



Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication 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, refer to the *Cisco IOS Master Command List*, Release 12.2SX at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html

This chapter consists of these sections:

- [Understanding 802.1X Port-Based Authentication, page 46-1](#)
- [Default 802.1X Port-Based Authentication Configuration, page 46-6](#)
- [802.1X Port-Based Authentication Guidelines and Restrictions, page 46-6](#)
- [Configuring 802.1X Port-Based Authentication, page 46-7](#)
- [Displaying 802.1X Status, page 46-16](#)



Tip

For additional information (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/routers/ps368/tsd_products_support_series_home.html

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client-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 router port and assigns the port to a VLAN before making available any services offered by the router 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.

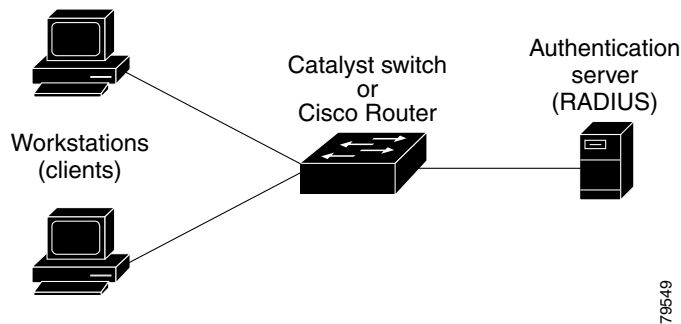
These sections describe IEEE 802.1X port-based authentication:

- [Device Roles, page 46-2](#)
- [Authentication Initiation and Message Exchange, page 46-3](#)
- [Ports in Authorized and Unauthorized States, page 46-4](#)
- [Supported Topologies, page 46-5](#)

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in [Figure 46-1](#).

Figure 46-1 802.1X Device Roles



The specific roles shown in [Figure 46-1](#) are as follows:

- *Client*—The device (workstation) that requests access to the LAN and router services and responds to requests from the router. 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 router whether or not the client is authorized to access the LAN and router services. Because the router 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, 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.
- *Router* (also called the *authenticator* and *back-end authenticator*)—Controls the physical access to the network based on the authentication status of the client. The router 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 router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the router 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 router 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.

Authentication Initiation and Message Exchange

The router or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, the router must initiate authentication when it determines that the port link state transitions from down to up. The router then sends an EAP-request/identity frame to the client to request its identity (typically, the router 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 router during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the router to request the client's identity.



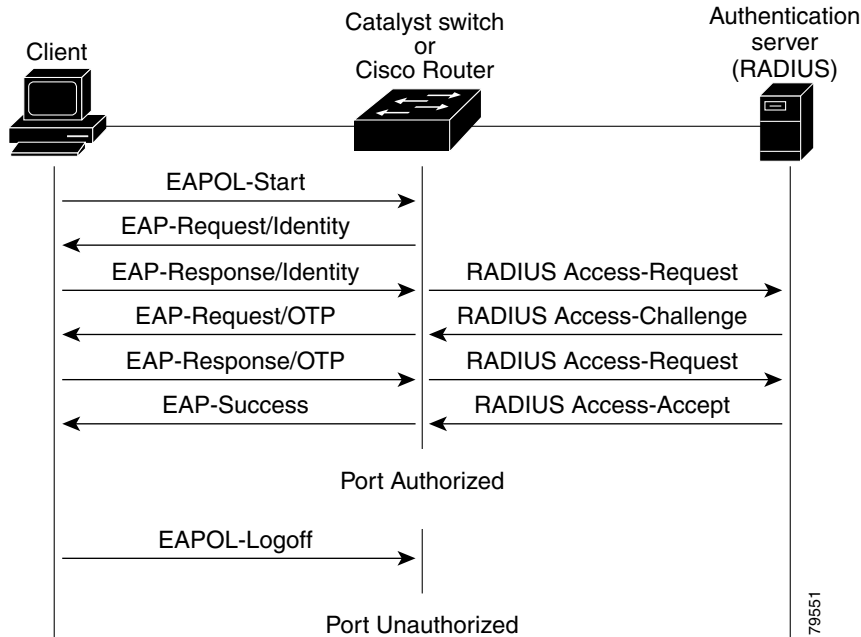
Note

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 46-4](#).

