Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication in Cisco IOS Release 12.2SX to prevent unauthorized devices (clients) from gaining access to the network.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

This chapter consists of these sections:

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client and server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port and assigns the port to a VLAN before making available any services offered by the switch or the LAN.

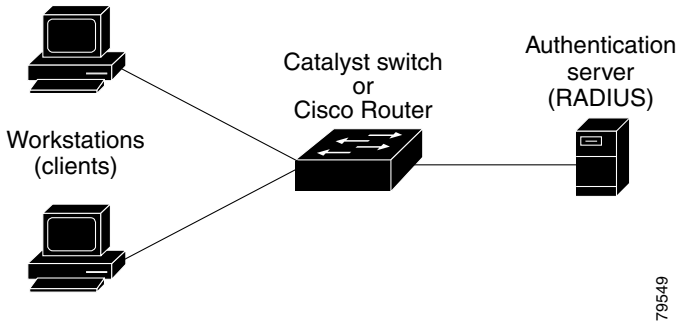
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

These sections describe the role of 802.1X port-based authentication as a part of a system of authentication, authorization, and accounting (AAA):

Understanding 802.1X Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 60-1.

Figure 60-1 802.1X Device Roles



The specific roles shown in Figure 60-1 are as follows:

- *Client*—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch.The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1X specification.)

  Note To resolve Windows XP network connectivity and 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL: http://support.microsoft.com/kb/q303597/

- *Authentication server*—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. The Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server (ACS), version 3.0. RADIUS uses a client-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- *Switch* (also called the *authenticator* and *back-end authenticator*)—With Release 12.2(33)SXH and later releases, 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. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

  When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

Understanding the Port-based Authentication Process

When 802.1X port-based authentication is enabled, these events occur:

- If the client supports 802.1X-compliant client software and the client's identity is valid, the 802.1X authentication succeeds and the switch grants the client access to the network.
- If 802.1X authentication times out while waiting for an EAPOL message exchange, the switch can use a fallback authentication method, such as MAC authentication bypass (MAB) or web-based authentication (webauth), if either or both are enabled:
  - If MAC authentication bypass is enabled, the switch relays the client's MAC address to the AAA server for authorization. If the client's MAC address is valid, the authorization succeeds and the switch grants the client access to the network.
  - If web-based authentication is enabled, the switch sends an HTTP login page to the client. The switch relays the client's username and password to the AAA server for authorization. If the login succeeds, the switch grants the client access to the network.

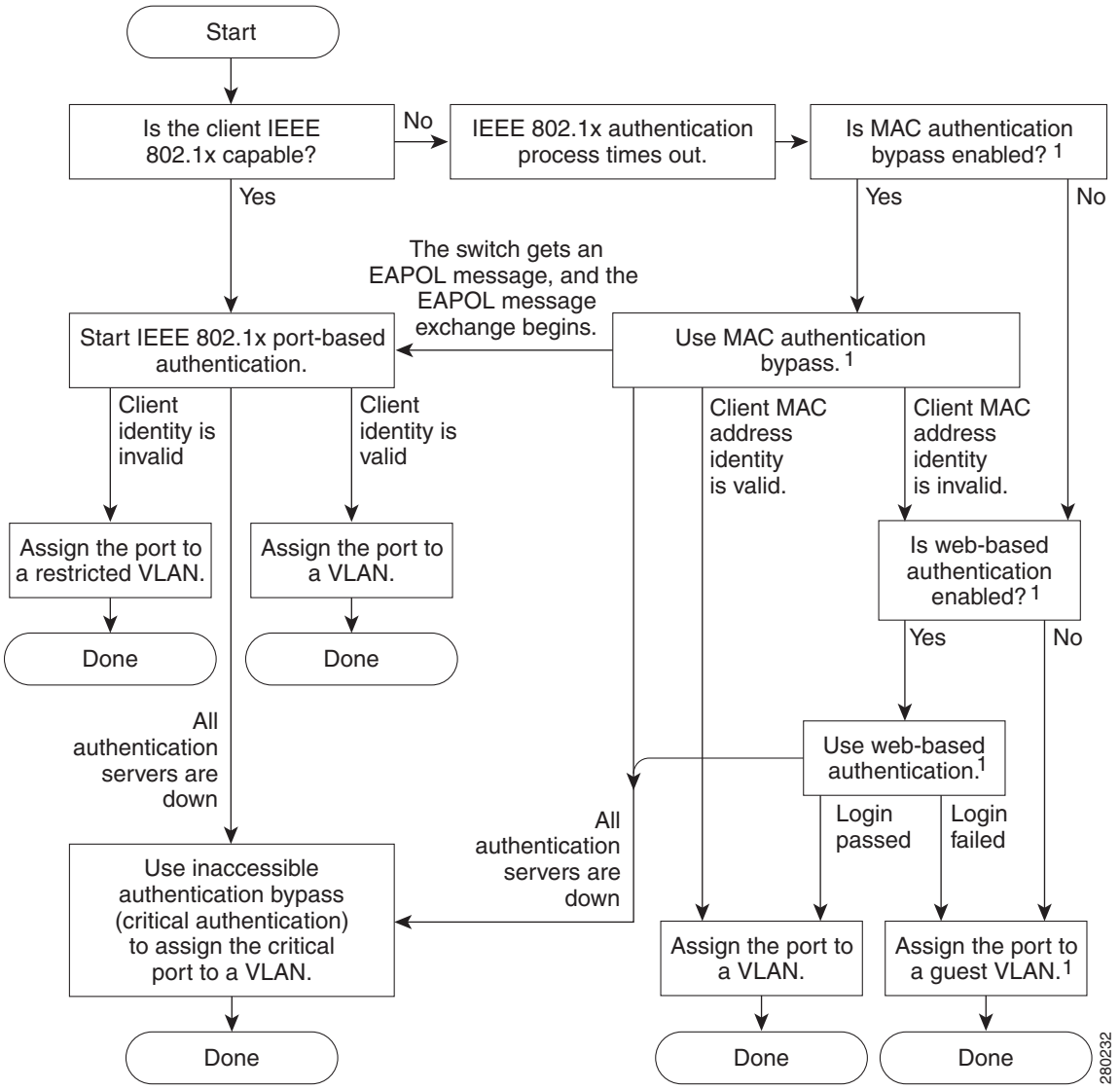
Note The default order for authentication methods is 802.1X, and then MAB, then web-based authentication. You can change the order, and you can disable any of these methods.

- If fallback authentication methods are not enabled or are not successful, and if a guest VLAN is configured, the switch assigns the client to a guest VLAN that provides limited services.
- If the switch receives an invalid identity from an 802.1X-capable client and a restricted VLAN is specified, the switch can assign the client to a restricted VLAN that provides limited services.
- If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the switch grants the client access to the network by putting the port in the critical-authentication state in the user-specified critical VLAN. Release 12.2(33)SXJ1 and later releases support configuration of critical voice and data VLANs.

Note Inaccessible authentication bypass is also referred to as critical authentication or the AAA fail policy.

Figure 60-2 shows the authentication process.

Figure 60-2 Authentication Flowchart



1 = This occurs if the switch does not detect EAPOL packets from the client.

The switch reauthenticates a client when one of these situations occurs:

- Periodic reauthentication is enabled, and the reauthentication timer expires.

  You can configure the reauthentication timer to use a switch-specific value or to be based on values from the RADIUS server.

  After 802.1X authentication using a RADIUS server is configured, the switch uses timers based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).

  The Session-Timeout RADIUS attribute (Attribute[27]) specifies the time after which reauthentication occurs.

The Termination-Action RADIUS attribute (Attribute [29]) specifies the action to take during reauthentication. The actions are Initialize and ReAuthenticate. When the Initialize action is set (the attribute value is DEFAULT), the 802.1X session ends, and connectivity is lost during reauthentication. When the ReAuthenticate action is set (the attribute value is RADIUS-Request), the session is not affected during reauthentication.

- You manually reauthenticate the client by entering the **dot1x re-authenticate interface** *type slot/port* privileged EXEC command (Cisco IOS Release 12.2(33)SXH and earlier releases).

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x pae authenticator** and **authentication port-control auto** interface configuration commands (**dot1x port-control auto** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the switch must initiate authentication when it determines that the port link state transitions from down to up. The switch then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). When the client receives the frame, it responds with an EAP-response/identity frame.

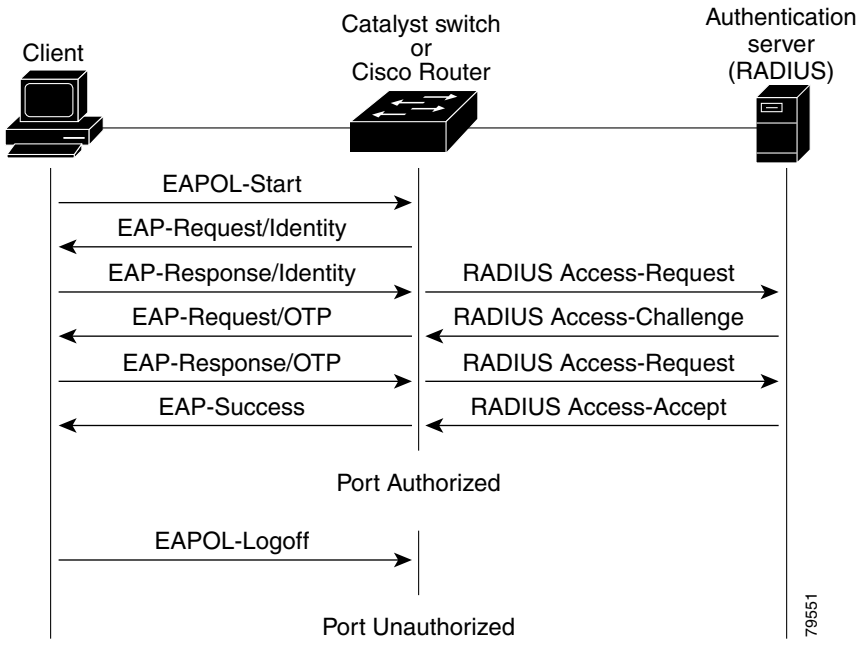
If the client does not receive an EAP-request/identity frame from the switch during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

Note If 802.1X is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated. For more information, see the "Ports in Authorized and Unauthorized States" section on page 60-8.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the port becomes authorized. If the authentication fails, authentication can be retried, the port might be assigned to a VLAN that provides limited services, or network access is not granted. For more information, see the "Ports in Authorized and Unauthorized States" section on page 60-8.

The specific exchange of EAP frames depends on the authentication method being used. Figure 60-3 shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

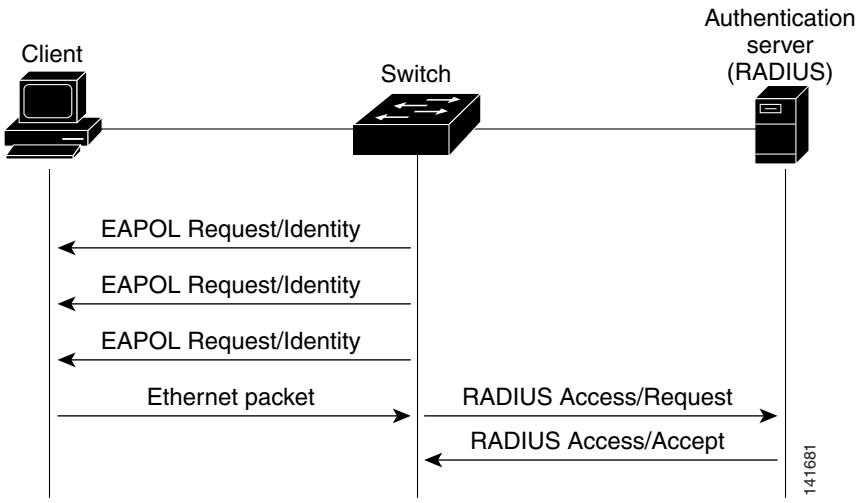
Figure 60-3 Message Exchange



If 802.1X authentication times out while waiting for an EAPOL message exchange, and MAC authentication bypass is enabled, the switch can authorize the client when the switch detects an Ethernet packet from the client. The switch uses the MAC address of the client as its identity and includes this information in the RADIUS-access/request frame that is sent to the RADIUS server. After the server sends the switch the RADIUS-access/accept frame (authorization is successful), the port becomes authorized. If MAB authorization fails and a guest VLAN is specified, the switch assigns the port to the guest VLAN. If the switch detects an EAPOL packet while waiting for an Ethernet packet, the switch stops the MAC authentication bypass process and starts 802.1X authentication.

Figure 60-4 shows the message exchange during MAC authentication bypass.

Figure 60-4 Message Exchange During MAC Authentication Bypass



Ports in Authorized and Unauthorized States

The switch port state determines whether or not the client is granted access to the network. The port starts in the *unauthorized* state. While in this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a client is successfully authenticated, the port transitions to the *authorized* state, allowing all traffic for the client to flow normally.

If a client that does not support 802.1X authentication connects to an unauthorized 802.1X port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You control the port authorization state by using the **authentication port-control** interface configuration command (**dot1x port-control auto** command in Cisco IOS Release 12.2(33)SXH and earlier releases) and these keywords:

- **force-authorized**—Disables 802.1X port-based authentication and causes the port to transition to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This is the default setting.
- **force-unauthorized**—Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.
- **auto**—Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

802.1X Host Modes

The 802.1X port's host mode determines whether more than one client can be authenticated on the port and how authentication will be enforced. You can configure an 802.1X port to use any of the four host modes described in the following sections. In addition, each mode may be modified to allow pre-authentication open access.

Single-Host Mode

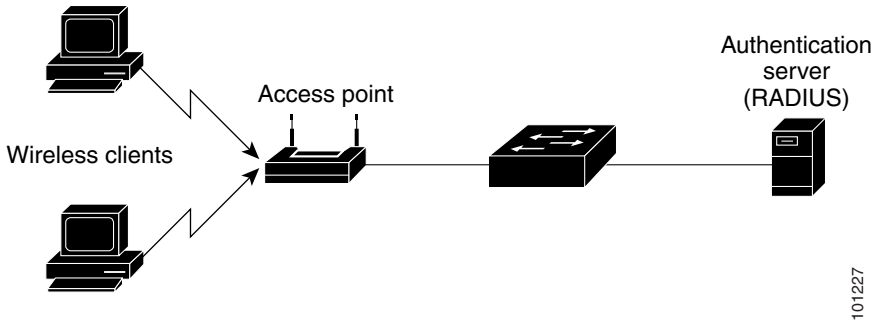
In single-host mode (see Figure 60-1 on page 60-2), only one client can be connected to the 802.1X-enabled port. The switch detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

Multiple-Hosts Mode

In multiple-hosts mode, you can attach multiple hosts to a single 802.1X-enabled port. Figure 60-5 shows 802.1X port-based authentication in a wireless LAN. In this mode, only one of the attached clients must be authorized for all clients to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies network access to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and it also acts as a client to the switch.

