5 Fix the translation for ICMP and IGMP through NAT translation. By default, fixups for both ARP and ICMP are enabled, so this command is not normally needed unless you change the defaults.

fixup arp | icmp | all

Note For ICMP, only fixups for ICMP Error messages are supported.

Step 6 (Optional) Permit untranslated unicast traffic (it is dropped by default):

permit { **multicast** | **igmp** | **all** }

Step 7 Exit config-l2nat mode:

exit

Step 8 Access interface configuration mode for the specified interface (gi1/1 and gi1/2 for IE3105, IE3300, and IE3400):

interface *interface-id*

Step 9 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 10 Exit interface configuration mode:

end

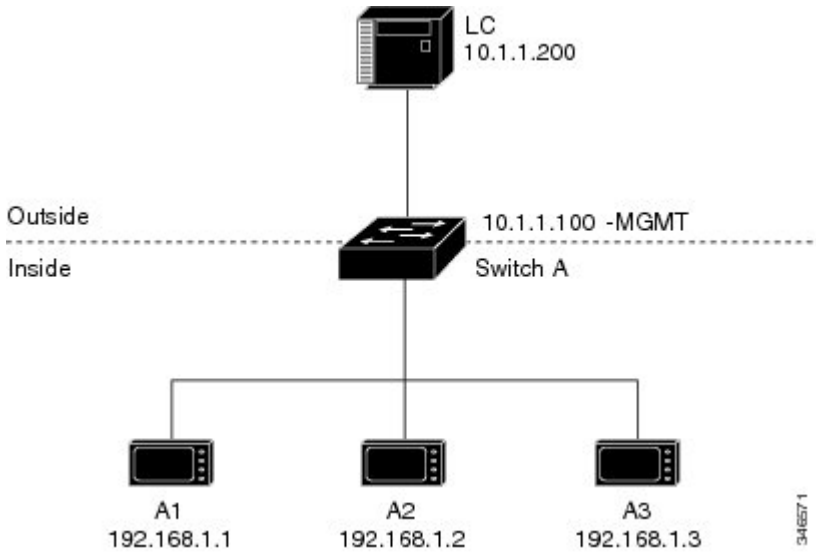
Verifying Configuration

Command	Purpose
show l2nat instance	Displays the configuration details for a specified Layer 2 NAT instance.
show l2nat interface	Displays the configuration details for Layer 2 NAT instances on one or more interfaces.
show l2nat statistics	Displays the Layer 2 NAT statistics for all interfaces.
show l2nat statistics interface	Displays the Layer 2 NAT statistics for a specified interface.
debug l2nat	Enables showing real-time Layer 2 NAT configuration details when the configuration is applied.

Basic Inside-to-Outside Communications Example

In this scenario, A1 needs to communicate with a logic controller (LC) that is directly connected to the uplink port. An 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 6: 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 It is a good practice to put the management interface of the switch on a different VLAN from the inside network 192.168.1.x.

The following table shows the configuration tasks for this scenario. The Layer 2 NAT instance is created, two translation entries are added, and the instance is applied to the interface. ARP fixups are enabled by default.

Table 9: Configuration of Cisco Switch A for Basic Inside-to-Outside Example

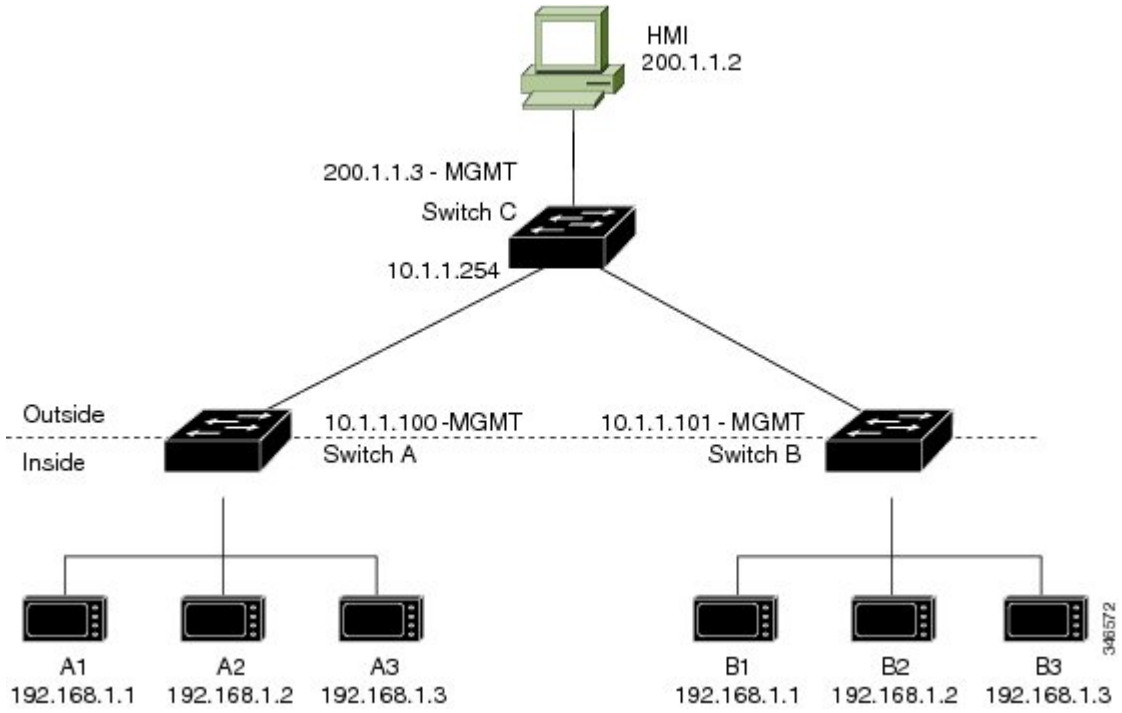
	Command	Purpose
1.	Switch# **configure**	Enters global configuration mode.
2.	Switch(config)# **l2nat instance A-LC**	Creates a new Layer 2 NAT instance called A-LC.
3.	Switch(config-l2nat)# **inside from host 192.168.1.1 to 10.1.1.1**	Translates A1's inside address to an outside address.
4.	Switch(config-l2nat)# **inside from host 192.168.1.2 to 10.1.1.2**	Translates A2's inside address to an outside address.

	Command	Purpose
5.	Switch(config-l2nat)# **inside from host 192.168.1.3 to 10.1.1.3**	Translates A3's inside address to an outside address.
6.	Switch(config-l2nat)# **outside from host 10.1.1.200 to 192.168.1.250**	Translates LC's outside address to an inside address.
7.	Switch(config-l2nat)# **exit**	Exits config-l2nat mode.
8.	Switch(config)# **interface Gi1/1**	Accesses interface configuration mode for the uplink port.
9.	Switch(config-if)# **l2nat A-LC**	Applies this Layer 2 NAT instance to the native VLAN on this interface. **Note** For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows: l2nat *instance vlan*
10.	Switch# **end**	Returns to privileged EXEC mode.

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 7: Duplicate IP Addresses



- For switch C to act as a gateway for the private network, 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. The10.1.1.32 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

Node	Address in Node A	Address in Outside Network	Address in Node B
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

Table 11: Configuration of Switch A for Duplicate Addresses Example, on page 85 shows the configuration tasks for Switch A. Table 12: Configuration of Switch B for Subnet Example, on page 86 shows the configuration tasks for Switch B.

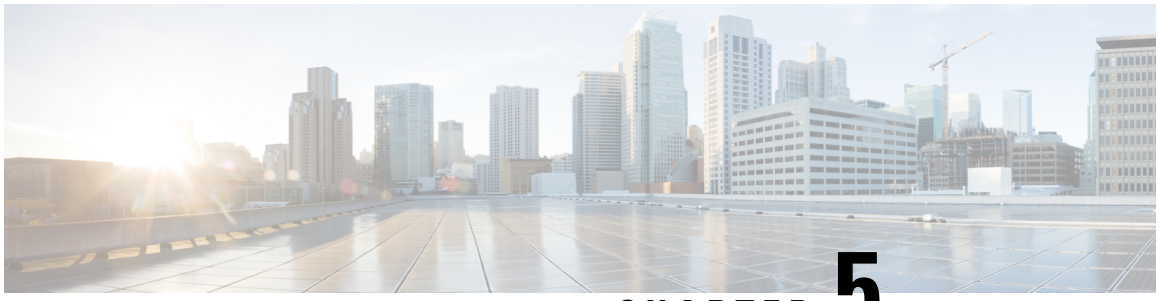
Note This example is based on the IE 2000 switch. For the IE3x00 and ESS3300 switches, the interface numbers may vary.

Table 11: Configuration of Switch A for Duplicate Addresses Example

	Command	Purpose
1	Switch# **configure**	Enters global configuration mode.
2	Switch(config)# **l2nat instance A-Subnet**	Creates a new Layer 2 NAT instance called A-Subnet.
3	Switch(config-l2nat)# **inside from network 192.168.1.0 to 10.1.1.16 mask 255.255.255.240**	Translates the Node A machines' inside addresses to addresses in the 10.1.1.16 255.255.255.240 subnet.
4	Switch(config-l2nat)# **outside from host 10.1.1.254 to 192.168.1.254**	Translates the outside address of Switch C to an inside address.
5	Switch(config-l2nat)# **outside from network 10.1.1.32 to 192.168.1.32 255.255.255.240**	Translates the Node B machines' outside addresses to their inside addresses.
6	Switch(config-l2nat)# **exit**	Exits config-l2nat mode.
7	Switch(config)# **interface Gi1/1**	Accesses interface configuration mode for the uplink port.
8	Switch(config-if)# **l2nat A-Subnet**	Applies this Layer 2 NAT instance to the native VLAN on this interface. **Note** For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows: l2nat *instance vlan*
9	Switch# **end**	Returns to privileged EXEC mode.

Table 12: Configuration of Switch B for Subnet Example

	Command	Purpose
1.	Switch# **configure**	Enters global configuration mode.
2.	Switch(config)# **l2nat instance B-Subnet**	Creates a new Layer 2 NAT instance called B-Subnet.
3.	Switch(config-l2nat)# **inside from network 192.168.1.0 to 10.1.1.32 255.255.255.240**	Translates the Node B machines' inside addresses to addresses in the 10.1.1.32 255.255.255.240 subnet.
4.	Switch(config-l2nat)# **outside from host 10.1.1.254 to 192.168.1.254**	Translates the outside address of Switch C to an inside address.
5.	Switch(config-l2nat)# **outside from network 10.1.1.16 to 192.168.1.16 255.255.255.240**	Translates the Node A machines' outside addresses to their inside addresses.
6.	Switch(config-l2nat)# **exit**	Exits config-l2nat mode.
7.	Switch(config)# **interface Gi1/1**	Accesses interface configuration mode for the uplink port.
8.	Switch(config-if)# **l2nat B-Subnet**	Applies this Layer 2 NAT instance to the native VLAN on this interface. **Note** For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows: l2nat *instance vlan*
9.	Switch# **show l2nat instance name1**	Shows the configuration details for the specified Layer 2 NAT instance.
10.	Switch# **show l2nat statistics**	Shows Layer 2 NAT statistics.
11.	Switch# **end**	Returns to privileged EXEC mode.



CHAPTER 5

MACsec Encryption

This chapter contains the following sections:

MACsec and the MACsec Key Agreement (MKA) Protocol

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The switch supports 802.1AE encryption with MACsec Key Agreement (MKA) on 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. The MKA protocol provides the required session keys and manages the required encryption keys.

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

Important On the ESS-3300, MACsec is supported on 1 gigabit ethernet downlink ports only.

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

Table 13: 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 and MACsec Key Agreement (MKA) are implemented after successful authentication using certificate-based MACsec or Pre Shared Key (PSK) framework.

MKA Policies

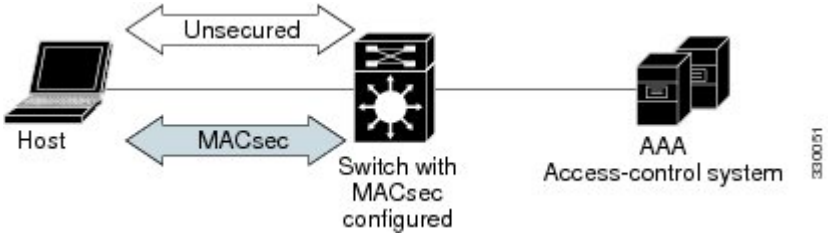
To enable MKA on an interface, a defined MKA policy should be applied to the interface. You can configure these options:

- Policy name, not to exceed 16 ASCII characters.
- Confidentiality (encryption) offset of 0, 30, or 50 bytes for each physical interface

Single-Host Mode

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

Figure 8: MACsec in Single-Host Mode with a Secured Data Session



Switch-to-Switch MKA MACsec Must Secure Policy

When MACsec is enabled on an interface, all interface traffic except EAPoL traffic is secured by default ("must-secure" is the default) on both the ingress and the egress. Unencrypted packets are dropped until the MKA session is secured. However, to enable MACsec on selected interfaces, you can choose to allow unencrypted packets to be transmitted or received from the same physical interface by setting **macsec access-control** to **should-secure**. This option allows unencrypted traffic to flow until the MKA session is secured. After the MKA session is secured, only encrypted traffic can flow. For configuration details, see Configuring MACsec MKA on an Interface using PSK, on page 102.

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 Etherchannel links that are formed as part of the port channel can either be congruent or disparate i.e. 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.

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.

This is an example of the show mka statistics command output:

```
Switch# 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
==========================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO                YES
43             c800.8459.e764/002a 1                Secured
01000000000000000000000000000000000000000000000000000000000000000000

Switch#show mka sessions interface G1/0/1
```

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

=============================================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN
=============================================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO                YES
43             c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000


Switch#show mka sessions interface G1/0/1 de

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)........... 0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89567
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

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  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
  ------------------------------------------------------------------------

Switch#show mka sessions de
Switch#show mka sessions detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)............ 01000000000000000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

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
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
  ------------------------------------------------------------------------

Switch#sh mka pol

MKA Policy Summary...
```

```
Policy            KS       Delay   Replay  Window Conf   Cipher          Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)        Applied
=============================================================================================
*DEFAULT POLICY*  0        FALSE   TRUE    0      0      GCM-AES-128

p1                1        FALSE   TRUE    0      0      GCM-AES-128

p2                2        FALSE   TRUE    0      0      GCM-AES-128     Gi1/0/1


Switch#sh mka poli
Switch#sh mka policy p2
Switch#sh mka policy p2 ?
  detail    Detailed configuration/information for MKA Policy
  sessions  Summary of all active MKA Sessions with policy applied
  |         Output modifiers
  <cr>

Switch#sh mka policy p2 de

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

Switch#sh 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     Gi1/0/1


Switch#sh mka se?
sessions

Switch#sh mka ?
  default-policy  MKA Default Policy details
  keychains       MKA Pre-Shared-Key Key-Chains
  policy          MKA Policy configuration information
  presharedkeys   MKA Preshared Keys
  sessions        MKA Sessions summary
  statistics      Global MKA statistics
  summary         MKA Sessions summary & global statistics

Switch#sh mka statis
Switch#sh mka statistics ?
  interface  Statistics for a MKA Session on an interface
  local-sci  Statistics for a MKA Session identified by its Local Tx-SCI
  |          Output modifiers
  <cr>

Switch#sh mka statistics inter
Switch#show mka statistics interface G1/0/1

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


Switch#show mka ?
  default-policy  MKA Default Policy details
  keychains       MKA Pre-Shared-Key Key-Chains
  policy          MKA Policy configuration information
  presharedkeys   MKA Preshared Keys
  sessions        MKA Sessions summary
  statistics      Global MKA statistics
  summary         MKA Sessions summary & global statistics

Switch#show mka summ
Switch#show mka summary

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
============================================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO                YES
43             c800.8459.e764/002a 1                Secured
01000000000000000000000000000000000000000000000000000000000000000000000000000000