When the client supplies its identity, the router 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 router port becomes authorized. For more information, see the [“Ports in Authorized and Unauthorized States” section on page 46-4](#).

The specific exchange of EAP frames depends on the authentication method being used. [Figure 46-2](#) shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 46-2 Message Exchange



Ports in Authorized and Unauthorized States

The router 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 is connected to an unauthorized 802.1X port, the router 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 **dot1x port-control** interface configuration command 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 router 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 router 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 router 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 router 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 router 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.

Supported Topologies

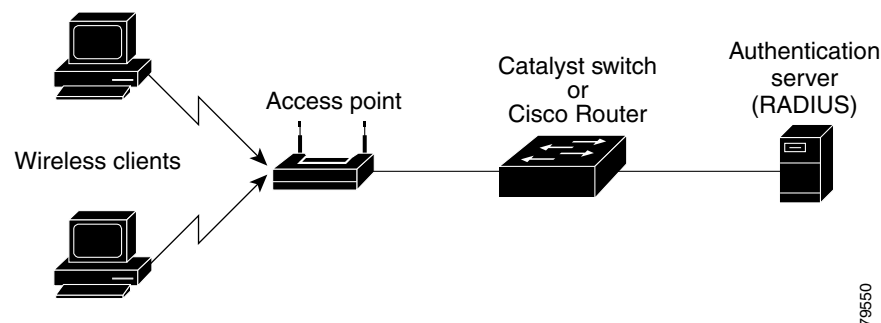
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see [Figure 46-1 on page 46-2](#)), only one client can be connected to the 802.1X-enabled router port. The router detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the router changes the port link state to down, and the port returns to the unauthorized state.

[Figure 46-3](#) shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one client is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the router denies access to the network to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and the wireless access point acts as a client to the router.

Figure 46-3 Wireless LAN Example



Default 802.1X Port-Based Authentication Configuration

Table 46-1 shows the default 802.1X configuration.

Table 46-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server IP address	None specified
RADIUS server UDP authentication port	1812
RADIUS server key	None specified
Per-interface 802.1X protocol enable state	Disabled (force-authorized) Note The port transmits and receives normal traffic without 802.1X-based authentication of the client.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet period	60 seconds (number of seconds that the router remains in the quiet state following a failed authentication exchange with the client)
Retransmission time	30 seconds (number of seconds that the router should wait for a response to an EAP request/identity frame from the client before retransmitting the request)
Maximum retransmission number	2 times (number of times that the router will send an EAP-request/identity frame before restarting the authentication process)
Multiple host support	Disabled
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the router 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 router waits for a reply before retransmitting the response to the server)

802.1X Port-Based Authentication Guidelines and Restrictions

When configuring 802.1X port-based authentication, follow these guidelines and restrictions:

- When 802.1X is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- The 802.1X protocol is supported on both Layer 2 static-access ports and Layer 3 routed ports, but it is not supported on these port types:

- Trunk port—If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.
- EtherChannel port—Before enabling 802.1X on the port, you must first remove it from the EtherChannel port-channel interface. If you try to enable 802.1X on an EtherChannel port-channel interface or on an individual active port in an EtherChannel, an error message appears, and 802.1X is not enabled. If you enable 802.1X on a not-yet active individual port of an EtherChannel, the port does not join the EtherChannel.
- Secure port—You cannot configure a secure port as an 802.1X port. If you try to enable 802.1X on a secure port, an error message appears, and 802.1X is not enabled. If you try to change an 802.1X-enabled port to a secure port, an error message appears, and the security settings are not changed.
- Switch Port Analyzer (SPAN) destination port—You can enable 802.1X on a port that is a SPAN destination port; however, 802.1X is disabled until the port is removed as a SPAN destination port. You can enable 802.1X on a SPAN source port.
- The 802.1X protocol is not supported on ports configured with voice VLAN.

