# Wired 802.1X Deployment Guide

Last Updated: September 6, 2011

Cisco Validated Design

Building Architectures to Solve Business Problems

# About Cisco Validated Design (CVD) Program

The CVD program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information visit http://www.cisco.com/go/designzone.



Wired 802.1X Deployment Guide

# Wired 802.1X Deployment Guide

Cisco IOS software enables standards-based network access control at the access layer by using the 802.1X protocol to secure the physical ports where end users connect. This document focuses on deployment considerations specific to 802.1X, and includes the following sections:



## IEEE 802.1X Overview

This section introduces IEEE 802.1X and includes the following topics:



### What is 802.1X?

802.1X offers unprecedented visibility and secure, identity-based access control at the network edge. With the appropriate design and well-chosen components, you can meet the needs of your security policy while minimizing the impact to your infrastructure and end users.

The need for secure network access has never been greater. Consultants, contractors, and guests now require access to network resources over the same LAN connections as regular employees, who may themselves bring unmanaged devices into the workplace. As data networks become increasingly

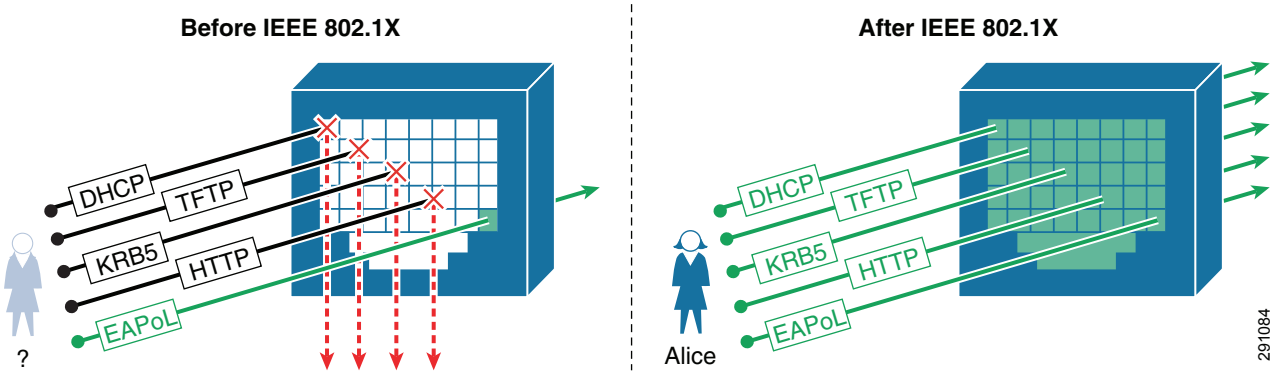

**Corporate Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

indispensable in day-to-day business operations, the possibility that unauthorized people or devices will gain access to controlled or confidential information also increases. The best and most secure solution to vulnerability at the access edge is to leverage the intelligence of the network.

802.1X is an IEEE standard for media-level (Layer 2) access control, offering the capability to permit or deny network connectivity based on the identity of the end user or device.

802.1X enables port-based access control using authentication. An 802.1X-enabled port can be dynamically enabled or disabled based on the identity of the user or device that connects to it. Figure 1 shows the default behavior of an 802.1X-enabled port.

***Figure 1 Default Network Access Before and After 802.1X***




Before authentication, the identity of the endpoint is unknown and all traffic is blocked. After authentication, the identity of the endpoint is known and all traffic from that endpoint is allowed. The switch performs source MAC filtering to ensure that only the authenticated endpoint is allowed to send traffic. To learn more about solution-level use cases, design, and a phased deployment methodology, see the following URL: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/whitepaper_C11-530469.html.

For step-by-step configuration guidance, see the following URL: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/Whitepaper_c11-532065.html.

# 802.1X Benefits

802.1X offers the following benefits on wired networks:

- Visibility—802.1X provides greater visibility into the network because the authentication process provides a way to link a username with an IP address, MAC address, switch, and port. This visibility is useful for security audits, network forensics, network use statistics, and troubleshooting.
- Security—802.1X is the strongest method for authentication and should be used for managed assets that support an 802.1X supplicant. 802.1X acts at Layer 2 in the network, allowing you to control network access at the access edge.
- Identity-based services—802.1X enables you to leverage an authenticated identity to dynamically deliver customized services. For example, a user might be authorized into a specific VLAN or assigned a unique access list that grants appropriate access for that user.
- Transparency—In many cases, 802.1X can be deployed in a way that is transparent to the end user.
- User and device authentication—802.1X can be used to authenticate devices and users.

# 802.1X Limitations

Although 802.1X enables unparalleled visibility and security, the following limitations must be addressed by your design:

- Legacy endpoint support—By default, 802.1X provides no network access to endpoints that cannot authenticate because they do not support 802.1X. Alternative mechanisms such as MAC Authentication Bypass (MAB) or Web Authentication must be provided for legacy endpoints.
- Delay—By default, 802.1X allows no access before authentication. Endpoints that need immediate network access must be capable of performing 802.1X at or near boot-up/link-up time, or alternative mechanisms must be used to grant the necessary access in a timely manner.

# 802.1X Components

802.1X defines the following three required components:

- Supplicant—A client that runs on the endpoint and submits credentials for authentication.

  Supplicants can be software applications such as the Cisco Secure Services Client; or they can be embedded in operating systems such as Microsoft Windows, or hardware such as Intel vPro.

- Authenticator—The network access device that facilitates the authentication process by relaying the credentials of the supplicant to the authentication server.

  The authenticator enforces both the locally configured network access policy and the dynamically assigned network access policy returned by the authentication server. In the context of this document, the authenticator is simply the access layer switch, and terms *authenticator* and *switch* should be considered interchangeable.

**Note** The authenticator is also often referred to as a policy enforcement point (PeP).

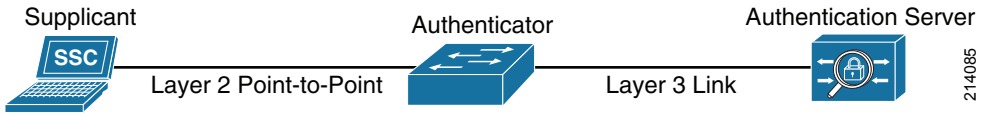
- Authentication server—A server that validates the credentials sent by the supplicant and determines what level of network access the end user or device should receive.

  The de facto industry standard is a RADIUS server, such as Cisco Access Control Solution (ACS). In this document, *RADIUS server* and *authentication server* are used interchangeably.

**Note** The authentication server is also often referred to as a policy decision point (PdP).

Figure 2 shows the three 802.1X components.

***Figure 2 802.1X Components***



In addition to the required components, additional components such as the following are almost always used:

- Backend identity databases—Centralized identity stores that the authentication server can query to validate credentials.

  Typical backend databases include Microsoft Active Directory, Novell eDirectory, or an LDAP server. By leveraging existing backend databases, the authentication server is relieved of the burden of internally maintaining credentials such as passwords.

**Note** Backend identity databases and certificate authorities are also commonly referred to as policy information point (PiPs).

- Public Key Infrastructure (PKI)—The set of technologies and processes that enables the distribution and maintenance of digital certificates.

  Because 802.1X often uses client and/or server certificates in the authentication process, many of the system components may need to be integrated with a PKI.

# 802.1X Protocols

802.1X uses the following protocols:

- Extensible Authentication Protocol (EAP)—The message format and framework defined by RFC 4187 that provides a way for the supplicant and the authenticator to negotiate an authentication method (the EAP method).
- EAP method—Defines the authentication method; that is. the credential type and how it is submitted from the supplicant to the authentication server using the EAP framework.

  Common EAP methods used in 802.1X networks are *EAP-Transport Layer Security (*EAP-TLS) and Protected EAP-Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MSCHAPv2).
- EAP over LAN (EAPoL)—An encapsulation defined by 802.1X for the transport of the EAP from the supplicant to the switch over IEEE 802 networks.

  EAPoL is a Layer 2 protocol.
- RADIUS—The de facto standard for communication between the switch and the authentication server.

  The switch extracts the EAP payload from the Layer 2 EAPoL frame and encapsulates the payload inside a Layer 7 RADIUS packet.

Figure 3 shows the 802.1X wire protocols.

***Figure 3 802.1X Wire Protocols***

# Sequence of Operations

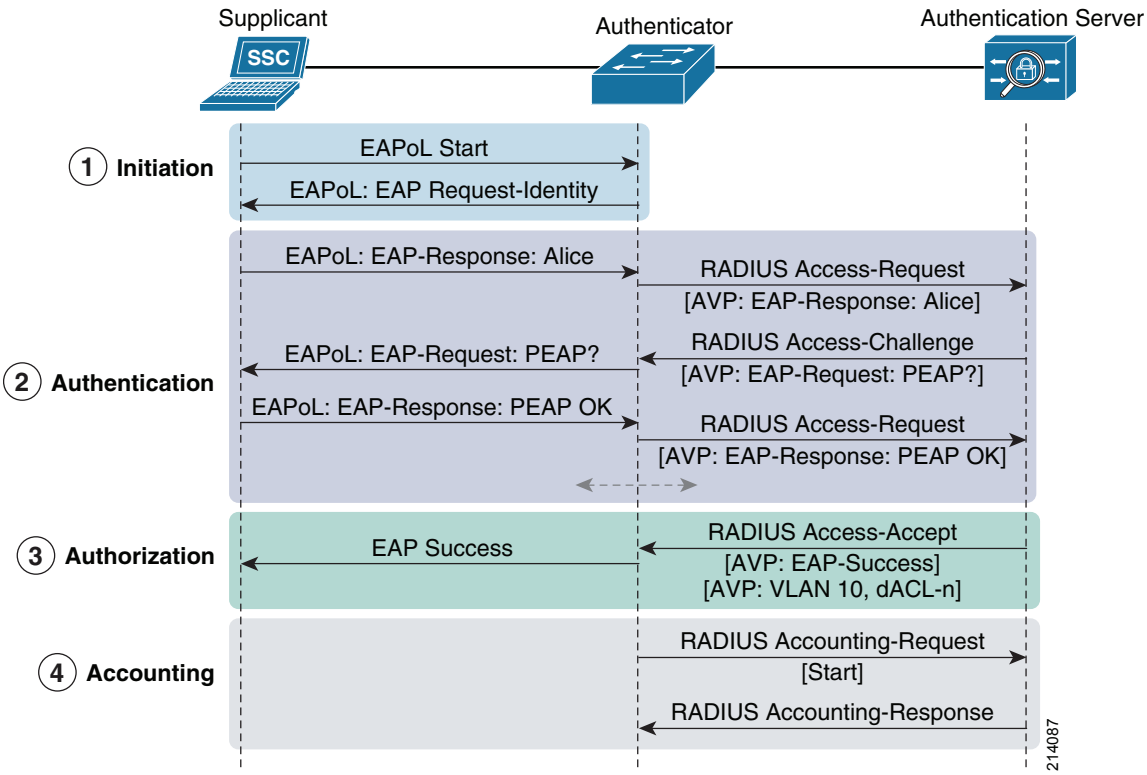
This section describes the stages of 802.1X operation, and includes the following topics:



## Operation Sequence Overview

The high-level functional sequence in Figure 4 shows how the components and protocols of 802.1X work together.