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

Switch#
```

Certificate Based MACsec

The Certificate based MACsec Encryption feature uses 802.1X port-based authentication with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) to carry Certificates for ports where MACsec encryption is required. EAP-TLS mechanism is used for the mutual authentication and to get the Master Session Key (MSK) from which the Connectivity Association Key (CAK) is derived for the MACsec Key Agreement (MKA) protocol.

This feature allows keys to be managed at a centralized server (CA) over PSK (Pre-Shared Key) based MACsec. Switch to switch MACsec is supported. See Configuring Certificate Based MACsec, on page 104 for more information.

How to Configure MACsec Encryption

Limitations and Restrictions

MACsec has these limitations and restrictions:

- Ports should be in access mode or trunk mode.
- MKA is not supported on port-channels. Individual links that comprise the port-channel can use MACsec.
- High Availability for MKA is not supported.
- Ports with **no switchport** are not supported.
- ESS3300 uplink ports do not have a PHY and hence do not support MACSec.

Prerequisites for MACsec Encryption

Prerequisites for MACsec Encryption:

- Ensure that 802.1x authentication and AAA are configured on your device.

Configuring MKA and MACsec

Default MACsec MKA Configuration

MACsec is disabled. No MKA policies are configured.

MKA-PSK: CKN Behavior Change

A change was made in Cisco IOS XE from how the CKN (the "key") was implemented in Cisco IOS Classic. When an IE switch running Cisco IOS XE needs to make a PreShared Key (PSK) MACSec connection with an IE switch running Cisco IOS Classic, the configured "key" value must be 64 hex characters long. Also, the "key" value must match the same on the IE switch running Cisco IOS Classic. The same "key" value on the Cisco IOS Classic side does not have to pad zeros.

This Cisco IOS XE example shows key chain configuration when connecting two Cisco IOS XE devices:

```
configure terminal
key chain KEYCHAINONE macsec
key 1234
cryptographic-algorithm aes-128-cmac
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
end
```

For the above example, following is the output for the two Cisco IOS XE connected devices for the **show mka session** command:

```
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
|
==========================================================================
Gi1/1         34c0.f983.6c81/0001    POLICYONE       NO             YES
1             54a2.7498.5b01/0001    1               Secured        1234
```

Note that the CKN key-string is exactly the same that has been configured for the key as hex-string. This is an example of Cisco IOS XE to Cisco IOS XE PSK where the CKN is not zero padded.

For interoperability between devices running Cisco IOS XE and devices running Cisco IOS Classic, the Cisco IOS XE devices must zero pad the CKN value (the "key") to match the Cisco IOS Classic where the configuration CKN key does not have to be zero padded. Cisco IOS Classic zero pads the CKN key value for the user, adding zeroes after the configured CKN value to make the length 64 hex characters long. The CKN key value must be 64 hex characters long.

The following example shows configuration of the CKN key-string on a Cisco IOS Classic device:

```
config t
key chain KEYCHAINONE macsec
key 1234
cryptographic-algorithm aes-128-cmac
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
```

For the above example, following is the output on the Cisco IOS Classic device for the **show mka session** command:

```
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
==========================================================================
Gi1/1         4c0.f983.6c81/0001     POLICYONE       NO             YES
1             54a2.7498.5b01/0001    1               Secured        123400000000000
                                                                    0000000000000000
                                                                    0000000000000000
                                                                    0000000000000000
```

This example shows the configuration on the Cisco IOS XE device for interoperability with Cisco IOS Classic devices:

```
config t
key chain KEYCHAINONE macsec
key 1234000000000000000000000000000000000000000000000000000000000000
cryptographic-algorithm aes-128-cmac
key-string 123456789ABCDEF0123456789ABCDEF0
lifetime local 12:21:00 Sep 9 2015 infinite
```

For the above example, following is the **show mka session** output on the Cisco IOS XE device:

```
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
==========================================================================
Gi1/1        34c0.f983.6c81/0001  POLICYONE       NO             YES
1            54a2.7498.5b01/0001  1               Secured        123400000000000000
                                                                 00000000000000000000
                                                                 000000000000000
                                                                 000000000000
```

Configuring an MKA Policy

SUMMARY STEPS

1. **configure terminal**
2. **mka policy** *policy name*
3. **send-secure-announcements**
4. **key-server** *priority*
5. **include-icv-indicator**
6. **macsec-cipher-suite** *gcm-aes-128*
7. **confidentiality-offset** *Offset value*
8. **end**
9. **show mka policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mka policy** *policy name*	Identify an MKA policy, and enter 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 "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 3	**send-secure-announcements**	Enabled secure announcements.

	Command or Action	Purpose
		Note By default, secure announcements are disabled.
Step 4	**key-server** *priority*	Configure 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**	Enables the ICV indicator in MKPDU. Use the no form of this command to disable the ICV indicator — **no include-icv-indicator**.
Step 6	**macsec-cipher-suite** *gcm-aes-128*	Configures cipher suite for deriving SAK with 128-bit encryption.
Step 7	**confidentiality-offset** *Offset value*	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	**end**	Returns to privileged EXEC mode.
Step 9	**show mka policy**	Verify your entries.

Example

This example configures the MKA policy:

```
Switch(config)# mka policy mka_policy
Switch(config-mka-policy)# key-server priority 200
Switch(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Switch(config-mka-policy)# confidentiality-offset 30
Switch(config-mka-policy)# end
```

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter the password if prompted.
Step 2	**configureterminal** **Example:** `Device>configure terminal`	Enters the global configuration mode.
Step 3	**interface** *type number* **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***vlan-id* **Example:** `Device(config-if)# switchport access vlan 1`	Configures the access VLAN for the port.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Configures the interface as an access port.
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** *vlan-id* **Example:**	(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.

	Command or Action	Purpose
	`Device(config-if)# authentication event linksec fail action authorize vlan 1`	
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 *should secure*.
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** *policy-name* **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).
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

	Command or Action	Purpose
Step 17	**end** **Example:** `Device(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 18	**show authentication session interface** *interface-id* **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.

Configuring MACsec MKA using Pre Shared Key (PSK)

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**key chain** *key-chain-name* **macsec**	Configures a key chain and enters the key chain configuration mode.
Step 3	**key** *hex-string*	Configures a unique identifier for each key in the keychain and enters the keychain's key configuration mode. **Note** For 128-bit encryption, use 32 hex digit key-string. For 256-bit encryption, use 64 hex digit key-string.
Step 4	**cryptographic-algorithm** {*gcm-aes-128*	*gcm-aes-256*}	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 5	**key-string** { [0|6|7] *pwd-string*	*pwd-string*}	Sets the password for a key string. Only hex characters must be entered..

	Command or Action	Purpose
Step 6	**lifetime local** [*start timestamp {hh::mm::ss | day | month | year}*] [**duration** *seconds* | *end timestamp {hh::mm::ss | day | month | year}*]	Sets the lifetime of the pre shared key.
Step 7	**end**	Returns to privileged EXEC mode.

Example

Following is an indicative example:

```
Switch(config)# Key chain keychain1 macsec
Switch(config-key-chain)# key 1000
Switch(config-keychain-key)# cryptographic-algorithm gcm-aes-128
Switch(config-keychain-key)# key-string 12345678901234567890123456789012
Switch(config-keychain-key)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016
Switch(config-keychain-key)# end
```

Configuring MACsec MKA on an Interface using PSK

Note To avoid traffic drop across sessions, the **mka policy** command must be configured before the **mka pre-shared-key key-chain** command.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **macsec access-control should-secure**
4. **macsec**
5. **mka policy** *policy-name*
6. **mka pre-shared-key key-chain** *key-chain name*
7. **macsec replay-protection window-size** *frame number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode.
Step 3	**macsec access-control should-secure**	(Optional) Allows unencrypted traffic to flow until the MKA session is secured. After the MKA session is secured, only encrypted traffic can flow. By default, traffic is dropped until the MKA session is secured.

	Command or Action	Purpose
		To revert to the default behavior, use the **no macsec access-control should-secure** command.
Step 4	**macsec**	Enables MACsec on the interface.
Step 5	**mka policy** *policy-name*	Configures an MKA policy.
Step 6	**mka pre-shared-key key-chain** *key-chain name*	Configures an MKA pre-shared-key key-chain name.
Step 7	**macsec replay-protection window-size** *frame number*	Sets the MACsec window size for replay protection.
Step 8	**end**	Returns to privileged EXEC mode.

Example

The following example configures an MKA policy and an MKA pre-shared-key key-chain name, and sets the MACsec window size for replay protection:

```
Switch(config)# interface GigabitEthernet 1/1
Switch(config-if)# mka policy mka_policy
Switch(config-if)# mka pre-shared-key key-chain key-chain-name
Switch(config-if)# macsec replay-protection window-size 10
Switch(config-if)# end
```

Note 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 configuration on each of the participating nodes using the **no macsec** command.
2. Configure the MKA policy on the interface on each of the participating nodes using the **mka policy policy-name** command.
3. Enable the new session on each of the participating node by using the **macsec** command.

The following examples show how to configure the interface to use **should-secure** instead of the default **must-secure** and how to change it back to the default **must-secure**.

Note Modifying **access-control** is not allowed when the session is up and running. You first need to remove the MACsec configuration by using the **no macsec** command, and then configure **access-control**.

Example 1: To change from **must-secure** to **should-secure**:

```
Switch(config-if)#no macsec
Switch(config-if)#macsec access-control should-secure
Switch(config-if)#macsec // this switches the access-control from must-secure & restarts
the macsec session with new behaviour.
```

Example 2: To change from **should-secure** to **must-secure**:

```
Switch(config-if)#no macsec
Switch(config-if)#no macsec access-control
Switch(config-if)#macsec
```

Configuring Certificate Based MACsec

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

- Generating Key Pairs
- Configuring Enrollment using SCEP
- Configuring Enrollment Manually
- Enabling 802.1x Authentication and Configuring AAA, on page 109
- Configuring EAP-TLS Profile and 802.1x Credentials, on page 111
- Applying the 802.1x MKA MACsec Configuration on Interfaces, on page 113

Prerequisites for Certificate Based MACsec

- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate or obtain a third-party certificate.
- Ensure that you have configured Cisco Identity Services Engine (ISE).
- Ensure that 802.1x authentication and AAA are configured on your device.

Generating 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

	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
	`Device# configure terminal`	
Step 3	**crypto key generate rsa label** *label-name* **general-keys modulus** *size* **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** *interface-id* **Example:** `Device# show authentication session interface gigabitethernet 0/1/1`	Verifies the authorized session security status.

Configuring 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.

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	**crypto pki trustpoint** *server name* **Example:** `Device(config)# crypto pki trustpoint ka`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url name pem* **Example:**	Specifies the URL of the CA on which your device should send certificate requests.

	Command or Action	Purpose
	`Device(ca-trustpoint)# enrollment url http://url:80`	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** *key-label key-sizeencryption-key-size* **Example:** `Device(ca-trustpoint)# rsakeypair exampleCAkeys`	Specifies which key pair to associate with the certificate. • A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The *key-size* and *encryption-key-size* must be the same size. Length of less than 2048 is not recommended. **Note** The **rsakeypair** name must match the trust-point name. **Note** If this command is not enabled, the FQDN key pair is used.
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	**auto-enroll** *percent* **regenerate** **Example:** `Device(ca-trustpoint)# auto-enroll 90 regenerate`	Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA. If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration. By default, only the Domain Name System (DNS) name of the device is included in the certificate. 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.

	Command or Action	Purpose
		Use the regenerate keyword to generate a new key for the certificate even if a named key already exists. 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." It is recommended that a new key pair be generated for security reasons.
Step 10	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 11	**crypto pki authenticate** *name* **Example:** `Device(config)# crypto pki authenticate myca`	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** *trustpoint name* **Example:** `Device# show crypto pki certificate ka`	Displays information about the certificate for the trust point.

Configuring 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:

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	**crypto pki trustpoint** *server name* **Example:**	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
	`Device# crypto pki trustpoint ka`	
Step 4	**enrollment url** *url-name* **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** *key-label key-sizeencryption-key-size* **Example:** `Device(ca-trustpoint)# rsakeypair exampleCAkeys`	Specifies which key pair to associate with the certificate. • A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The *key-size* and *encryption-key-size* must be the same size. Length of less than 2048 is not recommended. **Note** The **rsakeypair** name must match the trust-point name. **Note** If this command is not enabled, the FQDN key pair is used.
Step 6	**serial-number none** **Example:** `Device(ca-trustpoint)# serial-number none`	Specifies that serial numbers 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:**	Retrieves the CA certificate and authenticates it.

	Command or Action	Purpose
	`Device(config)# crypto pki authenticate myca`	
Step 11	**crypto pki enroll** *name* **Example:** `Device(config)# crypto pki enroll myca`	Generates certificate request and displays the request for copying and pasting into the certificate server. Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 12	**crypto pki import** *name* **certificate** **Example:** `Device(config)# crypto pki import myca certificate`	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. 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. The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the switch. **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 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** *trustpoint name* **Example:** `Device# show crypto pki certificate  ka`	Displays information about the certificate for the trust point.

Enabling 802.1x Authentication and Configuring 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

	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	aaa new-model **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	dot1x system-auth-control **Example:** `Device(config)# dot1x system-auth-control`	Enables 802.1X on your device.
Step 5	radius server *name* **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.
Step 6	address *ip-address* auth-port *port-number* acct-port *port-number* **Example:** `Device(config-radius-server)# address ipv4 <ISE ipv4 address> auth-port 1645 acct-port 1646`	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 7	automate-tester username username **Example:** `Device(config-radius-server)# automate-tester username dummy`	Enables the automated testing feature for the RADIUS server. With this practice, the device sends periodic test authentication messages to the RADIUS server. It looks

	Command or Action	Purpose
		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.
Step 8	key *string* **Example:** `Device(config-radius-server)# key dummy123`	Configures the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
Step 9	radius-server deadtime *minutes* **Example:** `Device(config-radius-server)# radius-server deadtime 2`	Improves RADIUS response time when some servers might be unavailable and skips unavailable servers immediately.
Step 10	exit **Example:** `Device(config-radius-server)# exit`	Returns to global configuration mode.
Step 11	aaa group server radius *group-name* **Example:** `Device(config)# aaa group server radius ISEGRP`	Groups different RADIUS server hosts into distinct lists and distinct methods, and enters server group configuration mode.
Step 12	server *name* **Example:** `Device(config-sg)# server name ISE`	Assigns the RADIUS server name.
Step 13	exit **Example:** `Device(config-sg)# exit`	Returns to global configuration mode.
Step 14	aaa authentication dot1x default group *group-name* **Example:** `Device(config)# aaa authentication dot1x default group ISEGRP`	Sets the default authentication server group for IEEE 802.1x.
Step 15	aaa authorization network default group *group-name* **Example:** `aaa authorization network default group ISEGRP`	Sets the network authorization default group.

Configuring EAP-TLS Profile and 802.1x Credentials

SUMMARY STEPS

1. enable
2. configure terminal
3. eap profile *profile-name*
4. method tls

5. pki-trustpoint *name*
6. exit
7. dot1x credentials *profile-name*
8. username *username*
9. pki-trustpoint *name*
10. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	eap profile *profile-name* **Example:** `Device(config)# eap profile EAPTLS-PROF-IOSCA`	Configures EAP profile and enters EAP profile configurationmode.
Step 4	method tls **Example:** `Device(config-eap-profile)# method tls`	Enables EAP-TLS method on the device.
Step 5	pki-trustpoint *name* **Example:** `Device(config-eap-profile)# pki-trustpoint POLESTAR-IOS-CA`	Sets the default PKI trustpoint.
Step 6	exit **Example:** `Device(config-eap-profile)# exit`	Returns to global configuration mode.
Step 7	dot1x credentials *profile-name* **Example:** `Device(config)# dot1x credentials EAPTLSCRED-IOSCA`	Configures 802.1x credentials profile and enters dot1x credentials configuration mode.
Step 8	username *username* **Example:** `Device(config-dot1x-cred)# username asr1000@polestar.company.com`	Sets the authentication user ID.
Step 9	pki-trustpoint *name* **Example:**	Sets the default PKI trustpoint.

	Command or Action	Purpose
	`Device(config-dot1x-cred)# pki-trustpoint POLESTAR-IOS-CA`	
Step 10	end **Example:** `Device(config-dot1x-cred)# end`	Returns to privileged EXEC mode.

Applying the 802.1x MKA MACsec Configuration on Interfaces

To apply MACsec MKA using certificate-based MACsec encryption to interfaces, perform the following task:

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 gigabitethernet 2/9`	Identifies the MACsec interface, and enters interface configuration mode. The interface must be a physical interface.
Step 4	macsec **Example:** `Device(config-if)# macsec`	Enables MACsec on the interface.
Step 5	authentication periodic **Example:** `Device(config-if)# authentication periodic`	(Optional) Enables reauthentication for this port.
Step 6	authentication timer reauthenticate interval **Example:** `Device(config-if)# authentication timer reauthenticate interval`	(Optional) Sets the reauthentication interval.
Step 7	access-session host-mode multi-domain **Example:** `Device(config-if)# access-session host-mode multi-domain`	Allows hosts to gain access to the interface.

	Command or Action	Purpose
Step 8	**access-session closed** **Example:** `Device(config-if)# access-session closed`	Prevents preauthentication access on the interface.
Step 9	access-session port-control auto **Example:** `Device(config-if)# access-session port-control auto`	Sets the authorization state of a port.
Step 10	dot1x pae both **Example:** `Device(config-if)# dot1x pae both`	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 11	dot1x credentials *profile* **Example:** `Device(config-if)# dot1x credentials EAPTLSCRED-IOSCA`	Assigns a 802.1x credentials profile to the interface.
Step 12	dot1x supplicant eap profile *name* **Example:** `Device(config-if)# dot1x supplicant eap profile EAPTLS-PROF-IOSCA`	Assigns the EAP-TLS profile to the interface.
Step 13	dot1x authenticator eap profile *name* **Example:** `Device(config-if)# dot1x authenticator eap profile EAPTLS-PROF-IOSCA`	Assigns the EAP profile to use during 802.1x authentication.
Step 14	service-policy type control subscriber *control-policy name* **Example:** `Device(config-if)# service-policy type control subscriber DOT1X_POLICY_RADIUS`	Applies a subscriber control policy to the interface.
Step 15	exit **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 16	show macsec interface interface-id **Example:** `Device# show macsec interface GigabitEthernet 2/9`	Displays MACsec details for the interface.
Step 17	**show access-session interface** *interface-id* **details** **Example:** `Device# show access-session interface GigabitEthernet 2/9 details`	Verifies successful dot1x authentication and authorization. This is the first thing to check. If dot1x authentication fails, then MKA will never start.

	Command or Action	Purpose
Step 18	**show mka session interface** *interface-id* **details** **Example:** `Device# show mka session interface GigabitEthernet 2/9 details`	Displays detailed MKA session status.

Example: Switch-to-Switch Certificate Based MACsec

An example configuration of switch-to-switch certificate based MACsec is shown below.