With the multiple-hosts mode enabled, you can use 802.1X authentication to authenticate the port and you can use port security to manage network access for all MAC addresses, including the client's MAC address.

Figure 60-5 Multiple Host Mode Example

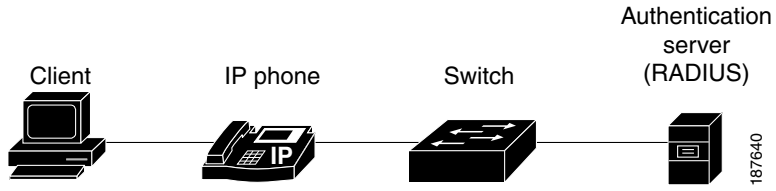


Multidomain Authentication Mode

Supported in Cisco IOS Release 12.2(33)SXI and later releases, multidomain authentication (MDA) mode allows an IP phone (Cisco or third-party) and a single host behind the IP phone to authenticate independently, using 802.1X, MAC authentication bypass (MAB), or (for the host only) web-based authentication. In this application, multidomain refers to two domains, data and voice, and only two MAC addresses are allowed per port. The switch can place the host in the data VLAN and the IP phone in the voice VLAN, though they appear on the same switch port. The data VLAN assignment can be obtained from the vendor-specific attributes (VSAs) received from the authentication, authorization, and accounting (AAA) server during authentication.

Figure 60-6 shows a typical MDA application with a single host behind an IP phone connected to the 802.1X-enabled port. Because the client is not directly connected to the switch, the switch cannot detect a loss of port link if the client is disconnected. To prevent the possibility of another device using the established authentication of the disconnected client, later Cisco IP phones send a Cisco Discovery Protocol (CDP) host presence type length value (TLV) to notify the switch of changes in the attached client's port link state.

Figure 60-6 Multidomain Authentication Mode Example



Multiauthentication Mode

Available in Cisco IOS Release 12.2(33)SXI and later releases, multiauthentication (multiauth) mode allows one 802.1X/MAB client on the voice VLAN and multiple authenticated 802.1X/MAB/webauth clients on the data VLAN. When a hub or access point is connected to an 802.1X port (as shown in Figure 60-5), multiauth mode provides enhanced security over the multiple-hosts mode by requiring authentication of each connected client. For non-802.1X devices, MAB or web-based authentication can be used as the fallback method for individual host authentications, which allows different hosts to be authenticated through different methods on a single port.

Multiauth also supports MDA functionality on the voice VLAN by assigning authenticated devices to either a data or voice VLAN depending on the data that the VSAs received from the authentication server.

Release 12.2(33)SXJ and later releases support the assignment of a RADIUS server-supplied VLAN in multiauth mode, by using the existing commands and when these conditions occur:

- The host is the first host authorized on the port, and the RADIUS server supplies VLAN information.
- Subsequent hosts are authorized with a VLAN that matches the operational VLAN.
- A host is authorized on the port with no VLAN assignment, and subsequent hosts either have no VLAN assignment, or their VLAN information matches the operational VLAN.
- The first host authorized on the port has a group VLAN assignment, and subsequent hosts either have no VLAN assignment, or their group VLAN matches the group VLAN on the port. Subsequent hosts must use the same VLAN from the VLAN group as the first host. If a VLAN list is used, all hosts are subject to the conditions specified in the VLAN list.
- After a VLAN is assigned to a host on the port, subsequent hosts must have matching VLAN information or be denied access to the port.
- The behavior of the critical-auth VLAN is not changed for multiauth mode. When a host tries to authenticate and the server is not reachable, all authorized hosts are reinitialized in the configured VLAN.

Note

- Only one voice VLAN is supported on a multiauth port.
- You cannot configure a guest VLAN or an auth-fail VLAN in multiauth mode.

Pre-Authentication Open Access

With Cisco IOS Release 12.2(33)SXI and later releases, any of the four host modes may be additionally configured to allow a device to gain network access before authentication. This pre-authentication open access is useful in an application such as the Pre-boot eXecution Environment (PXE), where a device must access the network to download a bootable image containing an authentication client.

Pre-authentication open access is enabled by entering the **authentication open** command after host mode configuration, and acts as an extension to the configured host mode. For example, if pre-authentication open access is enabled with single-host mode, then the port will allow only one MAC address. When pre-authentication open access is enabled, initial traffic on the port is restricted only by whatever other access restriction, independent of 802.1X, is configured on the port. If no access restriction other than 802.1X is configured on the port, then a client device will have full access on the configured VLAN.

Understanding 802.1X Authentication with DHCP Snooping

With Cisco IOS Release 12.2(33)SXH and later releases, when the Dynamic Host Configuration Protocol (DHCP) snooping option-82 with data insertion feature is enabled, the switch can insert a client's 802.1X authenticated user identity information into the DHCP discovery process, allowing the DHCP server to assign IP addresses from different IP address pools to different classes of end users. This feature allows you to secure the IP addresses given to the end users for accounting purposes and to allow services based on Layer 3 criteria.

After a successful 802.1X authentication between a supplicant and the RADIUS server, the switch puts the port in the forwarding state and stores the attributes that it receives from the RADIUS server. While performing DHCP snooping, the switch acts as a DHCP relay agent, receiving DHCP messages and regenerating those messages for transmission on another interface. When a client, after 802.1X authentication, sends a DHCP discovery message, the switch receives the packet. The switch adds to the packet a RADIUS attributes suboption section containing the stored RADIUS attributes of the client. The switch then submits the discovery broadcast again. The DHCP server receives the modified DHCP discovery packet and can, if configured to do so, use the authenticated user identity information when creating the IP address lease. The mapping of user-to-IP address can be on a one-to-one, one-to-many, or many-to-many basis. The one-to-many mapping allows the same user to authenticate through the 802.1X hosts on multiple ports.

The switch will automatically insert the authenticated user identity information when 802.1X authentication and DHCP snooping option-82 with data insertion features are enabled. To configure DHCP snooping option-82 with data insertion, see the "DHCP Snooping Option-82 Data Insertion" section on page 54-3.

For information about the data inserted in the RADIUS attributes suboption, see RFC 4014, "Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option."

Understanding 802.1X Accounting

The IEEE 802.1X standard defines how users are authorized and authenticated for network access but does not keep track of network usage. IEEE 802.1X accounting is disabled by default. With Release 12.2(33)SXH and later releases, you can enable 802.1X accounting to monitor the following activities on 802.1X-enabled ports:

- User successfully authenticates.
- User logs off.
- Link-down occurs.
- Reauthentication successfully occurs.
- Reauthentication fails.

The switch does not log IEEE 802.1X accounting information. Instead, it sends this information to the RADIUS server, which must be configured to log accounting messages.

The information sent to the RADIUS server is represented in the form of 802.1X Accounting Attribute-Value (AV) pairs. These AV pairs provide data for different applications. (For example, a billing application might require information that is in the Acct-Input-Octets or the Acct-Output-Octets attributes of a RADIUS packet.)

AV pairs are automatically sent by a switch that is configured for 802.1X accounting. Three types of RADIUS accounting packets are sent by a switch:

- START–Sent when a new user session starts.
- INTERIM–Sent during an existing session for updates.
- STOP–Sent when a session terminates.

Table 60-1 lists the AV pairs and indicates when they are sent are sent by the switch.

Table 60-1 Accounting AV Pairs

Attribute Number	AV Pair Name	START	INTERIM	STOP
Attribute[1]	User-Name	Always	Always	Always
Attribute[4]	NAS-IP-Address	Always	Always	Always
Attribute[5]	NAS-Port	Always	Always	Always
Attribute[8]	Framed-IP-Address	Never	Sometimes[1]	Sometimes[1]
Attribute[25]	Class	Always	Always	Always
Attribute[26]	Vendor-Specific[2]	—	—	—
Attribute[30]	Called-Station-ID	Always	Always	Always
Attribute[31]	Calling-Station-ID	Always	Always	Always
Attribute[40]	Acct-Status-Type	Always	Always	Always
Attribute[41]	Acct-Delay-Time	Always	Always	Always
Attribute[42]	Acct-Input-Octets	Never	Never	Always
Attribute[43]	Acct-Output-Octets	Never	Never	Always
Attribute[44]	Acct-Session-ID	Always	Always	Always
Attribute[45]	Acct-Authentic	Always	Always	Always
Attribute[46]	Acct-Session-Time	Never	Never	Always
Attribute[49]	Acct-Terminate-Cause	Never	Never	Always
Attribute[61]	NAS-Port-Type	Always	Always	Always

1. The Framed-IP-Address AV pair is sent only if a valid DHCP binding exists for the host in the DHCP snooping bindings table.
2. Vendor-specific attributes (VSAs) are used by other 802.1X features.

You can view the AV pairs that are being sent by the switch by entering the **debug radius accounting** privileged EXEC command. For more information about this command, see the *Cisco IOS Debug Command Reference, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/debug/command/reference/122debug.html

For more information about AV pairs, see RFC 3580, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines."

Understanding 802.1X Authentication with VLAN Assignment

After successful 802.1X authentication of a port, the RADIUS server sends the VLAN assignment to configure the port. The RADIUS server database maintains the username-to-VLAN mappings, assigning the VLAN based on the username of the client connected to the port. You can use this feature to limit network access for certain users.

When configured on the switch and the RADIUS server, 802.1X authentication with VLAN assignment has these characteristics:

- If 802.1X authentication is enabled on a port, and if all information from the RADIUS server is valid, the port is placed in the RADIUS server-assigned VLAN after authentication.
- If the multiple-hosts mode is enabled on an 802.1X port, all hosts on the port are placed in the same RADIUS server-assigned VLAN as the first authenticated host.
- If the multiauth mode is enabled on an 802.1X port, the VLAN assignment will be ignored.
- If no VLAN number is supplied by the RADIUS server, the port is configured in its access VLAN after successful authentication. An access VLAN is a VLAN assigned to an access port. All packets sent from or received on this port belong to this VLAN.
- If 802.1X authentication is enabled but the VLAN information from the RADIUS server is not valid, the port returns to the unauthorized state and remains in the configured access VLAN. This prevents ports from appearing unexpectedly in an inappropriate VLAN because of a configuration error.

  Configuration errors could include specifying a VLAN for a routed port, a malformed VLAN ID, a nonexistent or internal (routed port) VLAN ID, or an attempted assignment to a voice VLAN ID.
- If 802.1X authentication is disabled on the port, the port is returned to the configured access VLAN.

When the port is in the force-authorized, force-unauthorized, unauthorized, or shutdown state, the port is put into the configured access VLAN.

If an 802.1X port is authenticated and put in the RADIUS server-assigned VLAN, any change to the port access VLAN configuration does not take effect.

The 802.1X authentication with VLAN assignment feature is not supported on trunk ports, dynamic ports, or with dynamic-access port assignment through a VLAN Membership Policy Server (VMPS).

To configure VLAN assignment, perform this task:

Step 1 Enable AAA authorization by using the **network** keyword to allow interface configuration from the RADIUS server.

Step 2 Enable 802.1X authentication.

Step 3 The VLAN assignment feature is automatically enabled when you configure 802.1X authentication on an access port.

Step 4 Assign vendor-specific tunnel attributes in the RADIUS server. The RADIUS server must return these attributes to the switch:

- [64] Tunnel-Type = VLAN
- [65] Tunnel-Medium-Type = 802
- [81] Tunnel-Private-Group-ID = VLAN name or VLAN ID

Attribute [64] must contain the value *VLAN* (type 13). Attribute [65] must contain the value *802* (type 6). Attribute [81] specifies the *VLAN name* or *VLAN ID* assigned to the 802.1X-authenticated user.

Understanding Multiple VLANs and VLAN User Distribution with VLAN Assignment

In Cisco IOS Release 12.2(33)SXI1 and later releases, the RADIUS-supplied VLAN assignment can provide load balancing by distributing 802.1X-authenticated users among multiple VLANs.

In earlier releases, the RADIUS server can supply a single VLAN name or ID for the assignment of an authenticating user. In Cisco IOS Release 12.2(33)SXI1 and later releases, the RADIUS server can supply multiple VLAN names and IDs or the name of a VLAN group that contains multiple VLANs. Use either of the following two methods to load balance the users between the different VLANs:

- Configure the RADIUS server to send more than one VLAN ID or VLAN name as part of the response to the authenticating user. The 802.1X VLAN user group feature tracks the users in a particular VLAN and achieves load balancing by placing newly authenticated users in the least populated VLAN of the RADIUS-supplied VLAN IDs.

  Perform the steps shown in the "Understanding 802.1X Authentication with VLAN Assignment" section on page 60-13 with the following exception:

  Attribute [81] Tunnel-Private-Group-ID specifies multiple VLAN names or VLAN IDs

- Define a VLAN group that contains multiple VLANs. Configure the RADIUS server to supply the VLAN group name instead of a VLAN ID as part of the response to the authenticating user. If the supplied VLAN group name is found among the VLAN group names that you have defined, the newly authenticated user is placed in the least populated VLAN within the VLAN group.

  Perform the steps shown in the "Understanding 802.1X Authentication with VLAN Assignment" section on page 60-13 with the following exception:

  Attribute [81] Tunnel-Private-Group-ID specifies a defined VLAN group name

For more information, see the "Configuring VLAN User Distribution" section on page 60-49.

Understanding 802.1X Authentication with Guest VLAN

With Release 12.2(33)SXH and later releases, you can configure a guest VLAN for each 802.1X port on the switch to provide limited services to non-802.1X-compliant clients, such as for downloading the 802.1X client software. These clients might be upgrading their system for 802.1X authentication, and some hosts, such as Windows 98 systems, might not be 802.1X-capable.

When you enable a guest VLAN on an 802.1X port, the switch assigns clients to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client and no fallback authentication methods are enabled.

In addition, the switch maintains the EAPOL packet history. If an EAPOL packet is detected on the interface during the lifetime of the link, the switch determines that the device connected to that interface is an 802.1X-capable supplicant, and the interface will not change to the guest VLAN state. The EAPOL packet history is cleared if the interface link status goes down.

Use the **dot1x guest-vlan supplicant** global configuration command to allow an interface to change to the guest VLAN state regardless of the EAPOL packet history. That is, a host that is not 802.1X-capable will be assigned to the guest VLAN even if a previous host on that interface was 802.1X-capable.