***Figure 4 High-Level 802.1X Sequence***




The message exchange as shown in Figure 4 is divided into four stages:

- Session initiation
- Session authentication
- Session authorization
- Session accounting

A fifth stage, session termination, is not shown in Figure 4.

# Session Initiation

An 802.1X authentication can be initiated by either the switch or the supplicant. From the perspective of the switch, the authentication session begins when the switch detects a link up on a port. The switch initiates authentication by sending an EAP-Request-Identity message to the supplicant. If the switch does not receive a response, the switch retransmits the request at periodic intervals.

The supplicant can initiate authentication by sending an EAPoL-Start frame. The EAPoL-Start message enables supplicants to speed up the authenticate process without waiting for the next periodic EAP-Request-Identity from the switch. EAPoL-Start messages are required in situations where the supplicant is not ready to process an EAP-Request from the switch (for example, because the operating system is still booting); or where there is no physical link state change on the switch (for example, because the supplicant is indirectly connected via an IP phone or hub).

**Tip** **Best Practice Recommendation**—Ensure that your supplicants send EAPoL-Start messages. Not all supplicants send EAPoL-Starts by default (for example, the Microsoft XP SP2 native supplicant), but most can and should be configured to do so for the proper operation of 802.1X.

# Session Authentication

During this stage, the switch relays EAP messages between the supplicant and the authentication server, copying the EAP message in the EAPoL frame to an AV-pair inside a RADIUS packet and vice versa. In the first part of the exchange, the supplicant and the authentication server agree on an EAP method.

The rest of the exchange is defined by the specific EAP method. The EAP method defines the type of credential to be used to validate the identity of the supplicant and how the credential is submitted. Depending on the method, the supplicant may submit a password, certificate, token, or other credential. That credential can then be passed inside a TLS-encrypted tunnel, as a hash or in some other protected form.

# Session Authorization

If the supplicant submits a valid credential, the authentication server returns a RADIUS Access-Accept message with an encapsulated EAP-Success message. This indicates to the switch that the supplicant should be allowed access to the port. Optionally, the authentication server may include dynamic network access policy instructions (for example, a dynamic VLAN or ACL) in the Access-Accept message. In the absence of dynamic policy instructions, the switch simply opens the port.

If the supplicant submits an invalid credential or is not allowed to access the network for policy reasons, the authentication server returns a RADIUS Access-Reject message with an encapsulated EAP-Failure message. This indicates to the switch that the supplicant should not be allowed access to the port. Depending on how the switch is configured, it may retry authentication, deploy the port into the Auth-Fail VLAN, or try an alternative authentication method.

# Session Accounting

If the switch is able to successfully apply the authorization policy, the switch can send a RADIUS Accounting-Request message to the authentication server with details about the authorized session. Accounting-Request messages are sent for both dynamically authorized sessions as well as locally authorized sessions; for example, Guest VLAN and Auth-Fail VLAN. For more information about 802.1X accounting, see the "RADIUS Accounting" section on page 30.

# Session Termination

Session termination is an important part of the 802.1X authentication process. To ensure the integrity of the authenticated session, sessions must be cleared when the authenticated endpoint disconnects from the network. Sessions that are not terminated immediately can lead to security violations and security holes. Ideally, session termination happens as soon as the endpoint physically unplugs, but this is not always possible if the endpoint is connected indirectly; for example, via an IP phone or hub.

Multiple termination mechanisms may be needed to address all use cases. Figure 4 summarizes the various mechanisms and the appropriate applications.

***Table 1 Session Termination Mechanisms***

| Use Case | Typical Termination Mechanisms |
|---|---|
| All endpoints directly connected<br>• Single endpoint per port<br>• No IP Phones | Link down |
| Endpoints connected via IP Phone<br>• At most two endpoints per port (one phone, one data) | CDP enhancement for second port disconnect (Cisco phones)<br>Proxy EAPoL-Logoff + inactivity timer (non-Cisco phones) |
| Endpoints connected via hub<br>• Physical hub<br>• Bridged virtual hubs | Inactivity timer |

This section describes the ways in which an 802.1X session can be terminated and includes the following topics:

## Link down

The most direct way to terminate an 802.1X session is to unplug the endpoint. When the link state of the port goes down, the switch completely clears the session. If the original endpoint or a new endpoint plugs in, the switch restarts authentication from the beginning.

## EAPoL Logoff/proxy EAPoL Logoff

The EAPoL-Logoff message was designed to allow the supplicant to tell the switch to terminate the existing session. On receipt of an EAPoL-Logoff message, the switch terminates the existing session. However, there are not many practical applications of this message and many supplicants do not send EAPoL-Logoff messages.

Although EAPoL-Logoff itself does not have many applications, a proxy EAPoL-Logoff message can be very useful. For example, an IP phone can transmit a proxy EAPoL-Logoff message when the phone detects that an 802.1X-authenticated endpoint has unplugged from behind the phone. The phone substitutes the MAC address of the data endpoint, so the proxy EAPoL-Logoff message is indistinguishable from an actual EAPoL-Logoff message from the data endpoint itself. The switch immediately clears the session as soon as it receives the Logoff message.

To support this feature, your phone must be capable of sending proxy EAPoL-Logoff messages. All Cisco IP phones and some third-party phones provide this functionality. No special functionality is required from the switch because the EAPoL-Logoff message is fully supported as per the IEEE standard.

**Note** Although effective for 802.1X-authenticated endpoints, Proxy EAPoL-Logoff messages do not work for MAB or Web Authentication, because these authentication methods do not use EAP to authenticate.

## CDP Enhancement for Second Port Disconnect

For IP telephony deployments with Cisco IP phones, the best way to ensure that all 802.1X sessions are properly terminated is using Cisco Discovery Protocol (CDP). Cisco IP phones can send a CDP message to the switch indicating that the link state for the port of the data endpoint is down, which allows the switch to immediately clear the authenticated session of the data endpoint.

**Tip** **Best Practice Recommendation**—Use CDP Enhancement for Second Port Disconnect for IP telephony deployments. This feature works for all authentication methods, takes effect as soon as the endpoint disconnects, and requires no configuration. If you are using Cisco IP phones and Cisco Catalyst switches with the appropriate release of code, this is the simplest and most effective solution. No other method works as well to terminate authenticated sessions behind IP phones.

## Inactivity Timer

When the inactivity timer is enabled, the switch monitors the activity from authenticated endpoints. When the inactivity timer expires, the switch removes the authenticated session.

The inactivity timer for 802.1X can be statically configured on the switch port or it can be dynamically assigned using the RADIUS Idle-Timeout Attribute [28]. Cisco recommends setting the timer via the RADIUS attribute because this provides control over which endpoints are subject to this timer and the length of the timer for each class of endpoints. For example, if your phones are capable of Proxy-EAPoL-Logoff, there might be no need to assign an inactivity timer for 802.1X-authenticated sessions. Likewise, endpoints that are known to be quiet for long periods of time can be assigned a longer inactivity timer than endpoints in greater use.

The inactivity timer is an indirect mechanism the switch uses to infer that an endpoint has disconnected. An expired inactivity timer cannot guarantee that an endpoint has disconnected. Therefore, a quiet endpoint that does not send traffic for long periods of time, such as a network printer that services occasional requests but is otherwise silent, may have its session cleared even though it is still connected. That endpoint must then send traffic before it can be authenticated again and have access to the network.

**Tip** Enable IP Device Tracking with inactivity timers to keep quiet endpoints connected. When IP Device Tracking is enabled, the switch periodically sends ARP probes to endpoints in the IP Device Tracking table (which is initially populated by DHCP requests or ARP from the end point). As long as the endpoint is connected and responds to these probes, the inactivity timer is not triggered and the endpoint is not inadvertently removed from the network.

# EAP Methods

This section describes different EAP methods and provides information that will help with deployment:



## Overview

Inside the framework provided by 802.1X and EAP, the endpoint and/or user must authenticate to the authentication server using a secure and reliable EAP method. The EAP method determines the type of credential that is used and how that credential is submitted.

Although many EAP methods are defined, this document focuses on two of the most commonly used EAP methods: EAP-TLS and PEAP-MSCHAPv2. The following sections give a detailed technical overview of these two EAP methods, starting with a description of the basic functionality and ending with specific deployment considerations and recommendations for each method.

## EAP-TLS

This section describes the EAP-TLS method and includes the following topics:

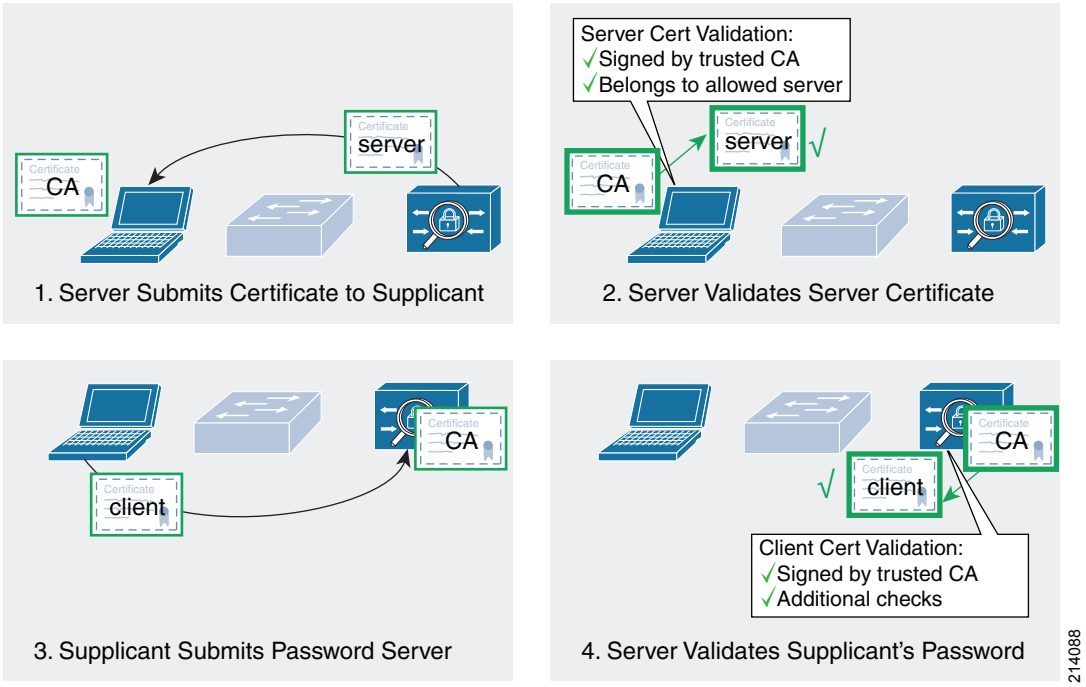
## Basic Functionality

EAP-TLS is an IETF standard defined in RFC 2716. EAP-TLS addresses a number of weaknesses in other EAP protocols by using X.509 certificates for secure authentication. In addressing these weaknesses, however, EAP-TLS increases the complexity of deployment. Unlike PEAP-MSCHAPv2 (which requires only server-side certificates), EAP-TLS requires client-side and server-side certificates for mutual authentication.

