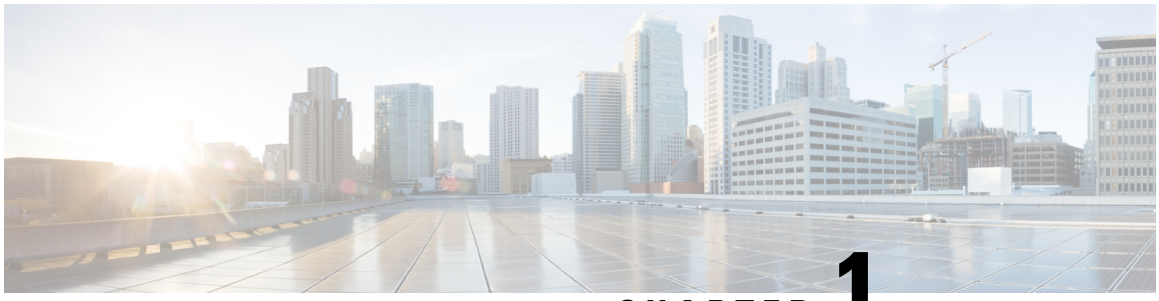
Security Configuration Guide, Cisco Catalyst IE31xx Series Switches

First Published: 2020-08-10

Last Modified: 2025-04-25

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

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 ISE), Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.
- To use the Change-of-Authorization (CoA) interface, a session must already exist on the switch. CoA can be used to identify a session and enforce a disconnect request. The update affects only the specified session.

RADIUS operation:

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

Restrictions for Configuring RADIUS

General:

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

RADIUS is not suitable in the following network security situations:

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

DSCP marking support for RADIUS packets:

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

Information about RADIUS

RADIUS and Switch Access

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

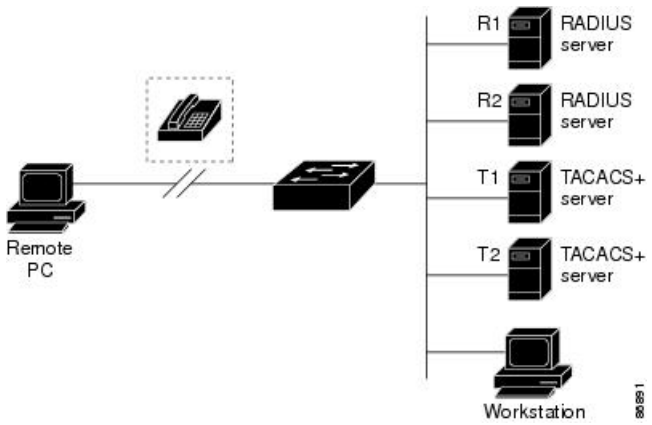
RADIUS Overview

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

Use RADIUS in these network environments that require access security:

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

Figure 1: Transitioning from RADIUS to TACACS+ Services



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

RADIUS Operation

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

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

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

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

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

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. When a policy changes for a user or user group in AAA, administrators can send RADIUS CoA packets from the AAA server such as a Cisco ISE to reinitialize authentication and apply the new policy. This section provides an overview of the RADIUS interface including available primitives and how they are used during a CoA.

- Change-of-Authorization Requests
- CoA Request Response Code
- CoA Request Commands
- Session Reauthentication

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 Identity Services Engine (ISE).

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

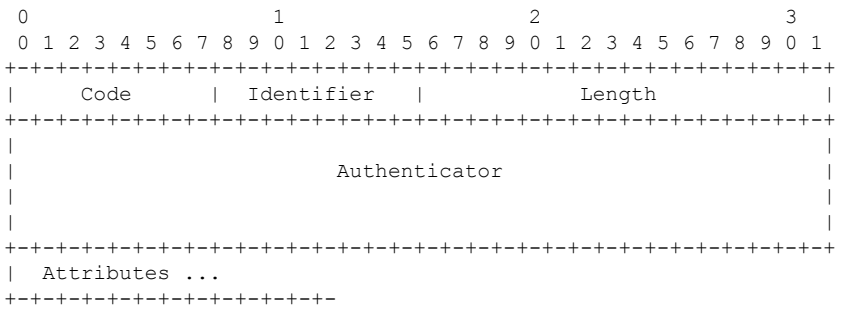
For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- 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)."

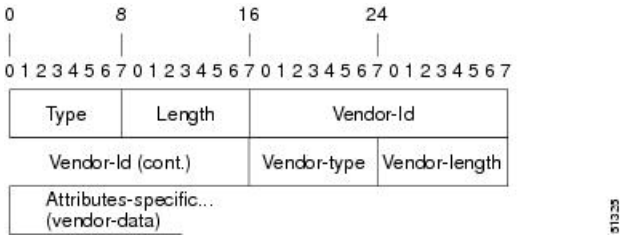
Attribute 26 contains the following three elements:

- 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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *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:**	(Optional) Specifies the RADIUS server parameters.

	Command or Action	Purpose
	`Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612`	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 configure the **ip http authentication aaa** global configuration command. By default, configuring AAA authentication does not secure the device for HTTP access by using AAA methods.

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	**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 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.

	Command or Action	Purpose
		• *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:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *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.
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 privileged access and network services:

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	**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:** `Device(config)# aaa authorization exec list1 radius`	Configures the device for user RADIUS authorization if the user has privileged EXEC access. The **exec** keyword might return user profile information (such as **autocommand** information).
Step 5	**end** **Example:** `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:

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *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:** `Device(config-radius-server)# key your_server_key`	Specifies the shared secret text string used between the switch and all RADIUS servers. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 5	**retransmit** *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:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server 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:

Procedure

	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	**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:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *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:**	Exits RADIUS server mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-radius-server)# end`	

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:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**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:** `Device(config-sg-radius)# dscp auth 10`	Configures DSCP marking for authentication and accounting on the radius server group. • **acct** configures radius DSCP marking value for accounting. The valid range is from 1 to 63. The default value is 0.

	Command or Action	Purpose
	`acct 20`	• **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.

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	**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.

	Command or Action	Purpose
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:** `Device(config-locsvr-da-radius)# exit`	Exits dynamic authorization local server configuration mode and returns to global configuration mode.
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.

	Command or Action	Purpose
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.

Feature History for RADIUS

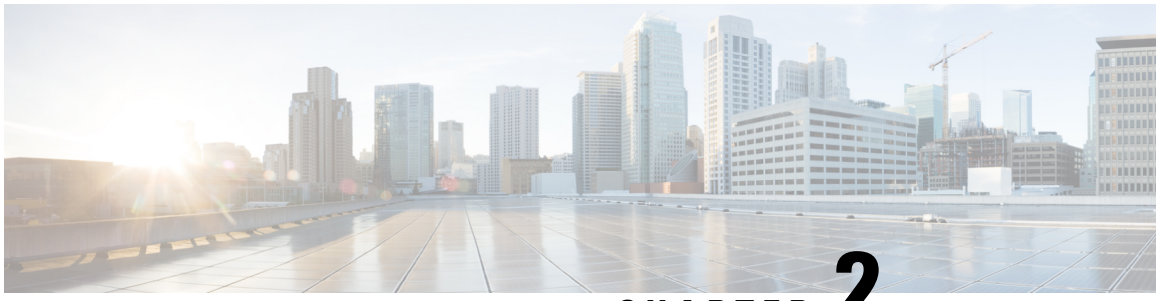
This table provides release and related information for the features explained in this module.

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

Feature	Feature Information	Release
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 when IPDT is 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.

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).

Note When DAI is enabled on the CPU, and if the limitation on ARP packets exceeds, IE3100 drops ARP packets randomly for few seconds.

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.

Configured Feature	ARP Packet 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.
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.
- IPDT policy can be attached or detached per interface or VLAN.

Example IPDT Configuration

The Delayless IPDT feature does not have any specific CLI to configure it. It will be automatically turned on when IPDT is 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

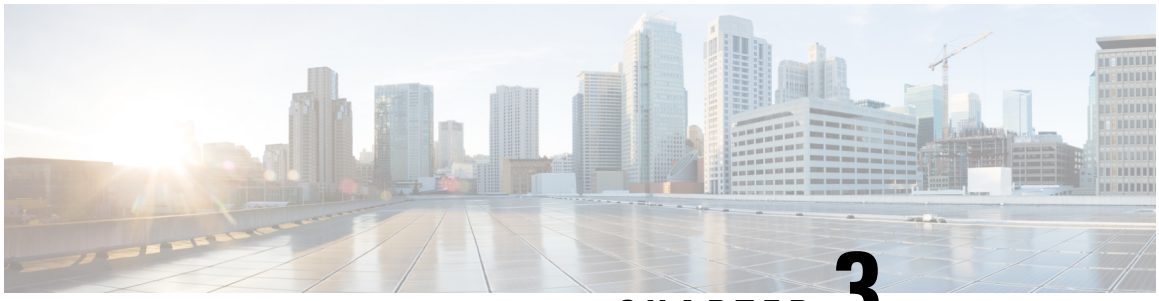
- show device-tracking database vlanid <vlanid>

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 Switch Integrated Security Features

Information About SISF

Overview

Switch Integrated Security Features (SISF) is a framework developed to optimize security in Layer 2 domains. It merges the IP Device Tracking (IPDT) and *certain* IPv6 first-hop security (FHS) functionality[2], to simplify the migration from IPv4 to IPv6 stack or a dual-stack.

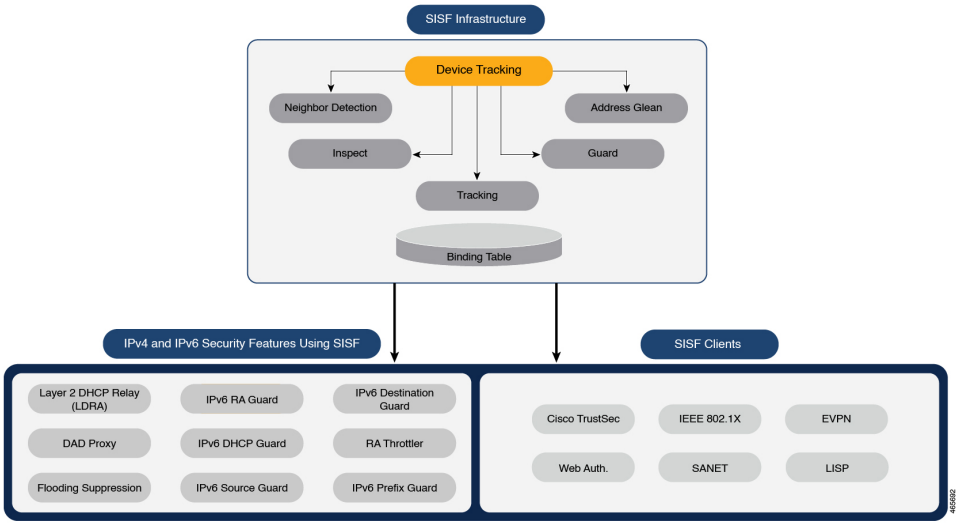
The SISF infrastructure provides a unified database that is used by:

- IPv6 FHS features: IPv6 Router Advertisement (RA) Guard, IPv6 DHCP Guard, Layer 2 DHCP Relay, IPv6 Duplicate Address Detection (DAD) Proxy, Flooding Suppression, IPv6 Source Guard, IPv6 Destination Guard, RA Throttler, and IPv6 Prefix Guard.
- Features like Cisco TrustSec, IEEE 802.1X, and Web Authentication, which act as clients for SISF.

The following figure illustrates this:

[2] IPv6 Snooping Policy, IPv6 FHS Binding Table Content, and IPv6 Neighbor Discovery Inspection

Figure 3: SISF Framework



Note The terms “SISF” “device-tracking” and “SISF-based device-tracking” are used interchangeably in this document and refer to the same feature. Neither term is used to mean or should be confused with the legacy IPDT or IPv6 Snooping features.

Understanding the SISF Infrastructure

This section explains the various elements of the SISF infrastructure as shown in the SISF Framework above.

The Binding Table

The SISF infrastructure is built around the binding table. The binding table contains information about the hosts that are connected to the ports of a switch and the IP and MAC address of these hosts. This helps to create a physical map of all the hosts that are connected to a switch.

Each entry in a binding table provides the following information about a connected host:

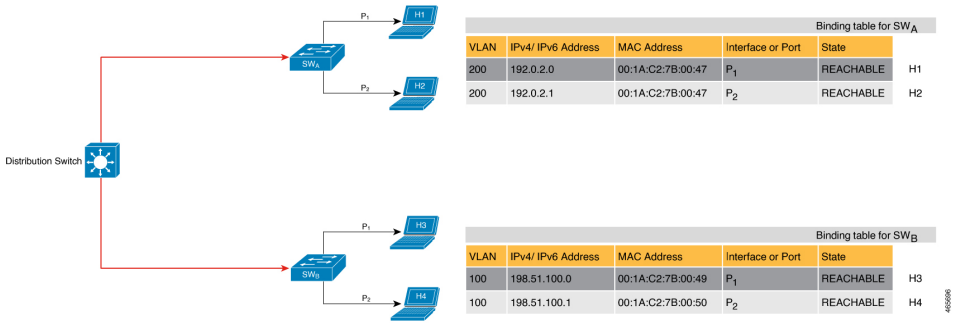
- IPv4 or IPv6 address of the host.
- MAC address of the host. The same MAC address may be linked to an IPv4 and IPv6 address.
- The interface or port on the switch that the host is connected to, and the associated VLAN.
- The state of the entry, which indicates the reachability of the entry.

