Network Virtualization—Access Control Design Guide

This document provides design guidance for enterprises that want to provide Internet and limited corporate access for their guests and partners. Several solutions for guest and partner access challenges are proposed and analyzed in this document, at both the architectural and functional levels. For related information, see the following documents:

- *Network Virtualization—Guest and Partner Access Deployment Guide* (OL-13635-01)
- *Network Virtualization—Network Admission Control Deployment Guide* (OL-13636-01)
- *Network Virtualization—Network Hosted Access Deployment Guide* (OL-13634-01)
- *Network Virtualization—Path Isolation Design Guide* (OL-13638-01)
- *Network Virtualization—Services Edge Design Guide* (OL-13637-01)

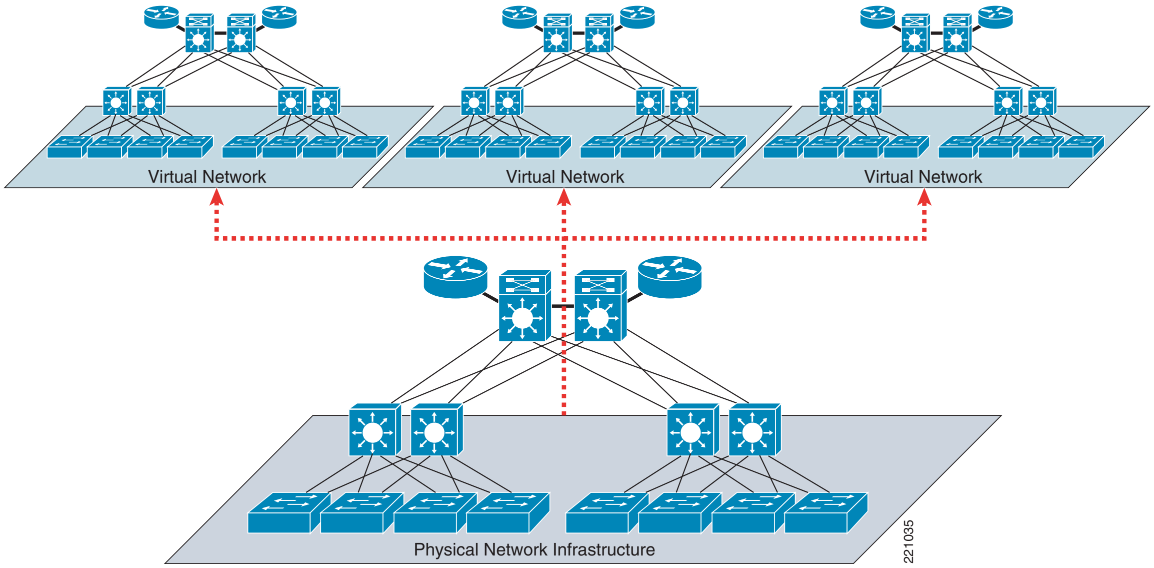
Contents

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

Introduction

The term *network virtualization* refers to the creation of logical isolated network partitions overlaid on top of a common physical infrastructure (see Figure 1). Each partition is logically isolated from the others, and must behave and appear as a fully dedicated network to provide privacy, security, and an independent set of policies, service levels, and even routing decisions.

Figure 1 Network Virtualization



Network virtualization provides multiple solutions to business problems and drivers that range from simple to complex. Simple scenarios include enterprises that want to provide Internet access to visitors (guest access). The stringent requirement in this case is to allow visitors external Internet access, while simultaneously preventing any possibility of unauthorized connection to the enterprise internal resources and services. This can be achieved by dedicating a logical "virtual network" to handle the entire guest communication path. Internet access can also be combined with connectivity to a subset of the enterprise internal resources, as is typical in partner access deployments.

Another simple driver for network virtualization is the creation of a logical partition dedicated to the machines that have been quarantined as a result of a Network Admission Control (NAC) posture validation. In this case, it is essential to guarantee isolation of these devices in a remediation segment of the network, where only access to remediation servers is possible until the process of cleaning and patching the machine is successfully completed.

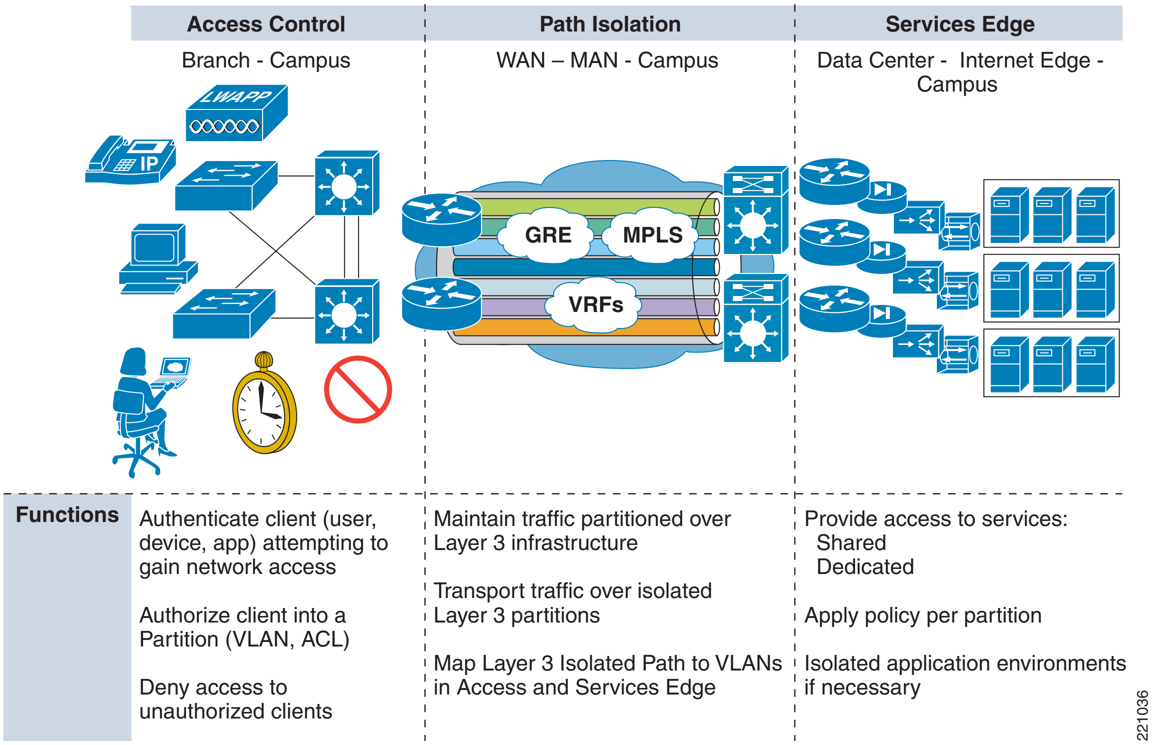
Complex scenarios include enterprise IT departments acting as a service provider, offering access to the enterprise network to many different "customers" that need logical isolation between them. In the future, users belonging to the same logical partitions will be able to communicate with each other and to share dedicated network resources. However, some direct inter-communication between groups may be prohibited. Typical deployment scenarios in this category include retail stores (for example, Best Buy, Albertson's, Wal-Mart, and so on) that provide on-location network access for kiosks or hotspot providers.

The architecture of an end-to-end network virtualization solution targeted to satisfy the requirements listed above can be separated in the following three logical functional areas:

- Access control
- Path isolation
- Services edge

Each area performs several functions and must interface with the other functional areas to provide the end-to-end solution (see Figure 2). This design guide focuses on the access control functional area to securely grant and control authorized access into any internal network system, while providing optional access to guests or partners.

Figure 2 Network Virtualization—Three Functional Areas



	Access Control	Path Isolation	Services Edge
	Branch - Campus	WAN – MAN - Campus	Data Center - Internet Edge - Campus
Functions	Authenticate client (user, device, app) attempting to gain network access Authorize client into a Partition (VLAN, ACL) Deny access to unauthorized clients	Maintain traffic partitioned over Layer 3 infrastructure Transport traffic over isolated Layer 3 partitions Map Layer 3 Isolated Path to VLANs in Access and Services Edge	Provide access to services: Shared Dedicated Apply policy per partition Isolated application environments if necessary

The access control functional area identifies the users or devices logging into the network so they can be successfully assigned to the corresponding groups. An identity is an indicator of a client in a trusted domain. In this architecture, it is used as a pointer to a set of rights or permissions to allow for client differentiation. The model described in this document demonstrates how to use identities as not only a security mechanism, but also how to use identity to provide permissions to service within a domain. Although network services are arbitrary, this represents a linkage to path isolation techniques to provide a holistic form of differentiation between various types of clients. Access control also promotes authentication: the process of establishing and confirming the identity of the client requesting services. Authentication is crucial for network-based security benefits, and to establish corresponding authorization as well.

When identified, the endpoints must be authorized onto the network. To achieve this, the enterprise LAN edge port on which an endpoint connects is activated and configured with certain characteristics and policies. Examples of authorization include the configuration of the VLAN membership of a port based on the results of an authentication process, and the dynamic configuration of port ACLs based on the authentication.

Note For wireless access, the concept of a port can be replaced by the association between client and access point (AP). When authorizing a wireless device, the association is customized to reflect the policy for the user or device. This customization can take the form of the selection of a different wireless LAN (WLAN), VLAN, or mobility group, depending on the wireless technology employed.

When an endpoint is authorized on the network, it can be associated to a specific group that typically corresponds to a separate partition or domain. Thus, the authorization method ultimately determines the mapping of the endpoint to an end-to-end virtual network. For example, when a VLAN is part of a virtual network, a user authorized onto that VLAN is therefore authorized onto the virtual network.

The main authentication scenarios for the enterprise are as follows:

- Client-based authentication for endpoints with client software
- Clientless authentication for endpoints with no client software

The current state of the technology provides broad support for VLAN assignment as an authorization alternative. In the cases where policy changes based on authentication are required and only VLAN assignment authorization is available, a static assignment of a policy to a VLAN provides the required linkage between the user authorization and the necessary policy. In effect, the policy is applied to the VLAN because users are subject to the policy when authorized onto the VLAN. The primary use of VLAN assignment promotes differentiation, and is critical to linkages to path isolation techniques. In essence, VLANs may be mapped into separate policy domains, which define the correct entrance criteria into the path isolation architecture alternatives.

Various access control technologies are discussed in this document: 802.1X, Guest-VLAN, Auth-Failed VLAN, MAC-Authentication Bypass (MAB), and so on. Note the following two important points:

- The various access control technologies are discussed in the context of network virtualization. This means, for example, that the reader should not expect to find here all the details regarding 802.1X deployments, but only the portions of that technology that have been validated and positioned as part of the network virtualization project to provide an answer to the business problems previously listed.
- Not all the technologies found in this design guide represent the right fit for each business problem. For example, the use of Guest and Auth-Failed VLAN features may be particularly relevant in guest and partner access scenarios, but not in deployments aiming to fulfill different business requirements (as for example, NAC quarantining). To properly map the technologies discussed here with each specific business problem, it is thus recommended to see the accompanying deployment guides:
  - *Network Virtualization—Guest and Partner Access Deployment Guide* (OL-13635-01)
  - *Network Virtualization—Network Admission Control Deployment Guide* (OL-13636-01)
  - *Network Virtualization—Network Hosted Access Deployment Guide* (OL-13634-01)

Technology Scope

The client-based framework focuses on 802.1X only as the access control method to provide holistic control over client access to the network. 802.1X always assumes a supplicant at the edge. 802.1X can give customers ubiquitous, port-based access control and provides them with the ability to manage access control on multiple levels for wired and wireless integration purposes. In support of network virtualization, 802.1X can also allow customers to leverage the notion of an authenticated identity with granular policy controls. Although out of this document scope, 802.1X can also provide auditing/accounting measures to network visibility and automate encryption techniques for end stations (wireless only today).