```
configure terminal
aaa new-model
aaa local authentication default authorization default
!
!
aaa authentication dot1x default group radius local
aaa authorization exec default local
aaa authorization network default group radius local
aaa authorization auth-proxy default group radius
aaa authorization credential-download default local
aaa accounting identity default start-stop group radius
!
!
aaa attribute list MUSTS
 attribute type linksec-policy must-secure
!
aaa attribute list macsec-dot1-credentials
 attribute type linksec-policy must-secure
!
aaa attribute list MUSTS_CA
 attribute type linksec-policy must-secure
!
aaa attribute list SHOULDS_CA
 attribute type linksec-policy should-secure
!
aaa attribute list mkadt_CA
 attribute type linksec-policy must-secure
!
aaa session-id common

username MUST aaa attribute list MUSTS_CA
username MUSTS.mkadt.cisco.com

crypto pki trustpoint demo
 enrollment terminal
 serial-number
 fqdn MUSTS.mkadt.cisco.com
 subject-name cn=MUSTS.mkadt.cisco.com,OU=CSG Security,O=Cisco Systems,L=Bengaluru,ST=KA,C=IN

 subject-alt-name MUSTS.mkadt.cisco.com
 revocation-check none
 rsakeypair demo 2048
 hash sha256

eap profile EAP_P
  method tls
 pki-trustpoint demo

dot1x system-auth-control
dot1x credentials MUSTS-CA
```

```
 username MUST
 password 0 MUST_CA
!
dot1x credentials MUSTS
 username MUSTS.mkadt.cisco.comcrypto pki authenticate demo

crypto pki authenticate
crypto pki enroll demo
crypto pki import demo certificate

policy-map type control subscriber MUSTS_1
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x both
 event authentication-failure match-all
  10 class always do-until-failure
   10 terminate dot1x
   20 authentication-restart 10
 event authentication-success match-all
  10 class always do-until-failure
   10 activate service-template DEFAULT_LINKSEC_POLICY_MUST_SECURE

interface GigabitEthernet2/9
 switchport mode access
 macsec
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae both
 dot1x authenticator eap profile EAP_P
 dot1x credentials MUSTS
 dot1x supplicant eap profile EAP_P
 service-policy type control subscriber MUSTS_1
```

The following example shows output of the **show mka sessions** command for Switch-to-Switch Certificate Based MACsec.

```
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

=============================================================================================
Gi2/14          40ce.24b7.617d/0002  pol_1             NO                 YES

2               f8b7.e2e5.ad88/0002  1                 Secured
80690202D09A9801BE98FC89D5380098



show mka sessions interface GigabitEthernet2/14 detail

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. 40ce.24b7.617d/0002
Interface MAC Address.... 40ce.24b7.617d
MKA Port Identifier...... 2
```

```
Interface Name.......... GigabitEthernet2/14
Audit Session ID........ 6514030B000000998FEDD629
CAK Name (CKN).......... 80690202D09A9801BE98FC89D538009800000000000000000000000000000000
Member Identifier (MI)... 534A6ECFBBA318B6423E49EB
Message Number (MN)..... 166
EAP Role................ Authenticator
Key Server.............. YES
MKA Cipher Suite........ AES-128-CMAC

Latest SAK Status....... Rx & Tx
Latest SAK AN........... 0
Latest SAK KI (KN)...... 534A6ECFBBA318B6423E49EB00000001 (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)
SAK Rekey Time.......... 0s (SAK Rekey interval not applicable)

MKA Policy Name......... pol_1
Key Server Priority..... 0
Delay Protection........ NO
Delay Protection Timer.......... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility....... 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO
SAK Cipher Suite........ 0080C20001000002 (GCM-AES-256)
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        RxSA          SSCI
                                                            Priority  Installed
  ------------------------------------------------------------------------------------
  96E534A06B405442034B846E  163         f8b7.e2e5.ad88/0002  0         YES            0

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

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

show access-session interface GigabitEthernet2/14 detail
            Interface:  GigabitEthernet2/14
               IIF-ID:  0x1398D40E
          MAC Address:  f8b7.e2e5.ad88
         IPv6 Address:  Unknown
         IPv4 Address:  Unknown
            User-Name:  MUST
               Status:  Authorized
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
```

```
      Session timeout:  1800s (local), Remaining: 1470s
       Timeout action:  Reauthenticate
    Common Session ID:  6514030B000000998FEDD629
      Acct Session ID:  0x0000000e
               Handle:  0x5900003a
       Current Policy:  MUSTS_1


Local Policies:
               Service Template: DEFAULT_LINKSEC_POLICY_MUST_SECURE (priority 150)

Server Policies:
      Security Policy:  Must Secure
      Security Status:  Link Secured


Method status list:
       Method           State
        dot1x           Authc Success
     dot1xSup           Authc Success
```

Configuring MKA/MACsec for Port Channel

Configuring MKA/MACsec for Port Channel Using PSK

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **macsec**
4. **mka policy** *policy-name*
5. **mka pre-shared-key key-chain** *key-chain-name*
6. **channel-group** *channel-group-number* **mode {active | passive } | {on }**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode.
Step 3	**macsec**	Enables MACsec on the interface. Supports layer 2 and layer 3 port channels.
Step 4	**mka policy** *policy-name*	Configures an MKA policy.
Step 5	**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 sub-interfaces and not on both.

	Command or Action	Purpose
Step 6	**channel-group** *channel-group-number* **mode {active | passive } | {on }**	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. The port channel associated with this channel group is automatically created if the port channel does not already exist.For mode, select one of the following keywords: • **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 7	**end**	Returns to privileged EXEC mode.

Configuring Port Channel Logical Interfaces for Layer 2 EtherChannels

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

SUMMARY STEPS

1. **configure terminal**
2. **[no] interface port-channel** *channel-group-number*
3. **switchport**
4. **switchport mode {access | trunk }**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**[no] interface port-channel** *channel-group-number*	Creates the port channel interface. **Note** Use the no form of this command to delete the port channel interface.
Step 3	**switchport**	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 4	**switchport mode {access | trunk }**	Assigns all ports as static-access ports in the same VLAN, or configure them as trunks.

	Command or Action	Purpose
Step 5	**end**	Returns to privileged EXEC mode.

Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels

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

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *interface-id*
3. **no switchport**
4. **ip address** *ip-address subnet_mask*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface port-channel** *interface-id*	Enters interface configuration mode.
Step 3	**no switchport**	Switches an interface that is in Layer 2 mode into Layer 3 mode for Layer 3 configuration.
Step 4	**ip address** *ip-address subnet_mask*	Assigns an IP address and subnet mask to the EtherChannel.
Step 5	**end**	Returns to privileged EXEC mode.

Example: Configuring MACsec MKA for Port Channel using PSK

Etherchannel Mode — Static/On

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

```
key chain KC macsec
  key 1000
    cryptographic-algorithm aes-128-cmac
    key-string FC8F5B10557C192F03F60198413D7D45
    end

mka policy POLICY
  key-server priority 0
  macsec-cipher-suite gcm-aes-128
  confidentiality-offset 0
  end

interface Te1/0/1
  channel-group 2 mode on
  macsec
  mka policy POLICY
  mka pre-shared-key key-chain KC
  end
```

```
interface Te1/0/2
  channel-group 2 mode on
  macsec
  mka policy POLICY
  mka pre-shared-key key-chain KC
  end
```

Layer 2 EtherChannel Configuration

Device 1

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

The following shows a sample output of **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

```
interface port-channel 2
 no switchport
```

```
ip address 10.25.25.3 255.255.255.0
no shutdown
end
```

Device 2

```
interface port-channel 2
 no switchport
 ip address 10.25.25.4 255.255.255.0
 no shutdown
 end
```

The following shows a sample output of **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 a sample configuration on Device 1 and Device 2 with EtherChannel Mode as LACP.

```
key chain KC macsec
  key 1000
    cryptographic-algorithm aes-128-cmac
    key-string FC8F5B10557C192F03F60198413D7D45
    end

mka policy POLICY
  key-server priority 0
  macsec-cipher-suite gcm-aes-128
  confidentiality-offset 0
  end

interface Te1/0/1
  channel-group 2 mode active
  macsec
```

```
  mka policy POLICY
  mka pre-shared-key key-chain KC
  end

interface Te1/0/2
  channel-group 2 mode active
  macsec
  mka policy POLICY
  mka pre-shared-key key-chain KC
  end
```

Layer 2 EtherChannel Configuration

Device 1

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 switchport
 switchport mode trunk
 no shutdown
 end
```

The following shows a sample output of **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      Te1/1/1(P)  Te1/1/2(P)
```

Layer 3 EtherChannel Configuration

Device 1

```
interface port-channel 2
 no switchport
 ip address 10.25.25.3 255.255.255.0
 no shutdown
 end
```

Device 2

```
interface port-channel 2
 no switchport
 ip address 10.25.25.4 255.255.255.0
 no shut
```

The following shows a sample output of **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      Te1/1/1(P)  Te1/1/2(P)
```

Displaying Active MKA Sessions

The following shows all the active MKA sessions.

```
# show mka sessions interface Te1/0/1
=======================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN

=======================================================================================
Te1/0/1        00a3.d144.3364/0025 POLICY           NO                NO

37             701f.539b.b0c6/0032 1                Secured
1000
```

Configuring MACsec Cipher Announcement

Configuring an MKA Policy for Secure Announcement

SUMMARY STEPS

1. **configure terminal**
2. **mka policy** *policy-name*
3. **key-server** *priority*
4. **[no] send-secure-announcements**
5. **macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}
6. **end**
7. **show mka policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mka policy** *policy-name*	Identify an MKA policy, and enter 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 "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 3	**key-server** *priority*	Configure 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 4	**[no] send-secure-announcements**	Enables sending of secure announcements. Use the no form of the command to disable sending of secure announcements. By default, secure announcements are disabled.
Step 5	**macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}	Configures cipher suite for deriving SAK with 128-bit or 256-bit encryption.
Step 6	**end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show mka policy**	Verify your entries.

Configuring Secure Announcement Globally (Across all the MKA Policies)

SUMMARY STEPS

1. **configure terminal**
2. **[no] mka defaults policy send-secure-announcements**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**[no] mka defaults policy send-secure-announcements**	Enables sending of secure announcements in MKPDUs across MKA policies. By default, secure announcements are disabled.
Step 3	**end**	Returns to privileged EXEC mode.

Configuring EAPoL Announcements on an interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **[no] eapol annoucement**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 3	**[no] eapol annoucement**	Enable EAPoL announcements. Use the no form of the command to disable EAPoL announcements. By default,EAPoL announcements are disabled.
Step 4	**end**	Returns to privileged EXEC mode.

Examples: Configuring MACsec Cipher Announcement

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

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

This example shows how to configure Secure Announcement globally:

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

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

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

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

```
# 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 a sample output of the **show mka sessions interface** *interface-name* **detail** command with secure announcement disabled.

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


MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI.............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
01000000000000000000000000000000000000000000000000000000000000000000
```

```
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89567
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

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  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
  ----------------------------------------------------------------------
```

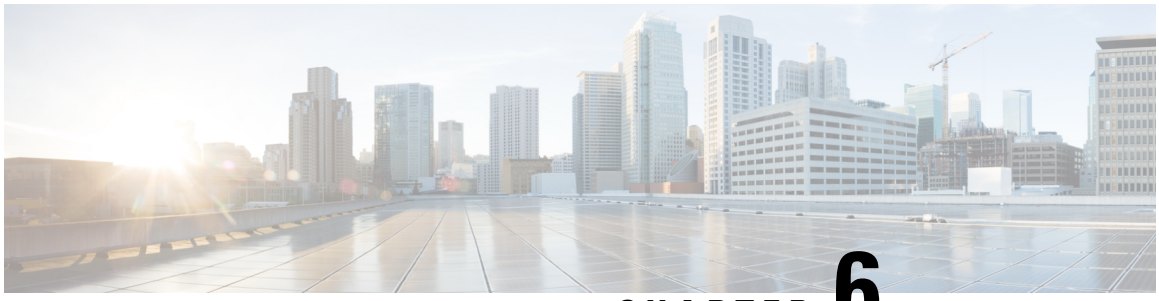
The following is a sample output of the **show mka sessions details** command with secure announcement disabled.

```
# show mka sessions details
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI............. 204c.9e85.ede4/002b
```

```
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

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 a sample output of the **show mka policy** *policy-name* **detail** command with secure announcement disabled.

```
# 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
```



CHAPTER 6

Configuring the Secure Cloud Analytics Connector

Configuring Cisco Connector for Secure Cloud Analytics

Cisco Secure Cloud Analytics (formerly Stealthwatch Cloud) provides the actionable security intelligence and visibility necessary to identify these kinds of malicious activities in real time. You can quickly respond before a security incident becomes a devastating breach. This guide will walk you through setting up the Cisco Cloud Connector in IOS-XE, on a Cisco Industrial Ethernet Switch.

Note For further information about **Cisco Secure Cloud Analytics (Stealthwatch Cloud)** or **Cisco Secure Network Analytics (Stealthwatch)** go to the following URL: https://www.cisco.com/c/en/us/products/security/stealthwatch/index.html.

Limitations and Restrictions

- Only a predetermined set of fields can be collected - These include 9-tuple flow data of Src IP, Src Port, Dst IP, Dst Port and Protocol along with Flow Start, Flow End, Number of Packets and Bytes
- The mandatory fields are not enforced through CLI restrictions. In case a record does not have all the mandatory fields and we are unable to collect 9-tuple data, we shall discard that flow.
- The StealthWatch Connector for Secure Cloud Analytics will rely on the Switch's routing functionality to send the packet to the Cloud Servers. No additional checks are done. Assumption is that appropriate routes exist.
- Monitor application restrictions inherent with Flexible Net Flow in terms of monitor application holds true with Secure Cloud Analytics as well. e.g no SVI, no VLAN, no egress monitor.
- The cloud exporter can't be used with other exporters.
- The uploaded file naming convention includes a random string to uniquely identify every file and to prevent file overwrites. Example:

https://sensor.ext.obsrvbl.com/sign/ios-xe-17-2/2019/7/5/00:00:00/hostname-random_suffix.csv.gz We will aggregate and upload every 1 minute.

Before you begin

The Secure Cloud Analytics Connector is supported on **IE3300, IE3400, IE3400H Switches** only.

- **Network Advantage** and **dna-advantage license**

SUMMARY STEPS

1. stealthwatch-cloud-monitor service-key <you service key> hostname my_sensor
2. flow record SWCRec
3. flow exporter SWCExp
4. interface gi1/0/3

DETAILED STEPS

	Command or Action	Purpose

Step 1	stealthwatch-cloud-monitor service-key <you service key> hostname my_sensor
	Example:
	`stealthwatch-cloud-monitor`
	`service-key <you service key>`
	`hostname my_sensor`
	`url https://sensor.ext.obsrvbl.com`
	`openssl s_client -showcerts -connect`
	`https://sensor.ext.obsrvbl.com:443`
	`openssl s_client -showcerts -connect`
	`s3.ap-southeast-2.amazonaws.com:443`
	Example:
	`openssl s_client -showcerts -connect`
	`https://sensor.ext.obsrvbl.com:443`
	`openssl s_client -showcerts -connect`
	`s3.ap-southeast-2.amazonaws.com:443`
		Please have valid root CAs installed based on your URL. Please use below CLI to figure out the ROOT CAs as per your URL
		Configuring the service-key and hostname, which is used for sensor registration. If no hostname is provided, the serial number of the box is used for registration.

Step 2	flow record SWCRec
	Example:
	`flow record SWCRec`
	`match ipv4 source address`
	`match ipv4 destination address`
	`match transport source-port`
	`match transport destination-port`
	`match ipv4 protocol`
	`collect counter bytes long`
	`collect counter packets long`
	`collect timestamp sys first`
	`collect timestamp sys last`
		Configure the fields in flow record for collecting data for Secure Cloud Analytics record.

Step 3	flow exporter SWCExp **Example:**	Configure a Secure Cloud Analytics exporter and attach it to a flow monitor to start exporting to Secure Cloud.

	Command or Action	Purpose
	flow exporter SWCExp destination stealthwatch-cloud flow monitor SWCMon flow record SWCRec flow exporter SWCExp	
Step 4	interface gi1/0/3 **Example:** Interface gi1/0/3 ip flow monitor SWCMon input	Identify the interface on which you want to monitor the flows and attach the monitor having Secure Cloud Analytics exporter to that interface

What to do next

For further Secure Cloud Analytics configuration information, refer to the appropriate configuration guide here: https://www.cisco.com/c/en/us/support/security/stealthwatch-cloud/products-installation-and-configuration-guides-list.html.

Troubleshooting

- debug logs can be enabled by using 'debug Stealthwatch' CLIs