The following figure shows a simple network topology and a representative binding table for each access switch in the network. SW_A and SW_B are the two access switches in the network. The two access switches are connected to the same distribution switch. H1, H2, H3, H4 are the hosts.

This is an example of a distributed binding table, that is, each access switch in the network has its own table. An alternative set-up could be one centralised binding table on the distribution switch with the entries of SW_A *and* SW_B.

Having a distributed or a centralised binding table is a key design choice in the process of implementing SISF in your network and is covered in greater detail in the Understanding Policy Parameters, on page 51 section in this chapter.

Figure 4: Binding Table



States and Lifetime of a Binding Table Entry

The state of an entry indicates if the host is reachable or not. The stable states of a binding table entry are: REACHABLE, DOWN, and STALE. When changing from one state to another, an entry may have other temporary or transitional states such as: VERIFY, INCOMPLETE, and TENTATIVE.

How long an entry remains in a given state is determined by its lifetime and by whether or not the entry is validated successfully. The lifetime of an entry can be policy-driven or configured globally.

To configure the REACHABLE, DOWN, and STALE lifetimes, enter the following command in global configuration mode:

device-tracking binding { **reachable-lifetime** { *seconds* | **infinite** } | **stale-lifetime** { *seconds* | **infinite** } | **down-lifetime** { *seconds* | **infinite** } }

State: Reachable

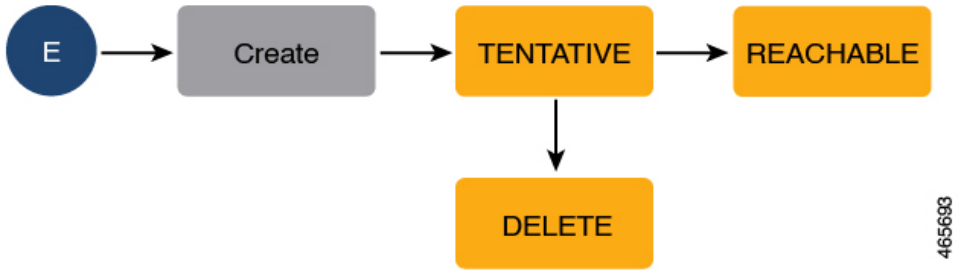
If an entry has this state, it means the host (IP and MAC address) from which a control packet was received, is a verified and valid host. A reachable entry has a default lifetime of 5 minutes. You can also configure a duration. By configuring a reachable-lifetime, you specify how long a host can remain in a REACHABLE state, after the last incoming control packet from that host.

If an event is detected before the entry's reachable lifetime expires, then the reachable lifetime is reset.

To qualify for the REACHABLE state, a new entry goes through the process illustrated in the figure below. The switch detects an event (E), such as an incoming control packet from a connected host and creates an entry. Various events cause the creation of an entry, and these are described in the Binding Table Sources section. The creation of an entry is followed by different transient states, such as TENTATIVE or INCOMPLETE. While in a transitional state, the switch validates and confirms the integrity of the binding entry. If the entry is found to be valid, then the state changes to REACHABLE.

But if an address theft or similar event is detected, then the entry is regarded as invalid and is deleted. For example, if an attacker sends unsolicited neighbor advertisement messages with the same IP as the target IP and its (attacker's) own MAC address to redirect traffic.

Figure 5: Creation of a Reachable Entry

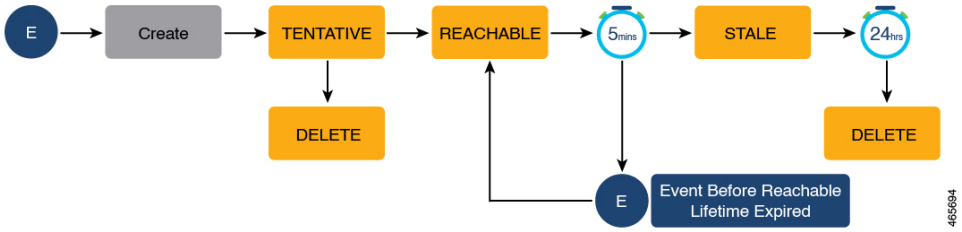


State: Stale

If an entry is in this state it means that the entry's reachable lifetime has expired and the corresponding host is still silent (no incoming packets from the host). A stale entry has a default lifetime of 24 hours. You can also configure a duration. An entry that remains in the STALE state beyond the stale lifetime, is deleted.

This is illustrated in the figure below which depicts the lifecycle of an entry.

Figure 6: Lifecycle of an Entry



State: Down

If an entry is in this state, it means that the host's connecting interface is down. A down entry has a default lifetime of 24 hours. You can also configure a duration. An entry that remains in the DOWN state beyond the down lifetime, is deleted.

Polling a Host and Updating the Binding Table Entry

Polling is a periodic and conditional checking of the host to see the state it is in, whether it is still connected, and whether it is communicating. In addition to determining an entry's state, you can use polling to reconfirm an entry's state.

You can enable polling with the **device-tracking tracking** command in global configuration mode. After you do, you still have the flexibility to turn polling on or off for a particular interface or VLAN. For this, configure the **tracking enable** or **tracking disable** keywords in the policy (the device-tracking configuration mode). When polling is enabled, the switch polls the host at the specified interval, thus reconfirming its reachability for the duration of its reachable lifetime.

When polling is enabled, the switch sends up to three polling requests, after the reachable lifetime expires, at system-determined intervals. You can also configure this interval with the **device-tracking tracking retry-interval** *seconds* command in global configuration mode.

The figure below depicts the lifecycle of an entry where the host is polled. Default reachable and stale lifetimes, and retry intervals are used in figure:

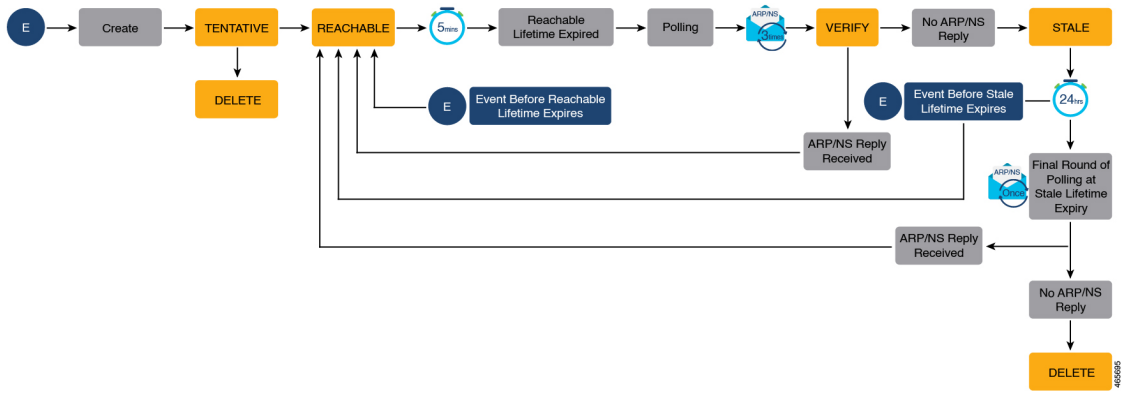
An event (E) is detected and a REACHABLE entry is created.

If an event is detected *during* the reachable lifetime, the reachable lifetime timer is reset.

The switch sends a polling request after the reachable lifetime expires. The switch polls the host up to three times at fixed, system-determined intervals. The polling request may be in the form of a unicast Address Resolution Protocol (ARP) probe or a Neighbor Solicitation message. During this time the state of the entry changes to VERIFY. If a polling response is received (thus confirming reachability of the host), the state of the entry changes back to REACHABLE.

If the switch does not receive a polling response after three attempts, the entry changes to the STALE state. It remains in this state for 24 hours. If an event is detected during the stale lifetime, the state of the entry is changed back to REACHABLE. At expiry of the stale lifetime, the device sends one final polling to ascertain reachability. If this final polling attempt receives a reply, the state of the entry is changed back to REACHABLE. If the final polling attempt does not receive a response, the entry is deleted.

Figure 7: Lifecycle of an Entry Where the Host is Polled



Binding Table Sources

The following are the sources of information and events that cause the creation and update of a binding table entry:

- Learning events that dynamically populate the binding table:
  - Dynamic Host Configuration Protocol (DHCP) negotiation (DHCP REQUEST, and DHCP REPLY). This includes DHCPv4 and DHCPv6.
  - Address Resolution Protocol (ARP) packets.

    ARP packets are throttled to mitigate high CPU utilization scenarios. In a five second window, a maximum of 50 ARP broadcast packets per binding entry are processed by SISF. When the limit is reached, incoming ARP packets are dropped. Note that the limit of 50 in five seconds is for each binding entry, that is, for each source IP. This limit is increased to a maximum of 100 ARP broadcast packets for each source IP. When the limit is reached, incoming ARP packets are dropped.
  - Neighbor Discovery Protocol (NDP) packets.
  - Multiple Identity Association-Nontemporary Address (IA_NA) and Identity Association-Prefix Delegation (IA_PD).

In some cases, a network device can request and receive more than one IPv6 address from the DHCP server. This may be done to provide addresses to multiple clients of the device, such as when a residential gateway requests addresses to distribute to its LAN clients. When the device sends out a DHCPv6 packet, the packet includes all of the addresses that have been assigned to the device.

When SISF analyzes a DHCPv6 packet, it examines the IA_NA (Identity Association-Nontemporary Address) and IA_PD (Identity Association-Prefix Delegation) components of the packet and extracts each IPv6 address contained in the packet. SISF adds each extracted address to the binding table.

Entries created through learning events like the ones listed above are called "dynamic entries". In the output of the **show device-tracking database details** privileged EXEC command, such entries are prefixed with an abbreviation that clarifies the kind of dynamic learning event it was. For example, ARP for ARP packets, ND for NDP packets, and so on.

- Configuring static binding entries.

  If there are silent but reachable hosts in the Layer 2 domain, you can create static binding entries to retain binding information even if the host becomes silent.

  A static binding entry is a binding entry that is manually added to the binding table, by configuring the following command in global configuration mode:

  device-tracking binding vlan *vlan_id* { *ipv4_add ipv6_add ipv6_prefix* } [**interface** *inteface_type_no*] [*48-bit-hardware-address*] [**reachable-lifetime** { *seconds* | **default** | **infinite** } | **tracking** { **default** | **disable** | **enable** [**retry-interval** { *seconds* | **default** }] } [**reachable-lifetime** { *seconds* | **default** | **infinite** }]]

  In the output of the **show device-tracking database details** privileged EXEC command, static entries are prefixed with the letter "S".

  You can configure a reachable lifetime for a static entry. The stale and down lifetime timer is fixed by the system as **infinite** (For an entry in the STALE or DOWN state, the output of the **show device-tracking database** command displays the `Time Left` column as “N/A”). This means that when a static entry enters the STALE or DOWN state it remains in this state, and in the binding table, indefinitely.

  A static entry can be removed from the binding table only by the actions listed below. It cannot be deleted from the binding table by using **clear** commands or by any other event:

  - You remove the entry by configuring the **no** form of the above command.
  - A local entry replaces the static entry.

    A local entry is an entry that is automatically created by the system when you configure an SVI on the device. When configuring the SVI, if you use the same IP address as the static entry then the static entry is replaced with the local entry, because the local entry has a higher priority.

    This replacement of a static entry by a local entry is introduced.

    In the output of the **show device-tracking database details** privileged EXEC command, local entries are prefixed with the letter "L".

  For more information about static binding entries, see the **device-tracking binding** command in the command reference.

Note In addition to the primary or key events listed above, there is a specific scenario in which a ping can result in a device-tracking entry. If a sender's ARP cache or IPv6 neighbor table doesn't have the target's IP address yet, then a ping triggers an ARP packet for IPv4, or ND packet for IPv6. This can result in a device-tracking entry.

But if the target IP is already in the ARP cache or IPv6 neighbour table, no ARP or ND packet is generated when you ping - in which case SISF cannot learn the IP address.

Device-Tracking

SISF-based device-tracking is disabled by default. You can enable the feature on an interface or VLAN.

When you enable the feature, the binding table is created, followed by subsequent maintenance of the binding table.

The events listed in the Binding Table Sources, on page 49 section act as triggers for SISF-based device-tracking, to track the presence, location, and movement of hosts in the network, to populate and maintain the binding table. For example, if information about a host is learnt by means of an ARP or ND packet, every subsequent ARP or ND packet from the same host acts as an alert for SISF-based device-tracking, to refresh the entry in the binding table, thus indicating if the host is still present in the same location or has moved.

The continuous process of snooping of packets that the switch receives, extraction of device identity (MAC and IP address), and storage of information in the binding table of the switch, ensures binding integrity and maintains the reachability status of the hosts in the binding table.

For information how to enable SISF-based device-tracking, see How to Configure SISF, on page 61.

Device-Tracking Policy

A device-tracking policy is a set of rules that SISF-based device-tracking follows. The policy dictates which events will be listened to, whether a host will be probed, the wait time before the host is probed, and so on. These rules are referred to as policy parameters.

Note The policy must be attached to an interface or VLAN. Only then is the binding table for that interface or VLAN populated - in accordance with policy parameters.

For information about the various ways in which you can create a policy, see How to Configure SISF, on page 61.