Configuring 802.1X Port-Based Authentication

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

- [Enabling 802.1X Port-Based Authentication, page 46-7](#)
- [Configuring Router-to-RADIUS-Server Communication, page 46-9](#)
- [Enabling Periodic Reauthentication, page 46-10](#)
- [Manually Reauthenticating the Client Connected to a Port, page 46-11](#)
- [Initializing Authentication for the Client Connected to a Port, page 46-11](#)
- [Changing the Quiet Period, page 46-12](#)
- [Changing the Router-to-Client Retransmission Time, page 46-12](#)
- [Setting the Router-to-Client Frame Retransmission Number, page 46-14](#)
- [Enabling Multiple Hosts, page 46-15](#)
- [Resetting the 802.1X Configuration to the Default Values, page 46-15](#)

Enabling 802.1X Port-Based 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 configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	Router(config)# aaa new-model	Enables AAA.
	Router(config)# no aaa new-model	Disables AAA.
Step 2	Router(config)# aaa authentication dot1x {default} method1 [method2...]	Creates an 802.1X port-based authentication method list.
	Router(config)# no aaa authentication dot1x {default list_name}	Clears the configured method list.
Step 3	Router(config)# dot1x system-auth-control	Globally enables 802.1X port-based authentication.
	Router(config)# no dot1x system-auth-control	Globally disables 802.1X port-based authentication.
Step 4	Router(config)# interface type¹ slot/port	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 5	Router(config-if)# dot1x port-control auto	Enables 802.1X port-based authentication on the interface.
	Router(config-if)# no dot1x port-control auto	Disables 802.1X port-based authentication on the interface.
Step 6	Router(config)# end	Returns to privileged EXEC mode.
Step 7	Router# show dot1x all	Verifies your entries. Check the Status column in the 802.1X Port Summary section of the display. An <i>enabled</i> status means the port-control value is set either to auto or to force-unauthorized .

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

When you enable 802.1X port-based authentication, note the following information:

- To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.
- Enter at least one of these keywords:
 - **group radius**—Use the list of all RADIUS servers for authentication.
 - **none**—Use no authentication. The client is automatically authenticated by the router without using the information supplied by the client.

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

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all

Dot1x Info for interface FastEthernet5/1
-----
```

```

AuthSM State      = FORCE UNAUTHORIZED
BendSM State      = IDLE
PortStatus        = UNAUTHORIZED
MaxReq            = 2
MultiHosts        = Disabled
Port Control      = Force Unauthorized
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
TxPeriod          = 30 Seconds

```

Configuring Router-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 <i>interface_name</i>	Specifies that the RADIUS packets have the IP address of the indicated interface.
	Router(config)# no ip radius source-interface	Prevents the RADIUS packets from having the IP address of the previously indicated interface.
Step 2	Router(config)# radius-server host { <i>hostname</i> <i>ip_address</i> }	Configures the RADIUS server host name or IP address on the router. If you want to use multiple RADIUS servers, reenter this command.
	Router(config)# no radius-server host { <i>hostname</i> <i>ip_address</i> }	Deletes the specified RADIUS server.
Step 3	Router(config)# radius-server key <i>string</i>	Configures the authorization and encryption key used between the router and the RADIUS daemon running on the RADIUS server.
Step 4	Router(config)# end	Returns to privileged EXEC mode.

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 router 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, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, publication and the *Cisco IOS Security Command Reference*, Release 12.2, publication at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html



Note

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

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

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

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600.

Automatic 802.1X client reauthentication is a global setting and cannot be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the “[Manually Reauthenticating the Client Connected to a Port](#)” section on page 46-11.

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	Selects an interface to configure.
Step 2	Router(config-if)# dot1x reauthentication	Enables periodic reauthentication of the client, which is disabled by default.
	Router(config-if)# no dot1x reauthentication	Disables periodic reauthentication of the client.

	Command	Purpose
Step 3	Router(config-if)# dot1x timeout reauth-period <i>seconds</i>	Sets the number of seconds between reauthentication attempts. The range is 1 to 65535; the default is 3600 seconds. This command affects the behavior of the router only if periodic reauthentication is enabled.
	Router(config-if)# no dot1x timeout reauth-period	Returns to the default reauthorization period.
Step 4	Router(config-if)# end	Returns to privileged EXEC mode.
Step 5	Router# show dot1x all	Verifies your entries.

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

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