Upon evaluation of 802.1X, a customer must take Guest-VLAN interoperability into account. This design guide discusses recent changes in this arena. It also addresses the Auth-Fail-VLAN to provide wired topologies a method to provide clients network access that is illegitimate and be otherwise failed on any connection attempt into the networked system. The Auth-Fail-VLAN is positioned here as a means to provide access for the 802.1X-enabled partner or guest. It is not positioned as a de facto recommendation for any 802.1X deployment. This design guide also introduces other clientless methods of access control to provide access as well. This form of access control is device-specific in nature, and is discussed in the wired context only. This functionality is MAC-Auth-Bypass. In all cases, Windows Active Directory was used as the backend identity store as the verified directory infrastructure.

This document does not discuss the following technology areas:

- Web-Auth
- IPsec authentication/remote access
- In-depth concepts on identity management and single sign-on
- Privacy issues—Packet confidentiality and integrity
- Topology-independent access control
- In-depth policy administration
- In-depth authorization techniques
- Specific EAP methods
- X.509 certificates and PKI
- EAP over UDP (EAPoUDP)
- NAC posture assessment/remediation

Client-Based Authentication

802.1X offers an efficient framework to a protected network for authenticating and administering user traffic. Together with technology extensions and supplemental authentication techniques, 802.1X builds on access control to establish a technology solution that can improve the security of physical and logical access to LANs.

802.1X Framework

The use of 802 LANs in public and semi-public places has dramatically increased. There is now a desire to provide a mechanism to associate identities with the port of access to the LAN to establish authorized access. 802.1X ties the Extensible Authentication Protocol (EAP) to both the wired and wireless LAN media and supports multiple authentication methods. 802.1X defines a generic framework that is able to use different authentication mechanisms without implementing these mechanisms outside the backend authentication infrastructure and client devices. 802.1X specifies a protocol framework between devices desiring access to a LAN (supplicants) and devices providing access to a LAN (authenticators). Various credentials, such as token cards, Kerberos, one-time password, certificates, and public key authentication can be used with 802.1X. Primarily, 802.1X is an encapsulation definition for EAP over an IEEE 802 media. This is known as EAP over LAN, or EAPOL. EAPOL transports authentication messages (EAP) between supplicant (user/PC) and authenticator (switch or access point). 802.1X always assumes a secure connection, and the actual enforcement is done via MAC-based filtering and port-state monitoring.

Although 802.1X is the recommended method to deploy access control in an enterprise environment, it is not the specific focus in this paper. The business problems that network virtualization is aimed to solve in this phase include the following:

- Guest access
- Partner access
- NAC remediation
- Hosted access

Hosted access and NAC remediation environments are not typically enabled for 802.1X at present. The need remains for some way to provide access to guest or partners when they are equipped with an unmanaged 802.1X supplicant. The 802.1X supplicant of the guest or partner may indeed be managed, but not by the IT staff that owns the network into which they plug. Thus, this design guide focuses only on what it takes to allow guest or partner online access in a virtualized environment when they are equipped with an 802.1X supplicant on the device.

Wireless Guest Access

Wireless users typically access the network differently than wired users. The paradigm of public access has extended to the enterprise. Mobility demands network connectivity. Enterprise guest access services are now a necessity in the corporate environment. The solution is made up of many components: access points, controllers, and management systems.

A detailed description and comparison of the various wireless deployment options is not within the scope of this document; a brief, high-level description of each scenario is provided in the following sections but only in the context of network access control. For more information on Cisco Integrated Wireless Networks, see the following URL:
http://www.cisco.com/en/US/netsol/ns340/ns394/ns348/ns337/networking_solutions_package.html

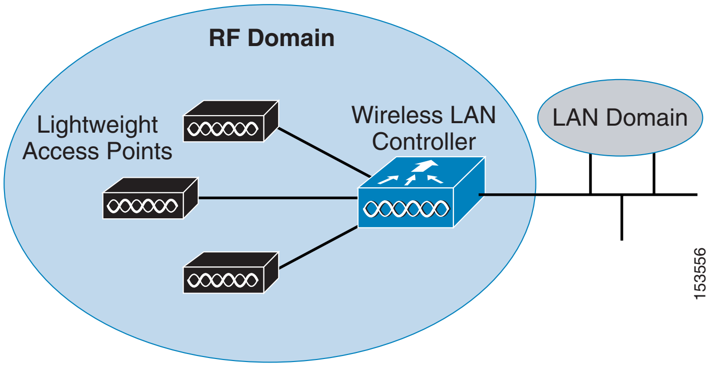
A typical company security policy most likely requires the implementation of various types of authentication and encryption for various types of users. For example, open authentication and no encryption are the typical choices when providing guest access, whereas 802.1X authentication and strong encryption are usually adopted for internal employees. This is achieved by defining multiple service set identifiers (SSIDs) on each access point, with each SSID characterized by its own security policies.

End users associate with the closest access point by selecting a specific SSID to access the enterprise network. After this point, the WLAN Controller allows traffic to be logically separated from the traffic for users belonging to different groups. This is described in more detail in the following section.

Lightweight Access Point Deployment with the Cisco WLAN Controller

A WLAN controller system is used to create and enforce policies across many different lightweight access points in this architecture (see Figure 3). Security, mobility, quality of service (QoS), and other functions essential to WLAN operations can be efficiently managed across an entire wireless enterprise by centralizing intelligence within a controller system. Furthermore, by splitting functions between the access point and the controller, IT staff can simplify management, improve performance, and increase security of large wireless networks.

Figure 3 Cisco Wireless LAN Controller



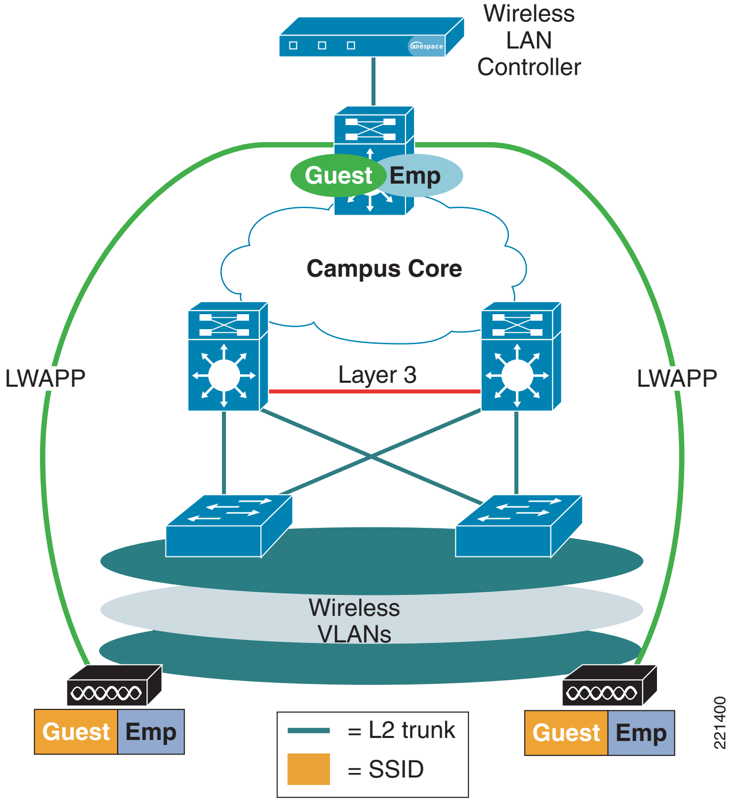
LWAPP revolutionizes the way WLAN deployments are managed with the concept of split MAC, which means the ability to separate the real-time aspects of the 802.11 protocol from most of its management aspects. In particular, real-time frame exchange and certain real-time portions of MAC management are accomplished within the access point, while authentication, security management, and mobility are handled by WLAN controllers. The Cisco Centralized WLAN Solution, which uses LWAPP, is the first centralized WLAN system to use the split MAC.

From a traffic handling perspective, all data traffic originating from wireless clients associated to the distributed lightweight access points is encapsulated on the access points themselves and carried to a centralized wireless LAN controller, which aggregates the traffic and represents the single point of ingress and egress for IP traffic to and from the wired network. Traffic is tunneled from the access points to the centralized controller, leveraging LWAPP. The LWAPP tunnel is a Layer 2 tunnel (the original Ethernet frame is LWAPP-encapsulated), which carries both control and data traffic. Data traffic uses UDP port 12222, control traffic is encapsulated in UDP port 12223, and Radio Resource Manager uses ports 16666/16667. In addition, the control traffic is AES-encrypted, while the data is in the clear.

There is not a separate logical tunnel for each defined SSID; only a single logical tunnel is built between each access point and the centralized WLAN controller. This LWAPP tunnel is used to carry the data traffic for all the wireless clients associated to the access point, independently from the SSID they are using for this association.

Figure 4 shows the deployment of the lightweight architecture in an enterprise campus network where two categories of users (employees and guests) are defined as an example.

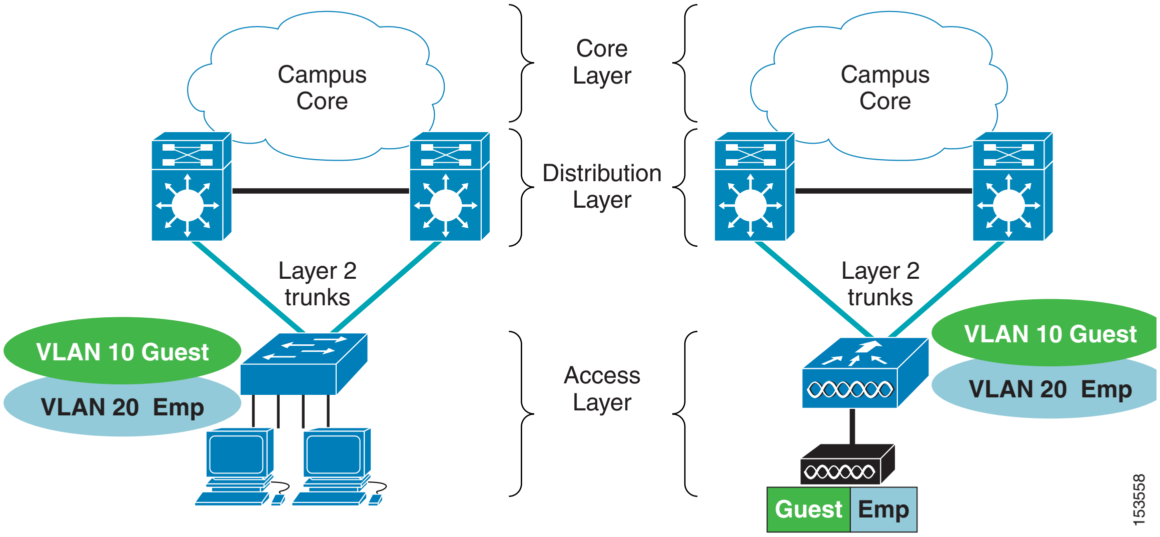
Figure 4 Lightweight Architecture Deployment



From the traffic isolation perspective, this scenario is very similar to the wired deployment in a traditional campus design. The reason is that traffic from various categories of users associating with their own SSID, after being aggregated to the main WLAN controller, is bridged to a corresponding VLAN and carried up to the first Layer 3 hop device.