To display a policy's settings, use the **show device-tracking policy** *policy_name* command in privileged EXEC mode.

Understanding Policy Parameters

Policy parameters are the keywords available for configuration in the device-tracking configuration mode. Each policy parameter addresses one or more aspects of network security.

This section explains the purpose of *some* of the important policy parameters so you can configure your policy to better suit your requirements.

```
Device(config)# device-tracking policy example_policy
Device(config-device-tracking)# ?
device-tracking policy configuration mode:

  device-role           Sets the role of the device attached to the port
  limit                 Specifies a limit
  security-level        setup security level
  tracking              Override default tracking behavior
  trusted-port          setup trusted port
```

For information about all the parameters displayed in the device-tracking configuration mode, see the command reference document of the corresponding release.

Glean versus Guard versus Inspect

When a packet enters the network, SISF extracts the IP and MAC address (the source of the packet) and subsequent action, is dictated by the security-level that is configured in the policy.

Glean, guard, and inspect are the options available under the security-level parameter. Glean is the least secure option, inspect, is moderately secure, and guard, is the most secure.

To configure this parameter in a policy, enter the **security-level** keyword in the device-tracking configuration mode.

Glean

When the security-level is set to **glean**, SISF extracts the IP and MAC address and enters them into the binding table, without any verification. This option therefore does not ensure binding integrity. It may for example, be suited to a set-up where client applications such as IEEE 802.1X or SANET want to only learn about the host and not rely on SISF for authentication.

The only factor that affects the addition of the binding entry for this security-level, is the address count limit. There are separate limits for the maximum number of IPs per port, IPv4 per MAC, and IPv6 per MAC. Entries are rejected once a limit is reached. For more information about this parameter, see Address Count Limits, on page 59.

Guard

This is the default value for the security-level parameter.

When the security-level is set to **guard**, SISF extracts and verifies the IP and MAC address of packets entering the network. The outcome of the verification determines if a binding entry is added, or updated, or if the packet is dropped and the client is rejected.

The process of verification starts with the search for a matching entry in the database. The database may be centralised or distributed. If a matching entry is not found, a new entry is added.

If a matching entry is found and the points of attachment (MAC, VLAN, or interface) are found to be the same, only the timestamp is updated. If not, the scope of verification is extended to include validation of address ownership. This may include host polling to determine if the change in the point of attachment (a different MAC, or VLAN) is valid. If the change is valid the entry is updated, or if it is a case of theft, the entry is not added to the binding table.

If a binding entry is added or updated, the corresponding client is granted access to the network. If an entry does not pass verification, the corresponding client is rejected.

Note The verification process affects the binding entry and the corresponding incoming packet.

The **guard** security-level supports the *prevention* of IPv4 spoofing. Detection and reporting of IPv4 spoofing is supported since the introductory release of SISF. Further, detection, reporting, and prevention of *IPv6 spoofing* is supported since the introductory release of SISF. For more information, see: Example: Detecting and Preventing Spoofing, on page 74 .

Inspect

Even though security-level **inspect** is available on the CLI, we recommend not using it. The **glean** and **guard** options described above address most use cases and network requirements.

The security level parameter affects how ARP and ND packets are handled.

Trusted-Port and Device-Role Switch

The **device-role switch** and **trusted-port** options help you design an efficient and scalable secure zone. When used together, these two parameters help you achieve an efficient distribution of the creation of entries in the binding table. This keeps the binding tables size under control.

The **trusted-port** option: Disables the guard function on configured targets. Bindings learned through a trusted-port have preference over bindings learned through any other port. A trusted port is also given preference in case of a collision while making an entry in the table.

The **device-role** option: Indicates the type of device that is facing the port and this can be a node or a switch. To allow the creation of binding entries for a port, you configure the device as a node. To stop the creation of binding entries, you configure the device as switch.

Configuring the device as a switch is suited to multi-switch set-ups, where the possibility of large device tracking tables is very high. Here, a port facing a device (an uplink trunk port) can be configured to stop creating binding entries, and the traffic arriving at such a port can be trusted, because the switch on the other side of the trunk port will have device-tracking enabled and that will have checked the validity of the binding entry.

Note While there are scenarios where configuring only either one of these options may be suitable, the more common use case is for both the **trusted-port** and **device-role switch** options to be configured on the port - the examples below explain this in detail. Possible scenarios where only either one of these options is suited or required have also been described, at the end of this section.

To configure these parameters in a policy, enter the **trusted-port** and **device-role** keywords in the device-tracking configuration mode.

Example: Using Trusted-Port and Device-Role Switch Options in a Multi-Switch Set-Up

The following example explains how the **device-role switch** and **trusted-port** options help to design an efficient and scalable “secure zone”.

In figure "*Multi-Switch Set-Ups Without Trusted-Port and Device-Role Switch Options*" below, SW_A, SW_B, and SW_C are three access switches. They are all connected to a common distribution switch. The only required configuration on the distribution switch in this scenario is to ensure that traffic of any kind is *not* blocked.

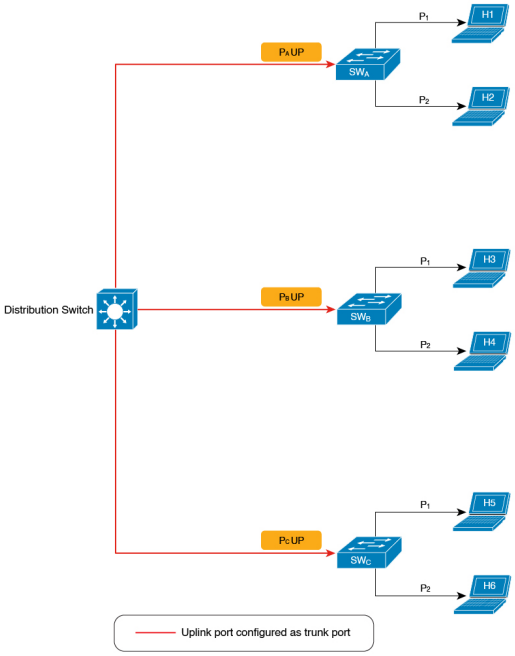
H1, H2, …H6 are the hosts. Each switch has two directly connected hosts. All hosts are communicating with each other, that is, control packets are being transmitted. All hosts are also within the same VLAN boundary. Each switch is receiving control packets from hosts that are directly connected to it, and also from hosts that are connected to other switches. This means SW_A is receiving control packets from H1, H2, …H6 similarly with SW_B and SW_C.

For each switch, the entries of directly connected hosts have interface or port P_1 and P_2 in the binding table. Entries originating from hosts that are connected to other switches have interface or port name P_xUP, to show that they have been learned through the uplink port (x represents the corresponding uplink port for each switch. For example, the entries that SW_A has learnt through its uplink port have interface or port name P_AUP and for SW_B it is P_BUP, and so forth.

The end result is that each switch learns and creates binding entries for all hosts in the set-up.

This scenario displays an inefficient use of the binding table, because each host is being validated multiple times, which does not make it more secure than if just one switch validates host. Secondly, entries for the same host in multiple binding tables could mean that the address count limit is reached sooner. After the limit is reached, any further entries are rejected and required entries may be missed this way.

Figure 8: Multi-Switch Set-Ups Without Trusted-Port and Device-Role Switch Options



Binding table for SW_A

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_1	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_2	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_AUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_AUP	REACHABLE	H4
300	192.0.2.5	00:1A:C2:7B:00:49	P_AUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_AUP	REACHABLE	H6

Binding table for SW_B

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
200	192.0.2.3	00:1A:C2:7B:00:48	P_1	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_2	REACHABLE	H4
100	192.0.2.1	00:1A:C2:7B:00:47	P_BUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_BUP	REACHABLE	H2
300	192.0.2.5	00:1A:C2:7B:00:49	P_BUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_BUP	REACHABLE	H6

Binding table for SW_C

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
300	192.0.2.5	00:1A:C2:7B:00:49	P_1	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_2	REACHABLE	H6
100	192.0.2.1	00:1A:C2:7B:00:47	P_CUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_CUP	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_CUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_CUP	REACHABLE	H4

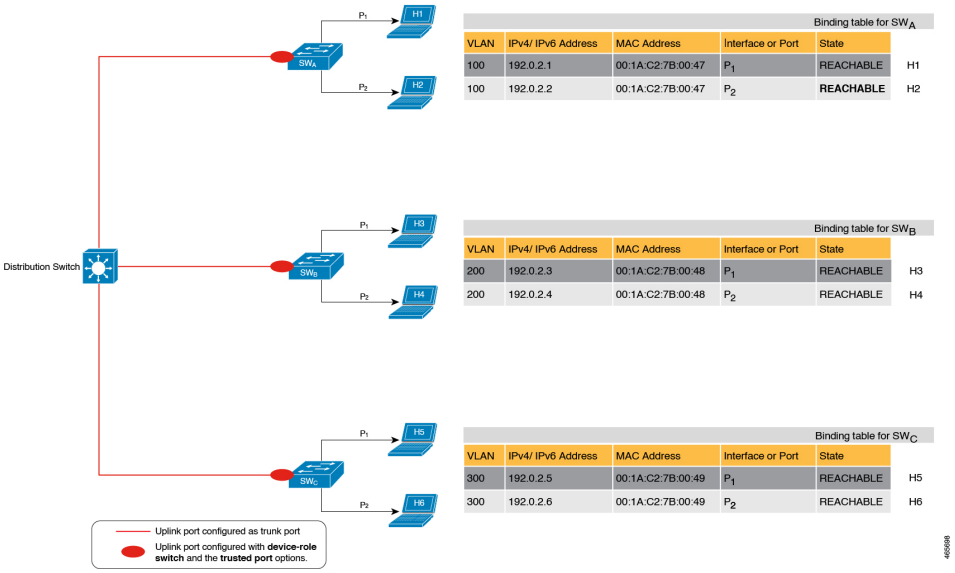
By contrast, see figure "Multi-Switch Set-Ups With Trusted-Port and Device-Role Switch Options" below. Here, when SW_A intercepts the packet of a host that is not attached to it (say H3 which is directly attached to SW_B), it does not create an entry because it detects that H3 is attached to a device that is configured as a switch (**device-role switch** option) and the uplink port of the switch (where the packet came from) is a trusted port (**trusted-port** option).

By creating binding entries only on switches where the host appears on an access port (port P_1 and P_2 of each switch), and not creating entries for a host that appears over an uplink port or trusted port (P_x UP), each switch

in the set-up validates and makes only the required entries, thus achieving an efficient distribution of the creation of binding table entries.

A second advantage of configuring **device-role switch** and **trusted-port** options in a multi-switch scenario is that it prevents duplicate entries when a host, say H1 moves from one switch to another. H1's IP and MAC binding in the earlier location (let's say SW_A) continues to remain there until it reaches the STALE state. But if H1 moves and connects to a second switch, say SW_C, then SW_A receives a duplicate binding entry through the uplink port. In such a situation, if the uplink port of the second switch (SW_C) is configured as a trusted port, SW_A deletes its stale entry. Further, it doesn't create another new binding entry because the SW_C will already have the latest entry and this entry is trusted.

Figure 9: Multi-Switch Set-Ups With Trusted-Port and Device-Role Switch Options



Binding table for SW_A

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_1	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_2	**REACHABLE**	H2

Binding table for SW_B

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
200	192.0.2.3	00:1A:C2:7B:00:48	P_1	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_2	REACHABLE	H4

Binding table for SW_C

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
300	192.0.2.5	00:1A:C2:7B:00:49	P_1	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_2	REACHABLE	H6

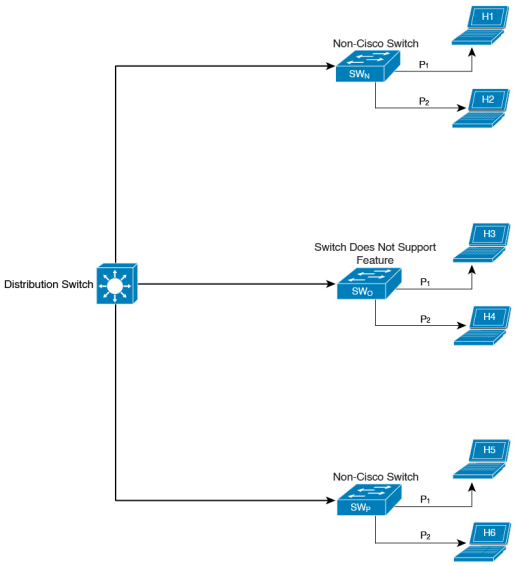
Example: When Not to Use Trusted-Port and Device-Role Switch Options

While the previous example clarifies how a multi-switch set-up with distributed binding tables stands to benefit from the **device-role switch** and **trusted-port** options, it may not suit networks of the following kinds:

- Networks where non-Cisco switches are being used
- Networks where the switch does not support the SISF-based device-tracking feature.

In both cases, we recommended that you not configure the **device-role switch** and **trusted-port** options. Further, we recommended that you maintain a centralised binding table - on the distribution switch. When you do, all the binding entries for all the hosts connected to non-Cisco switches and switches that do not support the feature, are validated by the distribution switch and still secure your network. The figure below illustrates the same.