Within 802.1X, the EAP-TLS exchange of messages provides mutual authentication, negotiation of the encryption method, and encrypted key determination between a supplicant and an authentication server.

Figure 5 shows high-level representation of the EAP-TLS process.

***Figure 5 High Level EAP-TLS Functionality***




After the server and supplicant agree to perform authentication using EAP-TLS, the server submits its certificate to the supplicant in Step 1. In Step 2, the supplicant validates the certificate of the server. After the identity of the server has been authenticated, the supplicant submits its certificate to the server in Step 3. The server validates the certificate of the supplicant, thus completing the process of mutual authentication in Step 4.

## Deployment Recommendations (Certificate Requirements)

One of the biggest challenges when deploying EAP-TLS is meeting the certificate requirements. EAP-TLS provides authentication through the exchange and verification of X.509 certificates. Therefore, installing the correct certificates on 802.1X supplicants and the authentication server is absolutely essential to a successful deployment.

Every end user and computer, including the authentication server, that participates in EAP-TLS must possess at least two certificates:

- A client certificate signed by the certificate authority (CA)

- A copy of the CA root certificate

The client certificate is like a passport that cannot be forged. A user or computer presents a client certificate as proof of identity. The client certificate is signed by the CA that issued it. Anyone in possession of a copy of that root certificate of the CA can validate the signature on the client certificate. Thus, the CA is a trusted third party that allows entities to authenticate each other.

In an EAP-TLS exchange, the authentication server must have a copy of the root certificate for the CA that signed the certificate of the supplicant. Conversely, the supplicant must have the root certificate for the CA that signed the certificate of the authentication server.

Although it is possible to manually install the required certificates on each endpoint, manual certificate enrollment does not scale well.

Tip **Best Practice Recommendation**—Automate the certificate enrollment process. By leveraging auto-enrollment capabilities in your PKI, you greatly simplify the deployment of certificates.

In Microsoft environments, it is possible to use Active Directory-based auto-enrollment mechanisms to simplify the deployment of PKI. The Active Directory default group policy automatically propagates the root CA certificate to the appropriate store of any device or user that joins the domain. Active Directory group policies can also be configured to auto-enroll machine and user client certificates and to renew all certificates in advance of expiration.

User auto-enrollment is supported only on Windows 2003 Server Enterprise Edition Certificate Authorities. Windows 2003 Server Standard Edition CA cannot be used for user auto-enrollment. Machine auto-enrollment is supported on both editions. Because user auto-enrollment greatly simplifies PKI deployment, using a Windows 2003 Server Enterprise Edition CA is the recommended best practice when deploying EAP-TLS in a Microsoft environment.

Note For detailed information on configuring user certificate auto-enrollment in Windows environments, see the following URL: http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/autoenro.mspx#EFD.

Another example of auto-enrollment comes from the Cisco Unified Communication Manager (CUCM). The CA Proxy Functionality (CAPF) of CUCM is capable of auto-enrolling certificates for Cisco IP phones that need to perform 802.1X.

## Initial Deployment

Machines or users attempting to connect to an 802.1X-protected network for the first time must have a valid certificate to gain full access to the network. Therefore, it is best to deploy certificates to all endpoints before enabling 802.1X in the network. After 802.1X is enabled and EAP-TLS is deployed, additional planning is required for certificate enrollment, expiration, and/or revocation.

After 802.1X has been enabled, there are several ways for new endpoints to acquire certificates. *Organizations that use pre-built images should build certificates into the image before deploying the endpoint*. This greatly simplifies certificate deployment and is recommended for organizations that deploy endpoints in this way. Otherwise, you may want to offer sufficient network access to allow these endpoints to acquire a certificate, either using a fallback authentication method such as MAC Authentication Bypass or Web Authentication, a fallback authorization such as the guest VLAN, or a deployment scenario such as low impact mode.

Note For more information on deployment scenarios, see the following URL: http://www.cisco.com/go/trustsec.

## Certification Expiration and Revocation

Certificates expire according to the date set by the CA that issued them. The duration of the lifetime of the certificate should be configured in accordance with the security policy of an organization. To prevent endpoints from losing network access, these certificates must be renewed before expiration. A best practice is to automate certificate renewal and design your PKI to enable certificate renewal well in advance of the expiration date.

Certificates can be revoked for various reasons. For example, the certificate may have been compromised or the person to whom the certificate was issued might have left the organization. These certificates must be revoked to prevent them being used to gain unauthorized access to the assets of an organization. Certificate revocation is achieved through a certificate revocation list (CRL). A CA periodically generates a CRL that contains a list of all certificates that should no longer be trusted.

When certificates are revoked or expire without renewal, EAP-TLS fails and network access is denied. Because network access is required to request a new certificate, these endpoints and/or users are permanently denied access by default.

In the absence of a manual process, you may want to offer sufficient network access to allow failed endpoints to acquire a valid certificate. The available mechanisms in this use case include a fallback authentication method such as MAC Authentication Bypass or Web Authentication, a fallback authorization such as the AuthFail VLAN, or a deployment scenario such as low impact mode that can allow a certain amount of access regardless of the authentication state of the port.

# PEAP-MSCHAPv2

This section describes the PEAP-MSCHAPv2 method and includes the following topics:



## Basic Functionality

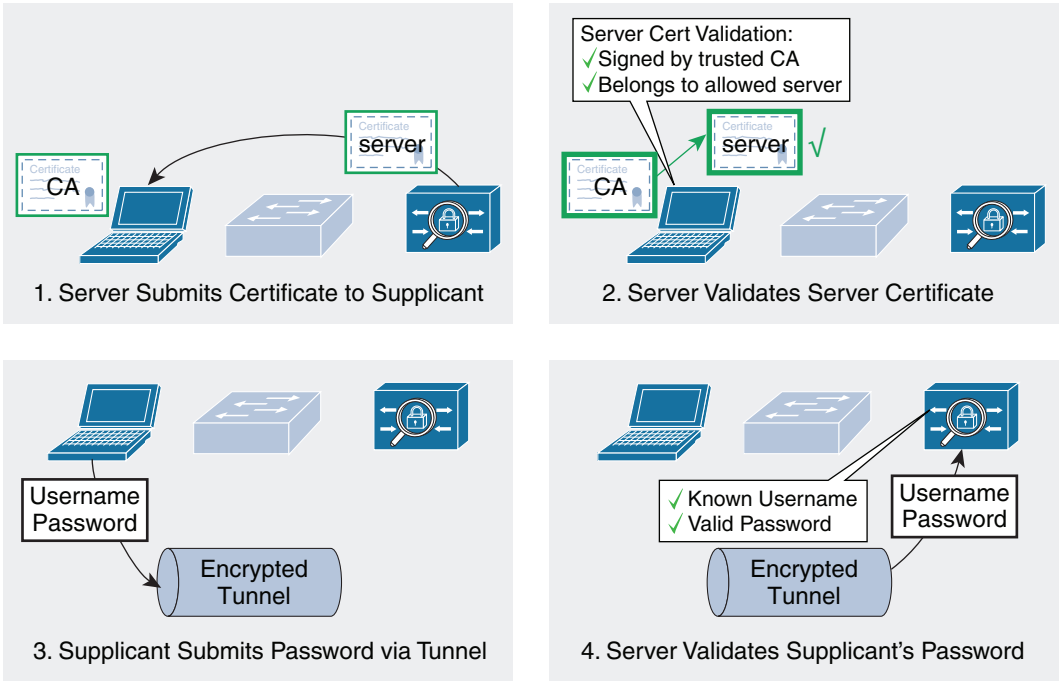
PEAP was developed by Cisco Systems, Microsoft Corporation, and RSA Security, Inc. PEAP is an EAP type that addresses security issues by first creating a secure channel that is both encrypted and integrity-protected with TLS. This tunnel is created using a valid server certificate that the authentication server sends to the supplicant at the beginning of the PEAP negotiation. Inside this secure channel, a new EAP negotiation takes place to authenticate the client. This second EAP negotiation can be virtually any EAP type.

Because the TLS channel protects EAP negotiation and authentication for the network access attempt, password-based authentication protocols that are normally susceptible to an offline dictionary attack can be safely used for authentication. By wrapping the EAP messages within TLS, any EAP method running within PEAP is provided with built-in support for key exchange, session resumption, fragmentation, and reassembly. Furthermore, because PEAP requires a certificate only on the authentication server, it is possible to securely authenticate LAN clients without requiring every client to have its own certificate. This greatly reduces the burden of deploying and maintaining a PKI.

MSCHAPv2 is commonly used as the second EAP type inside a PEAP tunnel. MS-CHAPv2 is a password-based, challenge-response, mutual authentication protocol that uses Message-Digest Algorithm (MD4) and Data Encryption Standard (DES) to encrypt responses. The authenticator challenges a supplicant and the supplicant can challenge the authentication server. If either challenge is not correctly answered, the connection can be rejected. Although MSCHAPv2 provides better protection than previous challenge-response authentication protocols, it is still susceptible to an offline dictionary attack. A malicious user can capture a successful MSCHAPv2 exchange and guess passwords until the correct one is determined. Used in the combination with PEAP, however, the MSCHAPv2 exchange is protected with the strong security of the TLS channel. PEAP-MSCHAPv2 is used primarily in Microsoft Active Directory environments.

Figure 6 shows a high-level representation of the PEAP-MSCHAPv2 process.

***Figure 6 High Level PEAP-MSCHAPv2 Functionality***




After the server and supplicant agree to perform authentication using PEAP-MSCHAPv2, the server submits its certificate to the supplicant in Step 1. The supplicant validates the certificate of the server in Step 2. After the identity of the server has been authenticated, the supplicant builds a TLS-encrypted tunnel. Inside that tunnel, the supplicant submits a username and password using MSCHAPv2 in Step 3. The server validates the password of the supplicant, thus completing the process of mutual authentication in Step 4.