Note If an EAPOL packet is detected after the interface has changed to the guest VLAN, the interface reverts to an unauthorized state, and 802.1X authentication restarts.

Any number of 802.1X-incapable clients are allowed access when the port is moved to the guest VLAN. If an 802.1X-capable client joins the same port on which the guest VLAN is configured, the port is put into the unauthorized state in the user-configured access VLAN, and authentication is restarted.

When operating as an 802.1X guest VLAN, a port functions in multiple-hosts mode regardless of the configured host mode of the port.

You can configure any active VLAN except an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as an 802.1X guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

The switch supports MAC authentication bypass in Release 12.2(33)SXH and later releases. When MAC authentication bypass is enabled on an 802.1X port, the switch can authorize clients based on the client MAC address when 802.1X authentication times out while waiting for an EAPOL message exchange. After detecting a client on an 802.1X port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is specified.

For more information, see the "Understanding 802.1X Authentication with MAC Authentication Bypass" section on page 60-23 and the "Configuring a Guest VLAN" section on page 60-49.

Understanding 802.1X Authentication with Restricted VLAN

You can configure a restricted VLAN (also referred to as an *authentication failed VLAN*) for each 802.1X port on a switch to provide limited services to clients that failed authentication and cannot access the guest VLAN. These clients are 802.1X-compliant and cannot access another VLAN because they fail the authentication process. A restricted VLAN allows users without valid credentials in an authentication server (typically, visitors to an enterprise) to access a limited set of services. The administrator can control the services available to the restricted VLAN.

Note You can configure a VLAN to be both the guest VLAN and the restricted VLAN if you want to provide the same services to both types of users.

Without this feature, the client attempts and fails authentication indefinitely, and the port remains in the spanning-tree blocking state. With this feature, you can configure the port to be in the restricted VLAN after a specified number of authentication attempts.

The authenticator counts the failed authentication attempts for the client. The failed attempt count increments when the RADIUS server replies with either an Access-Reject EAP failure or an empty response without an EAP packet. When this count exceeds the configured maximum number of authentication attempts, the port moves to the restricted VLAN, the failed attempt counter resets, and subsequent EAPOL-start messages from the failed client are ignored.

Users who fail authentication remain in the restricted VLAN until the next switch-initiated reauthentication attempt. A port in the restricted VLAN tries to reauthenticate at configured intervals (the default is 60 seconds). If reauthentication fails, the port remains in the restricted VLAN. If reauthentication is successful, the port moves either to the configured VLAN or to a VLAN sent by the RADIUS server. You can disable reauthentication. If you do this, the only way to restart the authentication process is for the port to receive a link down or EAP logoff event. We recommend that you keep reauthentication enabled if a client might connect through a hub. When a client disconnects from the hub, the port might not receive the link down or EAP logoff event.

When operating as an 802.1X restricted VLAN, a port functions in single-host mode regardless of the configured host mode of the port. Only the client that failed authentication is allowed access on the port. An exception is that a port configured in MDA mode can still authenticate a voice supplicant from the restricted VLAN.

You can configure any active VLAN except an RSPAN VLAN or a voice VLAN as an 802.1X restricted VLAN. The restricted VLAN feature is not supported on routed or trunk ports; it is supported only on access ports.

This feature works with port security. As soon as the port is authorized, a MAC address is provided to port security. If port security does not permit the MAC address or if the maximum secure address count is reached, the port becomes unauthorized and error disabled.

Other port security features such as dynamic ARP Inspection, DHCP snooping, and IP source guard can be configured independently on a restricted VLAN.

For more information, see the "Configuring a Restricted VLAN" section on page 60-51.

Understanding 802.1X Authentication with Inaccessible Authentication Bypass

With Release 12.2(33)SXH and later releases, when the switch cannot reach the configured RADIUS servers and hosts cannot be authenticated, you can configure the switch to allow network access to the hosts connected to critical ports. A critical port is enabled for the inaccessible authentication bypass feature, also referred to as critical authentication or the AAA fail policy.

When this feature is enabled, the switch checks the status of the configured RADIUS servers whenever the switch tries to authenticate a host connected to a critical port. If a server is available, the switch can authenticate the host. However, if all the RADIUS servers are unavailable, the switch grants network access to the host and puts the port in the critical-authentication state, which is a special case of the authentication state.

The behavior of the inaccessible authentication bypass feature depends on the authorization state of the port:

- If the port is unauthorized when a host connected to a critical port tries to authenticate and all servers are unavailable, the switch puts the port in the critical-authentication state in the user-specified critical VLAN.
- If the port is already authorized and reauthentication occurs, the switch puts the critical port in the critical-authentication state in the current VLAN, which might be the one previously assigned by the RADIUS server.
- If the RADIUS server becomes unavailable during an authentication exchange, the current exchanges times out, and the switch puts the critical port in the critical-authentication state during the next authentication attempt.

When a RADIUS server that can authenticate the host is available, all critical ports in the critical-authentication state are automatically reauthenticated.

Inaccessible authentication bypass interacts with these features:

- Guest VLAN—Inaccessible authentication bypass is compatible with guest VLAN. When a guest VLAN is enabled on 8021.x port, the features interact as follows:
  - If at least one RADIUS server is available, the switch assigns a client to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client.
  - If all the RADIUS servers are not available and the client is connected to a critical port, the switch authenticates the client and puts the critical port in the critical-authentication state in the user-specified critical VLAN.
  - If all the RADIUS servers are not available and the client is not connected to a critical port, the switch might not assign clients to the guest VLAN if one is configured.
  - If all the RADIUS servers are not available and if a client is connected to a critical port and was previously assigned to a guest VLAN, the switch keeps the port in the guest VLAN.
- Restricted VLAN—If the port is already authorized in a restricted VLAN and the RADIUS servers are unavailable, the switch puts the critical port in the critical-authentication state in the restricted VLAN.
- 802.1X accounting—Accounting is not affected if the RADIUS servers are unavailable.
- Private VLAN—You can configure inaccessible authentication bypass on a private VLAN host port. The access VLAN must be a secondary private VLAN.
- Voice VLAN—Inaccessible authentication bypass is compatible with voice VLAN, but the RADIUS-configured or user-specified access VLAN and the voice VLAN must be different.
- Remote Switched Port Analyzer (RSPAN)—Do not configure an RSPAN VLAN as the RADIUS-configured or user-specified access VLAN for inaccessible authentication bypass.

Understanding 802.1X Authentication with Voice VLAN Ports

A Multi-VLAN Access Port (MVAP) is a port that belong to two VLANs. A voice VLAN port is an MVAP that allows separating a port's voice traffic and data traffic on different VLANs. A voice VLAN port is associated with two VLAN identifiers:

- Voice VLAN identifier (VVID) to carry voice traffic to and from the IP phone. The VVID is used to configure the IP phone connected to the port.
- Port VLAN identifier (PVID) to carry the data traffic to and from the workstation connected to the switch through the IP phone. The PVID is the native VLAN of the port.

In releases earlier than Release 12.2(33)SXH, a switch in single-host mode accepted traffic from a single host, and voice traffic was not allowed. In multiple-hosts mode, the switch did not accept voice traffic until the client was authenticated on the primary VLAN, which makes the IP phone inoperable until the user logged in.

With Release 12.2(33)SXH and later releases, the IP phone uses the VVID for its voice traffic, regardless of the authorization state of the port. This allows the phone to work independently of 802.1X authentication.

In single-host mode, only the IP phone is allowed on the voice VLAN. In multiple-hosts mode, additional clients can send traffic on the voice VLAN after a supplicant is authenticated on the PVID. When multiple-hosts mode is enabled, the supplicant authentication affects both the PVID and the VVID.

In order to recognize an IP phone, the switch will allow CDP traffic on a port regardless of the authorization state of the port. A voice VLAN port becomes active when there is a link, and the device MAC address appears after the first CDP message from the IP phone. Cisco IP phones do not relay CDP messages from other devices. As a result, if several IP phones are connected in series, the switch recognizes only the one directly connected to it. When 802.1X authentication is enabled on a voice VLAN port, the switch drops packets from unrecognized IP phones more than one hop away.

When 802.1X authentication is enabled on a port, you cannot configure a port VLAN that is equal to a voice VLAN.

Note If you enable 802.1X authentication on an access port on which a voice VLAN is configured and to which a Cisco IP phone is connected, the Cisco IP phone loses connectivity to the switch for up to 30 seconds.

For voice VLAN configuration information, see Chapter 15, "Configuring Cisco IP Phone Support."

Understanding 802.1X Authentication Critical Voice VLAN Support

With normal network connectivity, when an IP phone successfully authenticates on a port, the authentication server puts the phone into the voice domain. If the authentication server becomes unreachable, IP phones cannot authenticate. In multidomain authentication (MDA) mode or multiauthentication mode, you can configure the critical voice VLAN support feature to put phone traffic into the configured voice VLAN of the port (see the "Enabling Critical Voice VLAN Support" section on page 60-56).

Understanding 802.1X Authentication with Port Security

With Release 12.2(33)SXH and later releases, you can configure an 802.1X port with port security in either single-host or multiple-hosts mode. (You also must configure port security on the port by using the **switchport port-security** interface configuration command.) When you enable port security and 802.1X authentication on a port, 802.1X 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 an 802.1X port.

These are some examples of the interaction between 802.1X authentication and port security on the switch:

- When a client is authenticated, and the port security table is not full, the client MAC address is added to the port security list of secure hosts. The port then proceeds to come up normally.

  When a client is authenticated and manually configured for port security, it is guaranteed an entry in the secure host table.

  A security violation occurs if the client is authenticated, but the port security table is full. This can happen if the maximum number of secure hosts has been statically configured or if the client ages out of the secure host table. If the client address is aged, its place in the secure host table can be taken by another host.

  If a security violation is caused by any host, the port becomes error-disabled and immediately shuts down.

  The port security violation modes determine the action for security violations. For more information, see the "Configuring the Port Security Violation Mode on a Port" section on page 62-6.

- When you manually remove an 802.1X client address from the port security table by using the **no switchport port-security mac-address** *mac-address* interface configuration command, you should reauthenticate the 802.1X client by using the **dot1x re-authenticate interface** *type slot/port* privileged EXEC command.
- When an 802.1X client logs off, the port changes to an unauthenticated state, and all dynamic entries in the secure host table are cleared, including the entry for the client. Normal authentication then takes place.
- If the port is administratively shut down, the port becomes unauthenticated, and all dynamic entries are removed from the secure host table.
- Port security and a voice VLAN can be configured simultaneously on an 802.1X port that is in either single-host or multiple-hosts mode. Port security applies to both the voice VLAN identifier (VVID) and the port VLAN identifier (PVID).

For more information about enabling port security on your switch, see the "Configuring Port Security" section on page 62-5.

Understanding 802.1X Authentication with ACL Assignments and Redirect URLs

With Cisco IOS Release 12.2(33)SXI and later releases, per-host policies such as ACLs and redirect URLs can be downloaded to the switch from the authentication server (AS) in a RADIUS Access-Accept packet at the end of an 802.1X, MAB, or web-based authentication exchange.

Per-host policies are activated during authentication as follows:

- Downloadable ACLs (DACLs) are defined in the Cisco Secure ACS and downloaded from the ACS to the switch using VSAs.
- Filter-ID ACLs are defined on the switch, and only the ACL name is downloaded from the AS to the switch using the RADIUS Filter-ID attribute. Filter-ID ACLs are supported in Cisco IOS Release 12.2(33)SXI2 and later releases.
- A redirection URL and an ACL name are downloaded from the ACS to the switch using VSAs. The redirection ACL is defined on the switch.

For information about configuring per-host policies, see the "Configuring the Switch for DACLs or Redirect URLs" section on page 60-60.

Downloadable ACLs Using the Cisco Secure ACS

Following a successful host authentication, the Cisco Secure ACS can use a VSA to download an ACL to the switch. The switch combines the DACL with the default ACL on the port to which the host has connected. Because the DACL definition resides on the authentication server, this feature allows for centralized policy management.

Two methods are provided in the Cisco Secure ACS for configuring DACLs:

- Downloadable IP ACL

  Downloading of the DACL is enabled by selecting Assign IP ACL in the ACS configuration, and the DACL is defined in the Downloadable IP ACL Content menu of the ACS. There is no restriction on the size of the DACL.
- Per-user ACL

In Cisco IOS Release 12.2(33)SXI2 and later releases, the ACS can use the CiscoSecure-Defined-ACL [009\001 cisco-av-pair] VSAs to deliver the DACL. Because the entire DACL is delivered in a single RADIUS packet, the maximum size is limited by the 4096-byte maximum size for a RADIUS packet. The DACL must be defined on the ACS using the following format:

protocol:inacl#*sequence_number=ace*

as shown in this example:

```
ip:inacl#10=permit ip any 67.2.2.0 0.0.0.255
```

These guidelines apply when using DACLs:

- The source address for all ACEs must be defined as ANY.
- When the 802.1X host mode of the port is MDA or multiauth, the DACL will be modified to use the authenticated host's IP address as the source address. When the host mode is either single-host or multiple-host, the source address will be configured as ANY, and the downloaded ACLs or redirects will apply to all devices on the port.
- If no DACLs are provided during the authentication of a host, the static default ACL configured on the port will be applied to the host. On a voice VLAN port, only the static default ACL of the port will be applied to the phone.

Filter-ID ACLs

In Cisco IOS Release 12.2(33)SXI2 and later releases, following a successful host authentication, the authentication server can use the RADIUS Filter-ID attribute (Attribute[11]) rather than a VSA to deliver only the name of an extended ACL to the switch in the following format:

acl_name.in

The suffix ".in" indicates that the ACL should be applied in the inbound direction.

In this method, the ACL must be already defined on the switch. The switch matches the Filter-ID attribute value to a locally configured ACL that has the same name or number as the Filter-ID (for example, Filter-ID=101.in will match the extended numbered ACL 101, and Filter-ID= guest.in will match the extended named ACL "guest"). The specified ACL is then applied to the port. Because the ACL definition resides on the switch, this feature allows for local variation in a policy.

These guidelines apply when using Filter-ID ACLs:

- The guidelines for using DACLs also apply to Filter-ID ACLs.
- The Filter-ID attribute may be a number (100 to 199, or 2000 to 2699) or a name.

Redirect URLs

Following a successful host authentication, the Cisco Secure ACS can use a VSA to download information to the switch for intercepting and redirecting HTTP or HTTPS requests from the authenticated host. The ACS downloads a redirection ACL and URL. When an HTTP or HTTPS request from the host matches the downloaded ACL, the host's web browser is redirected to the downloaded redirection URL.