Figure 10: Centralised Binding Table



Binding table for Distribution Switch

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_NUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_NUP	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_OUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_OUP	REACHABLE	H4
300	192.0.2.5	00:1A:C2:7B:00:49	P_PUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_PUP	REACHABLE	H6

465699

Creating an Efficient and Scalable Secure Zone

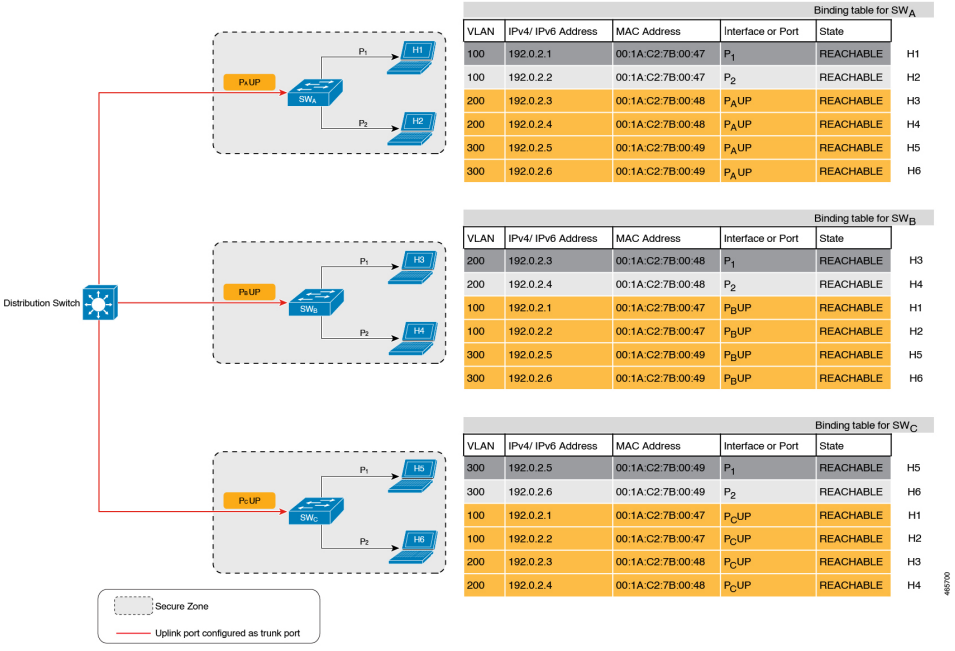
By using the **trusted-port** and **device-role switch** options in suitable networks and leaving them out in others, you can achieve an efficient and scalable secure zone.

Secure Zones 1, 2 and 3, display three different set-ups and the secure zone that is established in each case.

Secure Zone:	**Secure Zone 1 - Inefficient and Unscalable Secure Zone**	**Secure Zone 2 - Efficient and Scalable Secure Zone When Binding Tables are Decentralized**	**Secure Zone 3: Efficient Secure Zone When Binding Table is Centralized**
Scalability:	Unscalable; each switch has entries of all the hosts in the network	Scalable; each switch as entries of only directly connected hosts	Unscalable; the distribution switch has entries of all hosts in the network
Polling and its effect on the network: n = number of hosts m = number of switches total number of polling requests: = n X m	18 polling requests are being sent (6 hosts x 3 switches). Each host is polled by all the switches in the network (in the absence of the **trusted-port** and **device-role switch** options). Network load is very high.	6 polling requests are being sent (2 hosts x 1 switch for *each* switch). Minimal network load. (Polling requests are sent by the local access switches to directly connected hosts, each polling request passes through fewer points in the network.)	6 polling requests are being sent (6 hosts x 1 switch) Network load is higher than secure zone 2, but not as high as secure zone 1. (Polling requests come from the distribution switch and go through the access switch before reaching the host.)

Secure Zone:	Secure Zone 1 - Inefficient and Unscalable Secure Zone	Secure Zone 2 - Efficient and Scalable Secure Zone When Binding Tables are Decentralized	Secure Zone 3: Efficient Secure Zone When Binding Table is Centralized
Efficiency:	Inefficient binding table, because the binding table is duplicated on each switch.	Efficient binding table, because each host's binding information is entered only once, and in one binding table and this the binding table of the directly connected switch.	Efficient binding table, because the binding information for each host is entered only once, and this is in the central binding table, which is on the distribution switch.
Recommended Action:	Reapply suitable policies to make the secure zone like secure zone 2	None; this is an efficient and scalable secure zone.	None; this is the best possible secure zone given the type of set-up (where the other switches in the network are either non-Cisco or do not support the feature)

Figure 11: Secure Zone 1 - Inefficient and Unscalable Secure Zone



Binding table for SW_A

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_1	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_2	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_AUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_AUP	REACHABLE	H4
300	192.0.2.5	00:1A:C2:7B:00:49	P_AUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_AUP	REACHABLE	H6

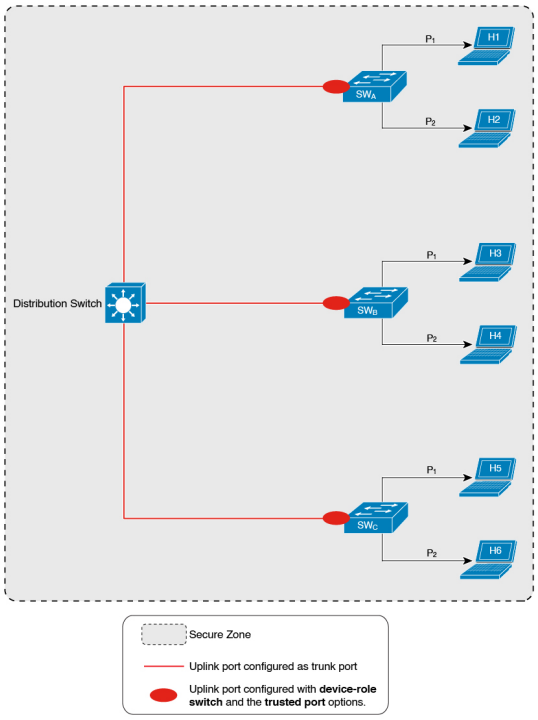
Binding table for SW_B

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
200	192.0.2.3	00:1A:C2:7B:00:48	P_1	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_2	REACHABLE	H4
100	192.0.2.1	00:1A:C2:7B:00:47	P_BUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_BUP	REACHABLE	H2
300	192.0.2.5	00:1A:C2:7B:00:49	P_BUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_BUP	REACHABLE	H6

Binding table for SW_C

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
300	192.0.2.5	00:1A:C2:7B:00:49	P_1	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_2	REACHABLE	H6
100	192.0.2.1	00:1A:C2:7B:00:47	P_CUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_CUP	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_CUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_CUP	REACHABLE	H4

Figure 12: Secure Zone 2 - Efficient and Scalable Secure Zone When Binding Tables are Decentralized



Binding table for SW_A

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_1	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_2	**REACHABLE**	H2

Binding table for SW_B

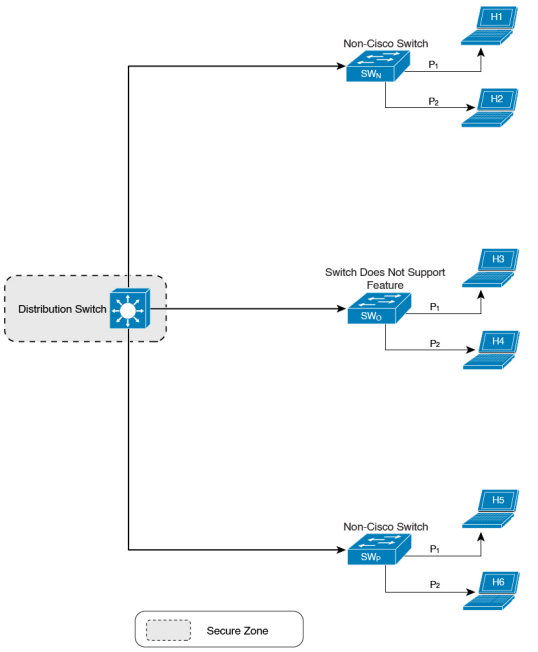
VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
200	192.0.2.3	00:1A:C2:7B:00:48	P_1	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_2	REACHABLE	H4

Binding table for SW_C

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
300	192.0.2.5	00:1A:C2:7B:00:49	P_1	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_2	REACHABLE	H6

465701

Figure 13: Secure Zone 3: Efficient Secure Zone When Binding Table is Centralized



Binding table for Distribution Switch

VLAN	IPv4/ IPv6 Address	MAC Address	Interface or Port	State	
100	192.0.2.1	00:1A:C2:7B:00:47	P_NUP	REACHABLE	H1
100	192.0.2.2	00:1A:C2:7B:00:47	P_NUP	REACHABLE	H2
200	192.0.2.3	00:1A:C2:7B:00:48	P_OUP	REACHABLE	H3
200	192.0.2.4	00:1A:C2:7B:00:48	P_OUP	REACHABLE	H4
300	192.0.2.5	00:1A:C2:7B:00:49	P_PUP	REACHABLE	H5
300	192.0.2.6	00:1A:C2:7B:00:49	P_PUP	REACHABLE	H6

465702

When to Use Only Trusted-Port or Only Device-Role Switch

Configuring only **device-role switch** is suited to situations when you want to listen but not learn entries. For example, for Duplicate Address Detection (DAD), or when you want to send IPv6 or Neighbor Solicitation (NS) message on a switch-facing port.

When you configure this option on a switch port (or interface), SISF-based device-tracking treats the port as a trunk port, implying that the port is connected to other switches. It does not matter whether the port is actually a trunk port or not. Therefore, when NS packets or queries are sent to switches in the network for new entry validation, only the secure ports (ports where the **device-role switch** is configured) receive the packet or query. This safeguards the network. If the command is not configured on any port, a general broadcast of the query is sent.

Configuring only **trusted-port** is suited to situations where an access port should be configured as a trusted port. If an access port is connected to a DHCP server or a similar service that the switch is consuming, configuring an access port as a trusted port ensures that the service is not disrupted because traffic from such a port is trusted. This also widens the secure zone, to include the access port.

Address Count Limits

The address count limit parameter specifies limits for the number of IP and MAC addresses that can be entered in a binding table. The purpose of these limits is to contain the size of the binding table based on the number of known and expected hosts, thus enabling you to take pre-emptive action against rogue hosts or IPs in the network.

At a policy level there are separate limits for the number of IP addresses per port, the number of IPv4 addresses per MAC, and IPv6 addresses per MAC. You can configure or change only the number of IP addresses per port.

IP per Port

The IP per port option is the total number of IP addresses allowed for a port. The address can be IPv4 or IPv6. When the limit is reached, no further IP addresses (i.e., entries) are added to the binding table.

To configure this parameter in a policy, enter the **limit address-count** *ip-per-port* keyword in device-tracking configuration mode. If you configure a limit that is lower than the currently configured one, then the new (lower) limit is applicable only to new entries. An existing entry remains in the binding table and goes through its binding entry lifecycle.

IPv4 per MAC and IPv6 per MAC

This refers to the number of IPv4 addresses that can be mapped to one MAC address and the number of IPv6 addresses that can be mapped to one MAC address. When the limit is reached, no further entries can be added to the binding table, and traffic from new hosts will be dropped

Note The IPv4 per MAC limit and the IPv6 per MAC limit that is effective on an interface or VLAN is as defined in the policy that is applied. If the policy does not specify a limit, this means that a limit does not exist. You cannot change or configure a limit for IPv4 per MAC or IPv6 per MAC for any kind of policy (programmatic, or custom policy, or default policy).

Enter the **show device-tracking policy** *policy name* to check if a limit exists.

Address Count Limit Considerations and Interactions with Other SISF Settings

- The limits do not have a hierarchy, but the threshold that is set for each limit affects the others.

  For example, if the IP per port limit is 100, and the IPv4 per MAC limit is one, the limit is reached with a single host's IPv4-MAC binding entry. No further IP entries, which are bound to the same MAC are allowed in the table even though the port has a provision for 99 more IP addresses. Similarly, if the IP per port limit is one, and the IPv4 per MAC limit is 100. The limit is reached with a single host's IPv4-MAC binding entry. No further IP entries are allowed in the table even though the MAC has a provision for 99 more IP addresses for *that* MAC.

- Global and policy-level limits

  The limits configured with the **device-tracking binding max-entries** command are at the global level, the limits configured with the **limit address-count** command in the device-tracking configuration mode are for a policy, which is at the interface or VLAN level.

  If a policy-level value *and* a globally configured value exists, the creation of binding entries is stopped when *a* limit is reached - this limit can be any one of the global values or the policy-level value.

  If only globally configured values exist, the creation of binding entries is stopped when *a* limit is reached.

  If only a policy-level value exists, the creation of binding entries is stopped when the policy-level limit is reached.

Tracking

The tracking parameter involves tracking of hosts in the network. In section #unique_68 unique_68_Connect_42_section_axm_sbk_ctb above, this is referred to as "polling". It also describes polling behaviour in detail.

To configure polling parameters at the global level, enter the **device-tracking tracking** command in global configuration mode. After you configure this command you still have the flexibility to turn polling on or off, for individual interfaces and VLANs. For this you must enable or disable polling in the policy.

To enable polling in a policy, enter the **tracking enable** keywords in the device-tracking configuration mode. By default, polling is disabled in a policy.

Guidelines for Policy Creation

- If multiple policies are available on a given target, a system-internal policy priority determines which policy takes precedence.

  A manually created policy has the highest priority. When you want to override the settings of a programmatically created policy, you can create a custom policy, so it has higher priority.