```
switch#debug stealthwatch-cloud ?
  all          All debugs for SWC
  cert         Certificate Validation
  error        errors
  event        Events
  file-events  File notifications
```

- For Platform level debugs you may use "debug platform software swc" CLIs

```
switch#debug platform software swc ?
  all         all
  errors      Stealthwatch Cloud errors
  events      Stealthwatch Cloud events
  pkt-events  Stealthwatch Cloud data collection events
```

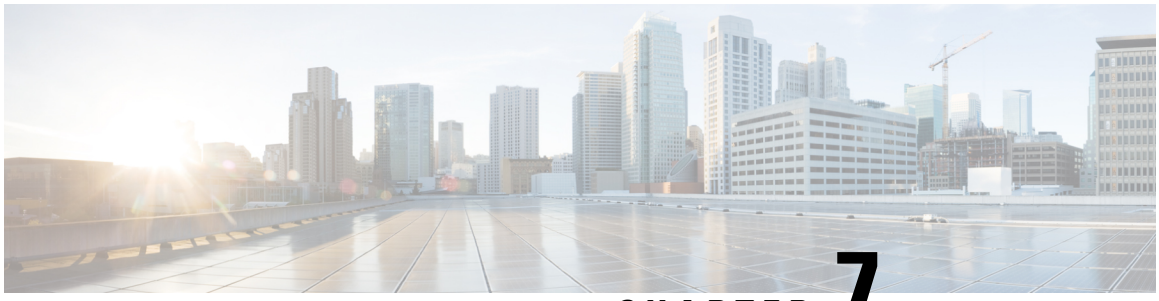
Show Commands

- **Switch-1# show stealthwatch-cloud detail**

```
=======================================
Stealthwatch Cloud Parameters
=======================================
   Service Key  : x8SS2q7e4twpcNWT35AsL6i6xHd24iXJvICo3N4sGx1U1pCqqs
   Sensor Name  : petra
   URL          : https://sensor.anz-prod.obsrvbl.com
=======================================
Stealthwatch Cloud Sensor Info
=======================================
   Sensor Status    : Registered
   Last heartbeat   : 2020-05-08T12:11:50
```

- **Switch-1# show platform software swc stats**

```
=========================
SWC Upload Statistics:
=========================
1 : Last file uploaded          :  202005081212_ufihi2
2 : Time of upload              :  202005081213 UTC
3 : Current file uploading      :
4 : Files queued for upload     :
5 : Number of files queued      :  0
6 : Last failed upload          :
7 : Files failed to upload      :  0
8 : Files successfully uploaded :  416
=========================
SWC File Creation Statistics:
=========================
9 : Last file created           :  202005081212_ufihi2
10: Time of creation            :  202005081212 UTC
=========================
SWC Flow Statistics:
=========================
11: Number of flows in prev file:  1
12: Number of flows in curr file:  0
13: Invalid dropped flows       :  0
=========================
SWC Flags:
=========================
14: Is Registered               :  Registered
15: File Delete         :  Enabled
16: Exporter            :  Enabled
```



CHAPTER 7

Configuring SGACL Logging

SGACL Logging

Security group-based access control list (SGACL) Logging is supported on Cisco IE3400 and IE3400H Series Switches in Cisco IOS XE Release 17.8.1 and later. Support for SGACL Logging also requires that one of the following FPGA Profiles be activated on the switch:

- Default Profile
- CTS-IPv6 Profile

For information about FPGA Profile, see System Management Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches.

Security group access control lists (SGACLs) are a policy enforcement method through which the administrator can control operations performed by a user, based on security group assignments and destination resources. SGACL is a component of the Cisco TrustSec security architecture, which builds secure networks by establishing domains of trusted network devices. For comprehensive information about TrustSec, including TrustSec prerequisites, guidelines and limitations, and configuration procedures, see Cisco TrustSec Configuration Guide.

Logging-enabled access control lists (ACLs) provide insight into traffic as it traverses the network or is dropped by network devices. The device can provide logging messages about packets permitted or denied by a role-based IPv4/v6 access list. That is, any packet that matches the SGACL causes an informational logging message about the packet to be sent to the console. Logging is triggered only when the Access Control Entry (ACE) includes the log keyword. The level of messages logged to the console is controlled by the **logging console** command controlling the syslog messages.

Following is an example syslog message that is generated after a specific ACE that is configured for logging is matched:

```
*Jun 18 10:17:22.205: %RBM-6-SGACLHIT: ingress_interface='' sgacl_name='testv4'
action='Permit'
 protocol='udp' src-vrf='default' src-ip='25.1.1.1' src-port='96' dest-vrf='default'
```

```
dest-ip='25.1.1.2'
dest-port='0' sgt='100' dgt='200' logging_interval_hits='12'
```

The logging message includes the access list name, whether the packet was permitted or denied, the source and destination IP addresses of the packet, and information regarding the security group tag (SGT) and destination group tag (DGT).

The following table shows the types of ACE operations in IPv4/v6 role-based ACLs supported on the switch. The log keyword applies to individual ACEs and causes packets that match the ACE to be logged. The first packet logged by the **log** keyword generates a syslog message.

Note SGACL Logging is supported for ACEs with the OR logical operator. SGACL Logging is not supported for operations with the AND logical operator.

SGACL command	Description
permit/deny tcp src eq <src-port> or dst eq <dst-port> log	Matches TCP packets based on the specified source port or destination port.
permit/deny udp src eq <src-port> or dst eq <dst-port> log	Matches UDP packets based on the specified source port or destination port.
permit/deny tcp src range <start-port> <end-port> or dst range <start-port> <end-port> log	Matches TCP packets based on the range specified for source ports or destination ports.
permit/deny udp src range <start-port> <end-port> or dst range <start-port> <end-port> log	Matches UDP packets based on the range specified for source ports or destination ports.
Permit/deny tcp src gt/lt <src-port> or dst gt/lt <dst-port> log	Matches TCP packets that are greater than or lesser than the specified source port or greater than or lesser than the specified destination port.
Permit/deny udp src gt/lt <src-port> or dst gt/lt <dst-port> log	Matches UDP packets that are greater than or lesser than the specified source port or greater than or lesser than the specified destination port.

Prerequisites

- SGACL Logging requires that one of the following FPGA Profiles be activated on the switch:
  - Default Profile
  - CTS-IPv6 Profile

  For information about FPGA Profile, see System Management Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches.
- Configure Security Group ACL Policies as described in How to Configure Security Group ACL Policies

Guidelines and Limitations

- FPGA can support a maximum of 254 entries for an instance.
- Syslog entries for ingress interfaces are empty due to hardware limitations.

Configuring SGACL Logging

To configure SGACL logging:

Step 1 Enter global configuration mode:

Example:

```
Switch# configure terminal
```

Step 2 Use the **cts role-based enforcement** command to globally enable or disable SGACL enforcement for Cisco TrustSec-enabled interfaces in the system.

Step 3 To configure a logging interval for an SGACL, enter:

cts role-based enforcement [logging-interval *interval*]

The valid values for the *interval* argument are from 5 to 86400 seconds. The default is 300 seconds.

Example:

```
Switch(config)# cts role-based enforcement logging-interval 90
```

Step 4 (Optional) Use the **logging rate-limit** command to limit the rate of messages logged per second.

Example:

```
Switch(config)# logging rate-limit <seconds>
```

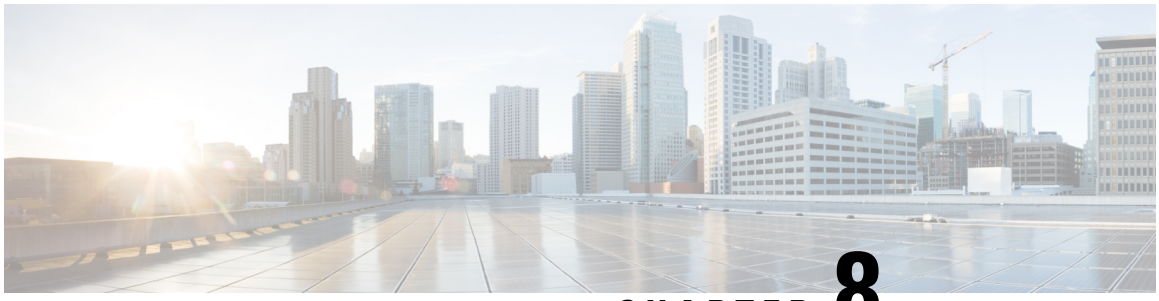
Example

The following is a sample log, displaying source and destination SGTs. ACE matches for deny action. The **logging rate-limit** command can be used to limit the rate of messages logged per second.

```
Switch(config)# cts role-based enforcement logging-interval 90
Switch(config)# logging rate-limit 20
May 27 10:19:21.509: %RBM-6-SGACLHIT:
ingress_interface='GigabitEthernet1/0' sgacl_name='sgacl2' action='Deny'
protocol='icmp' src-ip='16.16.1.3' src-port='8' dest-ip='17.17.1.2' dest-port='0'
sgt='101' dgt='202' logging_interval_hits='5'
```

Feature History

Feature Name	Release	Feature Information
SGACL Logging	Cisco IOS XE 17.8.1	Initial support on IE3400 and IE3400H



CHAPTER 8

Cisco TrustSec VRF-Aware SGT

VRF-Aware SXP

The Security Group Tag (SGT) Exchange Protocol (SXP) implementation of Virtual Routing and Forwarding (VRF) binds an SXP connection with a specific VRF. It is assumed that the network topology is correctly configured for Layer 2 or Layer 3 VPNs, with all VRFs configured before enabling Cisco TrustSec.

SXP VRF support can be summarized as follows:

- Only one SXP connection can be bound to one VRF.
- Different VRFs may have overlapping SXP peer or source IP addresses.
- IP–SGT mappings learned (added or deleted) in one VRF can be updated only in the same VRF domain. The SXP connection cannot update a mapping bound to a different VRF. If no SXP connection exits for a VRF, IP–SGT mappings for that VRF won't be updated by SXP.
- Multiple address families per VRF is supported. Therefore, one SXP connection in a VRF domain can forward both IPV4 and IPV6 IP-SGT mappings.
- SXP has no limitation on the number of connections and number of IP–SGT mappings per VRF.

IPv6 Support for VRF Aware SGT and SGACL

Beginning with the Cisco IOS XE Bengaluru 17.6.x release, IPv6 is supported for VRF Aware Security Group Tag (SGT) and SG Access Control List (SGACL). The feature extends for IPv6 the same functionality as for IPv4.

IPV6 support for SGT and SGACL feature enables the following features:

- SGT binding
  - Static binding between IPV6 addresses to SGT
  - VLAN-to-SGT bindings
  - Dynamic learning of mapping between IPv6 addresses and SGTs
- Enforcement
  - SGACL enforcement for IPV6 traffic based on UDP or TCP ports
  - SGACL enforcement for IPv6 traffic based on upper layer protocol type

Note

- SGT binding is not supported for link-local address.
- SGACL is not enforced on multicast traffic.

IPV6 SGT and SGACL scale numbers are the same for both IPv4 and IPv6, and most CLI commands are unchanged.

For more information about IPv6 support, see the following sections:

- How IPv4 and IPv6 Share SGT and SGACL Tables, on page 140
- SGT and SGACL Scale Numbers, on page 140

Also see the *Cisco TrustSec Configuration Guide, Cisco IOS XE 17* on Cisco.com.

How IPv4 and IPv6 Share SGT and SGACL Tables

IPv4 and IPv6 share SGT and SGACL tables in FPGA. The following list shows how that sharing is managed:

- If you enable either IPv4 *or* IPv6, it will use the entire table based on configurations.
- If you enable IPv4 *and* IPv6, the tables are shared based on which feature makes the first request.
- The appropriate syslog is generated if the SGT and SGACL tables are exceeded.
- The appropriate syslog is generated if you configure unsupported policies.

SGT and SGACL Scale Numbers

The following table shows scale numbers for IPv4 and IPv6.

Entry Type	Scale Number	Description
Host-SGT	1024	Host-to-SGT bind
Subnet-SGT	64	Network-to-SGT bind
SGT x DGT matrix	21 x 21	SGT-DGT mapping

Entry Type	Scale Number	Description
SGACL policy list size	15	Maximum ACE for each SGACL
Logging counters [31:0] Number of SGT and DGT pairs	32	Maximum pairs

Note By default, logging is enabled for only 32 SGT and DGT pairs. However, you can specify the pairs for which to enable logging. You can disable logging for any pairs among 32 and enable logging for different pairs.

- To see the SGT and DGT pairs for which logging is enabled, use the command **show platform hardware cts cell-logging**.
- To disable logging for specific SGT and DGT pairs, use the command **no platform cts logging**.
- To enable logging on specific SGT and DGT pairs, use the command **platform cts logging**.

The following text shows the options for the **no platform cts logging** command:

```
Device> enable
Device#configure terminal
Device(config)#no platform cts logging ?
all Disable logging for all the cells
default default logging list
from Source Group Tag (SGT) for enabling logging
```

How to Configure Cisco TrustSec VRF-Aware SGT

This section describes how to configure Cisco TrustSec VRF-Aware SGT.

Configuring VRF-to-SGT Mapping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts role-based sgt-map vrf** *vrf-name* {**ip4_netaddress** | *ipv6_netaddress* | **host** {*ip4_address* | *ip6_address*}} **sgt** *sgt_number*
4. **end**

DETAILED STEPS

	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	**cts role-based sgt-map vrf** *vrf-name* {**ip4_netaddress** | *ipv6_netaddress* | **host** {*ip4_address* | *ip6_address*}}] **sgt** *sgt_number*
	Example:
	`Device(config)# cts role-based sgt-map vrf red 10.0.0.3 sgt 23`
	Example:
	`Device(config)# cts role-based sgt-map vrf VRF_1 2405:201:c::f115 sgt 1201`
		Applies the SGT to packets in the specified VRF.
		The IP-SGT binding is entered into the IP-SGT table associated with the specified VRF and the IP protocol version implied by the type of IP address.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco TrustSec VRF-Aware SGT

The following sections show configuration examples of Cisco TrustSec VRF-Aware SGT:

Example: Configuring VRF-to-SGT Mapping

IPv4 example:

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-map vrf VRF_1 22.1.1.1 sgt 1204
Device(config)# end
```

IPv6 example:

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-map vrf VRF_1 2405:201:c::f115 sgt 1201
Device(config)# end
```

Example: Role-based Access List Commands

```
Switch(config)# ipv6 access-list role-based acl-name
Switch(config-rb-acl)#?
Role-based Access List configuration commands:
```