The ACS uses these cisco-av-pair VSAs to configure the redirection:

- url-redirect-acl

This AV pair contains the name or number of an ACL that specifies the HTTP or HTTPS traffic to be redirected. The ACL must be defined on the switch, and the source address must be defined as ANY. Traffic that matches a permit entry in the redirect ACL will be redirected.

- url-redirect

  This AV pair contains the HTTP or HTTPS URL to which the web browser will be redirected.

Static Sharing of ACLs

When a number of interfaces have the same PACL and VLAN-based features, you can use the **mls acl tcam share-acl** global configuration command to enable the static sharing feature. With static sharing, only one copy of the PACL and inherited VLAN-based feature ACLs is stored in the TCAM for all ports using the same ACL set, freeing TCAM space for more ACLs. With static sharing enabled, the switch will automatically evaluate all configured or enabled interfaces for static sharing when any of these events occur:

- When the **mls acl tcam share-acl** command is entered.
- When an interface is configured.
- When a state change occurs on an interface.

When enabling static sharing, consider the following guidelines and restrictions:

- Static sharing is not supported for interfaces enabled with IPv6.
- Static sharing is not supported with PFC3A-based supervisor engines or earlier, or in systems running in PFC3A mode or lower.
- Static sharing is supported only on switch ports in access mode with NAC or 802.1X DACL features configured.
- Static sharing is not supported on switch ports enabled with QOS, with the exception of VLAN-based QoS.
- When 802.1X is used with DACL, we recommend entering the **platform hardware acl downloadable setup static** command to avoid triggering a static sharing evaluation when the port is dynamically configured by the authentication server response. The static sharing evaluation may adversely affect the port/host linkup time.
- 802.1X interfaces with fallback authentication as active cannot form a static sharing group with interfaces on which fallback is not enabled or is not active.

Understanding 802.1X Authentication with Port Descriptors

With Release 12.2(33)SXI and later releases, you can associate descriptive text with an 802.1X client's authentication information by configuring the Cisco vendor-specific attribute (VSA) **aaa:supplicant-name** on the RADIUS server. During a successful 802.1X authentication of the client on the port, the switch will receive the descriptive information from the RADIUS server as part of the

Access-Accept packet and will display the information when the **show interface users** command is entered for the port. If the port is in a mode supporting multiple authenticated hosts, identity information for all the authenticated hosts will be displayed with the port description.

Understanding 802.1X Authentication with MAC Authentication Bypass

With Release 12.2(33)SXH and later releases, you can configure the switch to authorize clients based on the client MAC address (see Figure 60-4 on page 60-8) by using the MAC authentication bypass feature. For example, you can enable this feature on 802.1X ports connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the client, the switch tries to authorize the client by using MAC authentication bypass.

When the MAC authentication bypass feature is enabled on an 802.1X port, the switch uses the MAC address as the client identity. The authentication server has a database of client MAC addresses that are allowed network access. After detecting a client on an 802.1X port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is configured.

If an EAPOL packet is detected on the interface during the lifetime of the link, the switch determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the switch already authorized a port by using MAC authentication bypass and detects an 802.1X supplicant, the switch does not unauthorize the client connected to the port. When reauthentication occurs, the switch uses 802.1X authentication as the preferred reauthentication process if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN. If reauthentication fails, the switch assigns the port to the guest VLAN, if one is configured.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize, (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, “IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines.”

MAC authentication bypass interacts with the features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.
- Guest VLAN—If a client has an invalid MAC address identity, the switch assigns the client to a guest VLAN if one is configured.
- Restricted VLAN—This feature is not supported when the client connected to an 802.1x port is authenticated with MAC authentication bypass.
- Port security—See the “Understanding 802.1X Authentication with Port Security” section on page 60-19.

- Voice VLAN—See the "Understanding 802.1X Authentication with Voice VLAN Ports" section on page 60-18.
- VLAN Membership Policy Server (VMPS)— 802.1X and VMPS are mutually exclusive.
- Private VLAN—You can assign a client to a private VLAN.
- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

Understanding Network Admission Control Layer 2 IEEE 802.1X Validation

Cisco IOS Release 12.2(33)SXH and later releases support Network Admission Control (NAC) Layer 2 IEEE 802.1X validation, which checks the antivirus condition or *posture* of endpoint systems or clients before granting the devices network access. NAC Layer 2 IEEE 802.1X validation performs policy enforcement by assigning the authenticated port into a specified VLAN, which provides segmentation and quarantine of poorly postured hosts at Layer 2.

Configuring NAC Layer 2 IEEE 802.1X validation is similar to configuring 802.1X port-based authentication except that you must configure a posture token on the RADIUS server. You can view the NAC posture token, which shows the posture of the client, by using the **show dot1x** privileged EXEC command. For information about configuring NAC Layer 2 IEEE 802.1X validation, see the "Configuring NAC Layer 2 IEEE 802.1X Validation" section on page 60-58.

For more information about NAC, see the *Network Admission Control Software Configuration Guide*.

NAC Agentless Audit Support

With Cisco IOS Release 12.2(33)SXI and later releases, MAB support is added for the Cisco NAC Audit Architecture, which uses an external audit server to check the antivirus posture of clients that do not run a Cisco Trust Agent (CTA) and cannot respond to NAC queries. To audit and report an agentless client's antivirus posture, the NAC audit server must possess the client's IP address and a unique session identifier for the client's connection to the switch. To support the NAC audit architecture for agentless clients, the switch must snoop the client's IP address, create and assign a unique session identifier for the agentless client, and pass this information to the RADIUS server for sharing with the NAC audit server.

Because MAB operates at Layer 2, the MAB authenticator does not normally know the IP address of the supplicant, and the supplicant might not have an IP address when it first contacts the authenticator. A supplicant that requires a DHCP-assigned IP address must be allowed access to a DHCP server before authentication. You must enable ARP and DHCP snooping on the switch to allow the MAB authenticator to learn the IP address of the supplicant. To allow the IP address and unique session identifier information to be shared with the NAC audit server, you must enable the sending of certain RADIUS attributes. See the "Configuring NAC Agentless Audit Support" section on page 60-59.

The client IP address and unique session identifier are shared in RADIUS Access-Requests and Access-Accepts using the following RADIUS *cisco-av-pair* vendor-specific attributes (VSAs):

- Cisco-AVPair="identity-request=*ip-address*"

  ip-address is the client IP address obtained by the switch through ARP or DHCP snooping.
- Cisco-AVPair="audit-session-id=*audit session id string*"

  audit session id string is a UTF-8 encoding of a unique 96-bit identifier derived by the switch from the network access server (NAS) IP address, a session count, and the session start timestamp.

Understanding RADIUS Change of Authorization

With Cisco IOS Release 12.2(33)SXI4 and later releases, the switch can accept and execute unsolicited Change of Authorization (CoA) messages from the authentication server (AS). CoA is an extension to the RADIUS protocol that allows the AS to make dynamic and unsolicited changes to the authorization information of an active session hosted by a network access device, such as a switch. For more information about CoA, see RFC 5176.

The Catalyst 6500 series switch supports per-session and per-policy CoA commands relating to 802.1X, MAB, and web-based authentication sessions.

Per-Session CoA

Using per-session CoA commands, the AS can cause the switch to terminate a session or to force a reauthentication of the session. To terminate a session, the AS can instruct the switch to perform one of the following actions:

- End the session—The AS sends a CoA Disconnect-Request (see RFC 5176), causing the switch to delete all state information about the session.
- Shut down the port—The AS sends the following VSA to force an administrative shutdown of the port:

  Cisco-AVPair="subscriber:command=*disable-host-port*"
- Bounce the port—The AS sends the following VSA to force the switch link to be taken down, then up again:

  Cisco-AVPair="subscriber:command=*bounce-host-port*"

By default, the switch accepts and executes per-session CoA commands, but you can configure the switch to ignore CoA shutdown or bounce commands directed at specific ports.

The AS sends the following VSA to force a reauthentication of the session:

Cisco-AVPair="subscriber:command=*re-authenticate*"

Per-Policy CoA

Using per-policy CoA commands, the AS can instruct the switch to update the contents of a DACL or a Filter-ID ACL, and apply the updated policy information to all sessions that currently have the affected ACL applied.

Understanding 802.1X Authentication with Wake-on-LAN

With Release 12.2(33)SXH and later releases, the 802.1X authentication with wake-on-LAN (WoL) feature allows dormant PCs to be powered up when the switch receives a specific Ethernet frame, known as the *magic packet*. You can use this feature in environments where administrators need to connect to systems that have been powered down.

When a host that uses WoL is attached through an 802.1X port and the host powers off, the 802.1X port becomes unauthorized. The port can only receive and send EAPOL packets, and WoL magic packets cannot reach the host. When the PC is powered off, it is not authorized, and the switch port is not opened.

When the switch uses 802.1X authentication with WoL, the switch forwards traffic to unauthorized 802.1X ports, including magic packets. While the port is unauthorized, the switch continues to block ingress traffic other than EAPOL packets. The host can receive packets but cannot send packets to other devices in the network.

Note If PortFast is not enabled on the port, the port is forced to the bidirectional state.

When you configure a port as unidirectional by using the **authentication control-direction in** interface configuration command (**dot1x control-direction in** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the port changes to the spanning-tree forwarding state. The port can send packets to the host but cannot receive packets from the host.

When you configure a port as bidirectional by using the **authentication control-direction both** interface configuration command (**dot1x control-direction both** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the port is access-controlled in both directions. The port does not receive packets from or send packets to the host.

Understanding MAC Move

Release 12.2(33)SXI4 and later releases support the Mac move feature. When a MAC address is authenticated on one switch port, that address is not allowed on another authentication manager-enabled port of the switch. If the switch detects that same MAC address on another authentication manager-enabled port, the address is not allowed.

There are situations where a MAC address might need to move from one port to another on the same switch. For example, when there is another device (for example a hub or an IP phone) between an authenticated host and a switch port, you might want to disconnect the host from the device and connect it directly to another port on the same switch.

You can globally enable MAC move so the device is reauthenticated on the new port. When a host moves to a second port, the session on the first port is deleted, and the host is reauthenticated on the new port.

When a MAC address moves from one port to another, the switch terminates the authenticated session on the original port and initiates a new authentication sequence on the new port. Port security behavior remains the same when you configure MAC move.

Note

- MAC move is supported in all host modes. (The authenticated host can move to any port on the switch, no matter which host mode is enabled on the that port.)
- MAC move is supported with port security.
- The MAC move feature applies to both voice and data hosts.
- In open authentication mode, a MAC address is immediately moved from the original port to the new port, with no requirement for authorization on the new port.

For more information see the "Enabling MAC Move" section on page 60-71.

Understanding MAC Replace

Release 12.2(33)SXI4 and later releases support the Mac replace feature. The MAC replace feature can be configured to address the violation that occurs when a host attempts to connect to a port where another host was previously authenticated.

Note

- The Mac replace feature is not supported on ports in multiauth mode, because violations are not triggered in that mode.
- The Mac replace feature is not supported on ports in multiple host mode, because in that mode, only the first host requires authentication.

If you configure the **authentication violation** interface configuration command with the **replace** keyword, the authentication process on a port in multi-domain mode is:

- A new MAC address is received on a port with an existing authenticated MAC address.
- The authentication manager replaces the MAC address of the current data host on the port with the new MAC address.
- The authentication manager initiates the authentication process for the new MAC address.
- If the authentication manager determines that the new host is a voice host, the original voice host is removed.

If a port is in open authentication mode, any new MAC address is immediately added to the MAC address table.

For more information see the "Enabling MAC Replace" section on page 60-71.

Understanding 802.1x Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)

Release 12.2(33)SXJ and later releases support the Network Edge Access Topology (NEAT) feature. NEAT extends identity to areas outside the wiring closet (such as conference rooms). This allows any type of device to authenticate on the port.

- 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.

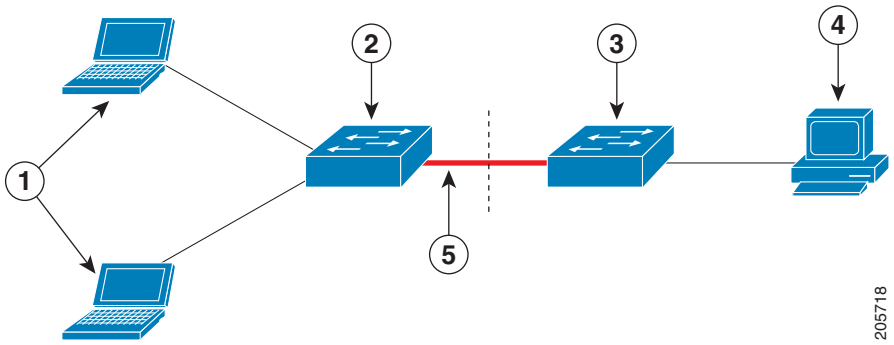
  Once the supplicant switch authenticates successfully the port mode changes from access to trunk.
- If the access VLAN is configured on the authenticator switch, it becomes the native VLAN for the trunk port after successful authentication.

You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. On the authenticator switch interface, multihost mode is not supported and in MDA mode voice client is not supported.

Use the **dot1x supplicant force-multicast** global configuration command on the supplicant switch for Network Edge Access Topology (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 Client Information Signalling Protocol (CISP) to send the MAC addresses connecting to the supplicant switch to the authenticator switch, as shown in Figure 60-7.
- 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 ACS. (You can configure this under the *group* or the *user* settings.)

Figure 60-7 Authenticator and Supplicant Switch using CISP



1	Workstations (clients)	**2**	Supplicant switch (outside wiring closet)
3	Authenticator switch	**4**	Access control server (ACS)
5	Trunk port		

When configuring NEAT and CISP, follow these guidelines and restrictions:

- You can configure NEAT ports with the same configurations as the other authentication ports. When the supplicant switch authenticates, the port mode is changed from *access* to *trunk* based on the switch vendor-specific attributes (VSAs). (device-traffic-class=switch).
- The VSA changes the authenticator switch port mode from access to trunk and enables 802.1x trunk encapsulation and the access VLAN if any would be converted to a native trunk VLAN. VSA does not change any of the port configurations on the supplicant

For more information, see the "Configuring an Authenticator and a Supplicant Switch with NEAT" section on page 60-64.

802.1X Authentication Feature Configuration Guidelines

This section has configuration guidelines for these features:

802.1X Authentication

When configuring 802.1X authentication, note the following guidelines:

- In releases where CSCtg01609 is not resolved, on ports with the **authentication port-control auto** command or the **dot1x pae supplicant** command configured, you cannot successfully enter the **no switchport** command. In releases where CSCtg01609 is resolved, on ports with any **authentication**, **dot1x**, or **mab** command configured, you cannot successfully enter the **no switchport** command.