- The parameters of a programmatically created policy cannot be changed. You can configure certain attributes of a custom policy.

Guidelines for Applying a Policy

- Multiple policies can be attached to the same VLAN.
- If a programmatic policy is attached to a VLAN and you want to change policy settings, create a custom device-tracking policy and attach it to the VLAN.

- When multiple policies with different priorities are attached to the same VLAN, the settings of the policy with the highest priority are effective. The exceptions here are the limit address-count for IPv4 per mac and limit address-count for IPv6 per mac settings - the settings of the policy with the lowest priority are effective.
- When a device-tracking policy is attached to an interface under a VLAN, the policy settings on the interface take precedence over those on its VLAN; exceptions here are the values for limit address-count for IPv4 per mac and limit address-count for IPv6 per mac, which are aggregated from the policy on both the interface and VLAN.
- A policy cannot be removed unless the device tracking client feature configuration is removed.

How to Configure SISF

SISF or SISF-based device-tracking, is disabled by default. You enable it by defining a device-tracking policy and attaching the policy to a specific target. The target could be an interface or a VLAN. There are multiple ways to define a policy and no single method is a preferred or recommended one - use the option that suits your requirements.

Method of Enabling SISF	Applicable Configuration Tasks	Result
Option 1: Manually, by using interface configuration commands to create and apply the **default** policy to a target.	Applying the Default Device Tracking Policy to a Target, on page 62	Automatically applies the **default** device tracking policy to the specified target. The **default** policy is a built-in policy with default settings; you cannot change any of the attributes of the default policy. See **Option 2** if you want to configure device tracking policy attributes.
Option 2: Manually, by using global configuration commands to create a custom policy and applying the custom policy to a target.	1. Creating a Custom Device Tracking Policy with Custom Settings, on page 63 2. Attach the custom policy to an interface or VLAN: Attaching a Device Tracking Policy to an Interface, on page 66 OR Attaching a Device Tracking Policy to a VLAN, on page 67	Creates a custom policy with the name and policy parameters you configure, and attaches the policy to the specified target.

Method of Enabling SISF	Applicable Configuration Tasks	Result
Option 3: Programmatically, by configuring the snooping command.	Enter the **ip dhcp snooping vlan** *vlan* command in global configuration mode. Example: Programatically Enabling SISF by Configuring DHCP Snooping, on page 71	When you configure the command, the system automatically creates policy `DT-PROGRAMMATIC`. Use this method if you want to enable SISF-based device tracking for these clients: IEEE 802.1X, Web authentication, Cisco TrustSec, IP Source Guard, and SANET.
Option 4: By using an interface template	Using an Interface Template to Enable SISF, on page 68	By adding the policy to an interface template, you can apply the same policy to multiple targets, without having to create it separately for each target.
Option 5: Migrating from legacy IPDT and IPv6 Snooping.	Migrating from Legacy IPDT and IPv6 Snooping to SISF-Based Device-Tracking, on page 70	Convert legacy IPDT and IPv6 Snooping configuration to the SISF-based device-tracking commands.

Applying the Default Device Tracking Policy to a Target

Beginning in privileged EXEC mode, follow these steps to apply the default device tracking policy to an interface or VLAN:

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	Specify an interface or a VLAN
	• **interface** *type number*
	• **vlan configuration** *vlan_list*
	Example:
	`Device(config)# interface gigabitethernet 1/1`
	`OR`
	`Device(config)# vlan configuration 333`
		interface *type number*—Specifies the interface and enters interface configuration mode. The device tracking policy will be attached to the specified interface.
		vlan configuration *vlan_list*—Specifies the VLANs and enters VLAN feature configuration mode. The device tracking policy will be attached to the specified VLAN.

	Command or Action	Purpose

Step 4	**device-tracking**
	Example:
	`Device(config-if)# device-tracking`
	`OR`
	`Device(config-vlan-config)# device-tracking`
		Enables SISF-based device tracking and attaches the default policy it to the interface or VLAN.
		The default policy is a built-in policy with default settings; none of the attributes of the default policy can be changed.

Step 5	**end**
	Example:
	`Device(config-if)# end`
	`OR`
	`Device(config-vlan-config)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.
		Exits VLAN feature configuration mode and returns to privileged EXEC mode.

Step 6	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy default`	Displays device-tracking policy configuration, and all the targets it is applied to.

Creating a Custom Device Tracking Policy with Custom Settings

Beginning in privileged EXEC mode, follow these steps to create and configure a device tracking policy:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	[**no**] **device-tracking policy** *policy-name* **Example:** `Device(config)# device-tracking policy example_policy`	Creates the policy and enters device-tracking configuration mode.
Step 3	[**data-glean** | **default** | **destination-glean** | **device-role** | **distribution-switch** | **exit** | **limit** | **no** | **prefix-glean** | **protocol** | **security-level** | **tracking** | **trusted-port** | **vpc**] **Example:** `Device(config-device-tracking)# destination-glean log-only`	Enter the question mark (?) at the system prompt to obtain a list of available options in this mode. You can configure the following for both IPv4 and IPv6: • (Optional) **data-glean**—Enables learning of addresses from a data packet snooped from a source inside the network and populates the binding table with the data traffic source address. Enter one of these options:

	Command or Action	Purpose
		• log-only—Generates a syslog message upon data packet notification • recovery—Uses a protocol to enable binding table recovery. Enter **NDP** or **DHCP**. • (Optional) **default**—Sets the policy attribute to its default value. You can set these policy attributes to their default values: **data-glean**, **destination-glean**, **device-role**, **limit**, **prefix-glean**, **protocol**, **security-level**, **tracking**, **trusted-port**. • (Optional) **destination-glean**—Populates the binding table by gleaning data traffic destination address. Enter one of these options: • **log-only**—Generates a syslog message upon data packet notification • **recovery**—Uses a protocol to enable binding table recovery. Enter **DHCP**. • (Optional) **device-role**—Sets the role of the device attached to the port. It can be a node or a switch. Enter one of these options: • node—Configures the attached device as a node. This is the default option. • switch—Configures the attached device as a switch. • (Optional) **distribution-switch**—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect. • **exit**—Exits the device-tracking policy configuration mode. • **limit** address-count—Specifies an address count limit per port. The range is 1 to 32000. • **no**—Negates the command or sets it to defaults. • (Optional) **prefix-glean**—Enables learning of prefixes from either IPv6 Router

	Command or Action	Purpose
		Advertisements or from DHCP-PD. You have the following option: • (Optional) **only**—Gleans only prefixes and not host addresses. • (Optional) **protocol**—Sets the protocol to glean; by default, all are gleaned. Enter one of these options: • **arp** [**prefix-list** *name*]: Gleans addresses in ARP packets. Optionally, enter the name of prefix-list that is to be matched. • **dhcp4** [**prefix-list** *name*]: Glean addresses in DHCPv4 packets. Optionally, enter the name of prefix-list that is to be matched. • **dhcp6** [**prefix-list** *name*]: Glean addresses in DHCPv6 packets. Optionally, enter the name of prefix-list that is to be matched. • **ndp** [**prefix-list** *name*]: Glean addresses in NDP packets. Optionally, enter the name of prefix-list that is to be matched. • (Optional) **security-level**—Specifies the level of security enforced by the feature. Enter one of these options: • **glean**—Gleans addresses passively. • **guard**—Inspects and drops un-authorized messages. This is the default. • **inspect**—Gleans and validates messages. • (Optional) **tracking**—Specifies a tracking option. Enter one of these options: • **disable** [**stale-lifetime** [*1-86400-seconds* | **infinite**]]—Turns of device-tracking. Optionally, you can enter the duration for which the entry is kept inactive before deletion, or keep it permanently inactive.

	Command or Action	Purpose
		• **enable** [**reachable-lifetime** [*1-86400-seconds* | **infinite**]] —Turns on device-tracking. Optionally, you can enter the duration for which the entry is kept reachable, or keep it permanently reachable. • (Optional) **trusted-port**—Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table. • (Optional) **vpc**—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
Step 4	**end** **Example:** `Device(config-device-tracking)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 5	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy example_policy`	Displays the device-tracking policy configuration.

What to do next

Attach the policy to an interface or VLAN.

Attaching a Device Tracking Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach a device tracking policy to an interface:

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface* **Example:** `Device(config-if)# interface gigabitethernet 1/1`	Specifies an interface and enters interface configuration mode.
Step 4	**device-tracking attach-policy** *policy name* **Example:** `Device(config-if)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the interface. Device tracking is also supported on EtherChannels. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show device-tracking policies** [**interface** *interface*] **Example:** `Device# show device-tracking policies interface gigabitethernet 1/1`	Displays policies that match the specified interface type and number.

Attaching a Device Tracking Policy to a VLAN

Beginning in privileged EXEC mode, follow these steps to attach a device-tracking policy to VLANs across multiple interfaces:

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 333`	Specifies the VLANs to which the device tracking policy will be attached; enters the VLAN interface configuration mode.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-vlan-config)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the specified VLANs across all switch interfaces. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 5	**do show device-tracking policies vlan** *vlan-ID* **Example:** `Device(config-vlan-config)# do show device-tracking policies vlan 333`	Verifies that the policy is attached to the specified VLAN, without exiting the VLAN interface configuration mode.
Step 6	**end** **Example:** `Device(config-vlan-config)# end`	Exits VLAN feature configuration mode and returns to privileged EXEC mode.

Using an Interface Template to Enable SISF

An interface template is a container of configurations or policies. When you apply the interface template to a target, all the configurations are applied to the target. This enables you to configure multiple commands or features on one or more targets.

You can add the **device-tracking policy** *policy_name* global configuration command to an interface template. SISF-based device-tracking is enabled and the policy is applied, wherever the template is applied.

You can also apply the template through 802.1x authentication. During the 802.1x authentication process, you can dynamically assign different templates (and therefore different policies) to different interfaces.

Note You can apply only one interface template to one port.

Before you begin

You have already created a custom policy. See Creating a Custom Device Tracking Policy with Custom Settings, on page 63.

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	**template interface** *template_name* **Example:** `Device(config)# template interface template_w_sisf`	Creates a template with the name you specify and enters template configuration mode. In the accompanying example, a template called "template_w_sisf" is created.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device (config-template)# device-tracking attach-policy sisf_policy_for_template`	Attaches a policy to the template. SISF-based device-tracking is enabled and the policy is applied wherever the template is applied.
Step 5	**exit** **Example:** `Device (config-template)# exit`	Exits the template configuration mode and enters the global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1`	Specifies an interface and enters interface configuration mode.
Step 7	**source template** *template_name* **Example:** `Device(config-if)# source template template_w_sisf`	Configures a static binding for an interface template. In the accompanying example, "template_w_sisf" is statically applied to an interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters the privileged EXEC mode.

Step 9	**show running-config interface** *type number*
	Example:
	`Device# show running-config interface gigabitethernet 1/1`
	`Building configuration...`
	`<output truncated>`
		Displays the contents of the running configuration.

	Command or Action	Purpose
	Current configuration : 71 bytes ! interface GigabitEthernet1/1 source template template_w_sisf end <output truncated>	

Migrating from Legacy IPDT and IPv6 Snooping to SISF-Based Device-Tracking

Based on the legacy configuration that exists on your device, the **device-tracking upgrade-cli** global configuration command upgrades your CLI differently. Consider the following configuration scenarios and the corresponding migration results before you migrate your existing configuration.

Note You cannot configure a mix of the old IPDT and IPv6 Snooping commands with the SISF-based device-tracking commands.

Only IPDT Configuration Exists

If your device has only IPDT configuration, running the **device-tracking upgrade-cli** command converts the configuration to use a SISF policy that is created and attached to the interface. You can then update this SISF policy.

If you continue to use the legacy commands, this restricts you to operate in a legacy mode where only the legacy IPDT and IPv6 Snooping commands are available on the device.

Only IPv6 Snooping Configuration Exists

On a device with existing IPv6 Snooping configuration, the old IPv6 Snooping commands are available for further configuration. The following options are available:

- (Recommended) Use the **device-tracking upgrade-cli** command to convert all your legacy configuration to the SISF-based device-tracking commands. After conversion, only the SISF-based device-tracking commands will work on your device.
- Use the legacy IPv6 Snooping commands for your future configuration and do not run the **device-tracking upgrade-cli** command. With this option, only the legacy IPv6 Snooping commands are available on your device, and you cannot use the SISF-based device-tracking commands.

Both IPDT and IPv6 Snooping Configuration Exist

On a device that has both legacy IPDT configuration and IPv6 Snooping configuration, you can convert legacy commands to the SISF-based device-tracking commands. However, note that only one snooping policy can be attached to an interface, and the IPv6 Snooping policy parameters override the IPDT settings.

Note If you do not migrate to the SISF-based device-tracking commands and continue to use the legacy IPv6 Snooping or IPDT commands, your IPv4 device-tracking configuration information may be displayed in the IPv6 Snooping commands, as the SISF-based device-tracking feature handles both IPv4 and IPv6 configuration. To avoid this, we recommend that you convert your legacy configuration to SISF-based device-tracking commands.

No IPDT or IPv6 Snooping Configuration Exists

If your device has no legacy IP Device Tracking or IPv6 Snooping configurations, you can use only the SISF-based device-tracking commands for all your future configuration. The legacy IPDT commands and IPv6 Snooping commands are not available.