## Deployment Recommendations (Credential Requirements)

Like EAP-TLS, PEAP-MSCHAPv2 requires that the authentication server present a certificate to the supplicant. To validate the server certificate, the supplicant must have the root certificate for the CA that signed the certificate of the authentication server. Unlike EAP-TLS, PEAP-MSCHAPv2 does not require that the supplicant have a certificate. This is because supplicant establishes its identity inside the tunnel via MSCHAPv2. MSCHAPv2 authentication relies on a password, not a certificate.

Every endpoint and user that participates in PEAP-MSCHAPv2 must possess the following credentials:

- Root CA certificate for the CA that signed the certificate of the authentication server
- MSCHAPv2 username and password

The authentication server must possess the following credentials:

- Server certificate signed by the root CA
- MSCHAPv2 password for every user and computer

Note For more details on certificate deployment and management, see the "EAP-TLS" section on page 11.

## Passwords

With PEAP-MSCHAPv2, clients use passwords to successfully complete the inner MSCHAPv2 challenge. If possible, re-use an existing password store. For example, in Microsoft Active Directory environments, the Active Directory passwords can be used for the MSCHAPv2 exchange. This is often referred to as *single-sign-on* because the end user enters the password only once (at the Microsoft login window). The 802.1X supplicant reuses this password for MSCHAPv2 without having to query the user again. Re-using existing credentials also reduces administrative overhead because there is only one password repository to manage.

Tip **Best Practice Recommendation**—Enable single sign-on if your security policy allows it. Enabling the supplicant for single sign-on reduces administrative overhead and eliminates the need for end users to change their behavior.

For machine authentication in Microsoft environments, machines need Active Directory passwords for the same reasons that users need Active Directory passwords. Active Directory automatically provisions machines with machine passwords suitable for MSCHAPv2 when the machine joins the domain. In most cases, no further action is required to provision the machine with suitable credentials.

Password aging is often enabled in Active Directory as part of a larger Windows security policy. To prevent network outages when Active Directory passwords expire, users can change passwords during PEAP authentication. When attempting PEAP-MSCHAPv2 using an expired Active Directory password, users receive a dialog box prompting them to change their passwords during first authentication after their passwords have expired. After the password is changed, the PEAP authentication session continues on as usual.

Expired machine passwords cannot be changed during the PEAP authentication process. Although users can update Active Directory passwords during a PEAP-MSCHAPv2 authentication, machines cannot when running Windows XP SP2. Therefore, machines with expired passwords fail authentication, and some other process must be employed to allow machines to update expired passwords. For example, if you are using PEAP with user *and* computer authentication, the machine password is reset when the user logs in. Other options are to configure longer password expiration times for machines or consider an EAP method (such as EAP-TLS) that does not use passwords.

Note For more information, see the following URL:
http://technet.microsoft.com/en-gb/library/cc512611.aspx.

# Choosing an EAP Method

Different EAP methods offer differing levels of security and complexity. When deploying 802.1X, it is essential to choose an EAP method that meets the security policy of your organization and that is supported by the available infrastructure. Important factors to consider when selecting any EAP method include the following:

- Credential type and EAP method—The EAP type you use is in part determined by the type of credential you want to use. EAP-TLS requires the client to have a digital certificate. PEAP-MSCHAPv2 uses passwords.

**Tip** **Best Practice Recommendation**—Re-use existing credentials. By re-using an existing credential system such as a WLAN or Active Directory credential, you eliminate the need to create and maintain a new set of credentials for 802.1X.

- Security policy and EAP method—Some security policies might require two-factor authentication, in which case a password-based EAP method such as PEAP-MSCHAPv2 is not appropriate.
- Mutual authentication and EAP method—To avoid supplicants authenticating to untrusted authentication servers, choose an EAP method that offers mutual authentication. Mutual authentication forces the supplicant to validate the identity of the authentication server before submitting its credentials, reducing the likelihood of a man-in-the-middle attack.

**Tip** **Best Practice Recommendation**—Choose an EAP method with mutual authentication. PEAP-MSCHAPv2 and EAP-TLS offer mutual authentication. EAP-MD5 does not.

- PKI and EAP method—Each EAP method makes different demands on the PKI of an organization. PKI refers to the infrastructure that creates, maintains, and revokes X.509 certificates for endpoints and users in the network. The ability of an organization to support PKI might influence the choice of an EAP method. Of the two EAP methods discussed in this document, EAP-TLS requires the most complex PKI (client and server certificates) while PEAP-MSCHAPv2 requires a less complex PKI (server certificates only).

  In some cases, the choice of a CA may also impact the choice of a supplicant. For example, the native Microsoft Windows XP supplicant requires that the certificate presented by the RADIUS server include an enhanced key usage (EKU) field in the certificate that is set to server authentication. Likewise, the Microsoft supplicant requires that client certificates have an EKU field set to client authentication. CAs that do not support the EKU field cannot be used with the Microsoft supplicant.
- Supplicant and EAP method—Not all supplicants support all EAP methods. When choosing a supplicant, be sure to verify that it supports the EAP method you wish to deploy.
- Authentication servers and EAP method—Not all authentication servers support all EAP methods. When choosing an authentication server, be sure to verify that it supports the EAP method you wish to deploy.

**Note** For more information on the EAP methods and backend data stores supported by the Cisco Secure ACS 5.2 Authentication Server, see the following URL: http://www.cisco.com/en/US/docs/net_mgmt/cisco_secure_access_control_system/5.2/user/guide/eap_pap_phase.html

- Backend data storage and EAP method—Not all backend data stores support all EAP methods. When choosing a backend data store, be sure to verify that it supports the EAP method you wish to deploy. Conversely, if you already have a backend data store, be sure to choose an EAP method that can leverage it. For example, an EAP type that uses MSCHAPv2 as the inner method, such as PEAP-MSCHAPv2, can use Active Directory as a backend database, but not a generic LDAP server.

# Choosing a Supplicant

Choose a supplicant or supplicants that can provide the needed functionality, minimize the administrative overhead, and can be easily deployed and maintained.

In Microsoft environments, the native supplicant is an attractive choice because it is pre-installed in the operating system. It supports both EAP-TLS and PEAP-MSCHAPv2. Starting in Windows XP SP3, there are two separate services for wired and wireless supplicants. However, consider the following limitations:

- Functional limitations in XP SP2—Before XP SP3, the wired supplicant had many functional limitations and could not be fully managed by GPOs.
- Upgrading from SP2—Because the implementation of the native supplicant changed significantly between XP SP2 and SP3/Vista/Win7, upgrading the OS can have unintended consequences for 802.1X. Where XP SP2 used the same service for wired and wireless, XP SP3/Vista/Win7 have separate services, and the wired service is disabled by default. Because of this, IEEE-802.1X authenticated-endpoints that upgrade from XP SP2 to SP3/Vista/Win7 can get removed from the wired network after upgrade.

**Note** For more information, see the following URL: http://support.microsoft.com/kb/953650.

- Single profile—The native supplicant allows only a single profile for user and machine authentication. For example, if you use EAP-TLS with soft certificates for machine authentication, you must also use EAP-TLS with soft certificates for user authentication.

The Cisco Secure Services Client (SSC) is another supplicant that works for Microsoft XP and Vista endpoints. The Cisco SSC is a full-featured supplicant with support for EAP-TLS, PEAP-MSCHAPv2, and many other EAP types. The SSC supports multiple profiles for wired and wireless authentication, with independent parameters for machine and user authentication. It comes with a management utility that simplifies the configuration and deployment, and also offers improved troubleshooting capabilities.

Mac OSX has a native supplicant that supports a broad array of EAP types for system authentication and user authentication.

There are several open source supplicants that can be used for Windows and Linux environments, including XSupplicant, WPA, and Secure W2.

Although relatively new to wired deployments, a hardware-based supplicant, such as Intel vPro, may be required for certain use cases. Hardware-based supplicants can perform 802.1X even if the operating system is unable to perform authentication. This is useful for endpoints in hibernate/standby mode that need to stay connected to the network (for example, to receive a Wake on LAN packet) and endpoints that use Pre eXecution Environment (PXE) to netboot an operating system.

# Choosing an Authentication Server

Many RADIUS servers can function as an 802.1X authentication server. First, choose a RADIUS server that can authenticate all of your endpoints (802.1X, MAB, Web Authentication) against all your identity stores (Active Directory, LDAP, and so on). Next, ensure that your RADIUS server supports the network access policies that you want to deploy. Lastly, your RADIUS server should be capable of monitoring, reporting, and troubleshooting.

Cisco ACS version 5.2 is a policy platform providing RADIUS and TACACS+ services. It provides a powerful, attribute-driven, rule-based policy model that addresses complex policy needs in a flexible manner. Cisco ACS has integrated advanced monitoring, reporting, and troubleshooting capabilities for maximum control and visibility; improved integration with external identity and policy databases, including Windows Active Directory and Lightweight Directory Access Protocol (LDAP)-accessible databases; and a distributed deployment model that enables large-scale deployments while providing a highly available solution.

In Microsoft environments, the Network Policy Server (NPS) on Windows Server 2008 and the Internet Authentication Service (IAS) on Window Server 2003 can perform authentication for endpoints and users that are part of the Active Directory domain. However, NPS and IAS do not support complex policy models, nor can they query backend databases for credential verification. Reporting and troubleshooting capabilities are limited.

Several open source RADIUS servers can be used for 802.1X, including FreeRADIUS and OpenRADIUS.

# Reauthentication

There is usually no need to re-authenticate a previously authenticated endpoint that remains connected to the network. After a successful 802.1X authentication, the port remains open until the session is terminated, most typically by a physical link-down event. Because physical connectivity is continuously maintained, the authenticated endpoint remains connected to the port. Under these circumstances, re-interrogating endpoint credentials serves no purpose.

In some situations, however, re-authentication can be used as a de-facto 802.1X keepalive mechanism. For example, if an endpoint is connected to the port via an IP phone that is not capable of proxy EAPoL-Logoff or CDP Enhancement for Second Port Disconnect, the switch does not know when to terminate the session. By re-authenticating or re-initializing the session, the switch can confirm that the authenticated endpoint is still connected. Because authentication and authorization are tightly coupled in 802.1X, re-authentication can also be used as a de-facto re-authorization technique. In the absence of explicit mechanisms to dynamically push policy updates to switches, such as RADIUS CoA, re-authentication provides a mechanism by which the switch can pull the latest authorization policy such as VLAN or ACL assignment for authenticated endpoints.