  Note Enter the **default interface** *type slot/port* command to revert to the default port configuration.

- When 802.1X authentication is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- If you try to change the mode of an 802.1X-enabled port (for example, from access to trunk), an error message appears, and the port mode is not changed.
- With Cisco IOS Release 12.2(33)SXH and later releases, you can configure port security and 802.1X port-based authentication on the same port. We do not recommend configuring both together.
- If the VLAN to which an 802.1X-enabled port is assigned changes, this change is transparent and does not affect the switch. For example, this change occurs if a port is assigned to a RADIUS server-assigned VLAN and is then assigned to a different VLAN after reauthentication.

  If the VLAN to which an 802.1X port is assigned to shut down, disabled, or removed, the port becomes unauthorized. For example, the port is unauthorized after the access VLAN to which a port is assigned shuts down or is removed.
- The 802.1X protocol is supported on Layer 2 static-access ports, voice VLAN ports, and Layer 3 routed ports, but it is not supported on these port types:
  - Trunk ports:

    With Release 12.2(33)SXJ and later releases, you can enter the commands to enable 802.1X authentication on a trunk port or change the mode of an 802.1X-enabled port to trunk, but 802.1X authentication works only on trunk ports configured to support a switch supplicant (SSw). Configure 802.1X authentication on trunk ports only to support NEAT (CSCtx16322).

    With releases earlier than Release 12.2(33)SXJ, if you try to enable 802.1X authentication on a trunk port, an error message appears, and 802.1X authentication is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, an error message appears, and the port mode is not changed.

- Dynamic ports—A port in dynamic mode can negotiate with its neighbor to become a trunk port. If you try to enable 802.1X authentication on a dynamic port, an error message appears, and 802.1X authentication is not enabled. If you try to change the mode of an 802.1X-enabled port to dynamic, an error message appears, and the port mode is not changed.
- Dynamic-access ports—If you try to enable 802.1X authentication on a dynamic-access (VLAN Query Protocol [VQP]) port, an error message appears, and 802.1X authentication is not enabled. If you try to change an 802.1X-enabled port to dynamic VLAN assignment, an error message appears, and the VLAN configuration is not changed.
- EtherChannel port—Do not configure a port that is an active or a not-yet-active member of an EtherChannel as an 802.1X port. If you try to enable 802.1X authentication on an EtherChannel port, an error message appears, and 802.1X authentication is not enabled.

Note In software releases earlier than Release 12.2(33)SXH, if 802.1X authentication is enabled on a not-yet active port of an EtherChannel, the port does not join the EtherChannel.

- Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) destination ports—You can enable 802.1X authentication on a port that is a SPAN or RSPAN destination port. However, 802.1X authentication is disabled until the port is removed as a SPAN or RSPAN destination port. You can enable 802.1X authentication on a SPAN or RSPAN source port.

Note In software releases earlier than Release 12.2(33)SXH, 802.1X authentication is not supported on voice VLAN ports.

- Before globally enabling 802.1X authentication on a switch by entering the **dot1x system-auth-control** global configuration command, remove the EtherChannel configuration from the interfaces on which 802.1X authentication and EtherChannel are configured.
- Because all traffic from unauthenticated hosts is forwarded to the switch processor, we recommend that you apply rate limiting to this traffic.

802.1X Host Mode

When configuring any host mode, note the following guidelines:

- In most cases when the host mode is changed on a port, any existing 802.1X authentications on that port are deleted. Exceptions are when changing from the single-host mode to any other mode, and when changing from multidomain mode to multiauth mode. In these two cases, existing 802.1X authentications are retained.
- If you enter the **authentication open** interface configuration command in Cisco IOS Release 12.2(33)SXI and later releases, any new MAC address detected on the port will be allowed unrestricted Layer 2 access to the network even before any authentication has succeeded. If you use this command, you should use static default ACLs to restrict Layer 3 traffic. For additional details, see the "Pre-Authentication Open Access" section on page 60-11.

When configuring multiple-hosts mode, note the following guideline:

- If the multiple-hosts port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

When configuring MDA host mode, note the following guideline:

- A third-party IP phone's MAC address will initially be assigned to the data VLAN. When tagged voice packets are observed, the device will be removed from the data VLAN and placed on the voice VLAN.

When configuring multiauth host mode, note the following guidelines:

- If one client on a multiauth port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received from that client), the authorization status of the other attached clients is not changed.
- RADIUS-assigned VLANs are not supported on multiauth ports, which can have only one data VLAN. If the authentication server sends VLAN-related attributes, the authentication will succeed but the VLAN assignment will be ignored.
- Although multiple hosts are allowed on the data VLAN, only one host is allowed on the voice VLAN. When one IP phone has been authenticated, further IP phones on the same port will be denied authentication.
- A multiauth port does not support a guest VLAN, authentication-fail VLAN, or with releases earlier than Release 12.2(33)SXJ1, a critical VLAN.

VLAN Assignment, Guest VLAN, Restricted VLAN, and Inaccessible Authentication Bypass

When configuring VLAN assignment, guest VLAN, restricted VLAN, and inaccessible authentication bypass, note the following guidelines:

- When 802.1X authentication is enabled on a port, you cannot configure a port VLAN that is equal to a voice VLAN.
- The 802.1X authentication with VLAN assignment feature is not supported on trunk ports, dynamic ports, or with dynamic-access port assignment through a VMPS.
- You can configure any VLAN except an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as an 802.1X guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.
- After you configure a guest VLAN for an 802.1X port to which a DHCP client is connected, you might need to get a host IP address from a DHCP server. You can change the settings for restarting the 802.1X authentication process on the switch before the DHCP process on the client times out and tries to get a host IP address from the DHCP server. Decrease the settings for the 802.1X authentication process (**dot1x timeout quiet-period** and **dot1x timeout tx-period** interface configuration commands). The amount to decrease the settings depends on the connected 802.1X client type.
- When configuring the 802.1X VLAN user distribution feature, follow these guidelines:
  - A maximum of 100 VLAN groups can be configured, and a maximum of 4094 VLANs can be mapped to a VLAN group.
  - A VLAN can be mapped to more than one VLAN group.
  - A guest VLAN, a critical VLAN, or a restricted VLAN can be mapped to a VLAN group.
  - A VLAN group name cannot be specified as a guest VLAN, a critical VLAN, or a restricted VLAN.

  - You can modify a VLAN group by adding or removing a VLAN, but at least one VLAN must be mapped to the VLAN group. If you remove the last VLAN from the VLAN group, the VLAN group is deleted.
  - Removing an existing VLAN from the VLAN group name does not revoke the authentication status of the ports in the VLAN, but the mappings are removed from the existing VLAN group.
  - Deleting an existing VLAN group name does not revoke the authentication status of the ports in any VLAN within the group, but the VLAN mappings to the VLAN group are removed.
- When configuring the inaccessible authentication bypass feature, follow these guidelines:
  - The inaccessible authentication bypass feature is supported on 802.1X ports in single-host mode, multiple-hosts mode, and MDA mode.
  - If the client is running Windows XP and the port to which the client is connected is in the critical-authentication state, Windows XP might report that the interface is not authenticated.
  - If the Windows XP client is configured for DHCP and has an IP address from the DHCP server, receiving an EAP-Success message on a critical port might not reinitiate the DHCP configuration process.
  - You can configure the inaccessible authentication bypass feature and the critical VLAN on an 802.1X port. If the switch tries to reauthenticate a critical port in a critical VLAN and all the RADIUS servers are unavailable, the switch changes the port state to the critical authentication state and the port remains in the critical VLAN.
  - You can configure the inaccessible bypass feature and port security on the same port.
- You can configure any VLAN except an RSPAN VLAN or a voice VLAN as an 802.1X restricted VLAN. The restricted VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

MAC Authentication Bypass

When configuring MAC authentication bypass, note the following guidelines:

- Unless otherwise stated, the MAC authentication bypass guidelines are the same as the 802.1X authentication guidelines. For more information, see the "802.1X Authentication" section on page 60-29.
- If you disable MAC authentication bypass from a port after the port has been authorized with its MAC address, the session will be removed.
- When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.
- If the port is in the unauthorized state and the client MAC address is not the authentication-server database, the port remains in the unauthorized state. However, if the client MAC address is added to the database, the switch can use MAC authentication bypass to reauthorize the port.
- If the port is in the authorized state, the port remains in this state until reauthorization occurs.
- To use MAC authentication bypass on a routed port, make sure that MAC address learning is enabled on the port.
- In Release 12.2(33)SXH and later releases, you can optionally configure a timeout period for hosts that are connected by MAC authentication bypass but are inactive. The range is 1 to 65535 seconds, but should be set to a value less than the reauthentication timeout. You must enable port security before configuring a timeout value. For more information, see the "Configuring Port Security" section on page 62-5.

Web-Based Authentication

When configuring web-based authentication, note the following guidelines:

- Fallback to web-based authentication is configured on switch ports in access mode. Ports in trunk mode are not supported.
- Fallback to web-based authentication is not supported on EtherChannels or EtherChannel members.
- Although fallback to web-based authentication is an interface-specific configuration, the web-based authentication fallback behavior is defined in a global fallback profile. If the global fallback configuration changes, the new profile will not be used until the next instance of authentication fallback.

For detailed information on configuring web-based authentication, see Chapter 61, "Configuring Web-Based Authentication."

Configuring 802.1X Port-Based Authentication

These sections describe how to configure 802.1X port-based authentication:

Default 802.1X Port-Based Authentication Configuration

Table 60-2 shows the default 802.1X configuration.

Table 60-2 Default 802.1X Configuration

Feature	Default Setting
Switch 802.1X enable state	Disabled.
Per-port 802.1X enable state	Disabled (force-authorized). The port sends and receives normal traffic without 802.1X-based authentication of the client.
AAA	Disabled.
RADIUS server • IP address • UDP authentication port • Key	 • None specified. • 1812. • None specified.
Host mode	Single-host mode.
Control direction	Bidirectional control.
Periodic reauthentication	Disabled.
Number of seconds between reauthentication attempts	3600 seconds.
Reauthentication number	2 times (number of times that the switch restarts the authentication process before the port changes to the unauthorized state).
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client).
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request).

Table 60-2 Default 802.1X Configuration (continued)

Feature	Default Setting
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process).
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before retransmitting the request to the client).
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before retransmitting the response to the server).
Inactivity timeout	Disabled.
Guest VLAN	None specified.
Inaccessible authentication bypass	Disabled.
Restricted VLAN	None specified.
Authenticator (switch) mode	None specified.
MAC authentication bypass	Disabled. **Note** When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.

Enabling 802.1X Authentication

To enable 802.1X port-based authentication, you must enable AAA and specify the authentication method list.

A method list describes the sequence and authentication methods to be queried to authenticate a user. 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, the authentication process stops, and no other authentication methods are attempted.

To allow VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

The 802.1X AAA process is as follows:

1. A user connects to a port on the switch.
2. Authentication is performed.
3. VLAN assignment is enabled, as appropriate, based on the RADIUS server configuration.
4. The switch sends a start message to an accounting server.
5. Reauthentication is performed, as necessary.

6. The switch sends an interim accounting update to the accounting server that is based on the result of reauthentication.
7. The user disconnects from the port.
8. The switch sends a stop message to the accounting server.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
Step 2	`Router(config)# aaa authentication dot1x {default} method1 [method2...]`	Creates an 802.1X port-based authentication method list. To create a default list that is used when a named list is *not* specified in the **aaa authentication** command, use the **default** keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports. For *method1*, enter the **group radius** keywords to use the list of all RADIUS servers for authentication. Though other keywords are visible in the command-line help string, only the **group radius** keywords are supported.
Step 3	`Router(config)# dot1x system-auth-control`	Globally enables 802.1X port-based authentication.
Step 4	`Router(config)# aaa authorization network {default} group radius`	(Optional) Configures the switch to use user-RADIUS authorization for all network-related service requests such as VLAN assignment.
Step 5	`Router(config)# radius-server host ip-address`	Specifies the IP address of the RADIUS server.
Step 6	`Router(config)# radius-server key string`	Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 7	`Router(config)# mls acl tcam static-share`	(Optional) Enables static sharing, which allows more efficient use of the TCAM when a number of interfaces have the same PACL and VLAN-based features.
Step 8	`Router(config)# interface type[1] slot/port`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 9	`Router(config-if)# switchport mode access`	Sets the port to access mode only if you configured the RADIUS server in previous steps.

Step 10	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the interface.
		The **no** form of the command disables port-based authentication on the interface.
		For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.

Step 11	`Router(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the interface.

	Command	Purpose
Step 12	`Router(config)# end`	Returns to privileged EXEC mode.
Step 13	`Router# show dot1x all`	Verifies your entries. Check the Status column in the 802.1X Port Summary section of the display. An *enabled* status means the port-control value is set either to **auto** or to **force-unauthorized**.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable AAA and 802.1X on Fast Ethernet port 5/1:

```
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# mls acl tcam static-share
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all

Sysauthcontrol              Enabled
Dot1x Protocol Version            2

Dot1x Info for GigabitEthernet1/7
----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by any of the following:

- 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, which 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 configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip radius source-interface interface_name`	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 2	`Router(config)# radius-server host {hostname | ip_address}`	Configures the RADIUS server host name or IP address on the switch. If you want to use multiple RADIUS servers, reenter this command.
Step 3	`Router(config)# radius-server key string`	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.

When you configure the RADIUS server parameters, note the following information:

- For *hostname* or *ip_address,* specify the host name or IP address of the remote RADIUS server.
- 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*, spaces within and at the end of the key are used. 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 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 retransmit**, and the **radius-server key** global configuration commands. For more information, see the *Cisco IOS Security Configuration Guide, Release 12.2,* publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

  and the *Cisco IOS Security Command Reference*, Release 12.2, publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, see the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46
Router(config)# radius-server key rad123
```

Configuring 802.1X Authenticator Host Mode

An 802.1X-enabled port can grant access to a single client or multiple clients as described in the "802.1X Host Modes" section on page 60-9.

To configure the host mode of an 802.1X-authorized port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the interface.
		The **no** form of the command disables port-based authentication on the interface.
		For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.