Configuration Examples for SISF

Example: Programatically Enabling SISF by Configuring DHCP Snooping

The following example shows how to configure the **ip dhcp snooping vlan** *vlan* command in global configuration mode to enable SISF-based device-tracking. When you enable SISF this way, the system creates the `DT-PROGRMMATIC` policy.

Enter the **show device-tracking policy** *policy_name* command in privileged EXEC mode, to display the settings for a `DT-PROGRMMATIC` policy.

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp snooping vlan 10
Device(config)# end
```

```
Device# show device-tracking policy DT-PROGRAMMATIC

Policy DT-PROGRAMMATIC configuration:
  security-level glean (*)
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  limit address-count for IPv4 per mac 1 (*)
  tracking enable
Policy DT-PROGRAMMATIC is applied on the following targets:
Target      Type    Policy              Feature            Target range
vlan 10     VLAN    DT-PROGRAMMATIC     Device-tracking    vlan all


  note:
  Binding entry Down timer: 24 hours (*)
  Binding entry Stale timer:   24 hours (*)
```

Example: Mitigating the IPv4 Duplicate Address Problem

This example shows how you can tackle the `Duplicate IP Address 0.0.0.0 error message` problem encountered by clients that run Microsoft Windows:

Configure the **device-tracking tracking auto-source** command in global configuration mode. This command determines the source IP and MAC address used in the ARP probe sent by the switch to probe a client, in order to maintain its entry in the device-tracking table. The purpose, is to avoid using 0.0.0.0 as source IP address.

Note Configure the **device-tracking tracking auto-source** command when a switch virtual interface (SVI) is not configured. You do not have to configure it when a SVI is configured with an IPv4 address on the VLAN.

Command	Action **(In order to select source IP and MAC address for device tracking ARP probe)**	Notes
device-tracking tracking auto-source global configuration command.	• Set source to VLAN SVI if present. • Look for IP and MAC binding in device-tracking table from same subnet. • Use 0.0.0.0	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping.
device-tracking tracking auto-source override global configuration command.	• Set source to VLAN SVI if present • Use 0.0.0.0	Not recommended when there is no SVI.
device-tracking tracking auto-source fallback 0.0.0.X 255.255.255.0 global configuration command.	• Set source to VLAN SVI if present. • Look for IP and MAC binding in device-tracking table from same subnet. • Compute source IP from client IP using host bit and mask provided. Source MAC is taken from the MAC address of the switchport facing the client*.	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping. The computed IPv4 address must not be assigned to any client or network device.

Command	Action (In order to select source IP and MAC address for device tracking ARP probe)	Notes
device-tracking tracking auto-source fallback 0.0.0.X 255.255.255.0 override global configuration command.	• Set source to VLAN SVI if present. Compute source IP from client IP using host bit and mask provided*. Source MAC is taken from the MAC address of the switchport facing the client*.	-

* Depending on the client IP address, an IPv4 address has to be reserved for the source IP.

A reserved source IPv4 address = (host-ip and mask) | client-ip

- Client IP = 192.0.2.25
- Source IP = (192.0.2.25 and 255.255.255.0) | (0.0.0.1) = 192.0.2.1

IP address 192.0.2.1 should not be assigned to any client or network device.

Example: Disabling IPv6 Device Tracking on a Target

By default, SISF-based device-tracking supports both IPv4 and IPv6. The following configuration examples show how you can disable IPv6 device-tracking where supported.

To disable device-tracking for IPv6, when a *custom* policy is attached to a target (all releases):

```
Device(config)# device-tracking policy example-policy
 Device(config-device-tracking)# no protocol ndp
 Device(config-device-tracking)# no protocol dhcp6
 Device(config-device-tracking)# end
```

Example: Enabling IPv6 for SVI on VLAN (To Mitigate the Duplicate Address Problem)

When IPv6 is enabled in the network and a switched virtual interface (SVI) is configured on a VLAN, we recommend that you add the following to the SVI configuration. This enables the SVI to acquire a link-local address automatically; this address is used as the source IP address of the SISF probe, thus preventing the duplicate IP address issue.

```
Device> enable
Device# configure terminal
Device(config)# interface vlan 10
Device(config-if)# ipv6 enable
Device(config-if)# end
```

Example: Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port

In a multi-switch network, SISF-based device tracking provides the capability to distribute binding table entries between switches running the feature. Binding entries are only created on the switches where the host appears on an access port. No entry is created for a host that appears over a trunk port. This is achieved by configuring a policy with the **trusted-port** and **device-role switch** options, and attaching it to the trunk port.

Note Both, the **trusted-port**, and **device-role switch** options, must be configured in the policy.

Further, we recommended that you apply such a policy on a port facing a device, which also has SISF-based device tracking enabled.

```
Device> enable
Device# configure terminal
Device(config)# device-tracking policy example_trusted_policy
Device(config-device-tracking)# device-role switch
Device(config-device-tracking)# trusted-port
Device(config-device-tracking)# exit
Device(config)# interface gigabitethernet 1/1
Device(config-if)# device-tracking attach-policy  example_trusted_policy
Device(config-if)# end
```

Example: Avoiding a Short Device-Tracking Binding Reachable Time

When migrating from an older release, the following configuration may be present:

```
device-tracking binding reachable-lifetime 10
```

Remove this by entering the **no** version of the command.

```
Device> enable
Device# configure terminal
Device(config)# no device-tracking binding reachable-lifetime 10
Device(config)# end
```

Example: Detecting and Preventing Spoofing

Address spoofing, is a man-in-the-middle attack that allows an attacker to intercept communication between network devices. These attacks attempt to divert traffic from its originally intended host to the attacker instead. For example, attacks are carried out by sending unsolicited Address Resolution Protocol (ARP) replies or with IPv6 Neighbor Advertisements carrying a mapping that is different from the legitimate one, such as <IPTARGET, MACTHIEF>. When the IPTARGET is of the default gateway, all traffic that is meant to leave the subnet is routed to the attacker.

The following example shows the required SISF configuration to enable the system to detect and prevent spoofing. It also shows the system messages that are logged when spoofing is detected, and the action that the system takes. It includes an excerpt of LISP configuration in an SDA setup for example purposes only. Actual LISP configuration may involve additional configuration.

Sample LISP configuration:

```
instance-id 100
  service ethernet
   eid-table vlan 100               <<< triggers creation of programmatic policy
"LISP-DT-GUARD-VLAN"
   database-mapping mac locator-set XTR11
   exit-service-ethernet
  !
  exit-instance-id
```

Settings of the programmatic policy:

```
Device# show device-tracking policy LISP-DT-GUARD-VLAN
Device-tracking policy LISP-DT-GUARD-VLAN configuration:
  security-level guard              <<< enables the detection and prevention of IPv4 and
IPv6 spoofing
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  limit address-count for IPv4 per mac 21
  limit address-count for IPv6 per mac 58
  origin fabric
  tracking enable reachable-lifetime 240
```

The following device-tracking counters show you that packet drops have occurred. However, the drops may be caused by reasons other than address spoofing as well. Use the information in the counters along with system messages to ascertain if spoofing has occurred.

```
Device# show device-tracking counters vlan 11
Received messages on vlan 11   :
Protocol        Protocol message
NDP             RS[4] RA[4] NS[1777] NA[2685]
DHCPv6
ARP             REQ[12] REP[1012]
DHCPv4
ACD&DAD         --[8]
:
Dropped messages on vlan 10   :
Feature             Protocol Msg [Total dropped]
Device-tracking:    ARP      REQ [23]
                    reason:  Packet accepted but not forwarded [23]
                             REP [450]
                    reason:  Silent drop [445]
                    reason:  Packet accepted but not forwarded [5] :
```

Required configuration to display system messages:

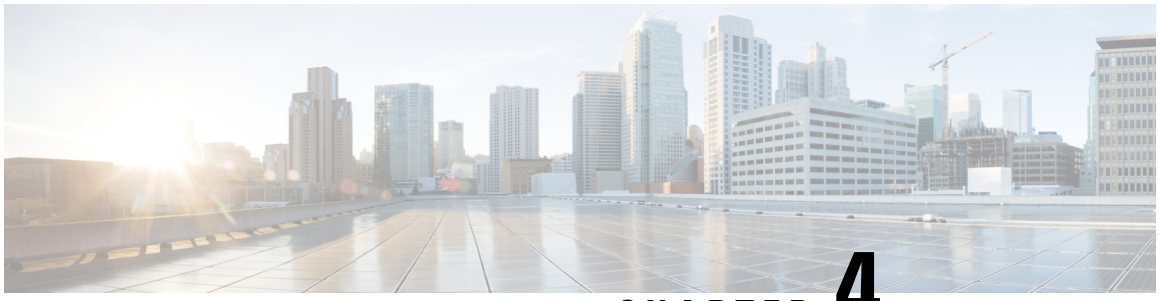
```
Device# device-tracking logging theft
Device# device-tracking logging packet drop
```

While the packet drops in the device-tracking counters do not conclusively prove that spoofing has occurred, the system messages help you ascertain this.