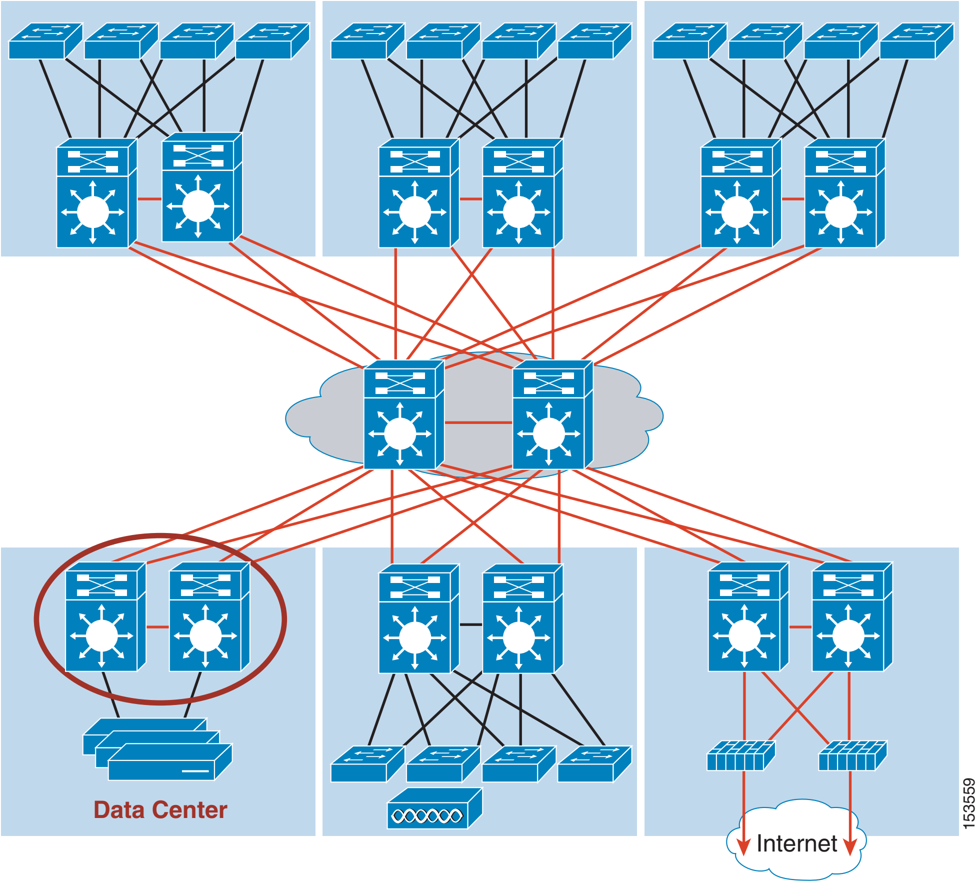
Figure 5 shows how the use of VLANs allows maintaining separation between the guest traffic and the enterprise internal traffic in the Layer 2 domain, in a very similar way to the wired scenario for a traditional campus deployment (Layer 2 in the access).

Figure 5 Similarities Between Wired and Wireless Deployments



Several alternative designs can be deployed when positioning the WLAN controllers in the campus network. Cisco recommends placing the WLAN controllers in a centralized location (for example, a data center) to leverage the high availability and continuous monitoring characteristic of such an environment. (See Figure 6.)

Figure 6 Cisco Wireless LAN Controller Deployment in a Campus Network

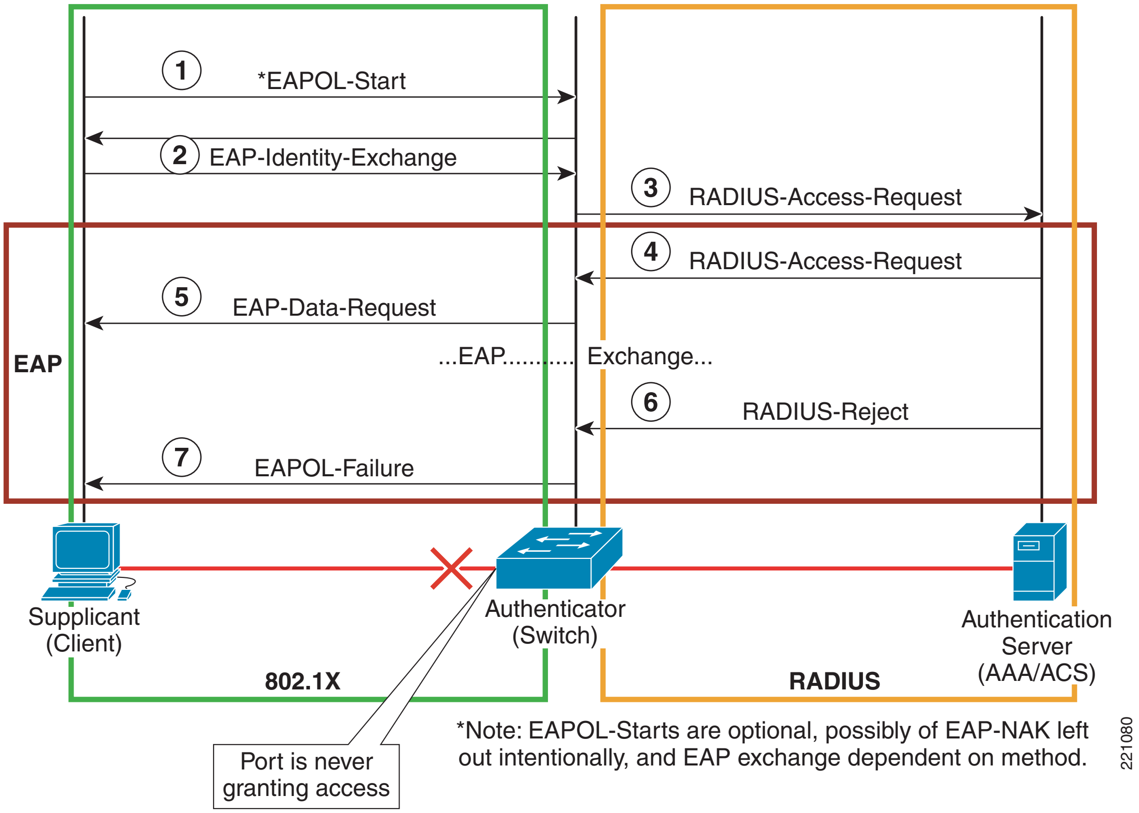


For more information on the design and implementation of the Cisco Unified Wireless Network based on the unified wireless architecture, which includes products operation with LWAPP, see the *Enterprise Mobility 3.0 Design Guide* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/solution/emblty30.pdf

802.1X Authentication Failure VLAN (Wired)

On a traditional 802.1X wired port, the switch does not provide access to the network until the supplicant connected to a port is authenticated, by verifying its identity information with an authentication server. There is no concept of an SSID for wired topologies today. For both media types, authentication failures work great in preventing rogue access to a network. This is a primary reason that some enterprises seek to enable 802.1X pervasively at the LAN edge. This default behavior is shown in Figure 7.

Figure 7 Typical 802.1X Authentication Failures



However, for wired topologies, there must be a way to deal with the fact that an 802.1X-enabled guest or partner can plug into the enterprise LAN via wired ports.

The Auth-Fail-VLAN can be configured for an 802.1X port to provide limited services to clients. These clients are 802.1X-compliant and cannot access another VLAN because they fail the authentication process. A restricted VLAN allows users without valid credentials in an authentication server (typically, visitors to an enterprise) to access a limited set of services. The administrator can control the services available to the restricted VLAN.

Note The same VLAN can be configured as both the Guest-VLAN and the Auth-Fail-VLAN when providing the same services to both types of users. The Guest-VLAN is discussed in detail in Clientless-Based Authentication, page 18.

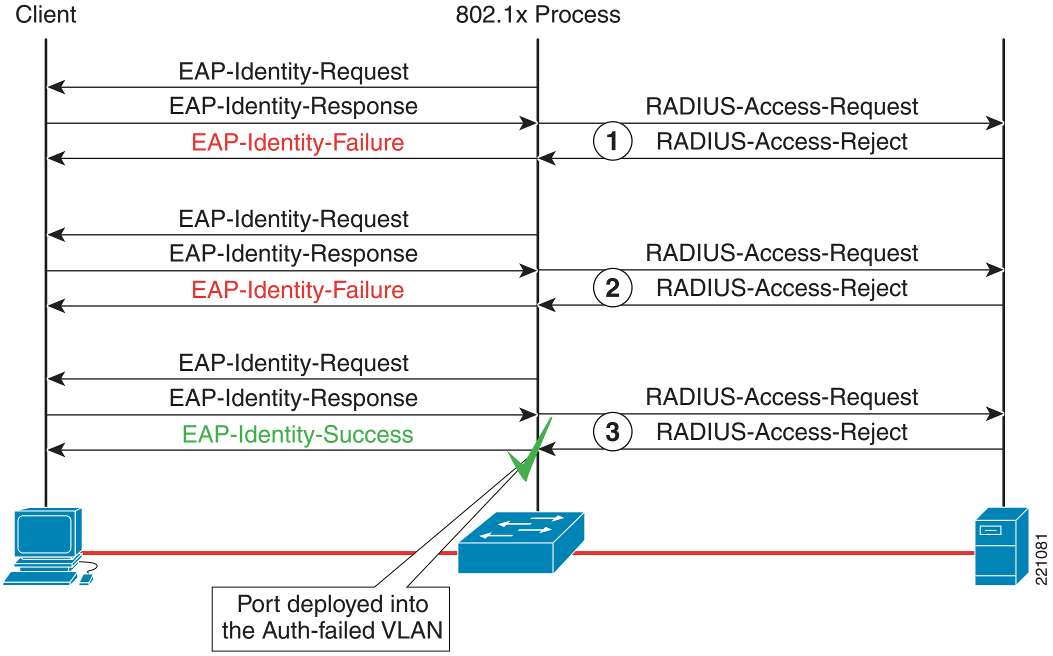
With the Auth-Fail-VLAN feature, you can configure the VLAN on a per-port basis and is enabled (by default) after three 802.1X authentication attempts. The port is then enabled and port forwarding is allowed in a VLAN where the supplicant can access the network. The Auth-Fail-VLAN can be configured for an 802.1X port to provide limited services to clients that are 802.1X-compliant and cannot access another VLAN because they fail the authentication process.

There may be several reasons why a user fails the 802.1X authentication. In addition, refer to an over-arching security policy to evaluate the deployment of the Auth-Fail-VLAN. The Auth-Fail-VLAN ultimately grants access to a device or end user that fails authentication. Although this authentication failure event can be differentiated from authorized devices, there is no chance to differentiate an 802.1X-enabled guest or partner who needs some form of network access from a hacker or illegitimate user. The same principle exists in wireless topologies. If 802.11 opens with no authentication provided by a separate SSID, there is no way to keep an illegitimate user off the network. Wireless uses an SSID to differentiate the entire session. Wired can only use an actual authentication failure to attempt accomplish a similar task.

Auth-Fail-VLAN Operational Overview

The authenticator (access switch) counts the failed authentication attempts for a client. When this count exceeds the configured maximum number of authentication attempts (the default is 3), the port is deployed into the Auth-Fail-VLAN. After a port is moved to the Auth-Fail-VLAN, an EAP success message is sent to the client, as shown in Figure 8.