Step 3	`Router(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the interface.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication host-mode single-host`
	`Router(config-if)# authentication host-mode multi-host`
	`Router(config-if)# authentication host-mode multi-domain`
	`Router(config-if)# authentication host-mode multi-auth`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x host-mode {single-host | multi-host}`
		Allows a single authenticated host (client) on an authorized port.
		Allows multiple clients on an authorized port when one client is authenticated.
		Allows a single IP phone and a single data client to independently authenticate on an authorized port.
		Allows a single IP phone and multiple data clients to independently authenticate on an authorized port.

Step 5	`Router(config-if)# authentication open`	(Optional) With Cisco IOS Release 12.2(33)SXI or later releases, enables pre-authentication open access.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface` *type*[1] *slot/port*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable 802.1X on Fast Ethernet interface 5/1 and to allow multiple hosts:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# authentication host-mode multi-host
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x host-mode multi-host
```

Enabling Fallback Authentication

On a port in multiauth mode, either or both of MAB and web-based authentication can be configured as fallback authentication methods for non-802.1X hosts (those that do not respond to EAPOL). You can configure the order and priority of the authentication methods.

For detailed configuration information for MAB, see the "Configuring MAC Authentication Bypass" section on page 60-57.

For detailed configuration information for web-based authentication, see Chapter 61, "Configuring Web-Based Authentication."

In Cisco IOS Release 12.2(33)SXI and later releases, to enable fallback authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip admission name rule-name proxy http`	Configures an authentication rule for web-based authentication.
Step 2	`Router(config)# fallback profile profile-name`	Creates a fallback profile for web-based authentication.
Step 3	`Router(config-fallback-profile)# ip access-group rule-name in`	Specifies the default ACL to apply to network traffic before web-based authentication.
Step 4	`Router(config-fallback-profile)# ip admission name rule-name`	Associates an IP admission rule with the profile, and specifies that a client connecting by web-based authentication uses this rule.
Step 5	`Router(config-fallback-profile)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.
Step 7	`Router(config-if)# authentication port-control auto`	Enables authentication on the port.
Step 8	`Router(config-if)# authentication order method1 [method2] [method3]`	(Optional) Specifies the fallback order of authentication methods to be used. The three values of *method*, in the default order, are **dot1x**, **mab**, and **webauth**. The specified order also determines the relative priority of the methods for reathentication, from highest to lowest.
Step 9	`Router(config-if)# authentication priority method1 [method2] [method3]`	(Optional) Overrides the relative priority of authentication methods to be used. The three values of *method*, in the default order of priority, are **dot1x**, **mab**, and **webauth**.
Step 10	`Router(config-if)# authentication event fail action next-method`	Specifies that the next configured authentication method will be used if authentication fails.
Step 11	`Router(config-if)# mab [eap]`	Enables MAC authentication bypass. The optional **eap** keyword specifies that the EAP extension is used during RADIUS authentication.
Step 12	`Router(config-if)# authentication fallback profile-name`	Enables web-based authentication using the specified profile.

	Command	Purpose
Step 13	`Router(config-if)# authentication violation [shutdown | restrict]`	(Optional) Configures the disposition of the port if a security violation occurs. The default action is to shut down the port. If the **restrict** keyword is configured, the port will not be shutdown, but trap entries will be installed for the violating MAC address, and traffic from that MAC address will be dropped.
Step 14	`Router(config-if)# authentication timer inactivity {seconds | server}`	(Optional) Configures the inactivity timeout value for MAB and 802.1X. By default, inactivity aging is disabled for a port. • *seconds*—Specifies inactivity timeout period. The range is from 1 to 65535 seconds. • server—Specifies that the inactivity timeout period value will be obtained from the authentication server.
Step 15	`Router(config-if)# authentication timer restart seconds`	(Optional) Specifies a period after which the authentication process will restart in an attempt to authenticate an unauthorized port. • *seconds*—Specifies the restart period. The range is from 1 to 65535 seconds.
Step 16	`Router(config-if)# exit`	Returns to global configuration mode.
Step 17	`Router(config)# ip device tracking`	Enables the IP device tracking table, which is required for web-based authentication.
Step 18	`Router(config)# ip device tracking [probe {count count | delay delay_interval | interval interval}]`	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.
Step 19	`Router(config)# exit`	Returns to privileged EXEC mode.
Step 20	`Router# show dot1x interface type[1] slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable 802.1X fallback to MAB, and then to enable web-based authentication, on an 802.1X-enabled port:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# ip admission name rule1 proxy http
Router(config)# fallback profile fallback1
Router(config-fallback-profile)# ip access-group default-policy in
Router(config-fallback-profile)# ip admission rule1
```

```
Router(config-fallback-profile)# exit
Router(config)# interface gigabit1/1
Router(config-if)# switchport mode access
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# authentication order dot1x mab webauth
Router(config-if)# mab eap
Router(config-if)# authentication fallback fallback1
Router(config-if)# exit
Router(config)# ip device tracking
Router(config)# exit
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# ip admission name rule1 proxy http
Router(config)# fallback profile fallback1
Router(config-fallback-profile)# ip access-group default-policy in
Router(config-fallback-profile)# ip admission rule1
Router(config-fallback-profile)# exit
Router(config)# interface gigabit1/1
Router(config-if)# switchport mode access
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x fallback fallback1
Router(config-if)# exit
Router(config)# ip device tracking
Router(config)# exit
```

Enabling Periodic Reauthentication

With Release 12.2(33)SXH and later releases, you can enable periodic 802.1X client reauthentication and specify how often it occurs. You can specify the reauthentication period manually or you can use the session-timeout period specified by the RADIUS server. If you enable reauthentication without specifying a time period, the number of seconds between reauthentication attempts is 3600.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication periodic`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x reauthentication`
		Enables periodic reauthentication of the client, which is disabled by default.

	Command	Purpose

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication timer reauthenticate [seconds | server]`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x timeout reauth-period [seconds | server]`
		Specifies the number of seconds between reauthentication attempts using these keywords:
		• *seconds*—Sets the number of seconds from 1 to 65535; the default is 3600 seconds.
		• **server**—Sets the number of seconds based on the value of the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).
		This command affects the operation of the switch only if periodic reauthentication is enabled.

Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication periodic
Router(config-if)# authentication timer reauthenticate 4000
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period 4000
```

Manually Reauthenticating the Client Connected to a Port

Note Reauthentication does not disturb the status of an already authorized port.

To manually reauthenticate the client connected to a port, perform this task:

	Command	Purpose
Step 1	`Router# dot1x re-authenticate interface type[1] slot/port`	Manually reauthenticates the client connected to a port.
Step 2	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 5/1:

```
Router# dot1x re-authenticate interface fastethernet 5/1
```

Initializing Authentication for the Client Connected to a Port

Note Initializing authentication disables any existing authentication before authenticating the client connected to the port.

To initialize the authentication for the client connected to a port, perform this task:

	Command	Purpose
Step 1	`Router# dot1x initialize interface` *type*[1] *slot/port*	Initializes the authentication for the client connected to a port.
Step 2	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to initialize the authentication for the client connected to Fast Ethernet port 5/1:

```
Router# dot1x initialize interface fastethernet 5/1
```

Removing 802.1X Client Information

To completely delete all existing supplicants from an interface or from all the interfaces on the switch, perform this task:

	Command	Purpose
Step 1	`Router# clear dot1x interface` *type*[1] *slot/port*	Removes 802.1X client information for the client connected to the specified port.
	`Router# clear dot1x all`	Removes 802.1X client information for all clients connected to all ports.
Step 2	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to remove 802.1X client information for the client connected to Fast Ethernet port 5/1:

```
Router# clear dot1x interface fastethernet 5/1
```

Clearing Authentication Sessions

To clear all or selected authentication sessions, perform this task:

Command	Purpose
`Router# clear authentication sessions [handle handle] [interface interface] [mac mac] [method method]`	Clears current authentication sessions. With no options specified, all current active sessions will be cleared. The keywords can be added and combined to clear specific sessions or subset of sessions.

This example shows how to clear all MAB authentication sessions connected to Fast Ethernet port 5/1:

```
Router# clear authentication sessions interface fastethernet 5/1 method mab
```

Changing 802.1X Timeouts

You can change several 802.1X timeout attributes using the **dot1x timeout {attribute}** *seconds* command form in the interface configuration mode. This section shows in detail how to change the quiet period timeout, followed by descriptions of how to change other 802.1X timeouts using the same command form.

Setting the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time and then tries again. The **dot1x timeout quiet-period** interface configuration command controls the idle period. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a number smaller than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# dot1x timeout quiet-period seconds`	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set the quiet period on the switch to 30 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout quiet-period 30
```

This example shows how to restore the default quiet period on the switch:

```
Router(config-if)# no dot1x timeout quiet-period
```

Setting the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time), and then retransmits the frame.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To change the amount of time that the switch switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request, use the **dot1x timeout tx-period** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x timeout tx-period** command.

This example shows how to set 60 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout tx-period 60
```

Setting the Switch-to-Client Retransmission Time for EAP-Request Frames

The client notifies the switch that it received the EAP-request frame. If the switch does not receive this notification, the switch waits a set period of time, and then retransmits the frame.

To set the amount of time that the switch waits for notification, use the **dot1x timeout supp-timeout** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x supp-timeout** command.

This example shows how to set the switch-to-client retransmission time for the EAP-request frame to 25 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout supp-timeout 25
```

Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets

The authentication server notifies the switch each time it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the switch waits a set period of time and then retransmits the packet.

To set the value for the retransmission of Layer 4 packets from the switch to the authentication server, use the **dot1x timeout server-timeout** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x server-timeout** command.

This example shows how to set the switch-to-authentication-server retransmission time for Layer 4 packets to 25 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout server-timeout 25
```

Setting the Switch-to-Client Frame Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To set the switch-to-client frame retransmission number, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# dot1x max-req` *count*	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set 5 as the number of times that the switch sends an EAP-request/identity request before restarting the authentication process:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x max-req 5
```

Setting the Reauthentication Number

You can also change the number of times that the switch restarts the authentication process before the port changes to the unauthorized state.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To set the reauthentication number, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# dot1x max-reauth-req` *count*	Sets the number of times that the switch restarts the authentication process before the port changes to the unauthorized state. The range is 0 to 10; the default is 2.

	Command	Purpose
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set 4 as the number of times that the switch restarts the authentication process before the port changes to the unauthorized state:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x max-reauth-req 4
```

Configuring IEEE 802.1X Accounting

Enabling AAA system accounting with 802.1X accounting allows system reload events to be sent to the accounting RADIUS server for logging. The server then can determine that all active 802.1X sessions are closed.

Because RADIUS uses the unreliable UDP transport protocol, accounting messages might be lost due to poor network conditions. If the switch does not receive the accounting response message from the RADIUS server after a configurable number of retransmissions of an accounting request, this system message appears:

```
Accounting message %s for session %s failed to receive Accounting Response.
```

When the stop message is not sent successfully, this message appears:

```
00:09:55: %RADIUS-3-NOACCOUNTINGRESPONSE: Accounting message Start for session
172.20.50.145 sam 11/06/03 07:01:16 11000002 failed to receive Accounting Response.
```

Note You must configure the RADIUS server to perform accounting tasks, such as logging start, stop, and interim-update messages and time stamps. To turn on these functions, enable logging of "Update/Watchdog packets from this AAA client" in your RADIUS server Network Configuration tab. Next, enable "CVS RADIUS Accounting" in your RADIUS server System Configuration tab.

To configure 802.1X accounting after AAA is enabled on your switch, perform this task:

	Command	Purpose
Step 1	`Router(config)# aaa accounting dot1x default start-stop group radius`	Enables 802.1X accounting using the list of all RADIUS servers.
Step 2	`Router(config)# aaa accounting system default start-stop group radius`	(Optional) Enables system accounting (using the list of all RADIUS servers) and generates system accounting reload event messages when the switch reloads.
Step 3	`Router(config)# end`	Returns to privileged EXEc mode.
Step 4	`Router# show running-config`	Verifies your entries.

Use the **show radius statistics** privileged EXEC command to display the number of RADIUS messages that do not receive the accounting response message.

This example shows how to configure 802.1X accounting. The first command configures the RADIUS server, specifying 1813 as the UDP port for accounting:

```
Router(config)# radius-server host 172.120.39.46 auth-port 1812 acct-port 1813 key rad123
Router(config)# aaa accounting dot1x default start-stop group radius
Router(config)# aaa accounting system default start-stop group radius
```

Configuring VLAN User Distribution

With Cisco IOS Release 12.2(33)SXI1 and later releases, you can define a VLAN group that contains multiple VLANs. For VLAN load balancing, you can then configure the RADIUS server to supply a VLAN group name as part of the response to a user during 802.1X authentication. If the supplied VLAN group name is found among the VLAN group names that you have defined, the newly authenticated user is placed in the least populated VLAN within the VLAN group.

To configure a VLAN group, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan group` *group-name* `vlan-list` *vlan-list*	Creates a VLAN group or adds VLANs to an existing VLAN group. • *group-name*—The name of the VLAN group. The name may contain up to 32 characters and must begin with a letter. • **vlan-list** *vlan-list*—The VLANs that belong to the VLAN group. Group members can be specified as a single VLAN ID, a list of VLAN IDs, or a VLAN ID range. Multiple entries are separated by a hyphen (-) or a comma (,).
Step 2	`Router(config)# no vlan group` *group-name* `vlan-list` *vlan-list*	(Optional) Removes the members specified by *vlan-list* from a VLAN group. **Note** When no VLANs remain in the VLAN group, the VLAN group is deleted.
Step 3	`Router# show vlan group [group-name` *group-name*`]`	Displays the VLANs and VLAN ranges that are members of the specified VLAN group or of all VLAN groups.

This example shows how to map VLANs 7 through 9 and 11 to a VLAN group:

```
Router(config)# vlan group ganymede vlan-list 7-9,11
```

This example shows how to remove VLAN 7 from a VLAN group:

```
Router(config)# no vlan group ganymede vlan-list 7
```

Configuring a Guest VLAN

With Cisco IOS Release 12.2(33)SXH and later releases, when you configure a guest VLAN, clients that are not 802.1X-capable are put into the guest VLAN when the server does not receive a response to its EAP request/identity frame. Clients that are 802.1X-capable but that fail authentication are not granted network access. When operating as a guest VLAN, a port functions in multiple-hosts mode regardless of the configured host mode of the port.

To configure a guest VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	`Router(config-if)# switchport mode access`
	or
	`Router(config-if)# switchport mode private-vlan host`
		Sets the port to access mode.
		or
		Configures the port as a private VLAN host port.
		The guest VLAN is not supported on routed or trunk ports.

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables authentication on the port.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)# authentication event no-response action authorize vlan` *vlan-id* **Releases earlier than Release 12.2(33)SXI:** `Router(config-if)# dot1x guest-vlan` *vlan-id*	Specifies an active VLAN as a guest VLAN. The range is 1 to 4094. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as a guest VLAN.

Step 5	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# dot1x pae authenticator`
	or
	`Router(config-if)# mab`
		Specifies whether the port authentication method is 802.1X or MAC address bypass.
		In Cisco IOS Release 12.2(33)SXH and earlier releases, this command is not needed, and the method will be 802.1X.

Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface` *type slot/port*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable VLAN 2 as an 802.1X guest VLAN:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# authentication event no-response action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x guest-vlan 2
```