```
%SISF-4-IP_THEFT: IP Theft IP=3001::5 VLAN=10 Cand-MAC=aabb.cc00.6600 Cand-I/F=Et0/0 Known
 MAC aabb.cc00.6900 Known I/F Et0/1
%SISF-4-IP_THEFT: IP Theft IP=FE80::A8BB:CCFF:FE00:6900 VLAN=10 Cand-MAC=aabb.cc00.6600
Cand-I/F=Et0/0 Known MAC aabb.cc00.6900 Known I/F Et0/1
```

In the log, verified binding information (IP, MAC address, interface or VLAN) is preceded by the term "Known" . A suspicious IP address and MAC address is preceded by the term "New" or "Cand". Interface and VLAN information is also provided along with the suspicious IP or MAC address - this helps you identify where the suspicious traffic was seen.

For more information about how to interpret these system messages, in the command reference of the corresponding release, see the usage guidelines of the **device-tracking logging** command.



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.

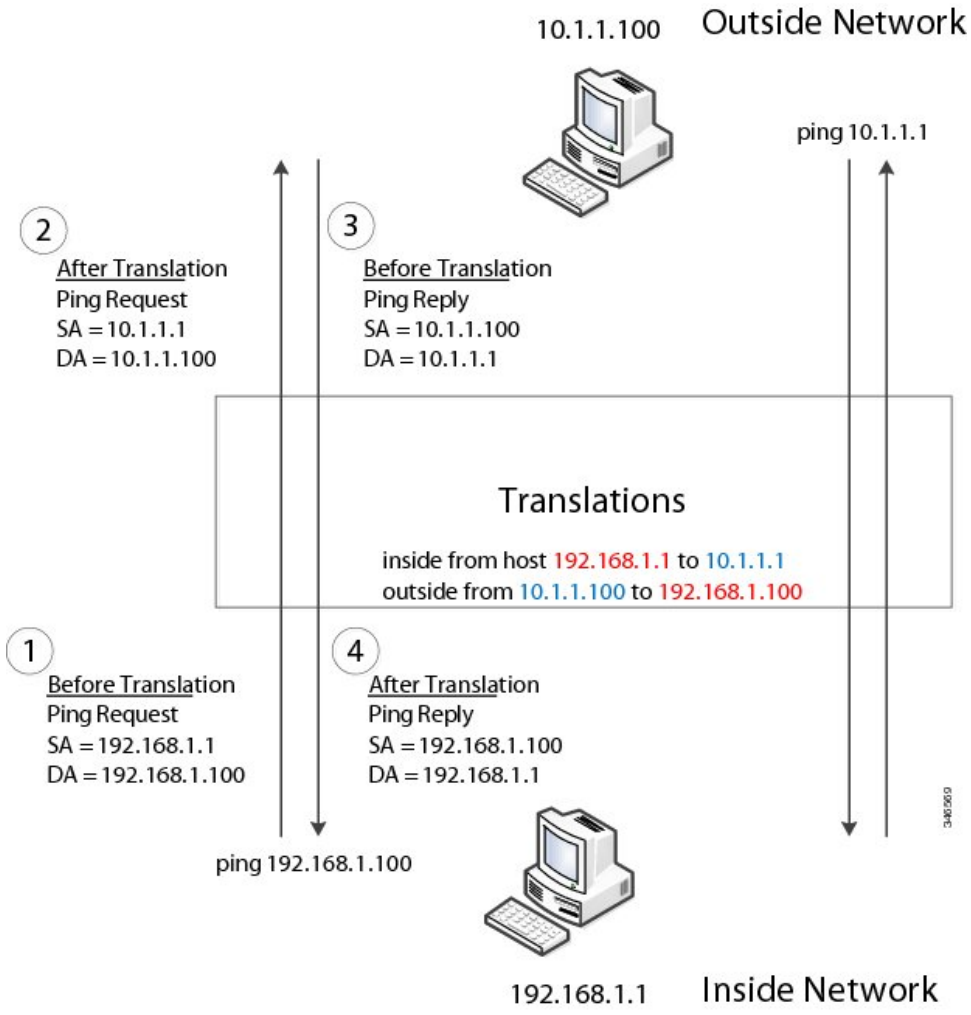
Layer 2 NAT uses a table to translate IPv4 addresses both public-to-private, and private-to-public at line rate. Layer 2 NAT is a hardware-based implementation that provides the same high level of (bump-on-the-wire) wire-speed performance. This implementation also supports multiple VLANs through the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

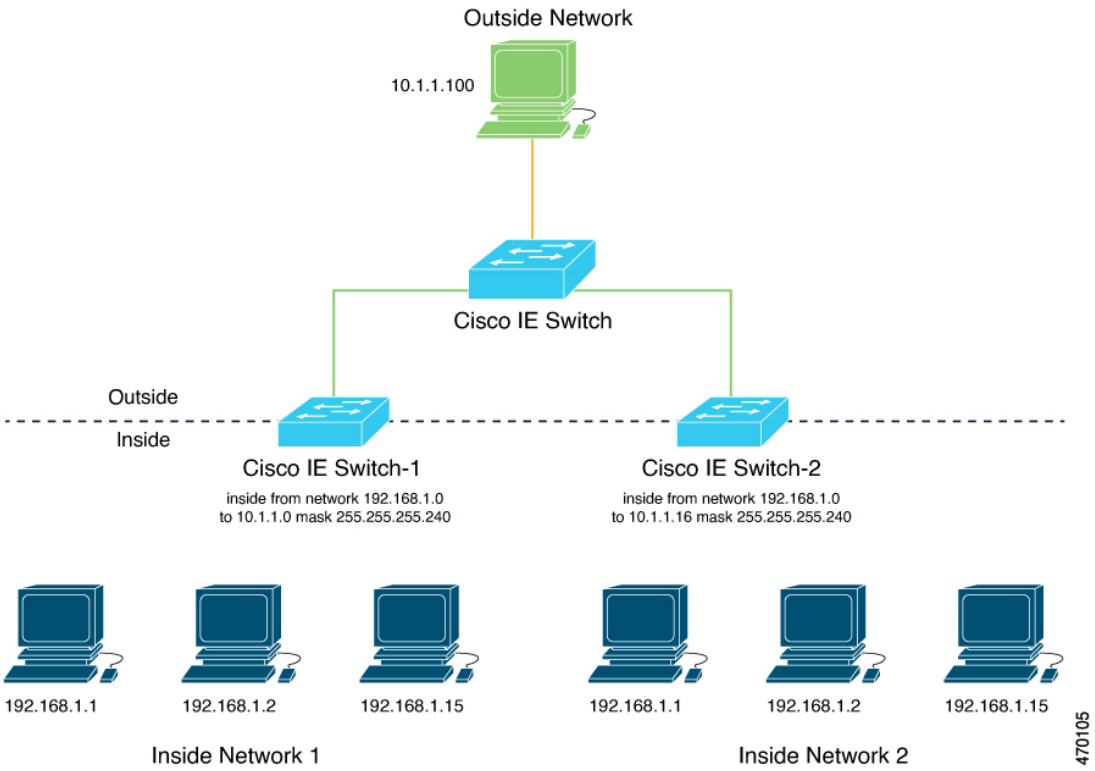
5. When the packet is received on the internal network, Layer 2 NAT translates the source address to 192.168.1.100 and the destination address to 192.168.1.1.

Figure 14: 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 15: Inside-Outside Address Translation



The following figure shows a Cisco Catalyst IE3100 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.

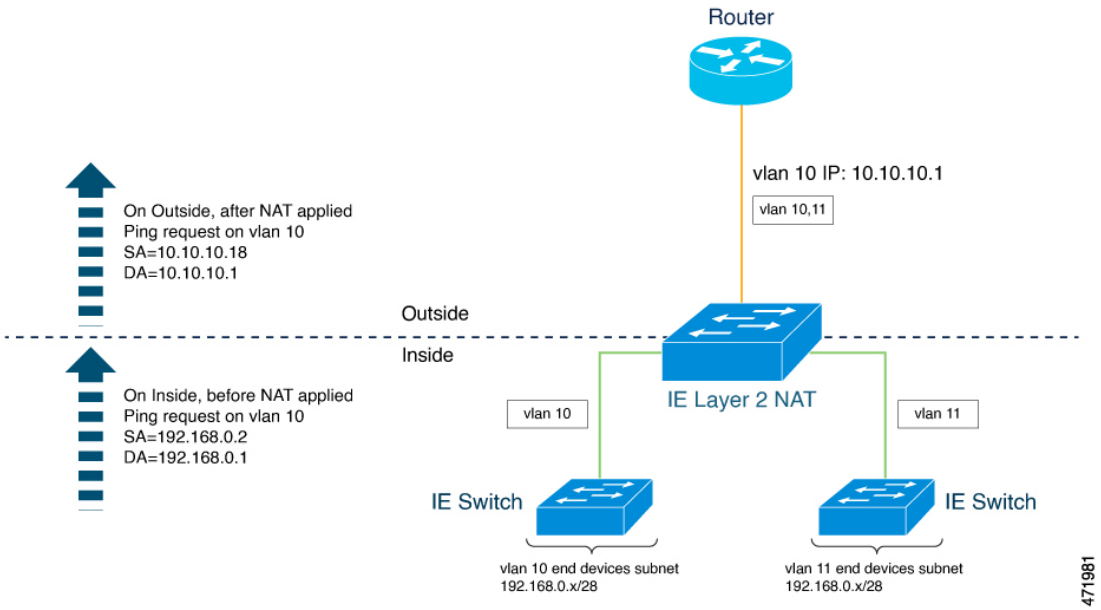
The L2NAT instance definitions use the `network` command to define a translation row for multiple devices in the same subnet. In this the case it's a /28 subnet. With this subnet mask, the last nibble in 4th byte of the inside IP address will not change. The last byte will be in range 16 – 31 because 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 16: NAT on the IE3100 Switch



The IE3100 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.

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 84).

To configure Layer 2 NAT, follow these steps. Refer to the examples in Basic Inside-to-Outside Communications Example, on page 85 and Duplicate IP Addresses Example, on page 86 for more details.

Procedure

Step 1 Enter global configuration mode:

configure terminal

Step 2 Create a new Layer 2 NAT instance:

l2nat instance *instance_name*

After creating an instance, you use this same command to enter the 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 IE3100):

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

Configure Layer 2 NAT support on Port Channel

Note Layer 2 NAT is supported on logical interface of port-channel but not on member interface.

The LACP is defined in IEEE 802.3ad and enables Cisco devices to manage port-channels between devices that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of port-channels by exchanging LACP packets between Ethernet ports.

By using LACP, the switch or switch stack learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an port-channels, LACP adds the group to the spanning tree as a single device port.

LACP modes specify whether a port can send LACP packets or only receive LACP packets.

Active mode: Places a port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.

Passive mode: Places a port into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation. This setting minimizes the transmission of LACP packets.

Both the active and passive LACP modes enable ports to negotiate with partner ports to a port-channel based on criteria such as port speed, and for Layer 2 EtherChannels, based on trunk state and VLAN numbers.

When you specify the maximum number of bundled LACP ports allowed in a port channel, the remaining ports in the port channel are designated as hot-standby ports. Beginning in privileged EXEC mode, follow these steps to configure the maximum number of LACP ports in a port-channel. This procedure is optional.

Procedure

Step 1 Enter global configuration mode:

device configure

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

device # l2nat instance A-LC

Step 3 Translate A1's inside address to an outside address:

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

Step 4 Translate A2's inside address to an outside address:

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

Step 5 Translate A3's inside address to an outside address:

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

Step 6 Translate LC's outside address to an inside address:

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

Step 7 Exit config-l2nat mode:

Device(config-l2nat)# exit

Step 8 Access interface configuration mode for the port channel:

Device(config)# interface port-channel

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

Device(config-if)#l2nat A-LC

Note
For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:
`l2nat instance vlan`

Step 10 Return to privileged EXEC mode:

Device# end

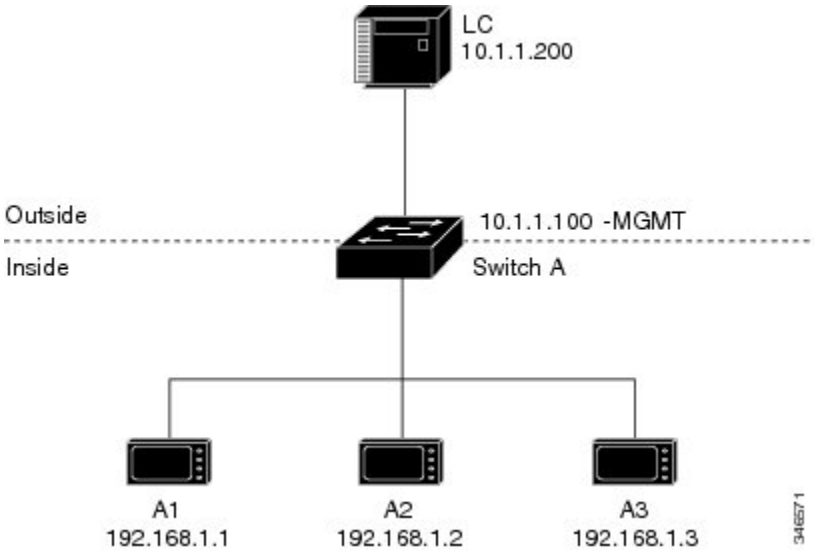
Verifying Configuration

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 17: 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

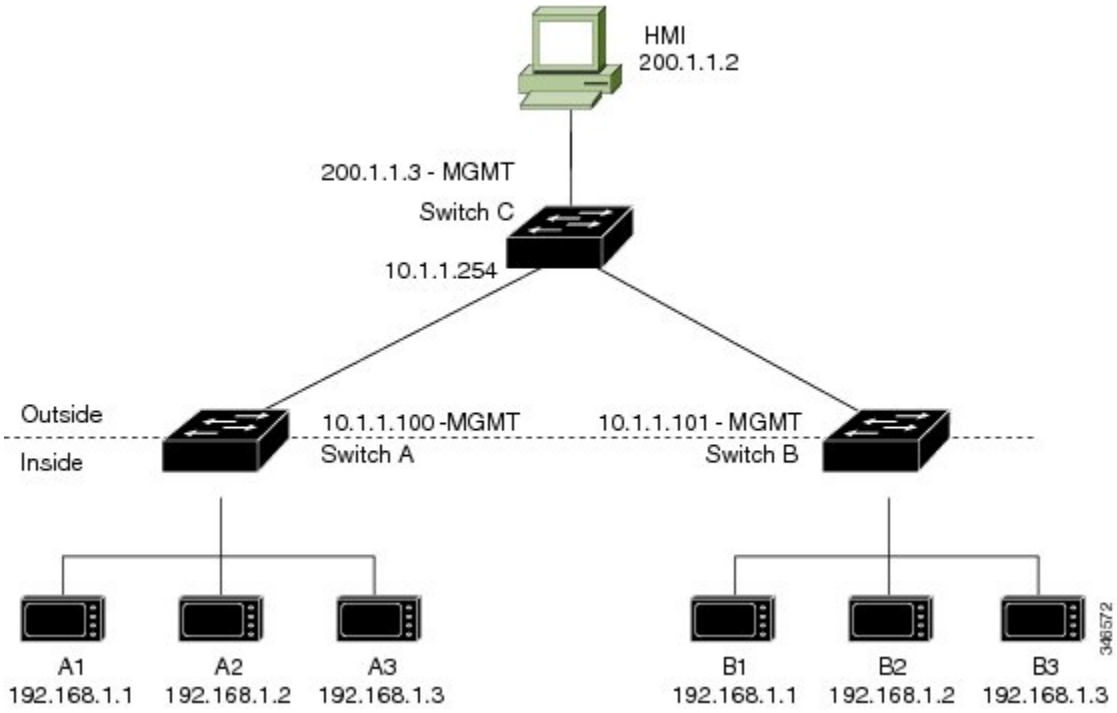
	Command	Purpose
1.	Switch# **configure**	Enters global configuration mode.

	Command	Purpose
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.
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 18: 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 88 shows the configuration tasks for Switch A. Table 12: Configuration of Switch B for Subnet Example, on page 89 shows the configuration tasks for Switch B.

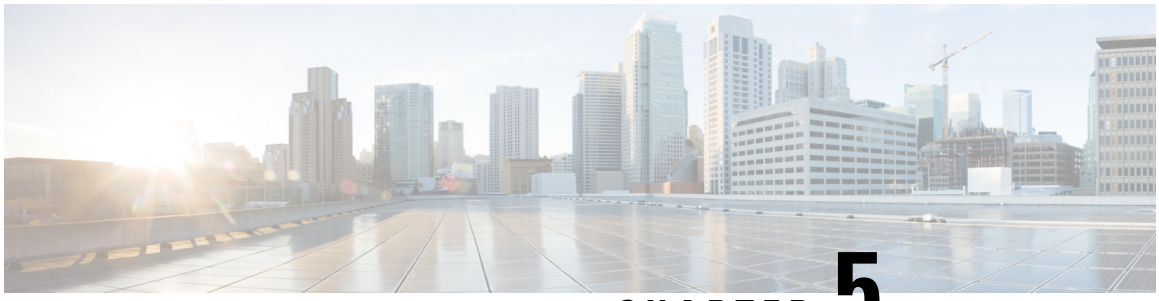
Note This example is based on the IE 2000 switch. For the IE3x00 and ESS3300 switches, the interface numbers may vary.

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

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 Signaling 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.

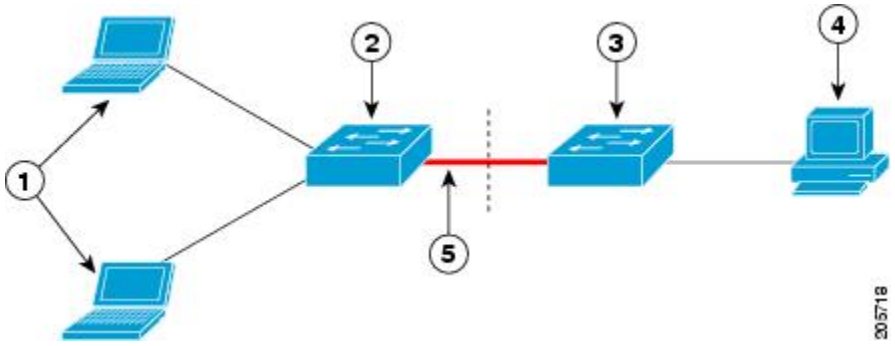
You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. Multihost mode is not supported on the authenticator switch interface.

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 19: 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:

Procedure

	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	**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:** `Device(config-if)# dot1x pae authenticator`	Configures the interface as a port access entity (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.

	Command or Action	Purpose
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:

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	**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.

	Command or Action	Purpose
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.
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 mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Configures the interface as a VLAN trunk port.
Step 13	**dot1x pae supplicant** **Example:** `Device(config-if)# dot1x pae supplicant`	Configures the interface as a port access entity (PAE) supplicant.
Step 14	**dot1x credentials** *profile-name* **Example:** `Device(config-if)# dot1x credentials`	Attaches the 802.1x credentials profile to the interface.

	Command or Action	Purpose
	test	
Step 15	**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 16	**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 Signaling 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 Signaling 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
```