The re-authentication timer for 802.1X can be statically configured on the switch port, or it can be dynamically assigned by sending the Session-Timeout Attribute [27] and the RADIUS Termination-Action Attribute [29] with a value of *RADIUS-Request* in the Access-Accept message from the RADIUS server. If you choose to enable re-authentication, Cisco recommends setting the timer via the RADIUS attribute because this gives you control over what endpoints are subject to this timer and the length of the timer for each class of endpoints. Network connectivity is maintained during the re-authentication. If re-authentication is successful, the current session remains active. If re-authentication is not successful, the session is terminated.

The session timer can be also used to terminate an 802.1X session, regardless of whether the authenticated endpoint remains connected or not. The session timer uses the same RADIUS Session-Timeout Attribute [27] as the server-based re-authentication timer described above, with the

RADIUS Termination-Action Attribute [29] set to Default. The switch terminates the session after the number of seconds specified by the Session-Timeout Attribute and immediately restarts authentication by sending an EAP Identity Request exactly as if a new endpoint had plugged into the port. If the previous endpoint remains connected, network connectivity is interrupted until the new authentication session is complete.

Tip **Best Practice Recommendation—**Use server-based re-authentication timeouts, if using timeouts at all. Depending on the length of the timers, periodic re-authentication can increase the authentication traffic load on the network infrastructure. More importantly, there are usually better ways to ensure the proper termination of authenticated sessions, such as link state, proxy EAPoL-Logoff, CDP Second Port Disconnect, inactivity timers, and so on. Therefore, you should configure the policy on your AAA server to dynamically assign re-authentication timers only to endpoints whose connectivity cannot be validated any other way.

Tip **Best Practice Recommendation—**Do not assign session or re-authentication timers to MAB endpoints.Re-authentication and session timers also impact MAB sessions. On MAB re-authentication (RADIUS Termination-Action Attribute [29] = “RADIUS-Request”), the switch does not re-learn the MAC address of the connected endpoint. It simply sends the previously learned MAC address to the RADIUS server. On MAB session timeout (RADIUS Termination-Action Attribute [29] = “Default”), the switch re-learns the MAC address, but network connectivity is disrupted until 802.1X times out and the MAB succeeds. This can result in significant network outage for MAB endpoints.

- RADIUS Change of Authorization

  RADIUS Change of Authorization (CoA) allows a RADIUS server to dynamically instruct the switch to alter an existing session. Cisco Catalyst switches support the following four actions for CoA:

  - Re-authenticate
  - Terminate
  - Port shutdown
  - Port bounce

  The re-authenticate and terminate actions terminate the authenticated session in the same way as the re-authentication and session timeout actions discussed in the previous section. The port shutdown and port bounce actions clear the session immediately, because they result in link-down events.

# Design Considerations

This section discusses a variety of design considerations that you should evaluate before deploying 802.1X. It includes the following topics:

# Deployment Scenarios

When deploying 802.1X, Cisco recommends a phased deployment model that gradually deploys identity-based access control to the network. The three scenarios for phased deployment are as follows:

- Monitor mode
- Low impact mode
- High security mode

Each scenario identifies combinations of authentication and authorization techniques that work well together to achieve a particular set of use cases. By developing your 802.1X design in the context of a comprehensive deployment scenario, you can leverage well-understood blueprints to address common design issues.

For more information about scenario-based deployments, see the "References" section on page 34.

# User and Machine Authentication

802.1X can authenticate an endpoint based on user credentials, machine credentials, or both. Deciding which type of authentication to support is an important step in the design process.

For endpoints that do not support any form of user login, such as a printer, the choice is obvious. You must authenticate the machine credentials of the printer or the printer never gets on the network.

For endpoints that do support user login, such as a corporate laptop, the choice is made more difficult by the fact that the endpoint may need network access long before a user logs in. If the endpoint does not have network access, critical features such as Dynamic Host Configuration Protocol (DHCP), Network File System (NFS), and Active Directory Group Policy Objects (GPOs) do not behave correctly. By performing 802.1X with machine credentials, endpoints can gain access to the network without a user logged in.

**Tip** **Best Practice Recommendation—**Enable machine authentication in managed desktop environments. Managed desktop environments, such as Microsoft Active Directory, require that endpoints have access to the network very near to initial boot-up time. By enabling machine authentication, you can ensure that endpoints have timely access in an 802.1X network.

Many organizations find that their visibility and access control objectives can be met by enabling machine authentication only. By enabling only machine authentication, you can ensure that only corporate assets are allowed onto the network. If your policy does not allow user credentials for 802.1X

authentication, users cannot bring a laptop from home and attach it to the network with their user credentials. In organizations where each employee has the exclusive use of a laptop, there is a one-to-one correspondence between the employee and the laptop, so there is typically no need to authenticate both onto the network. Higher-layer authentications, such as a user logging in to the Active Directory domain, are still performed, offering additional levels of security for the user.

Enabling user authentication allows you to differentiate access between different users on the same machine. This may be desirable when there is no one-to-one correspondence between users and endpoints, or you need the additional visibility of tracking users as they log into a machine.

# Open Access

By default, 802.1X drops all traffic before a successful 802.1X (or MAB) authentication or Web Authentication initialization. This is sometimes referred to as *closed mode*. Cisco switches can also be configured for open access, which allows all traffic while still performing 802.1X and MAB.

Open access has many applications, including increasing network visibility as part of a monitor mode deployment scenario. It can be combined with other features to provide incremental access control as part of a low impact mode deployment scenario. For more information on these deployment scenarios, see the "References" section on page 34.

# Multiple Endpoints Per Port

By default, an IEEE-802.1X-enabled port allows only a single endpoint per port. Any additional MAC addresses seen on the port cause a security violation.

**Tip** **Best Practice Recommendation**—Configure the *restrict* security violation action. Cisco Catalyst switches can be configured to err-disable the port (*shutdown*), drop all traffic from the new MAC address (*restrict*), or terminate the existing session and attempt to authenticate the new MAC address (*replace*). Restricting traffic from additional MACs on the port usually achieves the desired behavior.

Frequently, the limitation of a single endpoint per port does not fit all the network requirements. Cisco Catalyst switches allow you to address multiple use cases by modifying the default behavior. The host mode on a port determines the number and type of endpoints allowed on a port. The various host modes and their applications are as follows:

- Single-host mode

  In single-host mode, only a single MAC or IP address can be authenticated by any method on a port. If a different MAC address is detected on the port after an endpoint has authenticated with 802.1X, MAB, or Web Authentication, a security violation is triggered on the port. This is the default behavior.

- Multi-domain-authentication (MDA) host mode

  MDA was specifically designed to address the requirements of IP telephony in an 802.1X environment. When MDA is configured, two endpoints are allowed on the port: one in the voice VLAN, and one in the data VLAN. Additional MAC addresses trigger a security violation.

- Multi-auth host mode

  If the port is configured for multi-auth mode, multiple endpoints can be authenticated in the data VLAN. Each new MAC address that appears on the port is separately authenticated. Multi-auth can be used for bridged virtual environments or to support hubs.

- Multi-host mode

  Unlike multi-auth host mode, which authenticates every MAC address, multi-host mode authenticates the first MAC address and then allows an unlimited number of other MAC addresses. Because of the security implications of multi-host, multi-auth is typically a better choice than multi-host.

**Tip** **Best Practice Recommendation—**Use the most restrictive host mode that addresses your use cases. Limiting the number of MAC addresses allowed on the port helps ensure the validity of the authenticated session and discourages casual port-piggybacking.

# Wake On LAN

Wake on LAN (WoL) is an industry standard power management feature that allows a hibernating endpoint to be woken by sending a "magic packet" over the network. Most WoL endpoints flap the link when going into hibernate or standby mode, thus clearing any existing 802.1X-authenticated session. By default, traffic through the unauthorized port is blocked in both directions and the magic packet never gets to the sleeping endpoint.

To support WoL in an 802.1X environment, you can configure a Cisco Catalyst switch to modify the control direction of the port, allowing traffic *to* the endpoint while still controlling traffic *from* the endpoint. This allows the hibernating endpoint to receive the WoL packet while still preventing the unauthorized endpoint from sending any traffic into the network. When woken, the endpoint can authenticate and gain full access to the network.

An alternative to modifying the control direction is to use a hardware supplicant that can perform 802.1X even when the endpoint itself is sleeping.

When a port is configured for open access mode, magic packets are not blocked, even on unauthorized ports, so no special configuration for WoL endpoints is necessary.

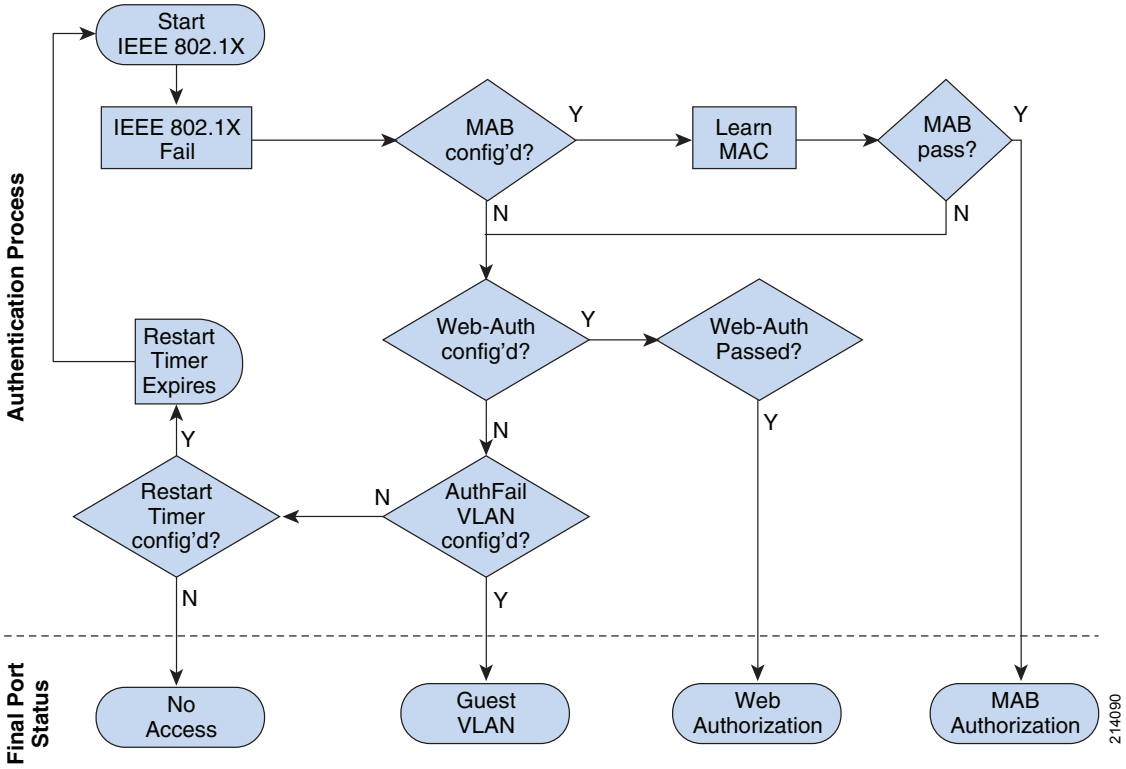
# Non-IEEE-802.1X-Capable Endpoints

If an endpoint without an 802.1X supplicant attempts to connect to a port that is enabled for 802.1X, it is subjected to the default security policy, which is no access until authentication.