This example shows how to set 3 seconds as the client notification timeout on the switch, to set 15 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before resending the request, and to enable VLAN 2 as an 802.1X guest VLAN when an 802.1X port is connected to a DHCP client:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config-if)# dot1x timeout supp-timeout 3
Router(config-if)# dot1x timeout tx-period 15
Router(config-if)# authentication event no-response action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config-if)# dot1x timeout supp-timeout 3
Router(config-if)# dot1x timeout tx-period 15
Router(config-if)# dot1x guest-vlan 2
```

Configuring a Restricted VLAN

When you configure a restricted VLAN on a switch, clients that are 802.1X-compliant are moved into the restricted VLAN when the authentication server does not receive a valid username and password. When operating as a restricted VLAN, a port functions in single-host mode regardless of the configured host mode of the port.

To configure a restricted VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	`Router(config-if)# switchport mode access`
	or
	`Router(config-if)# switchport mode private-vlan host`
		Sets the port to access mode.
		or
		Configures the port as a private-VLAN host port.

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the port.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)# authentication event fail [retry` *retries*`] action authorize vlan` *vlan-id* **Releases earlier than Release 12.2(33)SXI:** `Router(config-if)# dot1x auth-fail vlan` *vlan-id*	Specifies an active VLAN as a restricted VLAN. The range for *vlan-id* is 1 to 4094. (Optional) The **retry** keyword specifies a number of authentication attempts to allow before a port moves to the restricted VLAN. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as a restricted VLAN.

	Command	Purpose
Step 5	**Releases earlier than Release 12.2(33)SXI:** `Router(config-if)# dot1x auth-fail max-attempts max-attempts`	(Optional) The **max-attempts** keyword specifies a number of authentication attempts to allow before a port moves to the restricted VLAN.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To disable and remove the restricted VLAN, use the **no** form of the **authentication event fail** command or the **dot1x auth-fail** command. The port returns to the unauthorized state.

You can configure the maximum number of authentication attempts allowed before a user is assigned to the restricted VLAN.

- In Cisco IOS Release 12.2(33)SXI and later releases, you can set the number of attempts by using the **retry** keyword in the **authentication event fail** [**retry** *retries*] **action authorize vlan** command. The range of *retries* (allowable authentication attempts) is 1 to 5. The default is 2 attempts.
- In Cisco IOS Release 12.2(33)SXH, you can set the number of attempts by using the **dot1x auth-fail max-attempts** *max-attempts* interface configuration command. The range of *max-attempts* (allowable authentication attempts) is 1 to 3. The default is 3 attempts.

This example shows how to enable VLAN 2 as a restricted VLAN, with assignment of a host after 3 failed attempts:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# authentication event fail retry 3 action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x auth-fail vlan 2
Router(config-if)# dot1x auth-fail max-attempts 3
```

Configuring the Inaccessible Authentication Bypass Feature

You can configure the inaccessible bypass feature, also referred to as critical authentication or the AAA fail policy.

To configure the port as a critical port and enable the inaccessible authentication bypass feature, perform this task:

	Command	Purpose
Step 1	`Router(config)# radius-server dead-criteria time` *time* **tries** *tries*	(Optional) Sets the conditions that are used to decide when a RADIUS server is considered unavailable or *dead*. The range for *time* is from 1 to 120 seconds. The switch dynamically determines the default *seconds* value that is 10 to 60 seconds. The range for *tries* is from 1 to 100. The switch dynamically determines the default *tries* parameter that is 10 to 100.
Step 2	`Router(config)# radius-server deadtime` *minutes*	(Optional) Sets the number of minutes that a RADIUS server is not sent requests. The range is from 0 to 1440 minutes (24 hours). The default is 0 minutes.

	Command	Purpose
Step 3	`Router(config)# radius-server host ip-address [acct-port udp-port] [auth-port udp-port] [key string] [test username name [idle-time time] [ignore-acct-port] [ignore-auth-port]]`	(Optional) Configures the RADIUS server parameters by using these keywords: • **acct-port** *udp-port*—Specifies the UDP port for the RADIUS accounting server. The range for the UDP port number is from 0 to 65536. The default is 1646. • **auth-port** *udp-port*—Specifies the UDP port for the RADIUS authentication server. The range for the UDP port number is from 0 to 65536. The default is 1645. **Note** You should configure the UDP port for the RADIUS accounting server and the UDP port for the RADIUS authentication server to nondefault values. • **key** *string*—Specifies the authentication and encryption key for all RADIUS communication between the switch and the RADIUS daemon. **Note** You can also configure the authentication and encryption key by using the **radius-server key** {**0** *string* | **7** *string* | *string*} global configuration command. • **test username** *name*—Enables automated testing of the RADIUS server status, and specify the username to be used. • **idle-time** *time*—Sets the interval of time in minutes after which the switch sends test packets to the server. The range is from 1 to 35791 minutes. The default is 60 minutes (1 hour). • **ignore-acct-port**—Disables testing on the RADIUS server accounting port. • **ignore-auth-port**—Disables testing on the RADIUS server authentication port.
Step 4	`Router(config)# dot1x critical eapol`	(Optional) Specifies that the switch sends an EAPOL-Success message when the switch successfully authenticates the critical port.
Step 5	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 6	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication critical recovery delay milliseconds`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config)# dot1x critical recovery delay milliseconds`
		(Optional) Sets the recovery delay period during which the switch waits to reinitialize a critical port when a RADIUS server that was unavailable becomes available. The range is from 1 to 10000 milliseconds. The default is 1000 milliseconds (a port can be reinitialized every second).

	Command	Purpose

Step 7	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication event server dead action authorize [vlan vlan-id]`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x critical [vlan vlan-id]`
		Enables the inaccessible authentication bypass feature, authorizing ports on the specified VLAN when the AAA server is unreachable. If no VLAN is specified, the access VLAN will be used.
		
		Note The **vlan** keyword is only available on a switch port.

Step 8	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication event server alive action reinitialize`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x critical recovery action reinitialize`
		Configures the inaccessible authentication bypass recovery feature, specifying that the recovery action is to authenticate the port when an authentication server becomes available.

Step 9	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 10	`Router# show dot1x [interface type slot/port]`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To return to the RADIUS server default settings, use the **no radius-server dead-criteria**, the **no radius-server deadtime**, and the **no radius-server host** global configuration commands. To return to the default settings of inaccessible authentication bypass, use the **no dot1x critical eapol** global configuration command. To disable inaccessible authentication bypass, use the **no authentication event server dead action authorize** (or **no dot1x critical**) interface configuration command.

This example shows how to configure the inaccessible authentication bypass feature:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# radius-server dead-criteria time 30 tries 20
Router(config)# radius-server deadtime 60
Router(config)# radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 key abc1234 test
username user1 idle-time 30
Router(config)# dot1x critical eapol
Router(config)# authentication critical recovery delay 2000
Router(config)# interface gigabitethernet 1/1
Router(config-if)# authentication event server dead action authorize vlan 123
Router(config-if)# authentication event server alive action reinitialize
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# radius-server dead-criteria time 30 tries 20
Router(config)# radius-server deadtime 60
Router(config)# radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 key abc1234 test
username user1 idle-time 30
Router(config)# dot1x critical eapol
Router(config)# dot1x critical recovery delay 2000
Router(config)# interface gigabitethernet 1/1
Router(config-if)# dot1x critical vlan 123
Router(config-if)# dot1x critical recovery action reinitialize
```

Enabling Critical Voice VLAN Support

Release 12.2(33)SXJ1 and later releases support the critical voice VLAN support feature (see the "Understanding 802.1X Authentication Critical Voice VLAN Support" section on page 60-19).

Note When enabling critical voice VLAN support, follow these guidelines and restrictions:

- Use different VLANs for voice and data.
- The voice VLAN must be configured on the switch (see "Configuring Voice Traffic Support" section on page 15-5).

Enabling Critical Voice VLAN Support in Multidomain Authentication Mode

To enable critical voice VLAN support in multidomain authentication (MDA) mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# authentication event server dead action reinitialize vlan` *critical_data_vlan_id*	Configures a critical data VLAN. **Note** Only required if the **authentication event server dead action authorize vlan** *critical_data_vlan_id* command is not configured on the port (see the "Configuring the Inaccessible Authentication Bypass Feature" section on page 60-53).
Step 3	`Router(config-if)# authentication event server dead action authorize voice`	Enables the critical voice VLAN support feature, which puts phone traffic into the configured voice VLAN of a port if the authentication server becomes unreachable.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable the critical voice VLAN support feature in MDA mode when the **authentication event server dead action authorize vlan** *critical_data_vlan_id* command is also configured on the port as part of the inaccessible authentication bypass feature:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action authorize voice
```

This example shows how to enable the critical voice VLAN support feature in MDA mode when inaccessible authentication bypass is not configured on the port:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action reinitialize vlan 10
Router(config-if)# authentication event server dead action authorize voice
```

Enabling Critical Voice VLAN Support in Multiauthentication Mode

To enable critical voice VLAN support in multiauthentication mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# authentication event server dead action reinitialize vlan` *critical_data_vlan_id*	Configures a critical data VLAN.
Step 3	`Router(config-if)# authentication event server dead action authorize voice`	Enables the critical voice VLAN support feature, which puts phone traffic into the configured voice VLAN of a port if the authentication server becomes unreachable.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable the critical voice VLAN support feature in multiauthentication mode:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action reinitialize vlan 10
Router(config-if)# authentication event server dead action authorize voice
```

Configuring MAC Authentication Bypass

To configure MAC authentication bypass on an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control {auto | force-authorized | force-unauthorized}`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control {auto | force-authorized | force-unauthorized}`
		Enables 802.1X authentication on the port.
		The keywords have these meanings:
		• **auto**—Allows only EAPOL traffic until successful authentication.
		• **force-authorized**—Allows all traffic, requires no authentication.
		• **force-unauthorized**—Allows no traffic.

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# mab [eap]`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x mac-auth-bypass [eap]`
		Enables MAC authentication bypass on the interface.
		(Optional) Use the **eap** keyword to configure the switch to use EAP for authorization.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# no mab eap`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# no dot1x mac-auth-bypass eap`
		(Optional) Disables the use of EAP for authorization if EAP was previously configured using the **mab eap** or the **dot1x mac-auth-bypass eap** command.

	Command	Purpose

Step 5	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# no mab`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# no dot1x mac-auth-bypass`
		(Optional) Disables MAC authentication bypass on the interface.
		Note When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.

Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface` *type slot/port*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note To use MAC authentication bypass on a routed port, ensure that MAC address learning is enabled on the port.

This example shows how to enable MAC authentication bypass on a port:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# mab
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x mac-auth-bypass
```

Configuring NAC Layer 2 IEEE 802.1X Validation

With Cisco IOS Release 12.2(33)SXH and later releases, you can configure NAC Layer 2 IEEE 802.1X validation, which is also referred to as 802.1X authentication with a RADIUS server. NAC Layer 2 IEEE 802.1X configuration is the same as 802.1X configuration with the additional step of configuring the RADIUS server with a posture token and VLAN assignment.

To configure NAC Layer 2 IEEE 802.1X validation, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the interface.
		The **no** form of the command disables port-based authentication on the interface.
		For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.

	Command	Purpose

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication periodic`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x reauthentication`
		Enables periodic reauthentication of the client, which is disabled by default.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication timer reauthenticate [seconds | server]`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x timeout reauth-period [seconds | server]`
		Specifies the number of seconds between reauthentication attempts using these keywords:
		• *seconds*—Sets the number of seconds from 1 to 65535; the default is 3600 seconds.
		• **server**—Sets the number of seconds based on the value of the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).
		This command affects the operation of the switch only if periodic reauthentication is enabled.

Step 5	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 6	`Router# show dot1x interface type slot/port`	Verifies your 802.1X authentication configuration. Verify that a NAC posture token is displayed with the 802.1X authentication configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure NAC Layer 2 IEEE 802.1X validation:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config)# authentication port-control auto
Router(config-if)# authentication periodic
Router(config-if)# authentication timer reauthenticate server
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config)# dot1x port-control auto
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period server
```

Configuring NAC Agentless Audit Support

To support the NAC audit architecture for agentless clients, the switch must snoop an authenticating 802.1X client's IP address, create and assign a unique session identifier for the agentless client, and pass this information to the RADIUS server for sharing with the NAC audit server. To allow the switch to obtain and share this information, you must enable ARP and DHCP snooping on the switch and you must enable the sending of certain RADIUS attributes.

To configure the RADIUS and tracking settings to support NAC agentless audit, perform this task:

	Command	Purpose
Step 1	`Router(config)# radius-server attribute 8 include-in-access-req`	Configures the switch to send the Framed-IP-Address RADIUS attribute (Attribute[8]) in access-request or accounting-request packets.
Step 2	`Router(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes (VSAs) (specifically audit-session-id) in RADIUS Access-Requests generated by the switch during the authentication phase.
Step 3	`Router(config)# radius-server vsa send accounting`	Allows VSAs to be included in subsequent RADIUS Accounting-Requests.
Step 4	`Router(config)# ip device tracking`	Enables the IP device tracking table.
Step 5	`Router(config)# ip device tracking [probe {count count | delay delay_interval | interval interval}]`	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.

Configuring the Switch for DACLs or Redirect URLs

To configure switch ports to accept DACLs or redirect URLs from the RADIUS server during authentication of an attached host, perform this task:

	Command	Purpose
Step 1	`Router# config terminal`	Enters global configuration mode.
Step 2	`Router(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes (VSAs) in RADIUS Access-Requests generated by the switch during the authentication phase. **Note** This step is necessary only with redirect URLs or when DACLs are downloaded using VSAs rather than the Filter-ID attribute.
Step 3	`Router(config)# ip device tracking`	Enables the IP device tracking table.