Figure 8 Auth-Fail-VLAN Operation



Any active VLAN can be configured as Auth-Fail-VLAN with the exception of an RSPAN VLAN, or a voice VLAN (VVID). In addition, the Auth-Fail-VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

Note Authentication has to actually fail for this process to complete. Any sort of timeout condition (supplicant/authenticator or authenticator/authentication server) is not addressed by the Auth-Fail-VLAN feature.

Auth-Fail-VLAN Configuration

Following are configuration samples that enable the Auth-Failed VLAN feature on IOS and CatOS authenticators:

- IOS:

```
interface FastEthernet0/1
```

```
switchport access vlan 2
switchport mode access
dot1x pae authenticator
dot1x port-control auto
dot1x auth-fail vlan 5
spanning-tree portfast
spanning-tree bpduguard enable
```

- CatOS:

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port dot1x 2/1 auth-fail-vlan 5
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

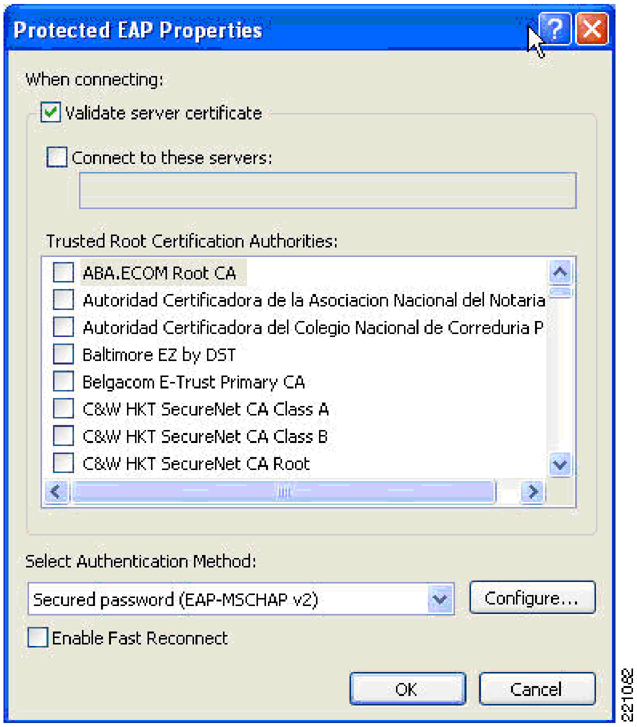
Note Although not verified as part of the solution, the number of failures to deploy a port into the Auth-Fail-VLAN can be configured in IOS via the **dot1x auth-fail max-attempts** command. The default value for this parameter is 3, and 3 is the hard-coded parameter in CatOS.

Auth-Fail-VLAN Verification

Reasons why a user may fail the 802.1X authentication, causing the switch port to be deployed in the Auth-Fail-VLAN, include the following:

- A tunneled EAP method (PEAP or EAP-FAST) is used for authentication, and the supplicant is configured for validating the server certificate. In this case, there are the following two scenarios:
  - Most supplicants are configured to use the Certificate Trust List (CTL) to validate a server certificate with a tunneled method. In this case, the authentication fails, unless the certificate sent by the RADIUS server is trusted by the supplicant. This means the supplicant trusts the intermediary that has signed or issued the server certificate. An example of a pre-populated CTL is shown in Figure 9. This is the Trusted Root Certification Authorities list available with Microsoft supplicants.

Figure 9 Microsoft Supplicant CTL Example



Protected EAP Properties

When connecting:

☑ Validate server certificate

□ Connect to these servers:

Trusted Root Certification Authorities:

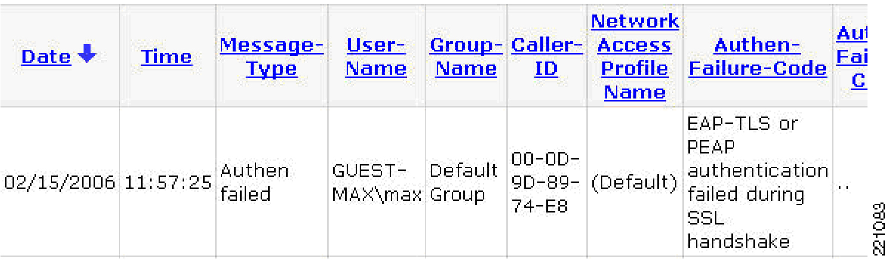
- □ ABA.ECOM Root CA
- □ Autoridad Certificadora de la Asociacion Nacional del Notaria
- □ Autoridad Certificadora del Colegio Nacional de Correduria P
- □ Baltimore EZ by DST
- □ Belgacom E-Trust Primary CA
- □ C&W HKT SecureNet CA Class A
- □ C&W HKT SecureNet CA Class B
- □ C&W HKT SecureNet CA Root

Select Authentication Method: Secured password (EAP-MSCHAP v2)

□ Enable Fast Reconnect

In many situations, including guest access scenarios, the certificate authority (CA) that provided the certificate sent by the RADIUS server is most likely not part of the client CTL (especially in deployments where a private CA is used). As a consequence, the TLS handshake tried in the tunnel establishment phase fails. The client denies the authentication attempt by being unable to verify the backend server to establish an SSL tunnel between client and server. On ACS, the message appears as indicated in Figure 10.

Figure 10 Authentication Failure From Client

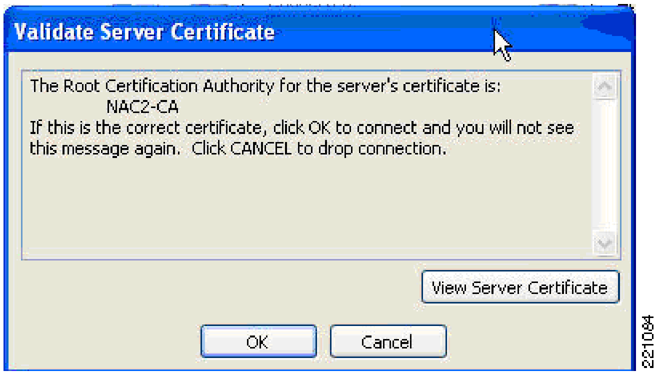


Date ⬇	Time	Message-Type	User-Name	Group-Name	Caller-ID	Network Access Profile Name	Authen-Failure-Code	Aut Fai C
02/15/2006	11:57:25	Authen failed	GUEST-MAX\max	Default Group	00-0D-9D-89-74-E8	(Default)	EAP-TLS or PEAP authentication failed during SSL handshake	..

Note that by default, the Microsoft Supplicant (WZCSVC) and the Cisco Secure Services Client (CSSC) validates the server certificate by default when tunneled methods are configured. Older versions of the Meetinghouse AEGIS client did not trust a server certificate by default.

- Alternatively, a non-default configuration for WZSVC with Windows XP SP2 enables the supplicant to conditionally validate the server certificate. This way, the end user is presented a popup window to inform the user to accept the certificate (similarly to what happens on HTTPS transactions). An example of this capability is offered WZCSVC is shown in Figure 11.

Figure 11 Conditional Trust for a Server Certificate



Validate Server Certificate

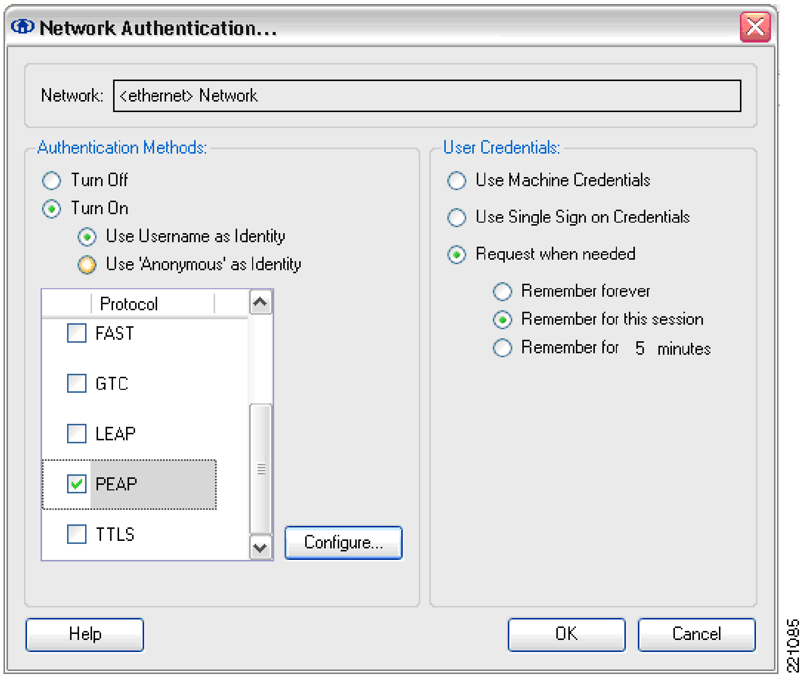
The Root Certification Authority for the server's certificate is:
NAC2-CA
If this is the correct certificate, click OK to connect and you will not see this message again. Click CANCEL to drop connection.

Note This functionality of conditional trust is not available with CSSC.

When the popup is displayed, an end user can manually accept the certificate sent by the authentication server, and avoiding failing the authentication because of SSL handshake. The authentication at this point may still fail for one of the two reasons discussed in bullets two and three below.

- The client is sending wrong credentials to the RADIUS server (or backend authentication server, as Active Directory). Note that most commonly the same credentials used for Windows are also used for 802.1X authentication. This is the default behavior for WCZSVC. For CSSC, the client can be configured to operate via Windows credentials or not, as shown in Figure 12.

Figure 12 CSSC Options for Authentication



Network Authentication...

Network: <ethernet> Network

Authentication Methods:
○ Turn Off
● Turn On
● Use Username as Identity
○ Use 'Anonymous' as Identity

Protocol
□ FAST
□ GTC
□ LEAP
☑ PEAP
□ TTLS

User Credentials:
○ Use Machine Credentials
○ Use Single Sign on Credentials
● Request when needed
○ Remember forever
● Remember for this session
○ Remember for 5 minutes

The supplicant is configured to try an EAP type that is not supported by the RADIUS server.

However, the client would be able to pull a network address from the pool corresponding to the auth-failed VLAN and to gain network connectivity.

In addition, if Active Directory (AD) is used as a backend database with the WZCSVC supplicant and the user is not found in AD, or the password sent is wrong, the supplicant may get stuck during the communication with the RADIUS server. The consequence is that the attempt does not even fail, preventing the deployment of the switch port into the Auth-Fail-VLAN. This implies that no network access at all can be provided in this scenario.

It was noticed that this supplicant does not return a failure response to the failure message of the server. For ACS, this prevents the EAP state machine from getting to the next stage of sending the final EAP-MSCHAP failure code and a RADIUS-Reject. As such, ACS does not send a RADIUS-Reject to the switch immediately. Nonetheless, it should result in a failure. Operationally, the first two authentication failures of each conversation result in a RADIUS challenge as opposed to a reject. Then, a reject is sent on the third unsuccessful attempt. With respect to the Auth-Fail-VLAN, this means an end user may actually have to fail nine times before the Auth-Fail-VLAN activates, because it is enabled upon the receipt of RADIUS-Rejects. In addition, note that the WZCSVC supplicant does not display meaningful messages such as “Account Expired”, “Bad Logon Hours”, and so on. The only failure scenarios that work with this supplicant are a bad password (where the user is otherwise known) or an expired password. This behavior occurs only with PEAP for machines and users who either blindly trust a server certificate from ACS, or who conditionally trust the server certificate and the credentials have actually been removed from a domain.