Although many endpoints increasingly support 802.1X, there are always endpoints that require network connectivity but do not or cannot support 802.1X, such as network printers, badge readers, legacy servers, and PXE boot machines. Some provision must be made for these endpoints.

Cisco provides features to accommodate non-802.1X endpoints, including MAB, Web Authentication, and Guest VLAN. These features provide fallback mechanisms when there is no 802.1X supplicant. After an 802.1X timeout on a port, the port can move to an authorized state if MAB or Web Authentication succeeds, or if the Guest VLAN is configured. Judicious application of these features is required for a successful 802.1X deployment. Figure 7 shows the interactions of these fallback mechanisms.

*Figure 7 Fallback Mechanisms for Non-802.1X Endpoints*




MAB, Web Authentication, and Guest VLAN are fallback mechanisms; that is, they get deployed only once 802.1X has timed out. If the PXE process of the endpoint times out or if DHCP gets deep into the exponential back-off process before the timeout occurs, the endpoint may not access the network even after the port has been opened. It appears to the end user as if network access has been denied. This problem can be alleviated by decreasing the 802.1X timeout value. For more information on relevant timers, see the "Timers and Variables" section on page 26.

In addition to or instead of modifying the timer, you could use a low impact deployment scenario that allows time-critical traffic such as DHCP before authentication. For more information on deployment scenarios, see the "References" section on page 34.

If your network has many non-IEEE-802.1X-capable endpoints that need instantaneous access to the network, a third option is to use the Flexible Authentication feature set that allows you to configure the order and priority of authentication methods. Instead of waiting for 802.1X to time out before performing MAB, you can configure the switch to perform MAB first and fallback to 802.1X only if MAB fails. For more information on Flexible Authentication, see the "References" section on page 34.

# 802.1X Endpoints with Invalid Credentials

802.1X protects the network by preventing users and endpoints without valid credentials from gaining access to the access port. However, there may be situations where legitimate users do not have valid credentials. For example, a business partner might attempt to connect to the network for guest access. The laptop of the partner is configured for 802.1X, but the credentials of the partner are valid only on the network of the partner. 802.1X authentication fails, preventing the partner from gaining guest access.

To provide for situations such as these, Cisco offers the following two options:

- Auth-Fail VLAN

  When 802.1X authentication fails, as opposed to a timeout because there is no supplicant on the endpoint, the port is moved to a configurable VLAN (the Auth Fail VLAN) where restricted access can be enforced. Using the Auth Fail VLAN, you can tailor network access for endpoints that have a supplicant but do not have valid credentials. For example, the Auth-Fail VLAN may be configured to permit access only to the Internet or to a CA for certificate renewal.

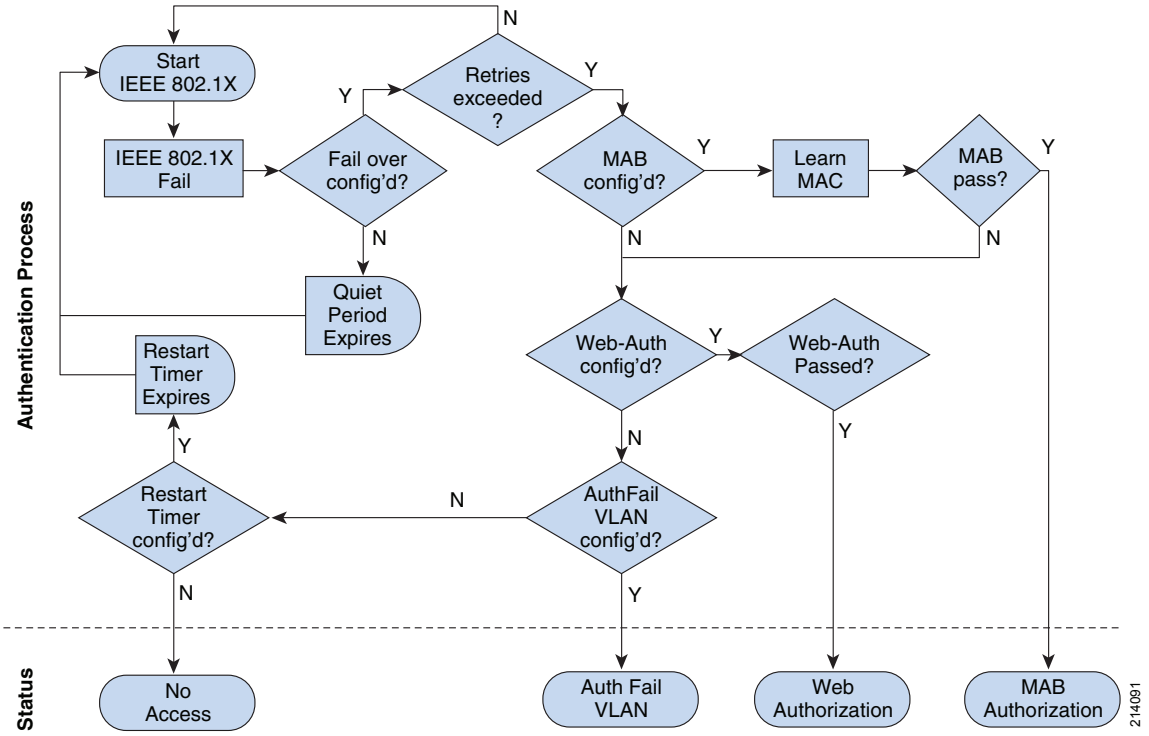
- Fail over to a secondary form of authentication

  Cisco Catalyst switches can be configured to attempt MAB and/or Web Authentication after 802.1X fails. Access to the network is granted based on the success or failure of the secondary authentication method.

Note All these failover mechanisms rely on the supplicant failing open as well.

Figure 8 shows the interactions of these failover mechanisms.

***Figure 8 Failover Mechanisms for Failed 802.1X Endpoints***




Note The Auth-Fail VLAN and failover authentication methods are optional. They should always be deployed in accordance with the security policy of an organization.

# Inaccessible Authentication Server

When the authentication server is unavailable, 802.1X fails and all endpoints are denied access by default. In a highly available enterprise campus environment, it is reasonable to expect that a switch is always able to communicate with the authentication server, so the default behavior may be perfectly acceptable. However, there may be some use cases, such as a branch office with occasional WAN outages, where the switch cannot reach the authentication server, but endpoints should be allowed access to the network.

If the switch already knows that the authentication server has failed, either through periodic probe or as the result of a previous authentication attempt, a port can be deployed in a configurable VLAN (sometimes called the critical VLAN) as soon as the link comes up. Because the switch has multiple mechanisms for learning that the AAA server has failed, this outcome is the most likely. If the switch determines that the authentication server has failed during an 802.1X or MAB authentication (for example, if this is the first endpoint to connect to the switch after connectivity to the authentication server has been lost), the port is moved to the critical VLAN after the authentication times out. Previously authenticated endpoints are not affected in any way; if a re-authentication timer expires when the authentication server is down, the re-authentication is deferred until the switch determines that the authentication server has returned.

The critical VLAN can be any VLAN except the voice VLAN. If no VLAN is specified, the port fails open into the switch data VLAN.

# Timers and Variables

802.1X relies on several timers and variables to control the timing of the authenticator functionality on the switch. This section outlines the timers on the switch that are relevant to 802.1X authentication process. Unless otherwise noted, these timers should be left at default values.

## dot1x timeout tx-period and dot1x max-reauth-req

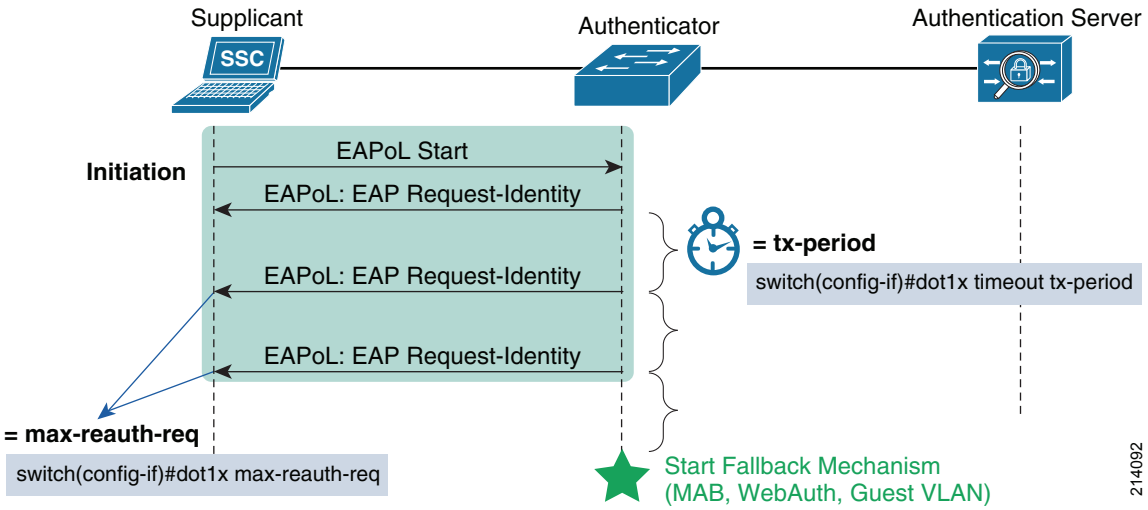
This section discusses the timers that control the timeout and retry behavior of an 802.1X-enabled port in the absence of a supplicant.

At link-up, the switch sends an EAP Request-Identity frame. It waits for a period of time defined by the *dot1x timeout tx-period* timer and then sends another Request-Identity frame. The number of times it resends the Request-Identity frames is defined by *dot1x max-reauth-req* variable.

Note Note that Request-Identity frames are sent only in the session initiation phase. During the subsequent authentication process, the retransmission of EAP Request frames are handled by *max-req*, not max-reauth-req.

Figure 9 shows the functions of the tx-period timer and the max-reauth-req variable.