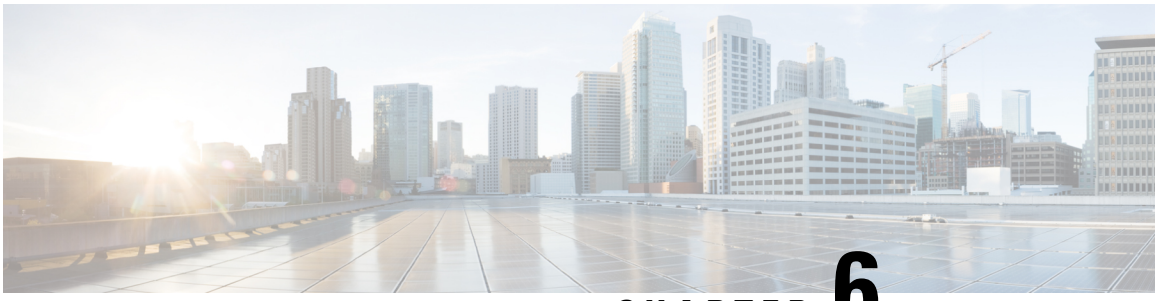
```
aaa authorization network default group radius
aaa server radius dynamic-author
client <ISE-IP> server-key cisco123
dot1x system-auth-control
policy-map type control subscriber Policy_dot1x
event session-started match-all
10 class always do-until-failure
10 authenticate using dot1x
exit

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 6

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 categorized 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.

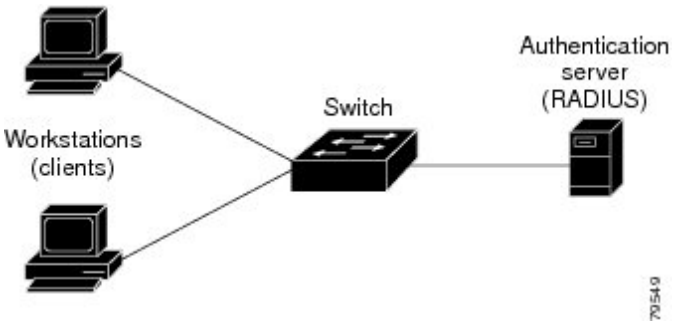
Device Roles

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.

This figure shows the roles of these devices in a network.

Figure 20: Web-Based Authentication Device Roles



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:

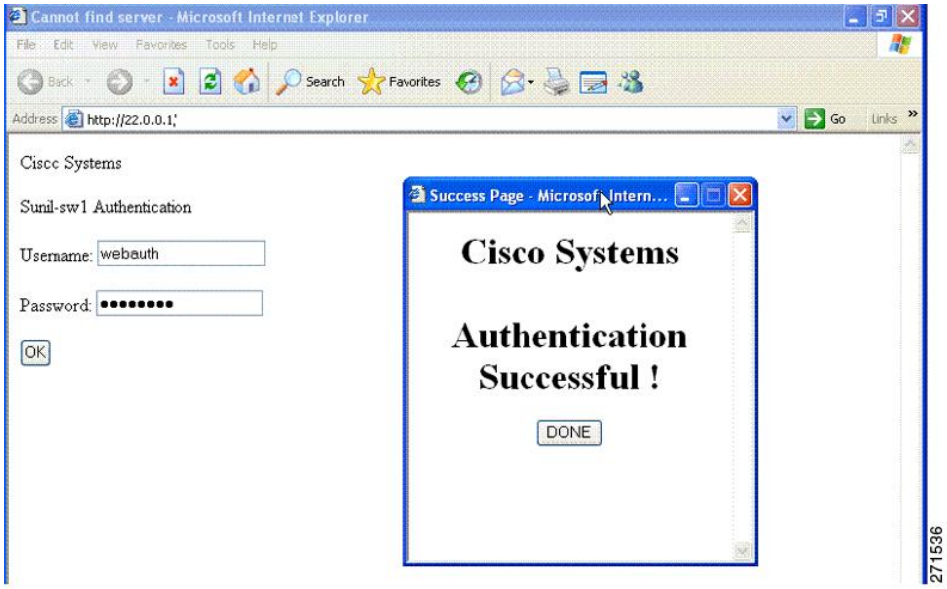
- *Authentication Successful*
- *Authentication Failed*
- *Authentication Expired*

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.

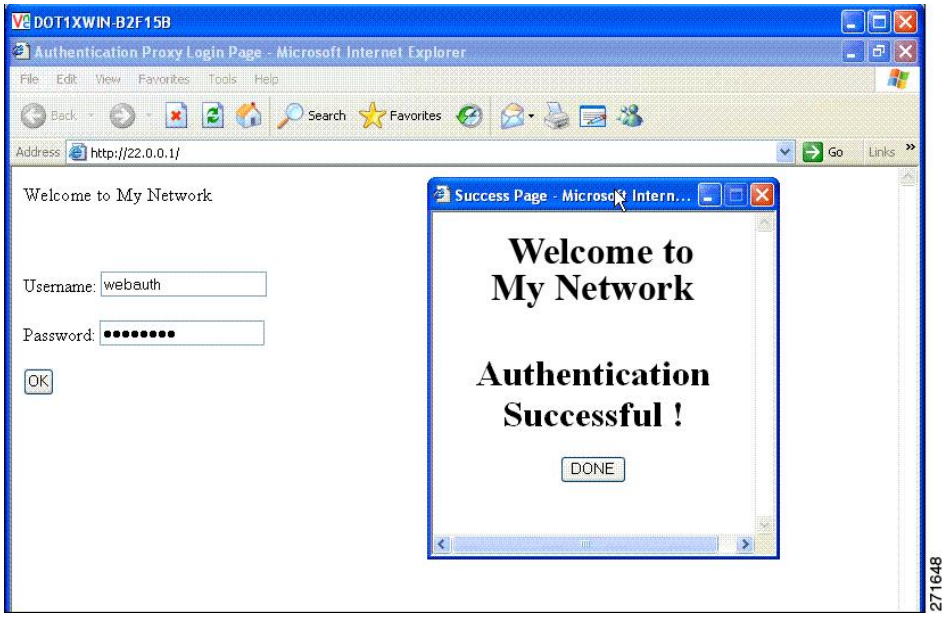
Figure 21: Authentication Successful Banner



The banner can be customized as follows:

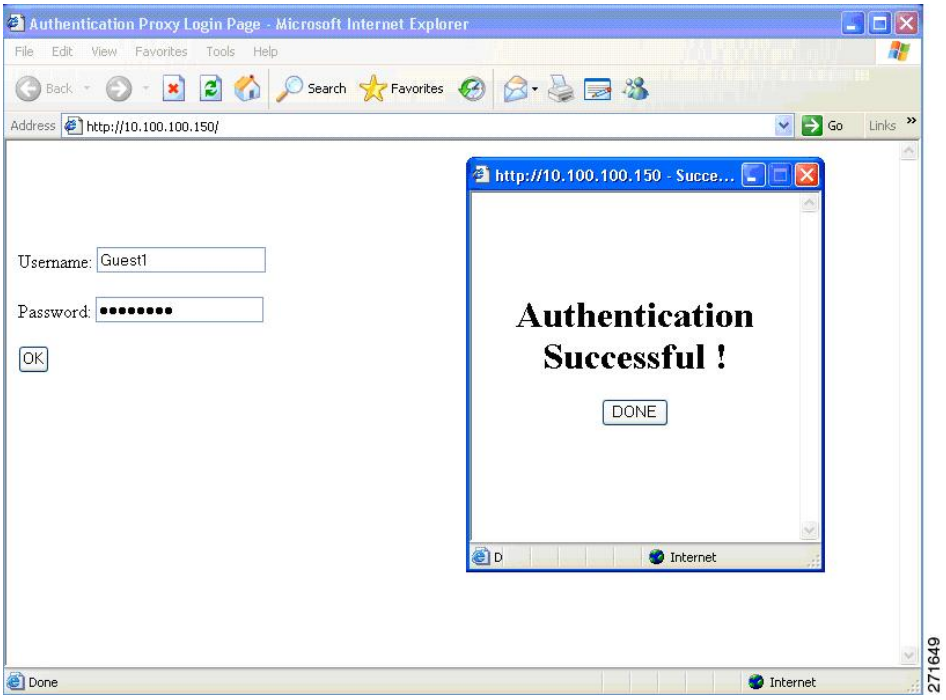
- Add a message, such as switch, router, or company name to the banner:
  - Legacy mode—Use the **ip admission auth-proxy-banner http** *banner-text* global configuration command.
  - New-style mode—Use the **parameter-map type webauth global banner** global configuration command.
- Add a logo or text 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 22: 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 23: 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:

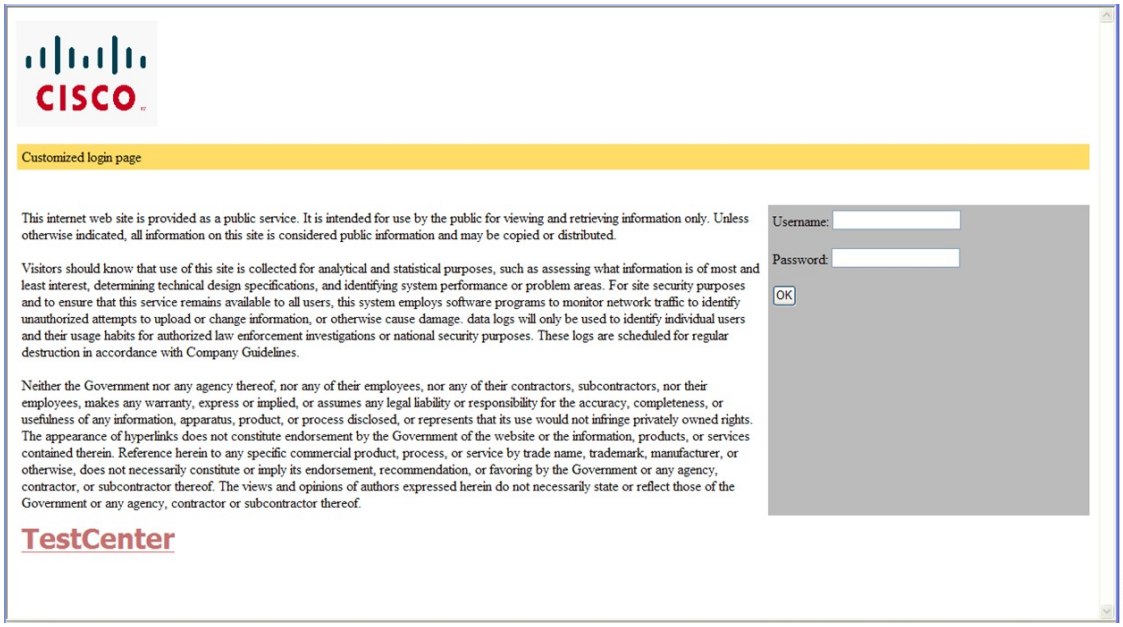
- Login—Your credentials are requested.
- Success—The login was successful.
- Fail—The login failed.
- Expire—The login session has expired because of excessive login failures.

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.
- 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 24: 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 13: 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.

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 *Configuring SISF-Based Tracking*.

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	**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 1/1`	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.
Step 5	**ip access-group** *name* **Example:** `Device(config-if)# ip access-group webauthag`	Applies the default ACL.
Step 6	**ip admission name** **Example:**	Configures an authentication rule for web-based authorization for the interface.

	Command or Action	Purpose
	`Device(config)# ip admission name`	
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:

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.

	Command or Action	Purpose
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.
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:

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	**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.

	Command or Action	Purpose
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 pseudo-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:

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.

	Command or Action	Purpose
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.
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.

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.

	Command or Action	Purpose
Step 3	**ip admission proxy http login page file** *device:login-filename* **Example:** `Device(config)# ip admission proxy http login page file disk1:login.htm`	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.
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:

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	**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.

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.

	Command or Action	Purpose
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:

Procedure

	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.
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 14: 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