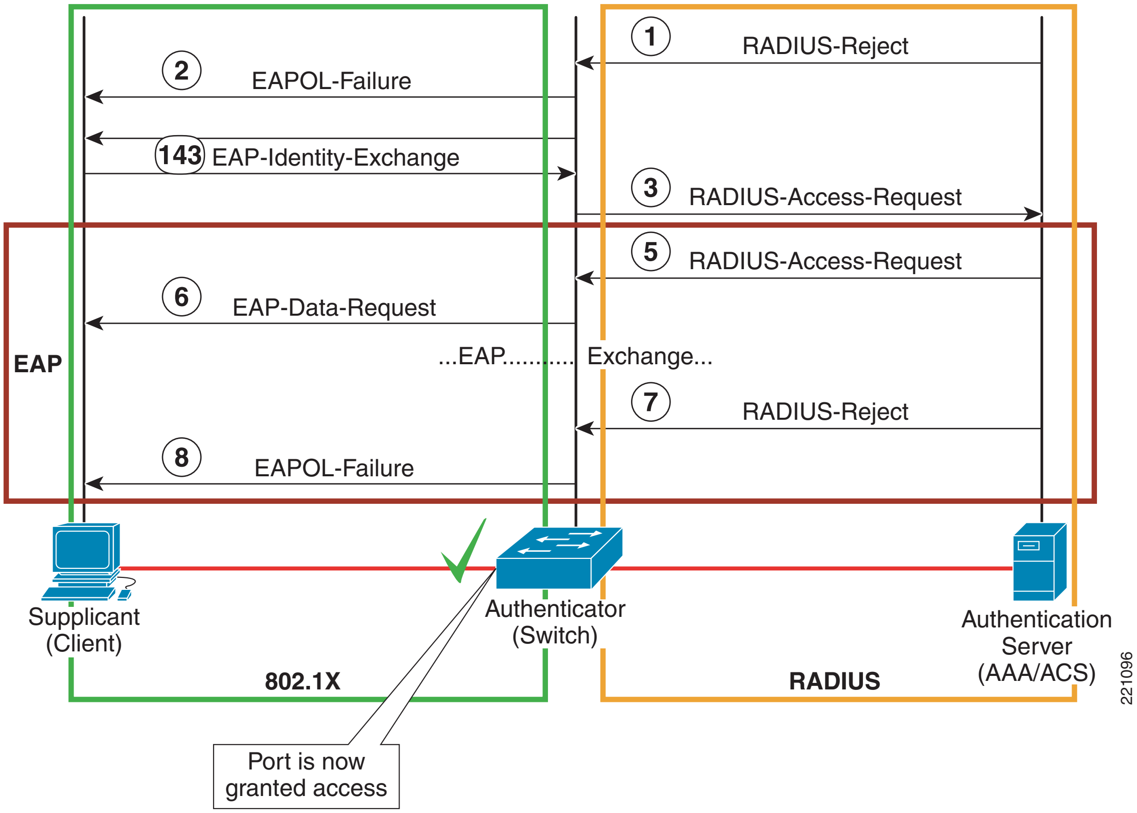
Auth-Fail-VLAN Summary and Recommendations

In the wired media for 802.1X, there exists a need to provide access to devices that fail authentication. This is the Auth-Fail-VLAN. This can also serve as a method to provide 802.1X-enabled guests and partners network access in a network virtualization architecture.

This is not a problem for wireless, because of culture, the secondary nature of the wireless media, and 802.11 station authentication. With 802.11 open, no authentication, and a broadcast SSID, the guest or partner problem can be solved easily without the need to attempt to provide access to devices that actually fail 802.1X authentication. It is typically not a problem for IPsec, PPP, or dial-up environments either. It is a problem for the wired media at the enterprise LAN edge, however. The Auth-Fail-VLAN can serve as a way to deal with the wired 802.1X-enabled entity that is unknown to the hosting enterprise.

However, there are some architectural problems with the Auth-Fail-VLAN as verified above. EAP is between a supplicant and an EAP-Server in the 802.1X framework. Although not precluded by the EAP architecture, Cisco switches today are not EAP-Servers, but authenticators only. Primarily, this means that they serve as an EAP transport via 802.1X and RADIUS, and rely on an authentication server to be an EAP-Server. Because switches operate in pass-through mode for EAP, attempting to modify the result of the authentication conversation from the authenticator alone can be challenging. This behavior is shown in Figure 13 (which begins at the end of a second consecutive failure).

Figure 13 Auth-Fail-VLAN—New Behavior



After a third consecutive failure, the port is enabled rather silently. There is no new CLI added for this change in functionality. At this point of enabling the port, any supplicant can choose to access the network or not (which is ultimately out of the control of a switch). This provides a predictable supplicant behavior, works with any EAP method, and provides a supplicant-agnostic solution. There remain systemic-level gaps (such as the agreement of end-to-end session state) but it should be deployable based on fixes to the DDTSs referenced above. In addition, not all customers run ACS or CSSC, so this should only impact internal deployments that attempt to get the Auth-Fail-VLAN to work. With CSSC, an end user also has the ability to temporarily stop the supplicant from the system tray anyway when they know they are traveling to a foreign network. This disables 802.1X on the supplicant, and it can be treated as a clientless session.

Clientless-Based Authentication

Currently, 802.1X is the recommended port-based authentication method at the access layer in enterprise networks. It has the following three primary components:

- Supplicant
- Authenticator
- Authentication server

Typically, the authenticator tries to authenticate the host device running the supplicant software to the authentication server. With some operating systems, the 802.1X supplicant capability is enabled by default (for example, Windows XP), but not all devices have this supplicant capability embedded into their operating system. For example, most printers, IP phones, fax machines, and so on, do not have this capability but still need to be allowed into the network even without 802.1X authentication. A supplemental authentication technique should be employed as the basis of the non-responsive host issue with 802.1X. This solution-based feature set is MAC Authentication Bypass (MAB). In addition, exception lists on routers or switches are not scalable for large enterprises. Thus, a method is needed for supporting these hosts.

For network virtualization, access control must also focus on clients who do not possess 802.1X capability, or whose 802.1X capability may be temporarily suspended to support mobility into environments where the end user/client may not be otherwise known to the authentication infrastructure in advance. When 802.1X is implemented in such an environment, a customer typically needs the ability to dynamically provision individual MAC addresses (without impacting service availability) for network authentication of non-responsive devices such as printers, video conferencing units, satellite receivers, faxes, and so on. MAB is intended to control network access based on a MAC address. The goals of MAB are to provide network access control on a port basis, based on a MAC address, and to dynamically apply policy to a client session based on a MAC address.

The Guest-VLAN may also be used to provide access for clients incapable of 802.1X and where the client MAC address may be unknown in advance. Although originally designed as a deployment enabled for 802.1X supplicant functionality on end stations, the Guest-VLAN provides an option for mobile guest users as well.

In addition, this document reflects updates to changes in recent functionality across the Cisco Catalyst switching product line that may impact the related architecture to support network virtualization.

Static VLAN Configuration

In this approach, each switch port on the access layer switches in the campus needs to be manually assigned to a VLAN. There are multiple drawbacks to this approach, including the lack of mobility capabilities across the enterprise network and the lack of any mechanism to identify the user before allowing connectivity to the network. In a design supporting multiple user groups that need to remain isolated from each other, there is also the drawback of increased costs because each switch port is reserved for a specific user group, even when not used to capacity.

802.1X Guest-VLAN

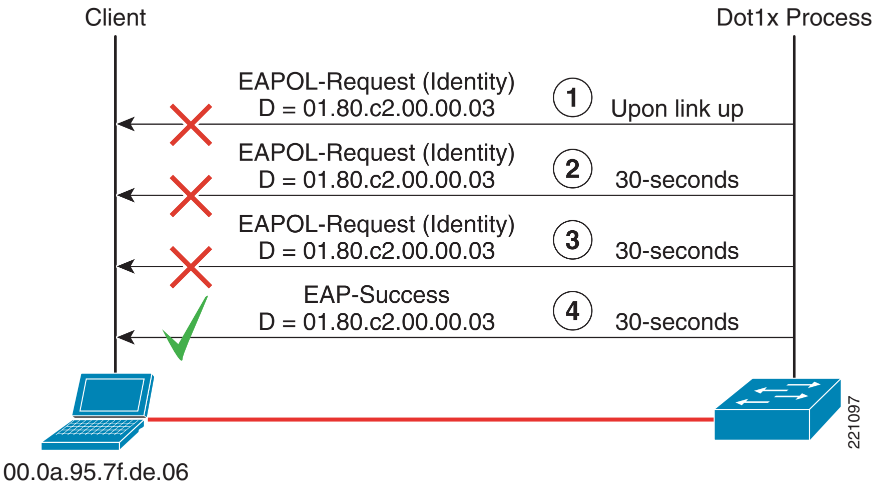
For enterprises that are starting to deploy 802.1X in their networks, leveraging Guest-VLAN functionality is a key element in providing network access to clients that are not equipped with an 802.1X supplicant. The 802.1X Guest-VLAN functionality was initially deployed as a migration tool to allow enterprises to easily migrate client devices to support 802.1X, while still providing network connectivity.

Any VLAN can be configured as the Guest-VLAN, voice VLANs (VVID), and the VLAN used for Remote SPAN (RSPAN). The Guest-VLAN feature is currently supported across all Cisco Catalyst platforms (4500, 3750, 3560, 2950 running Cisco IOS, and 6500 running CatOS); it will be integrated into Cisco IOS software releases for Catalyst 6500 platforms in the near future.

802.1X Guest-VLAN Functionality

Figure 14 shows the functionality of the 802.1X Guest-VLAN feature.

Figure 14 Guest-VLAN Feature



Currently, when a switch port initially receives a link, an EAP-Identity-Request message is transmitted to actively look for an 802.1X supplicant. This happens regardless of whether the device that connected to the port is actually equipped with the supplicant.

Assuming that the user does not have the 802.1X capability on their machine, the request from the switch goes unanswered. After the expiration of a timer (**tx-period**), the switch sends a new EAP-Identity-Request frame. This behavior is dictated by the 802.1X specification. This process continues until the third request from the switch goes unanswered. The number of retries is determined by the value of the **max-reauth-req** parameter. After the maximum number of retries is exceeded, and if the switch port has been configured with the 802.1X Guest-VLAN functionality, the port is moved to the Guest-VLAN and the switch sends an EAP-Success message to the client. This message is ignored and discarded by the client.

From the perspective of the 802.1X process, the port has become authorized and the 802.1X state machine has entered the authenticated state; no further security or authentication mechanisms are applied (the 802.1X state machine stops running). It is basically as if the administrator has disabled 802.1X and hard-set the port into that specific VLAN.

802.1X Guest-VLAN Configuration

The behavior illustrated in Figure 14 is valid when using default values for the 802.1X parameters that affect Guest-VLAN functionality: **max-reauth-req** and **tx-period**.