*Figure 9 tx-period and max-reauth-req*




The combination of tx-period and max-reauth-req is especially important to non-IEEE-802.1X-capable endpoints. Endpoints without a supplicant must wait until 802.1X times out before getting network access via a fallback mechanism. The total time it takes for 802.1X to time out is determined by the following formula:

Timeout = (max-reauth-req +1) * tx-period

Cisco Catalyst switches have default values of tx-period = 30 seconds and max-reauth-req = 2. Applying the above formula, it takes 90 seconds by default for an endpoint without a supplicant to get access via MAB, Web Authentication, or the Guest VLAN. By modifying these two settings, you can decrease the total timeout down to a minimum value of two seconds.

Because of the impact on endpoints without supplicants, most customers change the default values of tx-period and/or max-reauth-req to allow more rapid access to the network. When modifying these values, consider the following:

- A timer that is too short may result in IEEE-802.1X-capable endpoints being subject to a fallback authentication or authorization technique. Although supplicants can send an EAPoL-Start frame to restart 802.1X after a fallback has occurred, you may still be generating unnecessary control plane traffic. In addition, if the endpoint has been authorized by a fallback method, that endpoint may temporarily be adjacent to guest devices that have been similarly authorized. If your goal is to ensure that your IEEE-802.1X capable assets are always and exclusively on a trusted network, you should ensure that the timer is long enough to allow 802.1X-capable endpoints time to authenticate.
- A timer that is too long can subject endpoints without a supplicant to unnecessarily long delays in getting network access.

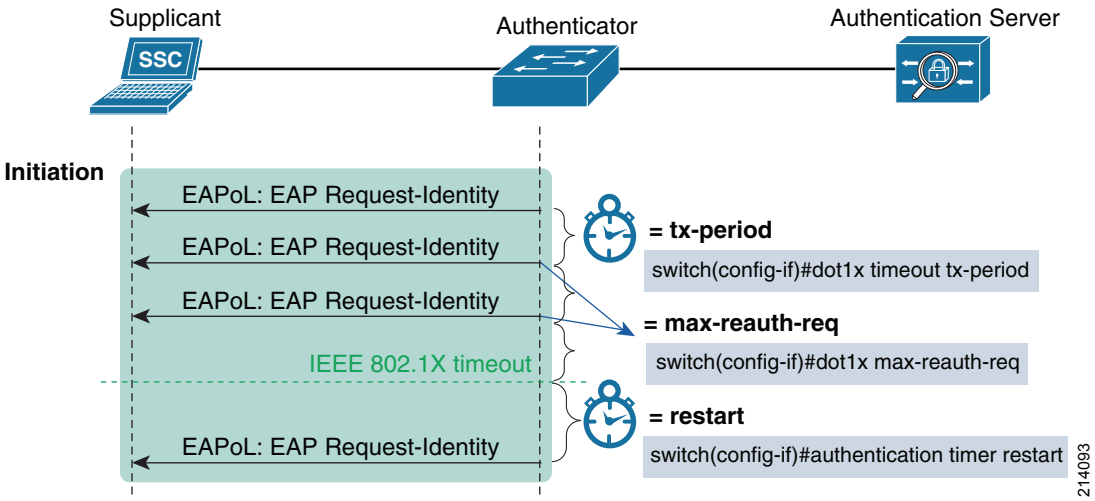
**Tip** **Best Practice Recommendation**—Test tx-period and max-reauth-req in your network. Because the optimal value for the timeout depends on the specific details of your network, Cisco recommends that you use the 802.1X deployment planning phase to test whichever value you select. Pay particular attention to DHCP clients, PXE clients, and the specifics of your managed desktop infrastructure.

## authentication timer restart

This section discusses the timer that controls when 802.1X restarts in the absence of an 802.1X authentication attempt.

If 802.1X times-out and a fallback mechanism has not been configured, or the configured fallback was not successful (that is, MAB failed), the switch waits a period of time defined by the *authentication timer restart* timer, after which it starts the authentication process over from the beginning (see Figure 10).

***Figure 10 authentication timer restart***



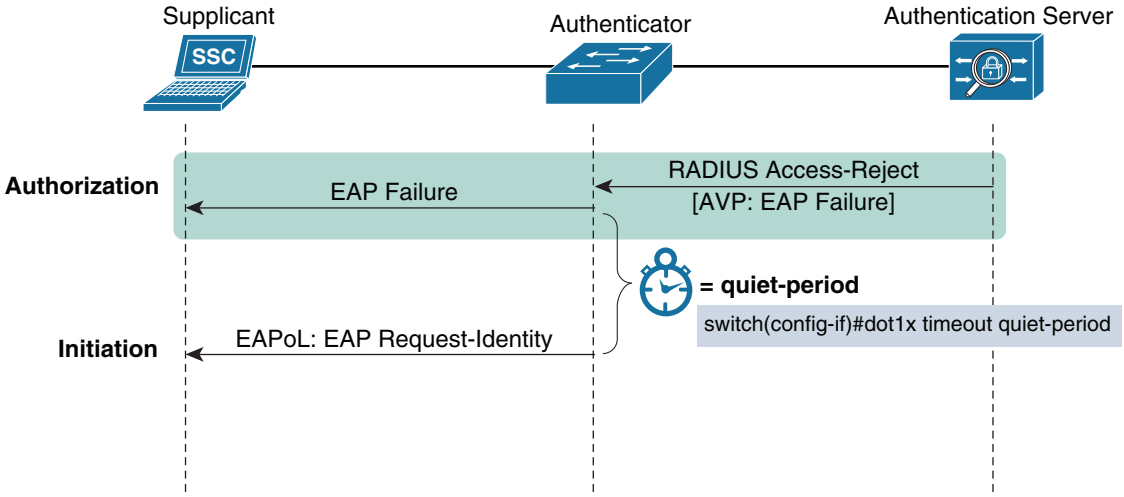

The default authentication timer restart timer is 60 seconds. There is usually no need to modify this timer.

## dot1x timeout quiet-period

This section discusses the timer that controls when 802.1X restarts after a failed 802.1X authentication attempt.

If 802.1X fails and there are no failover mechanisms enabled (MAB, Web Authentication, AuthFail VLAN), the switch waits for a period of time known as the *quiet-period*. Figure 11 shows the operation of the quiet-period timer.

***Figure 11 dot1x timeout quiet-period***



Most customers typically do have a failover mechanism enabled, so the quiet-period timer is rarely invoked.

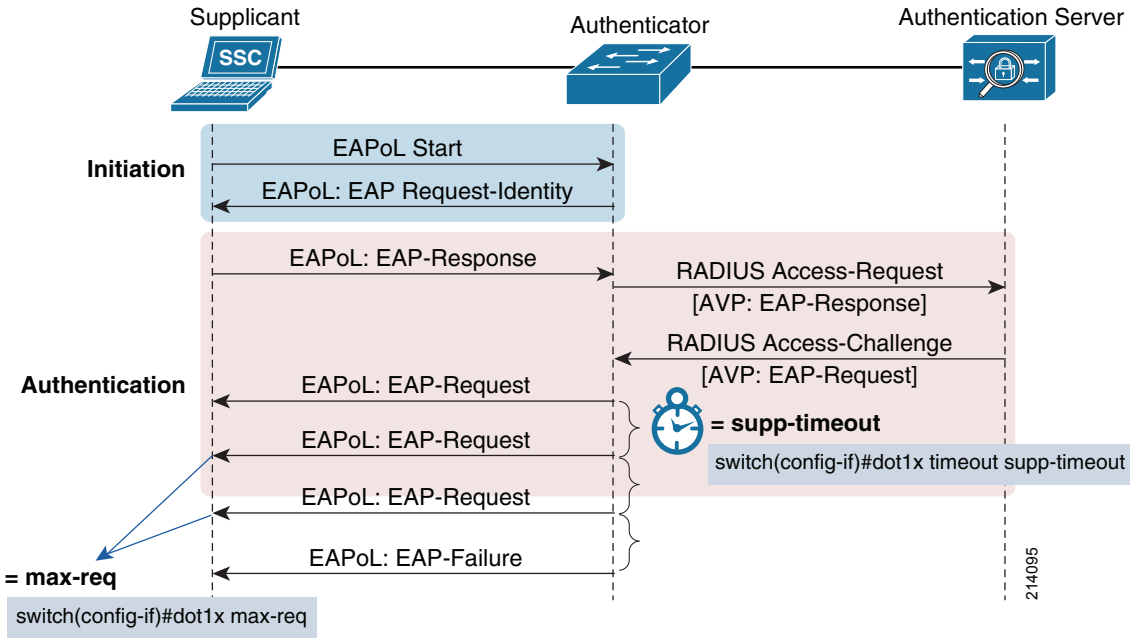
The default quiet-period is 60 seconds. There is usually no need to modify the quiet-period timer.

## dot1x timeout supp-timeout and dot1x max-req

This section discusses the timers that control the timeout and retry behavior of an 802.1X-enabled port when a supplicant stops responding in the middle of an authentication (see Figure 12).

*supp-timeout* and *max-req* are similar to tx-period and max-reauth-req except that they apply only after the supplicant has responded to the initial Request-Identity message. They are not commonly invoked, because they take effect only during rare events such as a supplicant that stops functioning mid-authentication or a transmission failure on the wire.

***Figure 12 dot1x timeout supp-timeout and dot1x timeout max-req***




The default dot1x timeout supp-timeout value is 30 seconds. The default dot1x max-req value is 2. In practice, there is almost never any need to modify either of these values.

## dot1x timeout server-timeout

The port-based configuration *dot1x timeout server-timeout* can influence the RADIUS retransmission behavior of the switch when the authentication server stops responding. In Cisco Catalyst switches, however, retransmission to the server is best handled through the global RADIUS configuration. Therefore, the port-based dot1x timeout server-timeout configuration is redundant. The default value is 0 (which disables this timer in favor of the global RADIUS configuration) and should not be changed.

**Tip** **Best Practice Recommendation**—Do not modify dot1x timeout server-timeout. Cisco does not recommend modifying the default value of dot1x timeout server-timeout. Instead, you should rely on the global RADIUS server timeout configuration to control retransmission to the authentication server.

# RADIUS Accounting

RADIUS Accounting is fully compatible with 802.1X and should be enabled as a best practice. RADIUS Accounting provides detailed information about the authenticated session and enables you to correlate a username with MAC address, IP address, switch, port, and even usage statistics.

Table 2 shows the relevant fields available in a typical Accounting Start message sent from the switch to the authentication server.