```
<1-2147483647> Sequence Number
default    Set a command to its defaults
deny       Specify packets to reject
exit       Exit from access-list configuration mode
no         Negate a command or set its defaults
permit     Specify packets to forward
remark     Access list entry comment
Switch(config-rb-acl)#
```

ACE Port Ranges

Starting with the Cisco IOS XE Bengaluru 17.6.x release, the TrustSec FPGA module supports the port-range option in policy elements to address some scaling issues.

As part of TrustSec, the FPGA module maintains IP-to-SGT bindings and SGACL policies. Cisco IE3400 switches support 21 x 21 SGT or DGT pairs and 15 policies at each cell, matching on IP protocol field, the L4 source port, and L4 destination port.

However, given the match criteria, you may not be able to scale user access permission. So the TrustSec FPGA module now supports the port range option in each policy element by keeping supported policies to 15 for each cell.

The enhancement enables you to combine multiple rules as shown in the following list:

- Match on IP protocol field
- Match on L4 source start port and end port
- Match on L4 destination start port and end port

Example: Role-based Access List Commands for ACE Port Ranges

You can use the following commands to configureACE port ranges for source and destination ports:

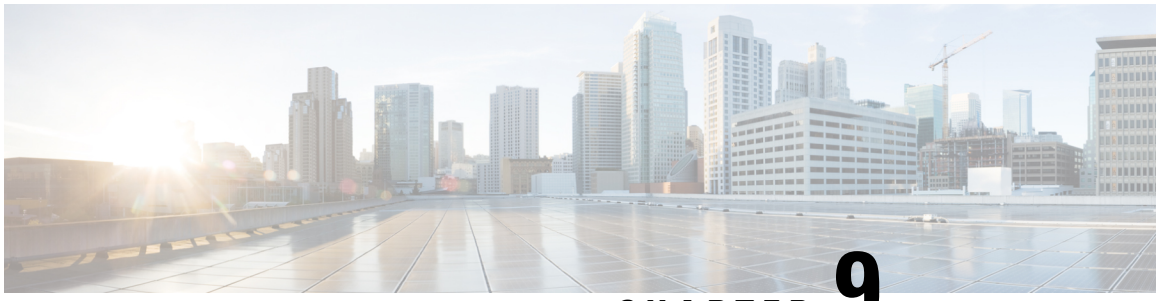
```
Switch(config)# ip access-list role-based rbacl
Switch(config-rb-acl)#10 deny tcp dst range ftp-data telnet
Switch(config-rb-acl)#20 permit tcp dst lt 10
Switch(config-rb-acl)#30 deny tcp dst gt 50
```

Feature History for Cisco TrustSec VRF-Aware SGT

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 Description
Cisco IOS XE Release 17.6.1	Extension of IPv6 support for SGT and SGACL	Enables host-to-SGT mapping and binding and subnet-to-SGT bindings.
	Extension of IPv6 support for SGACL enforcement	Enforces SGACL for IPV6 traffic based on UDP, TCP ports, and upper layer protocol type
	TrustSec FPGA module support for port-range option	Option is supported in policy elements to address scaling issues.
Cisco IOS XE Release 17.5.1	Cisco TrustSec VRF-Aware SGT	The Cisco TrustSec VRF-Aware SGT feature binds a SGT SXP connection with a specific VRF instance.



CHAPTER 9

Cisco Umbrella Integration

Prerequisites for Cisco Umbrella Integration

- Cisco Umbrella subscription license must be available. Go to https://umbrella.cisco.com/products/umbrella-enterprise-security-packages and click **Request a quote** to get the license.
- The device must be set as the default Domain Name System (DNS) server gateway and the domain name server traffic should go through the Cisco device.
- Communication for device registration to the Umbrella server is through HTTPS. This requires a root certificate to be installed on the device. You can download the certificate using this link: https://www.digicert.com/CACerts/DigiCertSHA2SecureServerCA.crt.
- The Cisco Industrial Ethernet switch runs the Cisco IOS XE release 17.2.1 software image or later.
- The Cisco Industrial Ethernet switch must have a DNA Advantage or higher license to enable Umbrella.

The following network requirements must be met:

- The device must be set as the default DNS server gateway and ensure that the Domain Name Server (DNS)traffic goes through the Cisco Industrial Ethernet switch.
- Communication for device registration to the Cisco Umbrella server is via HTTPS. This requires a root certificate to be installed on the router. To download this certificate directly from a link instead of pasting it in, you can find the certificate here:

  https://www.digicert.com/CACerts/DigiCertSHA2SecureServerCA.crt
- For initial registration, the interface configured as “umbrella out” must be able to access api.opendns.com over port 443 in order to complete initial registration.

Restrictions for Cisco Umbrella Integration

- Cisco Umbrella Integration does not work in the following scenarios:
  - If an application or host uses IP address instead of DNS to query domain names.
  - If a client is connected to a web proxy and does not send DNS query to resolve the server address.
  - If DNS queries are generated by a Cisco switching device.
  - If DNS queries are sent over TCP.
  - If DNS queries have record types other than address mapping and text.
- DNSv6 queries are not supported.
- DNS64 and DNS46 extensions are not supported.
- Extended DNS conveys only the IPv4 address of the host, and not the IPv6 address.
- Umbrella configurations on port-channel is not supported
- Umbrella may be configured to use 10G uplink ports as OUT only.
- No dscp markings are entertained for DNS traffic going via Umbrella interfaces. This is applicable to all punted traffic on Umbrella interfaces.
- For umbrella Interfaces, all egress ACL/s rules wouldn’t take affect for DNS traffic. This applies to CPU injected traffic for DNS.
- DNS packet fragmentation is not supported.
- QinQ and Security Group Tag (SGT) packets are not supported.
- When the Cisco Umbrella Integration policy blocks a DNS query, the client is redirected to a Cisco Umbrella block page. HTTPS servers provide these block pages and the IP address range of these block pages is defined by the Cisco Umbrella portal.
- User authentication and identity is not currently supported.
- Umbrella Connector maintains a list of IP address that is known for malicious traffic. When the Cisco Umbrella roaming client detects the destination of packets to those addresses, it forwards those addresses to Cisco Umbrella cloud for further inspection.
- Currently, there is no direct cloud access support.
- Updated resolver IPs do not take effect, DNS traffic is redirected to Cisco Umbrella cloud irrespective of user-configured resolver IPs.
- Network Address Translation (NAT) is not supported on interfaces that has Cisco Umbrella enabled on it.

Information About Cisco Umbrella Integration

The following sections provide details about the Cisco Umbrella Integration feature.

Benefits of Cisco Umbrella Integration

Cisco Umbrella Integration provides security and policy enforcement at the DNS level. It enables the administrator to split the DNS traffic and directly send some of the DNS traffic to a specific DNS server that is located within the enterprise network. This helps the administrator to bypass the Cisco Umbrella Integration.

Cloud-Based Security Service Using Cisco Umbrella Integration

The Cisco Umbrella Integration feature provides cloud-based security service by inspecting the DNS query that is sent to the DNS server through a Cisco device. When a host initiates the traffic and sends a DNS query, the Cisco Umbrella Connector in the device intercepts and inspects the DNS query. The Umbrella Connector is a component in the Cisco device that intercepts DNS traffic and redirects it to the Cisco Umbrella cloud for security inspection and policy application. The Umbrella cloud is a cloud-based security service that inspects the queries received from Umbrella Connectors, and based on the Fully Qualified Domain Name (FQDN), determines if the content provider IP addresses should be provided or not in the response.

If the DNS query is for a local domain, the query is forwarded without changing the DNS packet to the DNS server in the enterprise network. The Cisco Umbrella Resolver inspects the DNS queries that are sent from an external domain. An extended DNS record that includes the device identifier information, organization ID, and client IP address is added to the query and sent to the Umbrella Resolver. Based on all this information, the Umbrella Cloud applies different policies to the DNS query.

The Umbrella Integration cloud might take one of the following actions based on the policies configured on the portal and the reputation of the DNS FQDN:

- Blacklist action: If the FQDN is found to be malicious or blocked by the customized enterprise security policy, the IP address of the Umbrella Cloud's blocked landing page is returned in the DNS response.
- Whitelist action: If the FQDN is found to be nonmalicious, the IP address of the content provider is returned in the DNS response.
- Greylist action: If the FQDN is found to be suspicious, the intelligent proxy unicast IP addresses are returned in the DNS response.

When the DNS response is received, the device forwards the response back to the host. The host extracts the IP address from the response, and sends the HTTP or HTTPS requests to this IP address.

Handling of Traffic by Cisco Umbrella Cloud

With the aid of the Cisco Umbrella Integration feature, HTTP and HTTPs client requests are handled in the following ways:

- If the FQDN in the DNS query is malicious (falls under blacklisted domains), the Umbrella Cloud returns the IP address of the blocked landing page in the DNS response. When the HTTP client sends a request to this IP address, the Umbrella Cloud displays a page that informs the user that the requested page was blocked and the reason for the blocking.
- If the FQDN in the DNS query is nonmalicious (falls under whitelisted domains), the Umbrella Cloud returns the IP address of the content provider. The HTTP client sends the request to this IP address and gets the requested content.
- If the FQDN in the DNS query falls under greylisted domains, the Umbrella DNS resolver returns the unicast IP addresses of the intelligent proxy in the DNS response. All the HTTP traffic from the host to

the grey domain gets proxied through the intelligent proxy and undergoes Uniform Resource Locator (URL) filtering.

Note One potential limitation in using an intelligent proxy unicast IP addresses is the probability of the datacenter going down when a client tries to send the traffic to the intelligent proxy unicast IP address. In this scenario, the client has completed DNS resolution for a domain that falls under the grey-listed domain, and the client's HTTP or HTTPS traffic is sent to one of the obtained intelligent proxy unicast IP addresses. If that datacenter is down, the client has no way of knowing about it.

The Umbrella Connector does not act on the HTTP and HTTPS traffic. The Connector does not redirect any web traffic or alter any HTTP or HTTPS packets.

DNS Packet Encyrption

DNS packets sent from a Cisco device to the Cisco Umbrella Integration server must be encrypted if the extended DNS information in the packet contains information such as user IDs, internal network IP addresses, and so on. When the DNS response is sent back from the DNS server, the device decrypts the packet and forwards it to the host.

Note

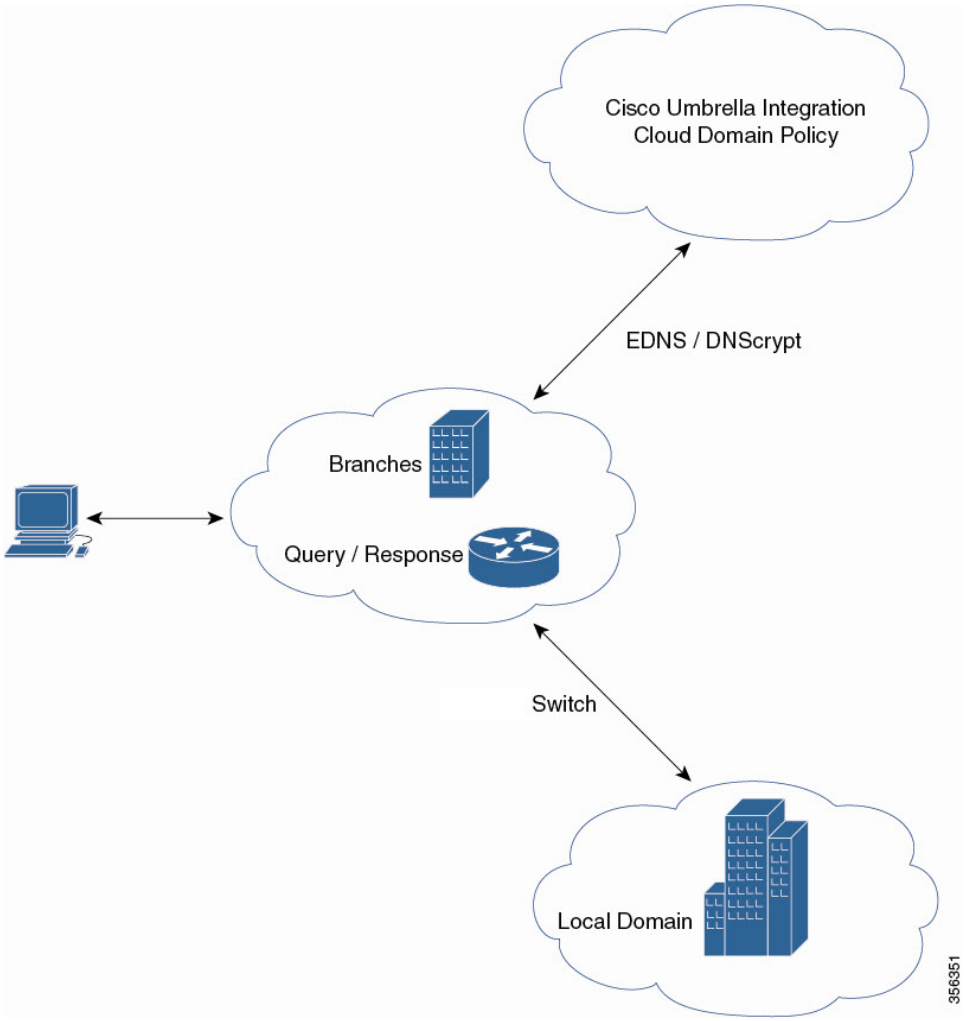
- You can encrypt DNS packets only when the DNScrypt feature is enabled on the Cisco device.
- The IP address of the client is exported to Umbrella Cloud for tracking statistics. We recommend that you do not disable DNScrypt, as the IP would then be sent out unencrypted.

Cisco devices use the following Anycast recursive Cisco Umbrella Integration servers:

- 208.67.222.222
- 208.67.220.220
- 2620:119:53::53
- 2620:119:35::35

The following figure displays the Cisco Umbrella Integration topology.

Figure 9: Cisco Umbrella Integration Topology



DNSCrypt and Public Key

The following subsections provide detailed information about DNScrypt and Public Key.

DNSCrypt

DNSCrypt is an encryption protocol to authenticate communications between a Cisco device and the Cisco Umbrella Integration feature. When the **parameter-map type umbrella** command is configured and the **umbrella out** command is enabled on a WAN interface, DNSCrypt gets triggered, and a certificate is downloaded, validated, and parsed. A shared secret key, which is used to encrypt DNS queries, is then negotiated. For every hour that this certificate is automatically downloaded and verified for an upgrade, a new shared secret key is negotiated to encrypt DNS queries.

When DNSCrypt is used, a DNS request packet's size is more than 512 bytes. Ensure that these packets are allowed through the intermediary devices. Otherwise, the response might not reach the intended recipients.

Public Key

Public key is used to download the DNSCrypt certificate from Umbrella Cloud. This value is preconfigured to B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79, which is the public key of the Cisco Umbrella Integration Anycast servers. If there is a change in the public key, and if you modify the **public-key** command, you have to remove the modified command to restore the default value.

⚠

Caution If you modify the value, the DNSCrypt certificate download might fail.

The **parameter-map type umbrella global** command configures a parameter-map type in umbrella mode. When you configure a device using this command, the DNSCrypt and public key values are autopopulated.

We recommend that you change the **parameter-map type umbrella global** parameters only when you perform certain tests in the lab. If you modify these parameters, it can affect the normal functioning of the device.

How to Configure Cisco Umbrella Integration

The following sections provide information about the various tasks that comprise Cisco Umbrella integration.

Configuring the Umbrella Connector

Before you begin

Get the application programming interface (API) token from the Cisco Umbrella registration server.

Have the root certificate establish the HTTPS connection with the Cisco Umbrella registration server. Import the root certificate of DigiCert into the device using the **crypto pki trustpool import terminal** command in global configuration mode.

There are two methods of importing the certificate.

1. Importing from a URL
2. Importing directly in a Terminal

To Import from a URL, Issue the command and allow Industrial Ethernet switch to fetch the cert:

crypto pki trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b

To import from a terminal, perform the following:

The following is the root certificate of DigiCert:

```
-----BEGIN CERTIFICATE-----
MIIElDCCA3ygAwIBAgIQAf2j627KdciIQ4tyS8+8kTANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0xMzAzMDgxMjAwMDBaFw0yMzAzMDgxMjAwMDBaME0xCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxJzAlBgNVBAMTHkRpZ2lDZXJ0IFNIQTIg
U2VjdXJlIFNlcnZlciBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ANyuWJBNwcQwFZA1W248ghX1LFy949v/cUP6ZCWA1O4Yok3wZtAKc24RmDYXZK83
nf36QYSvx6+M/hpzTc8zl5CilodTgyu5pnVILR1WN3vaMTIa16yrBvSqXUu3R0bd
KpPDkC55gIDvEwRqFDu1m5K+wgdlTvza/P96rtxcflUxDOg5B6TXvi/TC2rSsd9f
```