The **max-reauth-req** parameter sets the maximum number of times that the switch retransmits an EAP-Identity-Request frame on the wire before receiving a response from the connected client. This value is set to two by default. This is why Figure 14 shows two retries (at Steps 3 and 4) after the initial EAP-Identity-Request frame sent at link-up. The commands used to change this parameter (in CatOS and IOS) are as follows:

- CatOS

```
cat6500> (enable) set dot1x max-reauth-req ?
     <max-reauth-req>     maximum number of retries to supplicant (1..10)
```

- Cisco IOS

```
cat3750(config-if)#dot1x max-reauth-req ?
  <1-10>  Enter a value between 1 and 10
```

The **tx-period** parameter sets the number of seconds that the switch waits for a response to an EAP-Identity-Request frame from the client before retransmitting the request. The default value is 30 seconds and is configurable as follows:

- CatOS

```
cat6503> (enable) set dot1x tx-period ?
  <tx-period>                tx period (1..65535 seconds)
```

- Cisco IOS

```
cat3750(config-if)#dot1x timeout tx-period ?
     <1-65535>  Enter value between 1 and 65535
```

The **max-req** parameter is part of the configurable 802.1X parameter in Cisco IOS. The **max-req** parameter is different from the **max-reauth-req** parameter and represents the maximum number of retries a switch performs for EAP-Request frames of types other than EAP-Identity-Request. Basically, this parameter refers to EAP-Data frames, which are the EAP frames exchanged after the supplicant has replied to the initial EAP-Identity-Request frame. For this reason, the **max-req** parameter is effective only when there is a valid 802.1X supplicant connected, and it does not apply to Guest-VLAN services.

-

For a Catalyst 6500 running CatOS software, the situation is different; the main distinction is the fact that in CatOS releases earlier than 8.5, there is no **max-reauth-req** parameter. This implies that the same parameter described above (**max-req**) is used to tune both the number of retries for the EAP-Identity-Request and EAP-Data frames. Note also that the configurable values are consistent with the one detailed for Cisco IOS: **max-reauth-req** (and **max-req**) can vary from 1 to 10 and **tx-period** from 1 to 65535.

The overall configuration of the 802.1X Guest-VLAN is relatively simple but differs on switches running IOS and CatOS software releases, as follows:

- Cisco IOS

```
interface FastEthernet0/1
switchport access vlan 2
switchport mode access
dot1x pae authenticator
dot1x port-control auto
dot1x guest-vlan 10
dot1x max-reauth-req 2
dot1x timeout tx-period 30
spanning-tree portfast
spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port dot1x 2/1 guest-vlan 10
set spantree portfast 2/1 enable
set dot1x max-reauth-req 1
set dot1x tx-period 30
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

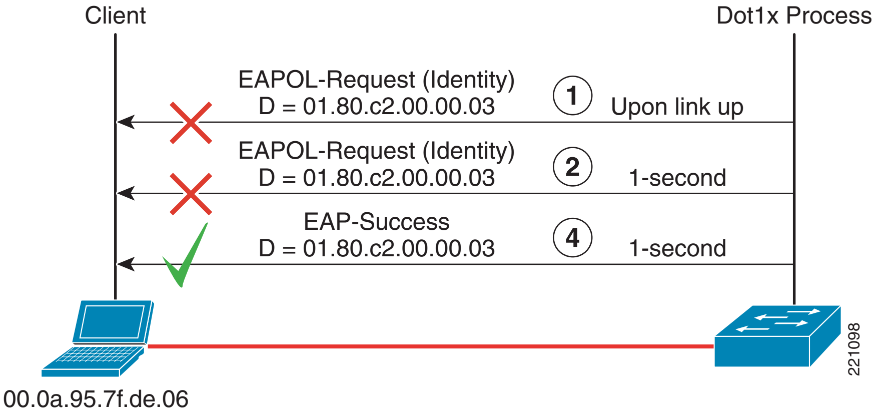
Note In CatOS systems, the values for *max-reauth-req* and *tx-period* are set at a global level, and not per port, as they are in Cisco IOS software.

The following formula calculates the time interval before the Guest-VLAN is enabled:

[(max-reauth-req + 1) * tx-period]

The end station must then attempt to send traffic into the network, so the specific time to ultimately authenticate the end device varies. The operation of tweaked timers to timeout 802.1X quickly as indicated above is shown in Figure 15.

Figure 15 Guest-VLAN with Tweaked Timers



This configuration should be attempted only after considering the consequences that this can have on the regular functionality of 802.1X. Analyzing the integration issues between 802.1X and DHCP at startup time helps in understanding this.

If a user starts up a machine equipped with an 802.1X supplicant, two possible scenarios can occur in relation to the use of 802.1X machine authentication after connecting to a switch port configured for Guest-VLAN. A complete description of machine authentication is not within the scope of this document. However, you can find more information for a deployment using the Microsoft supplicant at the following URL:
http://wwwin-eng.cisco.com/Eng/TME/TSE/IBNS/Understanding_Windows_Machine_Auth.doc

The following two scenarios are possible:

- The 802.1X supplicant is enabled for machine authentication and the switch port is configured with a *max-reauth-req* setting of 0 and *tx-period* setting of 1. At system startup, and subsequent port link-up, the switch immediately sends an EAP-Identity-Request frame in an attempt to find a supplicant online. As a consequence of the 802.1X parameter settings defined here, the switch port is deployed into the Guest-VLAN after two seconds and the 802.1X state machine stops before the supplicant can authenticate. At a certain point during the startup process, the supplicant on the clients initializes and, because machine authentication is enabled, it can send an EAPOL-Start frame to restart the authentication process. This message is sent by default with CSSC, but not with the native Windows XP 802.1X supplicant, which requires a specific setting of the Windows registry. This is described at the following URL:
  http://www.microsoft.com/WindowsServer2003/techinfo/overview/wififaq.mspx#EAAAA

  However, even assuming that the 802.1X supplicant is enabled to send EAPOL-Start frames, the DHCP and the 802.1X processes are completely asynchronous. Therefore, the machine can acquire an IP address from the DHCP pool associated to the Guest-VLAN even before sending the EAPOL-Start frame. In this case, the IP address must be released and renewed after the machine authentication process completes successfully, because the port can now be deployed in a different VLAN from the Guest-VLAN. In this case, things can break if the supplicant running on the machine is not able to trigger this DHCP renewal. The machine would not be able to get an IP address in the correct subnet.

Tests run with various supplicants showed that all of them are able to renew the IP address after the machine authentication process completes. However, this happens by default with CSSC but not with the Microsoft client. Windows XP requires the registry settings described previously or the machine does not send the EAPOL-Start and therefore is stuck in the Guest-VLAN. The same situation occurs when the user logs in through the Graphical Identification and Authentication (GINA) interface (which serves as the gateway for interactive logons).

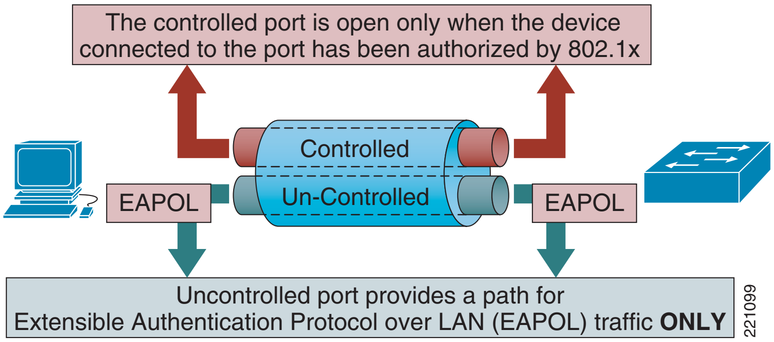
- The 802.1X supplicant is not enabled for machine authentication. In this case, during the startup process the switch port could be deployed into the Guest-VLAN and the machine could get an IP address from the Guest-VLAN pool. This happens not only when setting the *max-auth-req* and *tx-period* parameters at the minimum possible values, but also every time the startup process is longer than (max-reauth-req + 1) * tx-period seconds. A typical Guest-VLAN security policy limits communications to internal resources; this can break the startup process in all the scenarios where Windows clients need to participate in a Windows Active Directory (AD) networking environment. Even assuming the connectivity with an AD domain controller is not required, after the user logs in and successfully authenticates, there is the same need to renew the IP address as described previously. Validation tests run with various supplicants show that, by default, the CSSC clients are able to renew their IP address, whereas the Microsoft supplicant requires the registry setting to be modified to initiate the EAP authentication process after the user logs in.

In conclusion, it is possible to set the *tx-period* and *max-reauth-req* parameters to the minimum configurable values to reduce the time interval required for the deployment of a switch port in the Guest-VLAN. To avoid breaking the 802.1X functionality when using the Microsoft XP supplicant, Cisco recommends that you modify the default Windows registry values to allow the Microsoft supplicant to send EAPOL-Start frames. This is the default behavior for CSSC clients.

Wake-on-LAN Primer

Wake-On-LAN (WoL) is an industry standard, which is the result of the Intel-IBM Advanced Manageability Alliance. WoL creates a power management wake-up event. This is an advanced power management capability on many network interface cards (NICs) in the industry today. NICs that support WoL have an extra connector and cable to connect to the motherboard. After a machine goes into suspend mode, it can be automatically reactivated when data from the network is received by the NIC. This capability can be used to wake up a mail server machine to deliver mail, for software management pushes, to deploy patches overnight, and so on. By default, 802.1X and WoL are mutually exclusive, because of the architecture of 802.1X, as shown in Figure 16.

Figure 16 Standard 802.1X Operation


The controlled port is open only when the device connected to the port has been authorized by 802.1x
Controlled
Un-Controlled
EAPOL
EAPOL
Uncontrolled port provides a path for Extensible Authentication Protocol over LAN (EAPOL) traffic ONLY
221099

As indicated above, a switch exerts control over a virtual port in both directions. This is known as a bi-directional controlled port. This means only EAPOL should come into or go out of the switch port until authenticated. However, the operational direction of the controlled port can be changed per section

6.4(b) of the IEEE spec for 802.1X. Thus, in an effort to interoperate with WoL environments, most Cisco Catalyst switches provide unidirectional controlled port functionality as an optional configuration. WoL is a per-port feature. Operationally, the controlled port should then only operate in one direction. A WoL "magic packet" can now exit the network to wake a machine up if necessary. The machine still must 802.1X authenticate before successfully send traffic into the network. This corresponding configuration is as follows:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 dot1x pae authenticator
 dot1x port-control auto
 dot1x control-direction in
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port dot1x 2/2 port-control-direction in
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

The configuration above represents a weaker deployment of the technology because it allows outgoing traffic on a port before being secured by 802.1X, while still dropping all the incoming traffic on a port that has not yet authenticated. However, a subtle change is that spanning tree is now placed in a forwarding state for any ports that are not yet authorized.

Note A best practice is to enable WoL functionality along with 802.1X only on the ports where it is needed.

Minimum releases for the support of this per-port functionality on Catalyst switches are as follows:

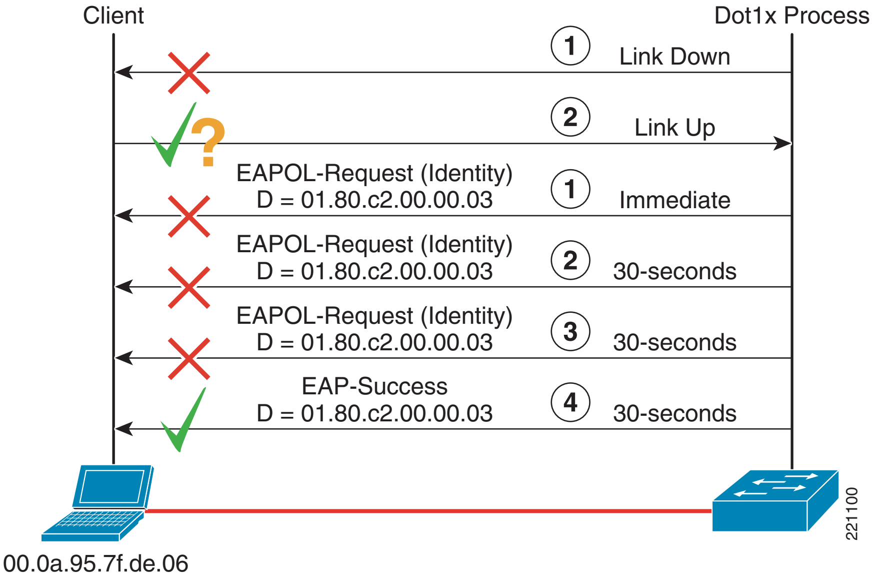
- Catalyst 6500—CatOS 8.3(1)
- Catalyst 4500—12.2(31)SG
- Catalyst 3750-2970—12.2(25)SEC
- Catalyst 2960—12.2(25)FX
- Catalyst 2940-2950—12.1(22)EA5

A recommended best practice for any deployment of 802.1X, MAB, the Guest-VLAN, and WoL are to plan ahead of time. Test how specific Network Driver Interface Specification (NDIS) functionalities or configurations residing on end devices should impact link change.

Guest-VLAN and WoL Interaction

A switch port is down conditionally after a link-down event is processed by an authenticator as a machine goes to sleep. The link should then come back up on the port immediately. The link-up event is then processed on the port as well. If the Guest-VLAN is configured, a port is enabled into the Guest-VLAN soon after the original "go to sleep" event. This process is shown in Figure 17.

Figure 17 Machine Going Into Power Save Mode with the Guest-VLAN



As shown above, a machine that goes into power save mode with the Guest-VLAN also enabled bounces link state, and then is deployed into the Guest-VLAN. There may be differences between "hibernate" and "standby" settings on end stations, so specific functionality must be examined in detail to evaluate the impact 802.1X may have on the environment. In addition, it is critical to understand whether an EAPOL-Logoff is, or needs to be, sent by an 802.1X supplicant on the specific implementation when going to sleep.

The operational behavior above exists on ports with the following configurations:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 dot1x pae authenticator
 dot1x port-control auto
 dot1x guest-vlan 22
 dot1x control-direction in
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
id1-6503-1> (enable) set port dot1x 2/2 guest-vlan 605
Port Control Direction set to IN on 2/2. Guest-VLAN can not be enabled.

id1-6503-1> (enable) set port dot1x 2/2 port-control-direction in
Port 2/2 is guest-vlan enabled, Port Control Direction can not be set to IN.
```