***Table 2 RADIUS Accounting Start Message Fields***

| RADIUS Attribute | Example Value | Significance |
|---|---|---|
| Acct-Session-Id(44) | 00000122 | A unique accounting identifier that makes it easy to match start, interim-update and stop records in a log file |
| Vendor-Specific(26) v=Cisco(9<br>Cisco-AVPair: audit-session-id | 0A640A0400000057193E518F | A unique session identifier derived by the switch from the IP address of the switch, a session count, and the session start timestamp |
| User-Name(1) | SEP001E4AA900A8 | Name of authenticated endpoint |
| Acct-Status-Type(40) | Start | Type of accounting message |
| NAS-Port-Type(61) | Ethernet | Port type to which authenticated endpoint is connected |
| NAS-Port(5) | 50248 | Numerical representation of the port to which the authenticated endpoint is connected |
| NAS-Port-Id(87) | FastEthernet2/48 | Port to which the authenticated endpoint is connected in human-readable format |
| Called-Station-Id(30) | 00-1F-6C-3E-56-8F | MAC address of switch |
| Calling-Station-Id(31) | 00-1E-4A-A9-00-A8 | MAC address of authenticated endpoint |
| Service-Type(6) | Framed-User | Authentication type:<br>• Framed-User (2)= 802.1X<br>• Call-Check (10)=MAB<br>• Outbound (5)=Wired WebAuth |
| NAS-IP-Address(4) | 10.100.10.4 | IP address of the switch |

If some information is either not available at the time of the Accounting Start message or changes at some point, the switch sends an Interim Update message with the new information. Table 3 shows relevant fields in an Interim-Update message once the authenticated endpoint acquired an IP address (RADIUS Attribute 8, Framed-IP-Address).

*Table 3 Interim-Update Fields*

| RADIUS Attribute | Example Value | Significance |
|---|---|---|
| Acct-Session-Id(44) | 00000122 | A unique accounting identifier that makes it easy to match start, interim-update and stop records in a log file |
| Vendor-Specific(26)<br>v=Cisco(9)<br>Cisco-AVPair: audit-session-id | 0A640A0400000057193E518F | A globally unique session identifier derived by the switch from the IP address of the switch, a session count, and the session start timestamp; included in all RADIUS messages, making it easier to match authentication and accounting records |
| Framed-IP-Address(8) | 10.100.41.200 | IP address of authenticated endpoint |
| User-Name(1) | SEP001E4AA900A8 | Name of authenticated endpoint |
| Acct-Session-Time(46) | 27 | Duration of current session (in seconds) |
| Acct-Input-Octets(42) | 2614 | Input octets for this session |
| Acct-Output-Octets(43) | 2469 | Output octets for this session |
| Acct-Input-Packets(47) | 7 | Input packets for this session |
| Acct-Output-Packets(48) | 18 | Output packets for this session |
| Acct-Status-Type(40) | Interim-Update | Type of accounting message |
| NAS-Port-Type(61) | Ethernet | Port type to which authenticated endpoint is connected |
| NAS-Port(5) | 50248 | Port to which authenticated endpoint is connected |
| NAS-Port-Id(87) | FastEthernet2/48 | Port to which authenticated endpoint is connected in human-readable format |
| Called-Station-Id(30) | 00-1F-6C-3E-56-8F | MAC address of switch |
| Calling-Station-Id(31) | 00-1E-4A-A9-00-A8 | MAC address of authenticated endpoint |
| Service-Type(6) | Framed-User | Authentication type:<br>• Framed-User (2)= 802.1X<br>• Call-Check (10)=MAB<br>• Outbound (5)=Wired WebAuth |
| NAS-IP-Address(4) | 10.100.10.4 | IP address of the switch |

When a session is terminated, the switch sends an Accounting-Stop record. Table 4 shows relevant fields in an Accounting-Stop record when the authenticated endpoint unplugs from the port. Note that complete usage information (Acct-Session-Time, Octets-Output, Octets-Input, and so on) is available.

*Table 4 Accounting Stop Fields*

| RADIUS Attribute | Example Value | Significance |
|---|---|---|
| Acct-Session-Id(44) | 00000122 | A unique accounting identifier that makes it easy to match start, interim-update and stop records in a log file |
| Vendor-Specific(26)<br>v=Cisco(9)<br>Cisco-AVPair:<br>audit-session-id | 0A640A0400000057193E518F | A globally unique session identifier derived by the switch from the IP address of the switch, a session count, and the session start timestamp; included in all RADIUS messages, making it easier to match authentication and accounting records. |

*Table 4 Accounting Stop Fields*

| | | |
|---|---|---|
| Framed-IP-Address(8) | 10.100.41.200 | IP address of authenticated endpoint |
| User-Name(1) | SEP001E4AA900A8 | Name of authenticated endpoint |
| Acct-Terminate-Cause(49) | Lost-Carrier | Indicates why the session was terminated |
| Acct-Session-Time(46) | 493992 | Duration of current session (in seconds) |
| Acct-Input-Octets(42) | 26644714 | Input octets for this session |
| Acct-Output-Octets(43) | 52824579 | Output octets for this session |
| Acct-Input-Packets(47) | 245767 | Input packets for this session |
| Acct-Output-Packets(48) | 590153 | Output packets for this session |
| Acct-Status-Type(40) | Stop | Type of accounting message |
| NAS-Port-Type(61) | Ethernet | Port type to which authenticated endpoint is connected |
| NAS-Port(5) | 50248 | Port to which authenticated endpoint is connected |
| NAS-Port-Id(87) | FastEthernet2/48 | Port to which authenticated endpoint is connected in human-readable format |
| Called-Station-Id(30) | 00-1F-6C-3E-56-8F | MAC address of switch |
| Calling-Station-Id(31) | 00-1E-4A-A9-00-A8 | MAC address of authenticated endpoint |
| Service-Type(6) | Framed-User | Authentication type:<br>• Framed-User (2)= 802.1X<br>• Call-Check (10)=MAB<br>• Outbound (5)=Wired WebAuth |
| NAS-IP-Address(4) | 10.100.10.4 | IP address of the switch |

# Using IP Telephony

Cisco Catalyst switches are fully compatible with IP telephony and 802.1X. For a full description of features and a detailed configuration guide, see the following URL: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/config_guide_c17-605524.html

# Using Cisco Catalyst Integrated Security Features

Note the following about Cisco Catalyst Integrated Security (CISF) features:

- Port Security—In general, Cisco does not recommend enabling Port Security when 802.1X is also enabled. Because 802.1X enforces a single MAC per port, or per VLAN when MDA is configured for IP telephony, Port Security is largely redundant and may in some cases interfere with the expected operation of 802.1X.
- DHCP Snooping—DHCP Snooping is fully compatible with 802.1X and should be enabled as a best practice.
- Dynamic ARP Inspection—Dynamic ARP Inspection is fully compatible with 802.1X and should be enabled as a best practice.

- IP Source Guard—IP Source Guard is compatible with 802.1X and should be enabled as a best practice.

# Deployment Summary for 802.1X

Address the following major design decisions before deploying 802.1X authentication:

- Evaluate your 802.1X design as part of a larger deployment scenario.
- Select EAP method(s) that meet the requirements of your security policy and the capabilities of your infrastructure.
- Re-use existing credentials.
- Plan for credential distribution (pre- and post- 802.1X deployment), maintenance, expiration, and renewal.
- Select a supplicant that provides the required functionality.
- Configure your supplicant to send EAPoL-Start messages.
- Select an authentication server that supports your EAP method and backend databases while providing sufficient monitoring and troubleshooting capabilities.
- Enable machine authentication.
- Enable user authentication if required.
- Disable re-authentication.
- Decide how many endpoints per port you must support and configure the most restrictive host mode.
- If your network includes WoL endpoints, use an open access-based deployment scenario, change the control direction to allow magic packets, or deploy a hardware-based supplicant to those endpoints.
- Identify a session termination method for indirectly connected endpoints:
  - CDP Enhancement for Second Port Disconnect (Cisco IP phones)
  - Proxy EAPoL-Logoff (third-party IP phones)
  - Inactivity timer with IP Device Tracking (physical/virtual hub)
- Identify endpoints that may fail 802.1X and, if your security policy permits it, plan for supplementary access (MAB, Web Authentication, AuthFail VLAN).
- Enable RADIUS accounting.
- Disable Port Security.
- Identify endpoints without supplicants and provide a mechanism to grant them network access (MAB, Web Authentication, Guest VLAN).
- Modify the tx-period timer and max-reauth-req variable to allow endpoints without supplicants to get faster access to the network using a supplementary method.

# References

## TrustSec 1.99 Documents

- Wired 802.1X Deployment Guide—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/Dot1X_Deployment/Dot1x_Dep_Guide.html
- IP Telephony for 802.1X Design Guide—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/IP_Tele/IP_Telephony_DIG.html
- MAC Authentication Bypass Deployment Guide—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/MAB/MAB_Dep_Guide.html
- TrustSec Phased Deployment Configuration Guide—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/Phased_Deploy/Phased_Dep_Guide.html
- Local WebAuth Deployment Guide—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/WebAuth/WebAuth_Dep_Guide.html
- Scenario-Based TrustSec Deployments Application Note—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/Scenario_based_AppNote/Scenario_based_AN.html
- TrustSec 1.99 Deployment Note: FlexAuth Order, Priority, and Failed Authentication—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/FlexAuthNote/flexauth-note.html
- TrustSec Planning and Deployment Checklist—
http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/TrustSec_1.99/TrustSec_Checklist/trustsec-199_checklist.html

## Related Documents

- Configuring WebAuth on the Cisco Catalyst 3750 Series Switches—
http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/sw8021x.html
- Configuring WebAuth on the Cisco Catalyst 4500 Series Switches—
http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst4500/12.2/53SG/configuration/webauth.html
- Configuring WebAuth on the Cisco Catalyst 6500 Series Switches—
http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/webauth.html
- Cisco IOS Firewall authentication proxy—
http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a0080094eb0.shtml

- WebAuth with Cisco Wireless LAN Controllers—http://www.cisco.com/en/US/partner/tech/tk722/tk809/technologies_configuration_example09186a008076f974.shtml#external-process
- 802.1X Quick Reference Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/whitepaper_c27-574041.pdf
- 802.1X Deployment Scenarios Design Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/whitepaper_C11-530469.html
- 802.1X Deployment Scenarios Configuration Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/Whitepaper_c11-532065.html
- MAC Authentication Bypass—http://www.cisco.com/univercd/cc/td/doc/solution/macauthb.pdf
- Basic Web Authentication Design and Configuration Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/app_note_c27-577494.html
- Advanced Web Authentication Design and Configuration Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/app_note_c27-577490.html
- Deploying IP Telephony in 802.1X Networks Design and Configuration Guide—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/config_guide_c17-605524.html
- Flexible Authentication, Order, and Priority App Note—http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/application_note_c27-573287_ps6638_Products_White_Paper.html