```
/ld0Uzs1gN2ujkSYs58O09rg1/RrKatEp0tYhG2SS4HD2nOLEpdIkARFdRrdNzGX
kujNVA075ME/OV4uuPNcfhCOhkEAjUVmR7ChZc6gqikJTvOX6+guqw9ypzAO+sf0
/RR3w6RbKFfCs/mC/bdFWJsCAwEAAaOCAVowggFWMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYYY
aHR0cDovL29jc3AuZGlnaWNlcnQuY29tMHsGA1UdHwR0MHIwN6A1oDOGMWh0dHA6
Ly9jcmwzLmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RDQS5jcmwwN6A1
oDOGMWh0dHA6Ly9jcmw0LmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RD
QS5jcmwwPQYDVR0gBDYwNDAyBgRVHSAAMCowKAYIKwYBBQUHAgEWHGh0dHBzOi8v
d3d3LmRpZ2ljZXJ0LmNvbS9DUFMwHQYDVR0OBBYEFA+AYRyCMWHVLyjnjUY4tCzh
xtniMB8GA1UdIwQYMBaAFAPeUDVW0Uy7ZvCj4hsbw5eyPdFVMA0GCSqGSIb3DQEB
CwUAA4IBAQAjPt9L0jFCpbZ+QlwaRMxp0Wi0XUvgBCFsS+JtzLHgl4+mUwnNqipl
5TlPHoOlblyYoiQm5vuh7ZPHLgLGTUq/sELfeNqzqPlt/yGFUzZgTHbO7Djc1lGA
8MXW5dRNJ2Srm8c+cftIl7gzbckTB+6WohsYFfZcTEDts8Ls/3HB40f/1LkAtDdC
2iDJ6m6K7hQGrn2iWZiIqBtvLfTyyRRfJs8sjX7tN8Cp1Tm5gr8ZDOo0rwAhaPit
c+LJMto4JQtV05od8GiG7S5BNO98pVAdvzr508EIDObtHopYJeS4d60tbvVS3bR0
j6tJLp07kzQoH3jOlOrHvdPJbRzeXDLz
-----END CERTIFICATE-----
```

Verify that the privacy-enhanced mail (PEM) import is successful. A confirmation message is displayed after importing the certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type umbrella global**
4. **dnscrypt**
5. **token** *value*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type umbrella global** **Example:** `Device(config)# parameter-map type umbrella global`	Configures the parameter map type as umbrella mode, and enters parameter-map type inspect configuration mode.
Step 4	**dnscrypt** **Example:**	Enables DNS packet encryption on the device.

	Command or Action	Purpose
	`Device(config-profile)# dnscrypt`	
Step 5	**token** *value* **Example:** `Device(config-profile)# token AABBA59A0BDE1485C912AFE472952641001EEECC`	Specifies the API token issued by the Cisco Umbrella registration server.
Step 6	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Registering the Cisco Umbrella Tag

Before you begin

- Configure the Umbrella Connector.
- Configure the **umbrella out** command before configuring the **umbrella in** command. Registration is successful only when port 443 is in Open state and allows the traffic to pass through the existing firewall.
- After you configure the **umbrella in** command with a tag, the device initiates the registration process by resolving api.opendns.com. Configure a name server by using the **ip name-server** command, and a domain lookup by using the **ip domain-lookup** command configured on the device to successfully resolve the FQDN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **umbrella out**
5. **exit**
6. **interface** *interface-type interface-number*
7. **umbrella in** *tag-name*
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface gigabitEthernet 1/1`	Specifies the WAN interface, and enters interface configuration mode.
Step 4	**umbrella out** **Example:** `Device(config-if)# umbrella out`	Configures Umbrella Connector on the interface to connect to the Umbrella Cloud servers.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode, and enters global configuration mode.
Step 6	**interface** *interface-type interface-number* **Example:** `Device(config)# interface gigabitEthernet 1/2`	Specifies the LAN interface, and enters interface configuration mode.
Step 7	**umbrella in** *tag-name* **Example:** `Device(config-if)# umbrella in mydevice_tag`	Configures the Umbrella Connector on the interface that is connected to the client. • The length of the Umbrella tag should not exceed 49 characters. • After you configure the **umbrella in** command with a tag, the device registers the tag to the Cisco Umbrella Integration server.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Cisco Device as a Pass-through Server

You can identify the traffic that is to be bypassed by using domain names. You can define these domains in the form of regular expressions on a Cisco device. If the DNS query that is intercepted by the device matches one of the configured regular expressions, the query is bypassed to the specified DNS server without being redirected to the Umbrella Cloud.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *expression*
5. **exit**
6. **parameter-map type umbrella global**
7. **token** *value*
8. **local-domain** *regex_param_map_name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Device(config)# parameter-map type regex dns_bypass`	Configures a parameter-map type to match the specified traffic pattern, and enters parameter-map type inspect configuration mode.

Step 4	**pattern** *expression*
	Example:
	`Device(config-profile)# pattern www.cisco.com`
	`Device(config-profile)# pattern .*example.cisco.*`
		Configures a local domain or URL that is used to bypass the Umbrella Cloud.

Step 5	**exit** **Example:**	Exits parameter-map type inspect configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-profile)# exit`	
Step 6	**parameter-map type umbrella global** **Example:** `Device(config)# parameter-map type umbrella global`	Configures the parameter map type as umbrella mode, and enters parameter-map type inspect configuration mode.
Step 7	**token** *value* **Example:** `Device(config-profile)# token AADDD5FF6E510B28921A20C9B98EEEFF`	Specifies the API token issued by the Cisco Umbrella registration server.
Step 8	**local-domain** *regex_param_map_name* **Example:** `Device(config-profile)# local-domain dns_bypass`	Attaches the regular expression parameter map with the Umbrella global configuration.
Step 9	**end** **Example:** `Device(config-profile)# end`	Exits parameter-map type inspect configuration mode and returns to privileged EXEC mode.

Verifying the Cisco Umbrella Integration Configuration

Use the following commands in any order to view the Cisco Umbrella Integration configuration.

The following example shows a sample output of the **show umbrella config** command:

```
Device# show umbrella config
Umbrella Configuration
=======================
Token: EB74330C50767B6A63770EA6C3408DCF00282D8E
API-KEY: NONE
OrganizationID: 2633102
Local Domain Regex parameter-map name: NONE
DNSCrypt: Enabled
Public-key: B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79
UDP Timeout: 5 seconds
Resolver address:
1. 208.67.220.220
2. 208.67.222.222
3. 2620:119:53::53
4. 2620:119:35::35
Umbrella Interface Config:
Number of interfaces with "umbrella out" config: 1
1. GigabitEthernet1/4
Mode : OUT
```

```
VRF : global(Id: 0)
Number of interfaces with "umbrella in" config: 2
1. GigabitEthernet1/9
Mode : IN
DCA : Disabled
Tag : IE_uniquetag
Device-id : 010a424c1597fe09
VRF : global(Id: 0)
2. GigabitEthernet2/3
Mode : IN
DCA : Disabled
Tag : IE_tag_2
Device-id : 010adaf012a36ad6
VRF : global(Id: 0)
Configured Umbrella Parameter-maps:
1. global
```

The following example shows a sample output of the **show umbrella deviceid** command:

```
Device# show umbrella deviceid

Device registration details
Interface Name Tag Status Device-id
GigabitEthernet1/9 IE_uniquetag 200 SUCCESS 010a424c1597fe09
GigabitEthernet2/3 IE_tag_2 200 SUCCESS 010adaf012a36ad6
```

The following example shows a sample output of the **show umbrella dnscrypt** command:

```
Device#show umbrella dnscrypt
DNSCrypt: Enabled
Public-key: B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79
Certificate Update Status:
Last Successfull Attempt: 20:01:18 IST Dec 17 2019
Certificate Details:
Certificate Magic : DNSC
Major Version : 0x0001
Minor Version : 0x0000
Query Magic : 0x7163373861576F6F
Serial Number : 1574811744
Start Time : 1574811744 (05:12:24 IST Nov 27 2019)
End Time : 1606347744 (05:12:24 IST Nov 26 2020)
Server Public Key :
88B4:E44B:35E9:64B4:90BD:DABA:E825:A24B:0415:A08B:E19D:7DDB:87A3:3CD7:7EDF:8E2F
Client Secret Key Hash:
0FB9:520E:5228:FB2C:D521:1E9E:2ACB:AC3D:B520:A795:F54C:C608:604B:A410:17F1:1284
Client Public key :
E42F:507E:F052:72DD:1BC8:4857:2AE0:2F9F:ED87:1687:AAE4:095D:D933:48F0:5D60:3662
NM key Hash : EDC3:25DD:4D21:103E:7E49:1EFA:75ED:4D6F:A450:107D:C6E8:1C41:9CF7:4039:FA89:2CED
```

The following example shows a sample output of the **show umbrella deviceid detailed** command:

```
Device# show umbrella deviceid detailed

Device registration details
1.GigabitEthernet1/9
Tag : IE_uniquetag
Device-id : 010a424c1597fe09
Description : Device Id recieved successfully
WAN interface : GigabitEthernet1/4
WAN VRF used : global(Id: 0)
2.GigabitEthernet2/3
Tag : IE_tag_2
```

```
Device-id : 010adaf012a36ad6
Description : Device Id recieved successfully
WAN interface : GigabitEthernet1/4
WAN VRF used : global(Id: 0)
```

The following is a sample output of the **show platform software dns-umbrella statistics** command. The command output displays traffic-related information such as the number of queries sent, number of responses received, and so on.

```
Device# show platform software dns-umbrella statistics

======================================
Umbrella Statistics
======================================
Total Packets : 7848
DNSCrypt queries : 3940
DNSCrypt responses : 0
DNS queries : 0
DNS bypassed queries(Regex) : 0
DNS responses(Umbrella) : 0
DNS responses(Other) : 3906
Aged queries : 34
Dropped pkts : 0
```

Troubleshooting Cisco Umbrella Integration

You can troubleshoot issues related to the Cisco Umbrella Integration feature configuration by using the following commands.

Table 14: debug Commands for Cisco Umbrella Integration Feature

Command	Purpose
debug umbrella config	Enables Umbrella configuration debugging.
debug umbrella device-registration	Enables Umbrella device registration debugging.
debug umbrella dnscrypt	Enables Umbrella DNSCrypt encryption debugging.

From the command prompt of a Windows machine, or the terminal window or shell of a Linux machine, run the **nslookup -type=txt debug.opendns.com** command. The IP address that you specify with the **nslookup -type=txt debug.opendns.com** command must be the IP address of the DNS server.

```
nslookup -type=txt debug.opendns.com 10.0.0.1
Server: 10.0.0.1
Address: 10.0.0.1#53
Non-authoritative answer:
debug.opendns.com text = "server r6.xx"
debug.opendns.com text = "device 010A826AAABB6C3D"
debug.opendns.com text = "organization id 1892929"
debug.opendns.com text = "remoteip 10.0.1.1"
debug.opendns.com text = "flags 436 0 6040 39FF000000000000000"
debug.opendns.com text = "originid 119211936"
debug.opendns.com text = "orgid 1892929"
debug.opendns.com text = "orgflags 3"
debug.opendns.com text = "actype 0"
debug.opendns.com text = "bundle 365396"
```

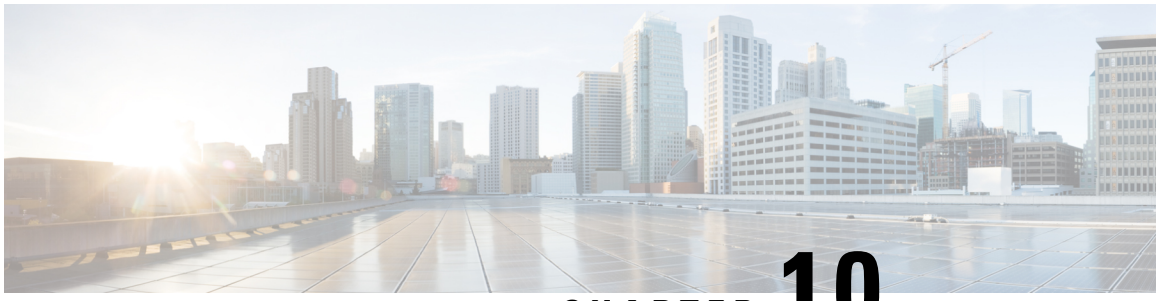
```
debug.opendns.com text = "source 10.1.1.1:36914"
debug.opendns.com text = "dnscrypt enabled (713156774457306E)"
```

Feature Information for Cisco Umbrella Integration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 15: Feature Information for Cisco Umbrella Integration

Feature Name	Releases	Feature Information
Cisco Umbrella Integration	Cisco IOS XE Amsterdam 17.2.1	The Cisco Umbrella Integration feature enables cloud-based security service by inspecting the DNS query that is sent to any DNS server through Cisco devices. The security administrator configures policies on the Cisco Umbrella Cloud to either allow or deny traffic towards the FQDN.



CHAPTER 10

Configuring Network Edge Access Topology (NEAT)

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 IE3x00 and ESS3300 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.

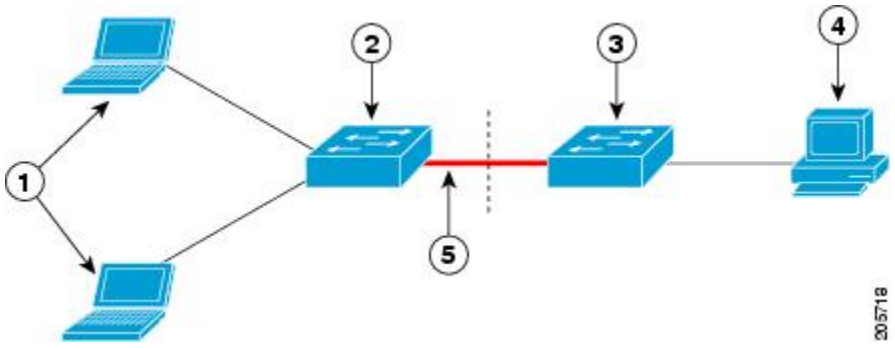
You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. Multihost mode is not supported on the authenticator switch interface.

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 10: 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.

Configuring 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

	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	**cisp enable** **Example:** `Device(config)# cisp enable`	Enables CISP.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the port mode to **access**.
Step 6	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	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.

Configuring 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

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

	Command or Action	Purpose
	`Device> enable`	
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** *profile-name* **Example:** `Device(config)# eap profile CISP`	Creates an Extensible Authentication Protocol (EAP) profile and enters EAP profile configuration mode.
Step 5	**method** *type* **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** *profile* **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** *suppswitch* **Example:** `Device(config)# username suppswitch`	Creates a username.
Step 9	**password** *password* **Example:** `Device(config)# password myswitch`	Creates a password for the new username.

	Command or Action	Purpose
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** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/1`	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** *profile-name* **Example:** `Device(config-if)# dot1x credentials test`	Attaches the 802.1x credentials profile to the interface.
Step 16	**dot1x supplicant eap profile** *profile-name* **Example:** `Device(config-if)# dot1x supplicant eap profile cisp`	Assigns the EAP-TLS profile to the 802.1X interface.
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 1/1 is configured as Authenticator, and GigabitEthernet 1/2 is configured as Supplicant.

```
Auth# show cisp interface Gi1/2

CISP Status for interface Gi1/2
 -------------------------------------
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 Gi1/10
6c03.09e7.3947 1 Gi1/10
6c03.09e7.3954 11 Gi1/10
6c03.09e7.4485 1 Gi1/10
 9077.ee4a.8567 1 Gi1/10
e41f.7ba1.bbd4 1 Gi1/10

Supplicant Client Table:
------------------------
MAC Address VLAN Interface
---------------------------------
9077.ee4a.856b 11 Vl11
9077.ee4a.8572 1 Ap1/1
e41f.7bc7.2f03 1 Gi1/9

Auth# show cisp summary

CISP is running on the following interface(s):
---------------------------------------------
Gi1/2 (Authenticator)

Supp# show cisp summary

CISP is running on the following interface(s):
---------------------------------------------
Gi1/1 (Supplicant)

Auth# show cisp registrations

Interface(s) with CISP registered user(s):
```

```
-----------------------------------------
Gi1/2
Auth Mgr (Authenticator)

Supp# show cisp registration

Interface(s) with CISP registered user(s):
-----------------------------------------
Gi1/1
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

Configuration Example

Following is an example of Client Information Signalling Protocol (CISP) and Network Edge Access Topology (NEAT) configuration on the authenticator switch.