Note The CatOS configuration example above represents an attempt to enable the Guest-VLAN on a port already enabled with WoL functionality (or vice versa). For CatOS switches, this configuration combination cannot be achieved.

For IOS-based switches, any combined deployment of WoL and the Guest-VLAN renders WoL specific functionality needless. Operationally, if the port is enabled into the Guest-VLAN already, a specific port configuration for the allowance of a unidirectional controlled port itself is not needed.

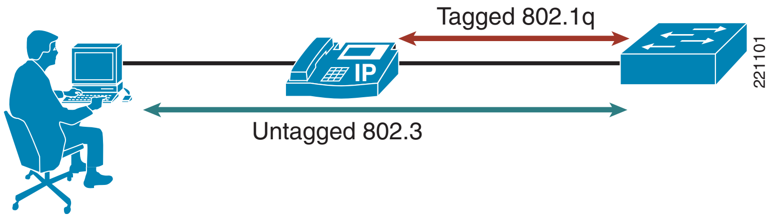
A best practice for a combined environment is to support WoL functionality from the Guest-VLAN itself, and not to bother configuring the unidirectional controlled port functionality. However, if the Guest-VLAN is being used as a means to provide third-party unauthenticated access, any WoL servers elsewhere in the network must be able to reach the Guest-VLAN. If the Guest-VLAN is configured as a separate VLAN from the access VLAN already configured on the port, this means that the WoL implementations may need to be re-configured to deploy 802.1X to reach the subnets associated to the Guest-VLAN deployed at the edge.

Network virtualization may also impact the operation of WoL functionality for PCs. For example, if the Guest-VLAN on a switch serves as entrance criteria to a separate VRF or VPN for guest access, this guest partition is not able to reach the rest of the enterprise network by design. However, this may conflict the requirement of a WoL server to be able to reach the Guest-VLAN, because it is safe to assume that the WoL server is privately owned and operated by enterprise IT staff. Thus, this needs to be planned for as part of the services edge design of a network virtualization architecture. See the *Services Edge Design Guide* for more details.

Interaction with VoIP Deployments

The integration of 802.1X and IP phones is based on the switch configuration of multi-VLAN access ports. Multi-VLAN ports belong to two VLANs: native VLAN (PVID) and auxiliary VLAN (VVID). This allows the separation of voice and data traffic and enables 802.1X authentication only on the PVID. Figure 18 shows the type of communication that occurs on these two VLANs.

Figure 18 Multi-VLAN Port



When 802.1X is enabled on a multi-VLAN access port, a supplicant must complete the authentication process before getting access to the data (native/PVID) VLAN. The IP phone can get access to the voice (auxiliary/VVID) VLAN after sending the appropriate Cisco Discovery Protocol (CDP) packets, regardless of the dot1x state of the port. The use of CDP with Cisco IP phones is typically required, given the lack of pervasive support for an embedded 802.1X supplicant. However, newer IP phones models have attained supplicant capability. In addition, multi-domain authentication (MDA) on switches provides a means to authenticate a phone and data client on the wire via 802.1X and/or MAB.

Note MDA and 802.1X supplicant capability on IP phones are not within the scope of this document.

The configuration commands for Cisco IOS and CatOS that are required to enable multi-VLAN functionality, in conjunction with 802.1X and Guest-VLAN functions, are as follows:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 switchport voice vlan 2
 dot1x pae authenticator
 dot1x port-control auto
 dot1x guest-vlan 10
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port dot1x 2/1 guest-vlan 10
set port auxiliaryvlan 2/1 2
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

The multi-VLAN and the Guest-VLAN features are not currently supported on Catalyst 6500 platforms running native Cisco IOS images. The Catalyst 6500 requires CatOS or Hybrid (CatOS and Cisco IOS). These features are supported on all other Catalyst platforms, beginning with the following minimum software releases:

- Catalyst 2950—12.1(12c)EA1
- Catalyst 3560—12.1(19)EA1
- Catalyst 3750—12.1(14)EA1
- Catalyst 4500—12.2(20)EWA
- Catalyst 6500—7.6(1)

This section describes the circumstances in which the Ethernet port on the IP phone can be shared between users equipped with 802.1X supplicants and users that do not have supplicants.

When accessing the network by connecting to the port on an IP phone, no link-down event occurs on the switch port when the PC is later removed. Therefore, the switch is unaware of this event, which poses security vulnerabilities.

Note The same considerations are valid in cases where hubs are deployed between the end users and the access layer switch. Cisco does not officially support the deployment of hubs in conjunction with 802.1X, so the topic here is limited to the use of IP phones.

To determine the conditions under which the same IP phone PC port can be sequentially used by various categories of users (users equipped with an 802.1X supplicant and those who are not), two main aspects must be considered:

- The capability of the IP phone to send an EAPOL-Logoff message on behalf of the client after it detects a link-down event on the PC port (this functionality is called proxy EAPOL-Logoff).

  Note Proxy EAPOL-Logoff was introduced in the Cisco 7940 and 7960 phones in firmware 7.2(2) and the Cisco 7911, 7941, 7961, 7970, and 7971 phones in firmware 7.0(1). Both images were released in June 2005. More information can be found at the following URL: http://www.cisco.com/en/US/partner/products/hw/phones/ps379/prod_release_note09186a0080461f84.html

- The mode of operation of the 802.1X process on Catalyst switch ports after receiving the EAPOL-Logoff message. This differs depending on the switch platform and on the specific software image. Based on this, it is possible to distinguish the two scenarios discussed in the next subsections.

Scenario 1

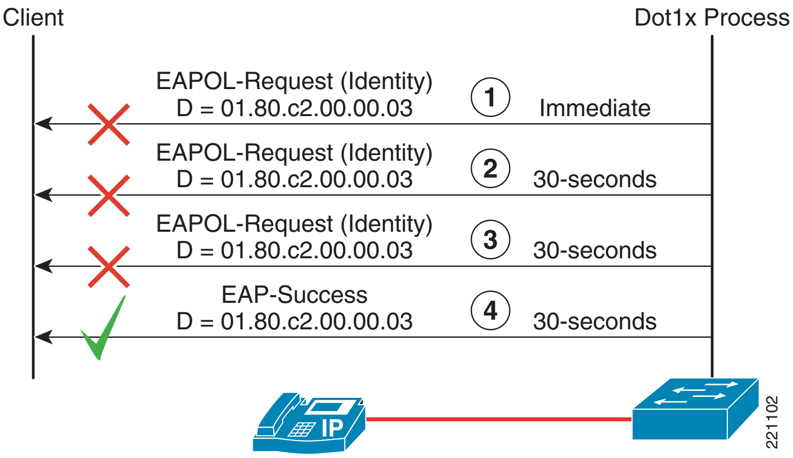
In describing the interaction of the Guest-VLAN with an IP phone, it is essential to distinguish between the two cases where the Cisco IP Phone supports EAPOL-Logoff.

IP Phone That Supports EAPOL-Logoff

When an IP Phone is plugged into a switch port configured as shown in the preceding configuration example, the port is assigned to the Guest-VLAN (see Figure 19).

Note that the Cisco IP phone is not able to reply to the EAP messages originated from the switch given the lack of 802.1X supplicant.