```
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 <i>type</i> ¹ <i>slot/port</i>	Manually reauthenticates the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, 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
Starting reauthentication on 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 <i>type</i> ¹ <i>slot/port</i>	Initializes the authentication for the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

1. *type* = ethernet, 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
Starting reauthentication on FastEthernet 5/1
```

Changing the Quiet Period

When the router cannot authenticate the client, the router remains idle for a set period of time, and then tries again. The idle time is determined by the quiet-period value. 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 smaller number than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout quiet-period <i>seconds</i>	Sets the number of seconds that the router remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
	Router(config-if)# no dot1x timeout quiet-period	Returns to the default quiet time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set the quiet time on the router to 30 seconds:

```
Router(config-if)# dot1x timeout quiet-period 30
```

Changing the Router-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the router with an EAP-response/identity frame. If the router 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 behavioral problems with certain clients and authentication servers.

To change the amount of time that the router waits for client notification, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout tx-period <i>seconds</i>	Sets the number of seconds that the router waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds; the default is 30.
	Router(config-if)# dot1x timeout tx-period	Returns to the default retransmission time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

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

```
Router(config)# dot1x timeout tx-period 60
```

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

The client notifies the router that it received the EAP-request frame. If the router does not receive this notification, the router waits a set period of time, and then retransmits the frame. You may set the amount of time that the router waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the router-to-client retransmission time for the EAP-request frames, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout supp-timeout <i>seconds</i>	Sets the router-to-client retransmission time for the EAP-request frame.
	Router(config-if)# no dot1x timeout supp-timeout	Returns to the default retransmission time.
Step 3	Router# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

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

```
Router(config-if)# dot1x timeout supp-timeout 25
```

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

The authentication server notifies the router each time it receives a Layer 4 packet. If the router does not receive a notification after sending a packet, the router waits a set period of time and then retransmits the packet. You may set the amount of time that the router waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the value for the retransmission of Layer 4 packets from the router to the authentication server, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout server-timeout <i>seconds</i>	Sets the router-to-authentication-server retransmission time for Layer 4 packets.
	Router(config-if)# no dot1x timeout server-timeout	Returns to the default retransmission time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

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

```
Router(config-if)# dot1x timeout server-timeout 25
```

Setting the Router-to-Client Frame Retransmission Number

In addition to changing the router-to-client retransmission time, you can change the number of times that the router 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 behavioral problems with certain clients and authentication servers.

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

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x max-req <i>count</i>	Sets the number of times that the router sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
	Router(config-if)# no dot1x max-req	Returns to the default retransmission number.

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

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

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

```
Router(config-if)# dot1x max-req 5
```

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1X-enabled port as shown in [Figure 46-3 on page 46-5](#). In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

To allow multiple hosts (clients) on an 802.1X-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x host-mode multi-host	Allows multiple hosts (clients) on an 802.1X-authorized port.
	Router(config-if)# dot1x host-mode single-host	Disables multiple hosts on the port.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x interface <i>type</i> ¹ <i>slot/port</i>	Verifies your entries.

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

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

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

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 <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x default	Resets the configurable 802.1X parameters to the default values.

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

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

Displaying 802.1X Status

To display global 802.1X administrative and operational status for the router, use the **show dot1x** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface *interface-id*** privileged EXEC command.

For detailed information about the fields in these displays, refer to the *Cisco IOS Master Command List*, Release 12.2SX.

**Tip**

For additional information (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/routers/ps368/tsd_products_support_series_home.html