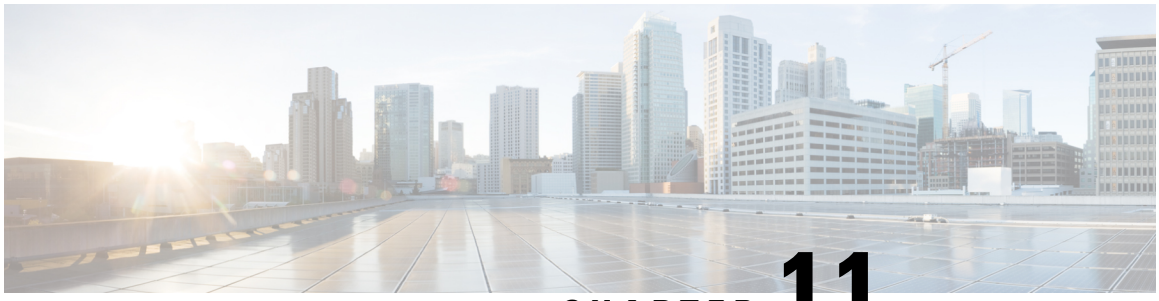
```
conf t
aaa new-model
cisp enable
radius server RADIUS_CWA
address ipv4 <ISE-IP> auth-port 1645 acct-port 1646
key <ISE KEY>
exit
aaa group server radius ISE
server name RADIUS_CWA
exit
aaa authentication dot1x default group radius
aaa authorization exec default group radius
aaa authorization network default group radius
aaa server radius dynamic-author
client <ISE-IP> server-key cisco123
dot1x system-auth-control
policy-map type control subscriber Policy_dot1x
event session-started match-all
10 class always do-until-failure
10 authenticate using dot1x
exit

interface <interface name>
switchport mode access
access-session closed
access-session port-control auto
dot1x pae authenticator
spanning-tree portfast
service-policy type control subscriber Policy_dot1x
exit
```

Following is an example of CISP and NEAT configuration on the supplicant switch.

```
conf t
cisp enable
eap profile CISP
method md5
exit
dot1x system-auth-control
dot1x supplicant controlled transient
dot1x credentials SWITCH
username <user configured in ISE>
password 0 <Password configured in ISE>
exit
interface <interface name>
switchport mode trunk
dot1x pae supplicant
dot1x credentials SWITCH
dot1x supplicant eap profile CISP
spanning-tree portfast trunk
exit
```

Feature History

Feature Name	Release	Feature Information
Network Edge Access Topology (NEAT)	Cisco IOS XE 17.8.1	Initial support on IE3x00



CHAPTER 11

Configuring Web-Based Authentication

Information About Web-Based Authentication

Web-Based Authentication Overview

Use the web-based authentication feature, known as web authentication proxy, to authenticate end users on host systems that do not run the IEEE 802.1x supplicant.

When you initiate an HTTP session, web-based authentication intercepts ingress HTTP packets from the host and sends an HTML login page to the users. The users enter their credentials, which the web-based authentication feature sends to the authentication, authorization, and accounting (AAA) server for authentication.

If authentication succeeds, web-based authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.

If authentication fails, web-based authentication forwards a Login-Fail HTML page to the user, prompting the user to retry the login. If the user exceeds the maximum number of attempts, web-based authentication forwards a Login-Expired HTML page to the host, and the user is placed on a watch list for a waiting period.

Note HTTPS traffic interception for central web authentication redirect is not supported.

Note You should use global parameter-map (for method-type, custom, and redirect) only for using the same web authentication methods like consent, web consent, and webauth, for all the clients and SSIDs. This ensures that all the clients have the same web-authentication method.

If the requirement is to use Consent for one SSID and Web-authentication for another SSID, then you should use two named parameter-maps. You should configure Consent in first parameter-map and configure webauth in second parameter-map.

Note The traceback that you receive when webauth client tries to do authentication does not have any performance or behavioral impact. It happens rarely when the context for which FFM replied back to EPM for ACL application is already dequeued (possibly due to timer expiry) and the session becomes 'unauthorized'.

Based on where the web pages are hosted, the local web authentication can be categorozied as follows:

- *Internal*—The internal default HTML pages (Login, Success, Fail, and Expire) in the controller are used during the local web authentication.
- *Customized*—The customized web pages (Login, Success, Fail, and Expire) are downloaded onto the controller and used during the local web authentication.
- *External*—The customized web pages are hosted on the external web server instead of using the in-built or custom web pages.

Note External web authentication is not supported in this release.

Based on the various web authentication pages, the types of web authentication are as follows:

- *Webauth*—This is a basic web authentication. Herein, the controller presents a policy page with the user name and password. You need to enter the correct credentials to access the network.
- *Consent or web-passthrough*—Herein, the controller presents a policy page with the Accept button. You need to click the Accept button to access the network.
- *Webconsent*—This is a combination of webauth and consent web authentication types. Herein, the controller presents a policy page with Accept button along with user name or password. You need to enter the correct credentials and click the Accept button to access the network.

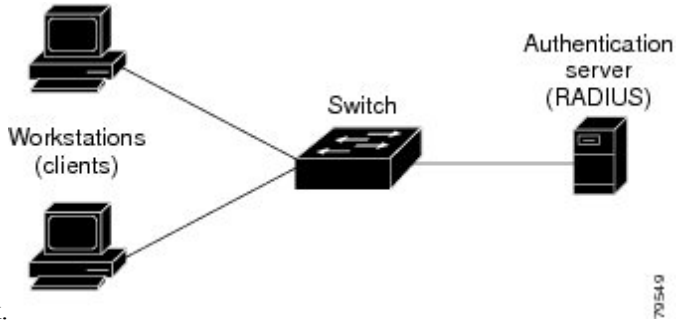
Device Roles

With web-based authentication, the devices in the network have these specific roles:

- *Client*—The device (workstation) that requests access to the LAN and the services and responds to requests from the switch. The workstation must be running an HTML browser with Java Script enabled.
- *Authentication server*—Authenticates the client. The authentication server validates the identity of the client and notifies the switch that the client is authorized to access the LAN and the switch services or that the client is denied.
- *Switch*—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client.

Figure 11: Web-Based Authentication Device Roles

This figure shows the roles of these devices in a



network.

Host Detection

The switch maintains an IP device tracking table to store information about detected hosts.

For Layer 2 interfaces, web-based authentication detects IP hosts by using these mechanisms:

- ARP based trigger—ARP redirect ACL allows web-based authentication to detect hosts with a static IP address or a dynamic IP address.
- Dynamic ARP inspection
- DHCP snooping—Web-based authentication is notified when the switch creates a DHCP-binding entry for the host.

Session Creation

When web-based authentication detects a new host, it creates a session as follows:

- Reviews the exception list.

  If the host IP is included in the exception list, the policy from the exception list entry is applied, and the session is established.

- Reviews for authorization bypass

  If the host IP is not on the exception list, web-based authentication sends a nonresponsive-host (NRH) request to the server.

  If the server response is access accepted, authorization is bypassed for this host. The session is established.

- Sets up the HTTP intercept ACL

  If the server response to the NRH request is access rejected, the HTTP intercept ACL is activated, and the session waits for HTTP traffic from the host.

Authentication Process

When you enable web-based authentication, these events occur:

- The user initiates an HTTP session.

- The HTTP traffic is intercepted, and authorization is initiated. The switch sends the login page to the user. The user enters a username and password, and the switch sends the entries to the authentication server.
- If the authentication succeeds, the switch downloads and activates the user's access policy from the authentication server. The login success page is sent to the user.
- If the authentication fails, the switch sends the login fail page. The user retries the login. If the maximum number of attempts fails, the switch sends the login expired page, and the host is placed in a watch list. After the watch list times out, the user can retry the authentication process.
- If the authentication server does not respond to the switch, and if an AAA fail policy is configured, the switch applies the failure access policy to the host. The login success page is sent to the user.
- The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface, or when the host does not send any traffic within the idle timeout on a Layer 3 interface.
- The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface.
- The feature applies the downloaded timeout or the locally configured session timeout.
- If the terminate action is RADIUS, the feature sends a nonresponsive host (NRH) request to the server. The terminate action is included in the response from the server.
- If the terminate action is default, the session is dismantled, and the applied policy is removed.

Local Web Authentication Banner

With Web Authentication, you can create a default and customized web-browser banners that appears when you log in to a switch.

The banner appears on both the login page and the authentication-result pop-up pages. The default banner messages are as follows:

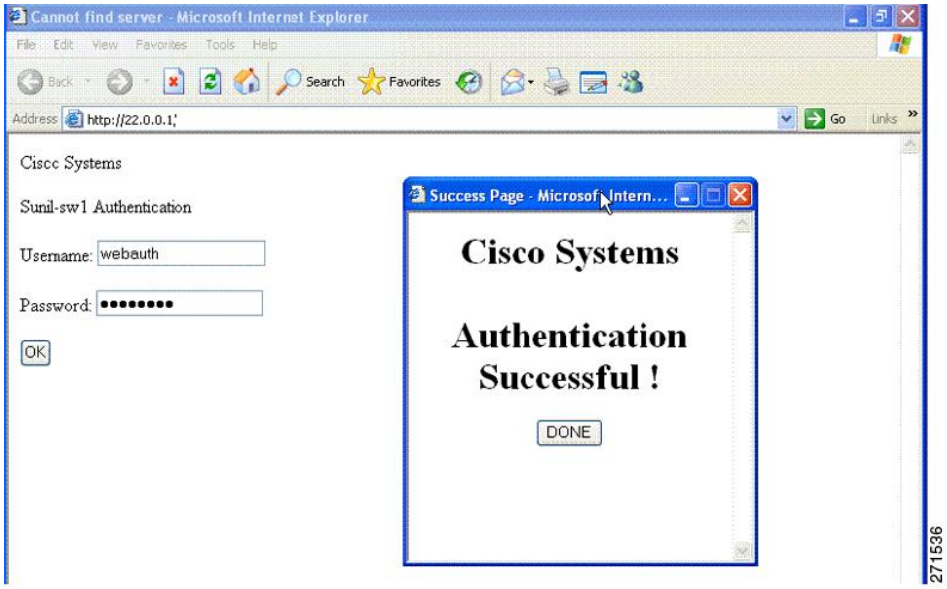
- *Authentication Successful*
- *Authentication Failed*
- *Authentication Expired*

The Local Web Authentication Banner can be configured in as follows:

- Legacy mode—Use the **ip admission auth-proxy-banner http** global configuration command.
- New-style mode—Use the **parameter-map type webauth global banner** global configuration command.

The default banner *Cisco Systems* and *Switch host-name Authentication* appear on the Login Page. *Cisco Systems* appears on the authentication result pop-up page.

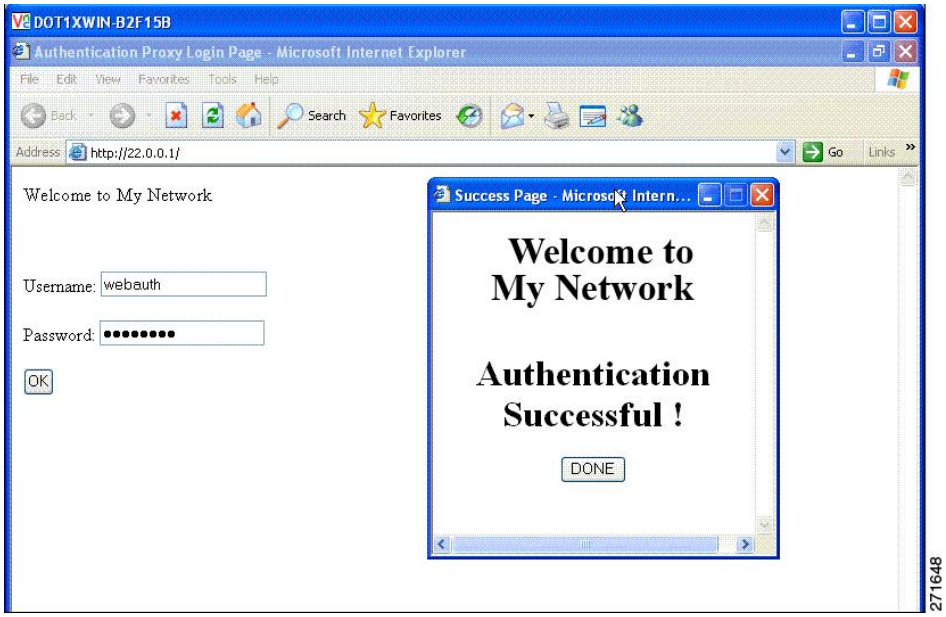
Figure 12: Authentication Successful Banner



The banner can be customized as follows:

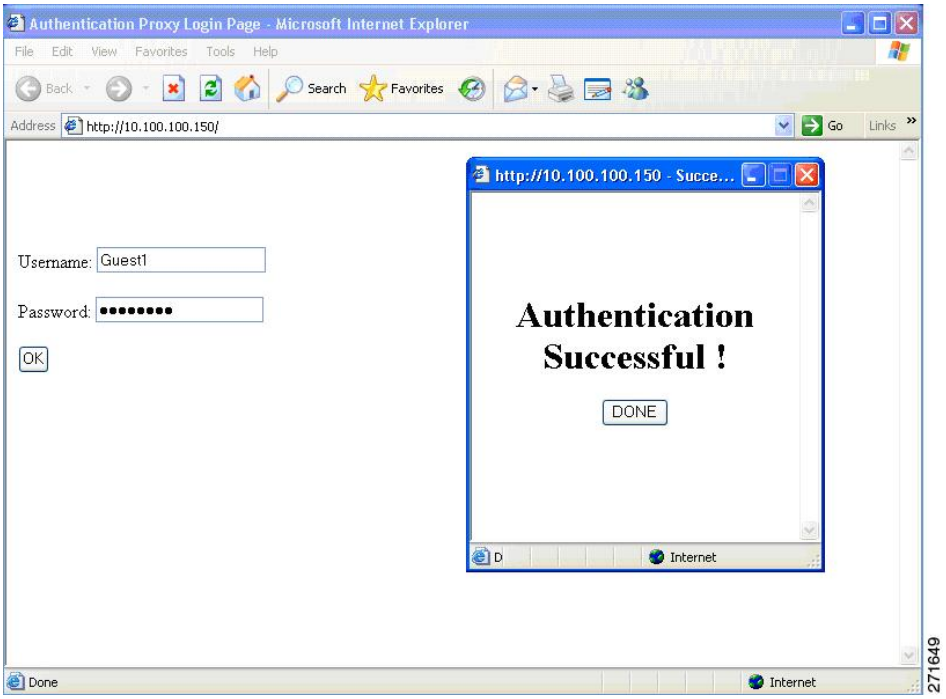
- Add a message, such as switch, router, or company name to the banner:
  - Legacy mode—Use the **ip admission auth-proxy-banner http** *banner-text*global configuration command.
  - New-style mode—Use the **parameter-map type webauth global banner** global configuration command.
- Add a logo or text file to the banner:
  - Legacy mode—Use the **ip admission auth-proxy-banner http** *file-path* global configuration command.
  - New-style mode—Use the **parameter-map type webauth global banner** global configuration command.

Figure 13: Customized Web Banner



If you do not enable a banner, only the username and password dialog boxes appear in the web authentication login screen, and no banner appears when you log into the switch.

Figure 14: Login Screen With No Banner



Web Authentication Customizable Web Pages

During the web-based authentication process, the switch internal HTTP server hosts four HTML pages to deliver to an authenticating client. The server uses these pages to notify you of these four-authentication process states:

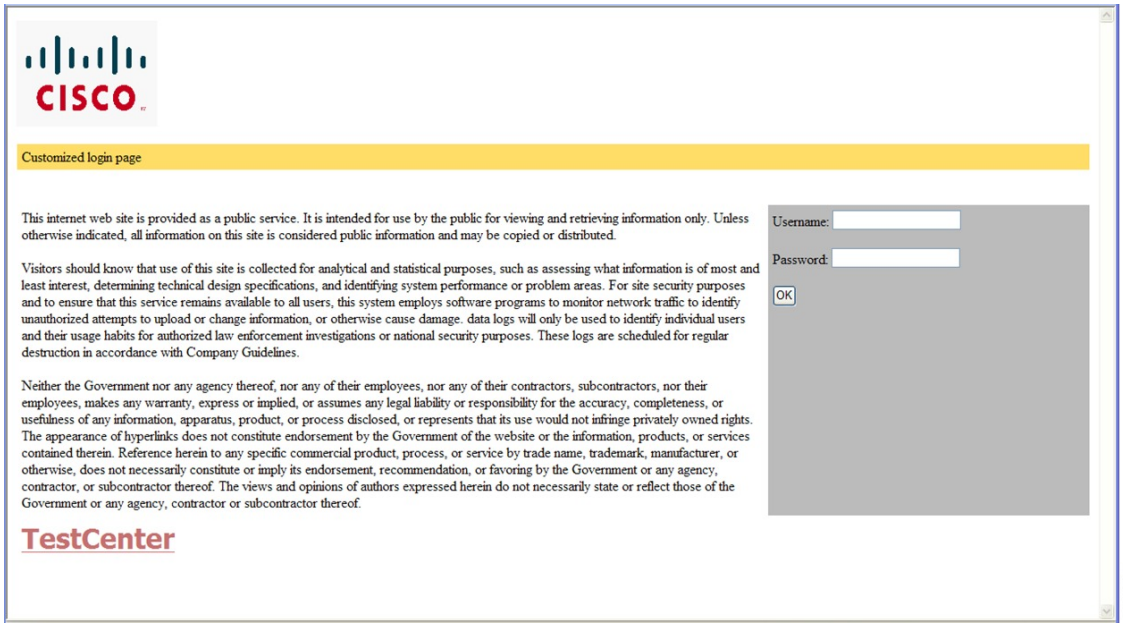
- Login—Your credentials are requested.
- Success—The login was successful.
- Fail—The login failed.
- Expire—The login session has expired because of excessive login failures.