	Command	Purpose
Step 4	`Router(config)# ip device tracking [probe {count count | delay delay_interval | interval interval}]`	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.
Step 5	`Router(config)# ip access-list extended dacl-name`	Configures an ACL that will be referenced by the VSA or Filter-ID attribute. **Note** This step is not necessary for DACLs defined on the RADIUS server and downloaded using VSAs.
Step 6	`Router(config-std-nacl)# {permit | deny} ...`	Defines the ACL. **Note** The source address must be ANY.
Step 7	`Router(config-std-nacl)# exit`	Returns to global configuration mode.
Step 8	`Router(config)# ip access-list extended acl-name`	Configures a default ACL for the ports.
Step 9	`Router(config-std-nacl)# {permit | deny} ...`	Defines the ACL.
Step 10	`Router(config-std-nacl)# exit`	Returns to global configuration mode.
Step 11	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.
Step 12	`Router(config-if)# ip access-group acl-name in`	Applies the default static ACL on the interface.
Step 13	`Router(config-if)# exit`	Returns to global configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure a switch for a downloadable policy:

```
Router# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# radius-server vsa send authentication
Router(config)# ip device tracking
Router(config)# ip access-list extended my_dacl
Router(config-ext-nacl)# permit tcp any host 10.2.3.4
Router(config-ext-nacl)# exit
Router(config)# ip access-list extended default_acl
Router(config-ext-nacl)# permit ip any any
Router(config-ext-nacl)# exit
Router(config)# interface fastEthernet 2/13
Router(config-if)# ip access-group default_acl in
Router(config-if)# exit
```

Configuring a Port to Ignore CoA Commands

To configure the switch to disregard a CoA command to shut down or bounce a specific port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# [no] authentication command disable-port ignore`	Configures the switch to ignore any CoA command requesting that this port be administratively shut-down.
	`Router(config-if)# [no] authentication command bounce-port ignore`	Configures the switch to ignore any CoA command requesting that this port be held down for a period of time.
Step 3	`Router(config-if)# exit`	Returns to global configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring 802.1X Authentication with WoL

Note Wake-on-LAN (WoL) is supported in multiauthentication (multiauth) mode only in releases where CSCti92970 is resolved.

To enable 802.1X authentication with WoL, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication control-direction {both | in}`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x control-direction {both | in}`
		Enables 802.1X authentication with WoL on the port, and uses these keywords to configure the port as bidirectional or unidirectional.
		• **both**—Sets the port as bidirectional. The port cannot receive packets from or send packets to the host. By default, the port is bidirectional.
		• **in**—Sets the port as unidirectional. The port can send packets to the host but cannot receive packets from the host.

Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x interface` *type slot/port*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To disable 802.1X authentication with WoL, use the **no authentication control-direction** (or the **no dot1x control-direction**) interface configuration command.

This example shows how to enable 802.1X authentication with WoL and set the port as bidirectional:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication control-direction both
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x control-direction both
```

Disabling 802.1X Authentication on the Port

You can disable 802.1X authentication on the port by using the **no dot1x pae** interface configuration command.

To disable 802.1X authentication on the port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# no dot1x pae authenticator`	Disables 802.1X authentication on the port.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x interface` *type slot/port*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To configure the port as an 802.1X port access entity (PAE) authenticator, which enables 802.1X on the port but does not allow clients connected to the port to be authorized, use the **dot1x pae authenticator** interface configuration command.

This example shows how to disable 802.1X authentication on the port:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# no dot1x pae authenticator
```

Resetting the 802.1X Configuration to the Default Values

To reset the 802.1X configuration to the default values, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# dot1x default`	Resets the configurable 802.1X parameters to the default values.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to reset a port's 802.1X authentication settings to the default values:

```
Router(config)# interface gigabitethernet 3/27
Router(config-if)# dot1x default
```

Configuring an Authenticator and a Supplicant Switch with NEAT

Release 12.2(33)SXJ and later releases support Network Edge Access Topology (NEAT), which requires one switch to be configured as a supplicant and to be connected to an authenticator switch.

For overview information, see the "Understanding 802.1x Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)" section on page 60-27.

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

To configure a switch as an authenticator, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# cisp enable`	Enables CISP.
Step 3	`Router(config)# interface` *interface-id*	Specifies the port to be configured, and enters interface configuration mode.
Step 4	`Router(config-if)# switchport mode access`	Sets the port mode to **access**.
Step 5	`Router(config-if)# authentication port-control auto`	Sets the port-authentication mode to auto.
Step 6	`Router(config-if)# dot1x pae authenticator`	Configures the interface as a port access entity (PAE) authenticator.
Step 7	`Router(config-if)# spanning-tree portfast`	Enables PortFast on an access port connected to a single workstation or server.
Step 8	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 9	`Router# show running-config interface` *interface-id*	Verifies your configuration.
Step 10	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch as an 802.1x authenticator:

```
Router# configure terminal
Router(config)# cisp enable
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport mode access
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# spanning-tree portfast trunk
```

To configure a switch as a supplicant, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# cisp enable`	Enables CISP.
Step 3	`Router(config)# dot1x credentials profile`	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 4	`Router(config)# username suppswitch`	Creates a username.
Step 5	`Router(config)# password password`	Creates a password for the new username.
Step 6	`Router(config)# dot1x supplicant force-multicast`	Forces the switch to send *only* multicast EAPOL packets when it receives either unicast or multicast packets, which allows NEAT to work on the supplicant switch in all host modes.
Step 7	`Router(config)# interface interface-id`	Specifies the port to be configured, and enters interface configuration mode.
Step 8	`Router(config-if)# switchport trunk encapsulation dot1q`	Sets the port to trunk mode.
Step 9	`Router(config-if)# switchport mode trunk`	Configures the interface as a VLAN trunk port.
Step 10	`Router(config-if)# dot1x pae supplicant`	Configures the interface as a port access entity (PAE) supplicant.
Step 11	`Router(config-if)# dot1x credentials profile-name`	Attaches the 802.1x credentials profile to the interface.
Step 12	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 13	`Router# show running-config interface interface-id`	Verifies your configuration.
Step 14	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch as a supplicant:

```
Router# configure terminal
Router(config)# cisp enable
Router(config)# dot1x credentials test
Router(config)# username suppswitch
Router(config)# password myswitch
Router(config)# dot1x supplicant force-multicast
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# dot1x pae supplicant
Router(config-if)# dot1x credentials test
Router(config-if)# end
```

Displaying Authentication Status and Information

This section describes the **show** commands used to display authentication status and information.

- Displaying 802.1X Status, page 60-66
- Displaying Authentication Methods and Status, page 60-67
- Displaying MAC Authentication Bypass Status, page 60-70

For detailed information about the fields in these displays, see the Cisco IOS Master Command List.

Displaying 802.1X Status

To display the global 802.1X administrative and operational status for the switch or the 802.1X settings for individual ports, perform this task:

Command	Purpose
`Router# show dot1x [all | interface type[1] slot/port]`	Displays the global 802.1X administrative and operational status for the switch. (Optional) Use the **all** keyword to display the global 802.1X status and the 802.1X settings for all interfaces using 802.1X authentication. (Optional) Use the **interface** keyword to display the 802.1X settings for a specific interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to view only the global 802.1X status:

```
Router# show dot1x
Sysauthcontrol              Disabled
Dot1x Protocol Version            2
Critical Recovery Delay         100
Critical EAPOL              Disabled

Router#
```

This example shows how to view the global 802.1X status and the 802.1X settings for all interfaces using 802.1X authentication:

```
Router# show dot1x all
Sysauthcontrol              Disabled
Dot1x Protocol Version            2
Critical Recovery Delay         100
Critical EAPOL              Disabled

Dot1x Info for GigabitEthernet3/27
----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = FORCE_AUTHORIZED
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0

Router#
```

Displaying Authentication Methods and Status

To display the authentication methods and status, perform any of these tasks:

Command	Purpose
`Router# show authentication registrations`	Displays details of all registered methods.
`Router# show authentication interface interface`	Displays authentication information for a specific interface
`Router# show authentication method method`	Lists current authentication sessions that were authorized using the specified method.
`Router# show authentication sessions [handle handle] [interface interface] [mac mac] [method method] [session-id session-id]`	Displays information about current authentication sessions. With no options specified, all current active sessions will be listed. The keywords can be added and combined to display detailed information about specific sessions or subset of sessions.

Table 60-3 shows the possible states of the authentication session.

Table 60-3 Authentication Session States

State	Description
Idle	The session has been initialized and no methods have run yet.
Running	A method is running for this session.
No methods	No method has provided a result for this session.
Authc Success	A method has provided a successful authentication result for the session.
Authc Failed	A method has provided a failed authentication result for the session.
Authz Success	All features have been successfully applied for this session.
Authz Failed	A feature has failed to be applied for this session.

Table 60-4 shows the possible states of the authentication methods.

Table 60-4 Authentication Method States

State	Description
Not run	The method has not run for this session
Running	The method is running for this session.
Failed over	The method has failed and the next method is expected to provide a result.

Table 60-4 Authentication Method States (continued)

State	Description
Success	The method has provided a successful authentication result for the session.
Authc Failed	The method has provided a failed authentication result for the session.

This example shows how to display the registered authentication methods:

```
Router# show authentication registrations
Auth Methods registered with the Auth Manager:
  Handle  Priority  Name
    3        0       dot1x
    2        1       mab
    1        2       webauth
```

This example shows how to display the authentication details for a given interface:

```
Router# show authentication interface g1/23
Client list:
  MAC Address     Domain   Status              Handle          Interface
  0123.4567.abcd  DATA     Authz Success  0xE0000000  GigabitEthernet1/23

Available methods list:
  Handle  Priority  Name
    3        0      dot1x
    2        1      mab
Runnable methods list:
  Handle  Priority  Name
    2        0      mab
    3        1      dot1x
```

This example shows how to display all authentication sessions on the switch:

```
Router# show authentication sessions

Interface  MAC Address     Method   Domain   Status         Session ID
Gi1/48     0015.63b0.f676  dot1x    DATA     Authz Success  0A3462B1000000102983C05C
Gi1/5      000f.23c4.a401  mab      DATA     Authz Success  0A3462B10000000D24F80B58
Gi1/5      0014.bf5d.d26d  dot1x    DATA     Authz Success  0A3462B10000000E29811B94
```

This example shows how to display sessions authorized using a specified authentication method:

```
Router# show authentication method dot1x
Interface  MAC Address     Method   Domain   Status         Session ID
Gi1/48     0015.63b0.f676  dot1x    DATA     Authz Success  0A3462B1000000102983C05C
Gi1/5      0014.bf5d.d26d  dot1x    DATA     Authz Success  0A3462B10000000E29811B94
```

This example shows how to display all authentication sessions on an interface:

```
Router# show authentication sessions interface f1/47

Interface:  FastEthernet1/47
          MAC Address:  Unknown
           IP Address:  Unknown
               Status:  Authz Success
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
        Authorized By:  Guest Vlan
          Vlan Policy:  20
```

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3462C8000000000002763C
      Acct Session ID:  0x00000002
               Handle:  0x25000000

Runnable methods list:
       Method   State
       mab      Failed over
       dot1x    Failed over

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

           Interface:  FastEthernet1/47
         MAC Address:  0005.5e7c.da05
          IP Address:  Unknown
           User-Name:  00055e7cda05
              Status:  Authz Success
              Domain:  VOICE
      Oper host mode:  multi-domain
     Oper control dir:  both
        Authorized By:  Authentication Server
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3462C8000000010002A238
      Acct Session ID:  0x00000003
               Handle:  0x91000001

Runnable methods list:
       Method   State
       mab      Authc Success
       dot1x    Not run
```

This example shows how to display the authentication session for a specified session ID:

```
Router# show authentication sessions session-id 0B0101C70000004F2ED55218

           Interface:  GigabitEthernet9/2
         MAC Address:  0000.0000.0011
          IP Address:  20.0.0.7
            Username:  johndoe
              Status:  Authz Success
              Domain:  DATA
      Oper host mode:  multi-host
     Oper control dir:  both
        Authorized By:  Critical Auth
          Vlan policy:  N/A
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0B0101C70000004F2ED55218
      Acct Session ID:  0x00000003
               Handle:  0x91000001

Runnable methods list:
       Method   State
       mab      Authc Success
       dot1x    Not run
```

This example shows how to display all clients authorized by the specified authentication method:

```
Router# show authentication sessions method mab

No Auth Manager contexts match supplied criteria
```

```
Router# show authentication sessions method dot1x

Interface  MAC Address     Domain   Status         Session ID
Gi9/2      0000.0000.0011  DATA     Authz Success  0B0101C70000004F2ED55218
```

Displaying MAC Authentication Bypass Status

To display the MAB status, perform this task:

Command	Purpose
Router# **show mab** {**all** | **interface** *type*[1] *slot/port*} [*detail*]	Displays MAB authentication details for all interfaces or for a specific interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Table 60-5 shows the possible states of the MAB authentication state machine.

Table 60-5 MAB States

State	Description
INITIALIZE	The authorization session is initialized.
ACQUIRING	The session is obtaining the client MAC address.
AUTHORIZING	The session is waiting for MAC-based authorization.
TERMINATE	The authorization session result has been obtained.

This example shows how to display the brief MAB status for a single interface:

```
Router# show mab interface fa1/1

MAB details for FastEthernet1/1
-------------------------------------
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None
```

This example shows how to display the detailed MAB status for a single interface:

```
Router# show mab interface fa1/1 detail

MAB details for FastEthernet1/1
-------------------------------------
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None

MAB Client List
---------------
Client MAC                = 000f.23c4.a401
MAB SM state              = TERMINATE
Auth Status               = AUTHORIZED
```

Enabling MAC Move

Release 12.2(33)SXI4 and later releases support the Mac move feature. To globally enable MAC move on the switch, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# authentication mac-move permit`	Enables MAC move on the switch.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show running-config`	(Optional) Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to globally enable MAC move on a switch:

```
Router(config)# authentication mac-move permit
```

Enabling MAC Replace

Release 12.2(33)SXI4 and later releases support the Mac replace feature. To enable MAC replace on an interface, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface` *interface-id*	Specifies the port to be configured, and enter interface configuration mode.
Step 3	`Router(config-if)# authentication violation {protect | replace | restrict | shutdown}`	Uses the **replace** keyword to enable MAC replace on the interface. The port removes the current session and initiates authentication with the new host. The other keywords have these effects: • **protect**: the port drops packets with unexpected MAC addresses without generating a system message. • **restrict**: violating packets are dropped by the CPU and a system message is generated. • **shutdown**: the port is error disabled when it receives an unexpected MAC address.
Step 4	`Router(config)# end`	Returns to privileged EXEc mode.
Step 5	`Router# show running-config`	Verifies your entries.
Step 6	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to enable MAC replace on an interface:

```
Router(config)# interface gigabitethernet2/2
Router(config-if)# authentication violation replace
```