Guidelines

- You can substitute your own HTML pages for the default internal HTML pages.
- You can use a logo or specify text in the *login*, *success*, *failure*, and *expire* web pages.
- On the banner page, you can specify text in the login page.
- The pages are in HTML.
- You must include an HTML redirect command in the success page to access a specific URL.
- The URL string must be a valid URL (for example, http://www.cisco.com). An incomplete URL might cause *page not found* or similar errors on a web browser.
- If you configure web pages for HTTP authentication, they must include the appropriate HTML commands (for example, to set the page time out, to set a hidden password, or to confirm that the same page is not submitted twice).
- The CLI command to redirect users to a specific URL is not available when the configured login form is enabled. The administrator should ensure that the redirection is configured in the web page.
- If the CLI command redirecting users to specific URL after authentication occurs is entered and then the command configuring web pages is entered, the CLI command redirecting users to a specific URL does not take effect.
- Configured web pages can be copied to the switch boot flash or flash.
- The login page can be on one flash, and the success and failure pages can be another flash (for example, the flash on the stack master or a member).
- You must configure all four pages.
- The banner page has no effect if it is configured with the web page.
- All of the logo files (image, flash, audio, video, and so on) that are stored in the system directory (for example, flash, disk0, or disk) and that must be displayed on the login page must use *web_auth_<filename>* as the file name.
- The configured authentication proxy feature supports both HTTP and SSL.

You can substitute your HTML pages for the default internal HTML pages. You can also specify a URL to which users are redirected after authentication occurs, which replaces the internal Success page.

Figure 15: Customizable Authentication Page



Authentication Proxy Web Page Guidelines

When configuring customized authentication proxy web pages, follow these guidelines:

- To enable the custom web pages feature, specify all four custom HTML files. If you specify fewer than four files, the internal default HTML pages are used.
- The four custom HTML files must be present on the flash memory of the switch. The maximum size of each HTML file is 8 KB.
- Any images on the custom pages must be on an accessible HTTP server. Configure an intercept ACL within the admission rule.
- Any external link from a custom page requires configuration of an intercept ACL within the admission rule.
- To access a valid DNS server, any name resolution required for external links or images requires configuration of an intercept ACL within the admission rule.
- If the custom web pages feature is enabled, a configured auth-proxy-banner is not used.
- If the custom web pages feature is enabled, the redirection URL for successful login feature is not available.
- To remove the specification of a custom file, use the **no** form of the command.

Because the custom login page is a public web form, consider these guidelines for the page:

- The login form must accept user entries for the username and password and must show them as **uname** and **pwd**.
- The custom login page should follow best practices for a web form, such as page timeout, hidden password, and prevention of redundant submissions.

Web-based Authentication Interactions with Other Features

Port Security

You can configure web-based authentication and port security on the same port. Web-based authentication authenticates the port, and port security manages network access for all MAC addresses, including that of the client. You can then limit the number or group of clients that can access the network through the port.

LAN Port IP

You can configure LAN port IP (LPIP) and Layer 2 web-based authentication on the same port. The host is authenticated by using web-based authentication first, followed by LPIP posture validation. The LPIP host policy overrides the web-based authentication host policy.

If the web-based authentication idle timer expires, the NAC policy is removed. The host is authenticated, and posture is validated again.

Gateway IP

You cannot configure Gateway IP (GWIP) on a Layer 3 VLAN interface if web-based authentication is configured on any of the switch ports in the VLAN.

You can configure web-based authentication on the same Layer 3 interface as Gateway IP. The host policies for both features are applied in software. The GWIP policy overrides the web-based authentication host policy.

ACLs

If you configure a VLAN ACL or a Cisco IOS ACL on an interface, the ACL is applied to the host traffic only after the web-based authentication host policy is applied.

For Layer 2 web-based authentication, it is more secure, though not required, to configure a port ACL (PACL) as the default access policy for ingress traffic from hosts connected to the port. After authentication, the web-based authentication host policy overrides the PACL. The Policy ACL is applied to the session even if there is no ACL configured on the port.

You cannot configure a MAC ACL and web-based authentication on the same interface.

You cannot configure web-based authentication on a port whose access VLAN is configured for VACL capture.

Context-Based Access Control

Web-based authentication cannot be configured on a Layer 2 port if context-based access control (CBAC) is configured on the Layer 3 VLAN interface of the port VLAN.

EtherChannel

You can configure web-based authentication on a Layer 2 EtherChannel interface. The web-based authentication configuration applies to all member channels.

How to Configure Web-Based Authentication

Default Web-Based Authentication Configuration

The following table shows the default web-based authentication configuration.

Table 16: Default Web-based Authentication Configuration

Feature	Default Setting
AAA	Disabled
RADIUS server • IP address • UDP authentication port • Key	• None specified • None specified
Default value of inactivity timeout	3600 seconds
Inactivity timeout	Enabled

Web-Based Authentication Configuration Guidelines and Restrictions

- Web-based authentication is an ingress-only feature.
- You can configure web-based authentication only on access ports. Web-based authentication is not supported on trunk ports, EtherChannel member ports, or dynamic trunk ports.
- External web authentication, where the switch redirects a client to a particular host or web server for displaying login message, is not supported.
- You cannot authenticate hosts on Layer 2 interfaces with static ARP cache assignment. These hosts are not detected by the web-based authentication feature because they do not send ARP messages.
- By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.
- You must enable SISF-Based device tracking to use web-based authentication. By default, SISF-Based device tracking is disabled on a switch.
- You must configure at least one IP address to run the switch HTTP server. You must also configure routes to reach each host IP address. The HTTP server sends the HTTP login page to the host.
- Hosts that are more than one hop away might experience traffic disruption if an STP topology change results in the host traffic arriving on a different port. This occurs because the ARP and DHCP updates might not be sent after a Layer 2 (STP) topology change.
- Web-based authentication does not support VLAN assignment as a downloadable-host policy.
- IPv6 Web-based authentication is not supported.

- Web-based authentication and Network Edge Access Topology (NEAT) are mutually exclusive. You cannot use web-based authentication when NEAT is enabled on an interface, and you cannot use NEAT when web-based authentication is running on an interface.
- Virtual IP configuration is not supported. As a result of this limitation, the Logout Page is not supported.
- Identify the following RADIUS security server settings that will be used while configuring switch-to-RADIUS-server communication:
  - Host name
  - Host IP address
  - Host name and specific UDP port numbers
  - IP address and specific UDP port numbers

  The combination of the IP address and UDP port number creates a unique identifier, that 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, authentication) the second host entry that is configured functions as the failover backup to the first one. The RADIUS host entries are chosen in the order that they were configured.
- When you configure the RADIUS server parameters:
  - Specify the **key** *string* on a separate command line.
  - For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
  - When you specify the **key** *string*, use spaces within and at the end of the key. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
  - You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using with the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, radius-server transmit, and the radius-server key global configuration commands.

Note You need to configure some settings on the RADIUS server, including: the switch IP address, the key string to be shared by both the server and the switch, and the downloadable ACL (DACL). For more information, see the RADIUS server documentation.

Note DACL download fails in IE3xxx platforms when the number of ACEs are 20 or more. This limitation causes fragmented RADIUS packets to not be processed, so the DACL from the ISE cannot be downloaded completely.

The workaround is to increase the MTU or keep the DACL less than 20 lines.

Configuring the Authentication Rule and Interfaces

Follow these steps to configure the authentication rule and interfaces:

Before you begin

SISF-Based device tracking is a prerequisite to Web Authentication. Ensure that you have enabled device tracking programmatically or manually.

For more information, see *Confguring SISF-Based Tracking*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission name** *name* **proxy http**
4. **interface** *type slot/port*
5. **ip access-group** *name*
6. **ip admission name**
7. **exit**
8. **show ip admission**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip admission name** *name* **proxy http** **Example:** `Device(config)# ip admission name webauth1 proxy http`	Configures an authentication rule for web-based authorization.
Step 4	**interface** *type slot/port* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode and specifies the ingress Layer 2 or Layer 3 interface to be enabled for web-based authentication. *type* can be FastEthernet, GigabitEthernet, or TenGigabitEthernet.

	Command or Action	Purpose
Step 5	**ip access-group** *name* **Example:** `Device(config-if)# ip access-group webauthag`	Applies the default ACL.
Step 6	**ip admission name** **Example:** `Device(config)# ip admission name`	Configures an authentication rule for web-based authorization for the interface.
Step 7	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show ip admission** **Example:** `Device# show ip admission`	Displays the network admission cache entries and information about web authentication sessions.

Configuring AAA Authentication

If a method-list is configured under VTY lines, the corresponding method list must be added to the AAA configuration:

```
Device(config)# line vty 0 4
Device(config-line)# authorization commands 15 list1
Device(config-line)# exit
Device(config)# aaa authorization commands 15 list1 group tacacs+
```

If a method-list is not configured under VTY lines, you must add the default method list to the AAA configuration:

```
Device(config)# line vty 0 4
Device(config-line)# exit
Device(config)# aaa authorization commands 15 default group tacacs+
```

Follow these steps to configure AAA authentication:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default group** {**tacacs+** | **radius**}

5. **aaa authorization auth-proxy default group** {**tacacs+** | **radius**}
6. **tacacs server** *server-name*
7. **address** {**ipv4 | ipv6**} *ip address*
8. **key** *string*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA functionality.
Step 4	**aaa authentication login default group** {**tacacs+**	**radius**} **Example:** `Device(config)# aaa authentication login default group tacacs+`	Defines the list of authentication methods at login. **named_authentication_list** refers to any name that is not greater than 31 characters. **AAA_group_name** refers to the server group name. You need to define the server-group **server_name** at the beginning itself.
Step 5	**aaa authorization auth-proxy default group** {**tacacs+**	**radius**} **Example:** `Device(config)# aaa authorization auth-proxy default group tacacs+`	Creates an authorization method list for web-based authorization.
Step 6	**tacacs server** *server-name* **Example:** `Device(config)# tacacs server yourserver`	Specifies an AAA server.

	Command or Action	Purpose
Step 7	**address** {**ipv4**	**ipv6**} *ip address* **Example:** `Device(config-server-tacacs)# address ipv4 10.0.1.12`	Configures the IP address for the TACACS server.
Step 8	**key** *string* **Example:** `Device(config-server-tacacs)# key cisco123`	Configures the authorization and encryption key used between the switch and the TACACS server.
Step 9	**end** **Example:** `Device(config-server-tacacs)# end`	Exits the TACACS server mode and returns to privileged EXEC mode.

Configuring Switch-to-RADIUS-Server Communication

Follow these steps to configure the RADIUS server parameters:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface vlan** *vlan interface number*
4. **radius server** *server name*
5. **address** {**ipv4** | **ipv6**} *ip address*
6. **key** *string*
7. **exit**
8. **radius-server dead-criteria tries** *num-tries*
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip radius source-interface vlan** *vlan interface number* **Example:** `Device(config)# ip radius source-interface vlan 80`	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 4	**radius server** *server name* **Example:** `Device(config)# radius server rsim address ipv4 172.16.0.1`	(Optional) Specifies the IP address of the RADIUS server.
Step 5	**address** {**ipv4** | **ipv6**} *ip address* **Example:** `Device(config-radius-server)# address ipv4 10.0.1.2 auth-port 1550 acct-port 1560`	Configures the IP address for the RADIUS server.
Step 6	**key** *string* **Example:** `Device(config-radius-server)# key rad123`	(Optional) Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 7	**exit** **Example:** `Device(config-radius-server)# exit`	Exits the RADIUS server mode and enters the global configuration mode.
Step 8	**radius-server dead-criteria tries** *num-tries* **Example:** `Device(config)# radius-server dead-criteria tries 30`	Specifies the number of unanswered sent messages to a RADIUS server before considering the server to be inactive. The range of *num-tries* is 1 to 100.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the HTTP Server

To use web-based authentication, you must enable the HTTP server within the device. You can enable the server for either HTTP or HTTPS.

Note The Apple psuedo-browser will not open if you configure only the **ip http secure-server** command. You should also configure the **ip http server** command.

Follow the procedure given below to enable the server for either HTTP or HTTPS:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http secure-server**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server. The web-based authentication feature uses the HTTP server to communicate with the hosts for user authentication.
Step 4	**ip http secure-server** **Example:** `Device(config)# ip http secure-server`	Enables HTTPS. You can configure custom authentication proxy web pages or specify a redirection URL for successful login. **Note** To ensure secure authentication when you enter the **ip http secure-server** command, the login page is always in HTTPS (secure HTTP) even if the user sends an HTTP request.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device# end`	Exits global configuration mode and returns to privileged EXEC mode.

Customizing the Authentication Proxy Web Pages

You can configure web authentication to display four substitute HTML pages to the user in place of the default HTML pages during web-based authentication.

Follow these steps to specify the use of your custom authentication proxy web pages:

Before you begin

Store your custom HTML files on the device flash memory.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission proxy http login page file** *device:login-filename*
4. **ip admission proxy http success page file** *device:success-filename*
5. **ip admission proxy http failure page file** *device:fail-filename*
6. **ip admission proxy http login expired page file** *device:expired-filename*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip admission proxy http login page file** *device:login-filename* **Example:** `Device(config)# ip admission proxy http login page`	Specifies the location in the device memory file system of the custom HTML file to use in place of the default login page. The *device:* is flash memory.

	Command or Action	Purpose
	`file disk1:login.htm`	
Step 4	**ip admission proxy http success page file** *device:success-filename* **Example:** `Device(config)# ip admission proxy http success page file disk1:success.htm`	Specifies the location of the custom HTML file to use in place of the default login success page.
Step 5	**ip admission proxy http failure page file** *device:fail-filename* **Example:** `Device(config)# ip admission proxy http fail page file disk1:fail.htm`	Specifies the location of the custom HTML file to use in place of the default login failure page.
Step 6	**ip admission proxy http login expired page file** *device:expired-filename* **Example:** `Device(config)# ip admission proxy http login expired page file disk1:expired.htm`	Specifies the location of the custom HTML file to use in place of the default login expired page.
Step 7	**end** **Example:** `Device# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Web-Based Authentication Parameters

Follow these steps to configure the maximum number of failed login attempts before the client is placed in a watch list for a waiting period:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission max-login-attempts** *number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip admission max-login-attempts** *number* **Example:** `Device(config)# ip admission max-login-attempts 10`	Sets the maximum number of failed login attempts. The range is 1 to 2147483647 attempts. The default is 5.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Web-Based Authentication Local Banner

Follow these steps to configure a local banner on a switch that has web authentication configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission auth-proxy-banner http** [*banner-text* | *file-path*]
4. **end**

DETAILED STEPS

	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	**ip admission auth-proxy-banner http** [*banner-text* | *file-path*] **Example:** `Device(config)# ip admission auth-proxy-banner http C My Switch C`	Enables the local banner. (Optional) Create a custom banner by entering *C banner-text C* (where *C* is a delimiting character), or *file-path* that indicates a file (for example, a logo or text file) that appears in the banner.
Step 4	**end** **Example:** `Device# end`	Exits global configuration mode and returns to privileged EXEC mode.

Removing Web-Based Authentication Cache Entries

Follow these steps to remove web-based authentication cache entries:

SUMMARY STEPS

1. **enable**
2. **clear ip auth-proxy cache** {* | *host ip address*}
3. **clear ip admission cache** {* | *host ip address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip auth-proxy cache** {*	*host ip address*} **Example:** `Device# clear ip auth-proxy cache 192.168.4.5`	Delete authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

	Command or Action	Purpose
Step 3	**clear ip admission cache** {* | *host ip address*} **Example:** `# clear ip admission cache 192.168.4.5`	Delete authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

Verifying Web-Based Authentication

Use the commands in this topic to display the web-based authentication settings for all interfaces or for specific ports.

Table 17: Privileged EXEC show Commands

Command	Purpose
show authentication sessions method webauth	Displays the web-based authentication settings for all interfaces for fastethernet, gigabitethernet, or tengigabitethernet
show authentication sessions interface *type slot/port*[**details**]	Displays the web-based authentication settings for the specified interface for fastethernet, gigabitethernet, or tengigabitethernet. In Session Aware Networking mode, use the **show access-session interface** command.

Additional References for Web-Based Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support