Figure 19 IP Phone and the Guest-VLAN



The 802.1X state machine running on the switch port has now entered into the authenticated state. At this point, what happens depends on who is connecting to the Ethernet port on the IP phone. The port remains configured in the Guest-VLAN if one of the following two situations occurs:

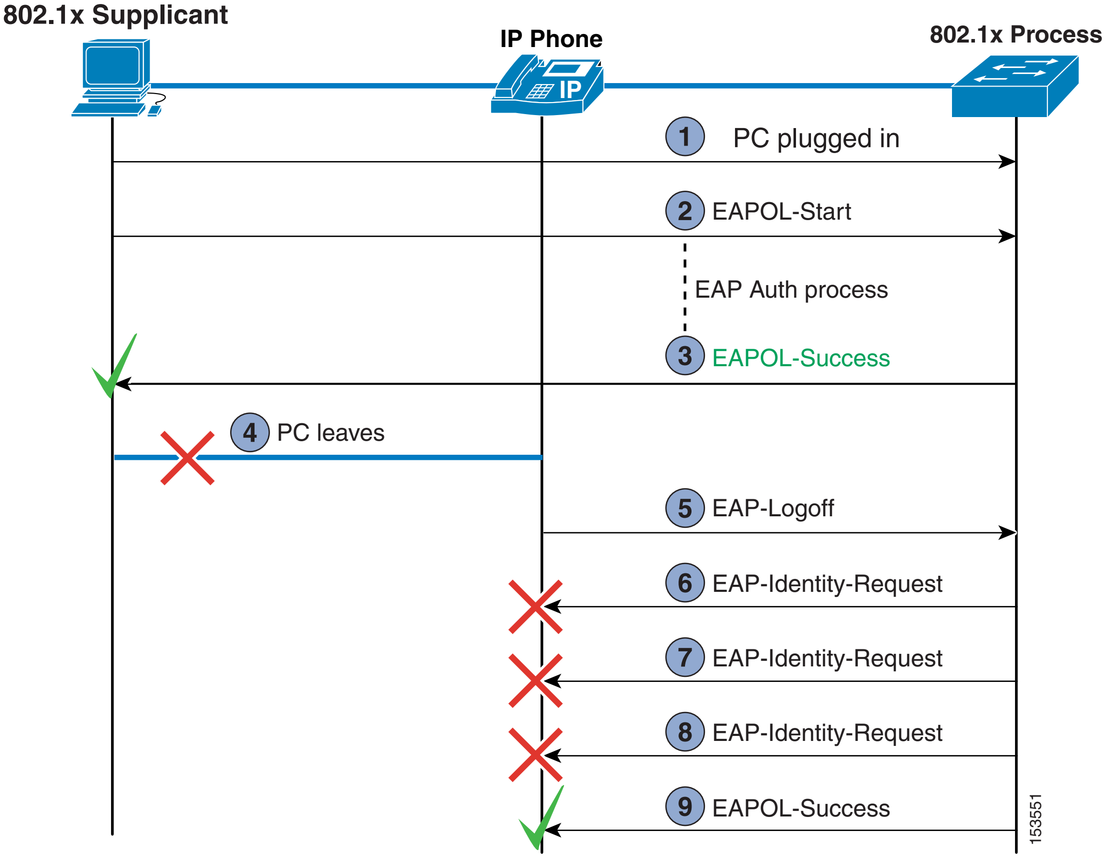
- A guest connects using a machine that does not have an 802.1X supplicant.
- An internal employee (or even a guest) connects using a machine that is equipped with an 802.1X supplicant that is not able to send EAPOL-Start (remember that by default Microsoft Windows XP supplicant does not send an EAPOL-Start). In this specific case, the user might wonder why the 802.1X authentication completes successfully when connecting to a switch port, but does not when plugging into the Ethernet port on the IP phone.

In either of these two situations, the 802.1X state machine remains in the authenticated state (it basically stops running); this is true even when the user disconnects from the IP phone.

A different situation occurs if an employee connects to a machine that is equipped with a supplicant that is able to send EAPOL-Start. The reception of this packet restarts the 802.1X state machine on the switch port, and assuming that the authentication is successful, the port is deployed in the proper data VLAN.

When the employee disconnects, the IP phone sends an EAPOL-Logoff message to the switch to inform the 802.1X process of this event. As a consequence, the 802.1X state machine on the switch port enters the unauthorized state and 802.1X stays active. After sending a few EAPOL-Identity-Request frames (this number is dictated by the value of the *max-reauth-req* parameter) looking for a supplicant, the switch port is finally deployed into the Guest-VLAN. This sequence of events is shown in Figure 20.

Figure 20 EAPOL-Logoff Capability with Older Software Releases



In summary, this case represents the best scenario because it allows deploying a user into the Guest-VLAN or into a data VLAN, depending on the capability of the user client to initiate an 802.1X conversation with the switch after connecting to the IP phone. In addition, because of the EAPOL-Logoff capability of the IP phone, the switch port is always deployed into the Guest-VLAN when a previously connected employee disconnects from the IP phone, making the port reusable by a guest.

IP Phone That Does Not Support EAPOL-Logoff

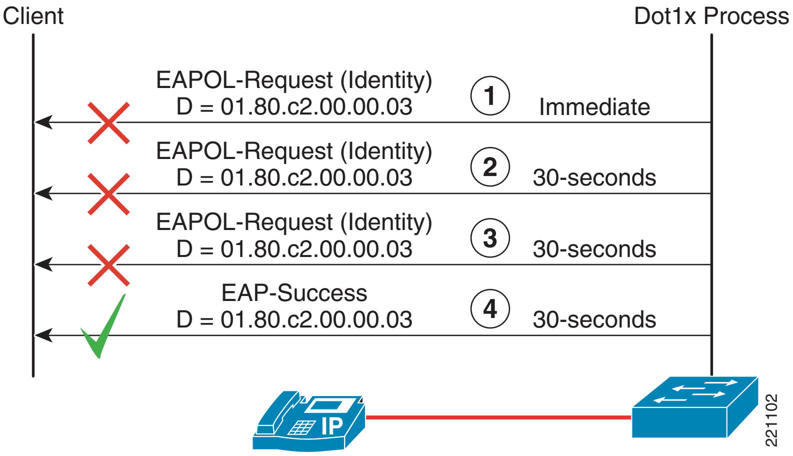
This situation is identical to what was previously described until Step 4 of Figure 20. After this point, because the IP phone is not capable of sending an EAPOL-Logoff frame to the switch, the port remains in the 802.1X authenticated state. This clearly opens a security hole, because an illegitimate user can now gain access to the port by spoofing the authenticated MAC address, and completely bypass the 802.1X authentication phase. The situation grows worse because the process breaks even if a legitimate user connects to the IP phone when the switch port is in the authenticated state. In fact, assuming that the MAC address of the new user is different from the one of the user previously authenticated, the switch could treat the event as a security violation. As a consequence, the switch port can be placed in an error disable state, in which case not only is the new user prevented from connecting, but the IP phone is disconnected. The only scenario where the switch port might still be deployed into the Guest-VLAN is when 802.1X re-authentication is enabled on that port, and it initializes before a user connects to the IP phone. This is recommended as a workaround condition only, and is fundamentally not a reason to enable re-authentication alone.

Scenario 2

IP Phone That Supports EAPOL-Logoff

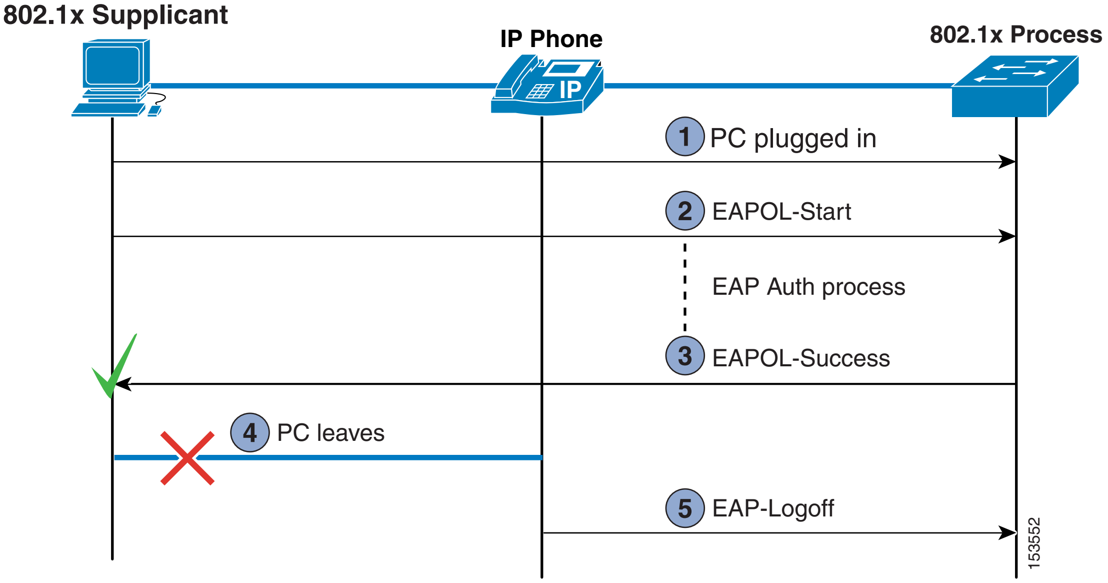
When an IP phone is connected to a switch port and configured as shown in the previous configuration example, the port is deployed into the Guest-VLAN. (See Figure 21.)

Figure 21 IP Phone and the Guest-VLAN



This situation is identical to that previously described at the first step of Scenario 1. However, a different situation occurs if, starting from the initial situation shown in that scenario, an employee connects with a machine that is equipped with a supplicant capable to send EAPOL-Starts. The reception of this packet restarts the 802.1X state machine on the switch port, and assuming that the authentication is successful, the port is deployed into the appropriate data VLAN. If the employee disconnects, the IP phone sends an EAPOL-Logoff message to the switch to inform the 802.1X process of this event. As a consequence, the 802.1X state machine on the switch port enters the unauthorized state and stops running. This sequence of events is shown in Figure 22.

Figure 22 EAPOL-Logoff Capability



Guests that attempt to connect a cable to the IP phone who are not equipped with a supplicant are not deployed into the Guest-VLAN because the state machine is stopped in the unauthorized state. The only event that unblocks this situation is when a machine with a supplicant that is able to send an EAPOL-Start message connects to the IP phone.

In summary, the IP phone capability of sending EAPOL-Logoff closes the security hole described in the following section, but at the same time jeopardizes the deployment of the switch port into the Guest-VLAN.

As previously mentioned, the description of events in Scenario 2 is valid for Catalyst switches running newer versions of Cisco IOS code. To maintain backward compatibility, a new global command, **dot1x guest-vlan supplicant,** was added in Cisco IOS software releases that allows you to manually configure the switches to behave as described in Scenario 1. This global command is available on Cisco IOS for all the Catalyst platforms, but it is not available on CatOS for Catalyst 6500, so these devices can only perform as described in this scenario when running software releases earlier than 8.4(4). In addition, the **dot1x guest-vlan supplicant** command has become hidden starting from the releases 12.2(31)SG for Catalyst 4500 and 12.2(25)SEE for Catalyst 3750. As of 12.2(35)SE, this command is still functional, but still hidden as well.

IP Phone That Does Not Support EAPOL-Logoff

This situation is identical to that shown in Steps 1 through 4 of the previous scenario. After this point, because the IP phone described here is not capable of sending an EAPOL-Logoff frame to the switch, the port remains in the 802.1X authenticated state. This clearly opens a security hole, because an illegitimate user can now gain access to the port by spoofing the authenticated MAC address, and completely bypassing the 802.1X authentication phase. Even worse, the process breaks even if a legitimate user connects to the IP phone when the switch port is in the authenticated state. In fact, assuming that the MAC address of the new user is different from the one of the user previously authenticated, the switch can treat the event as a security violation. As a consequence, the switch port can be placed in an error disable state, which not only prevents the new user from connecting, but also causes the IP phone to be disconnected. This differs from Scenario 1 in that, in this case, the switch port cannot be deployed into the Guest-VLAN even if re-authentication is enabled on that port and it initializes before a user connects to the IP phone.

Guest-VLAN Summary

In conclusion, when describing the integration of the 802.1X Guest-VLAN with IP telephony, the following are the recommended design choices:

- Use IP phones running a firmware version that enables them to send the EAPOL-Logoff frame when an authenticated user disconnects from the Ethernet port on the IP phone.
- Implement the Guest-VLAN functionality on Catalyst switches running either older IOS software versions or ones configured with the **dot1x guest-vlan supplicant** command. An improvement is to provide this command on a switch port level rather than globally. This allows individual ports to be configured for the operation and enhances operational flexibility by enabling this feature only on switch ports that are accessible from enterprise public areas (where guest access must be provided). It is not used in the other enterprise locations (a guest should not be able to achieve network connectivity when connecting, for example, to an IP phone that is located in an employee office). When using Catalyst 6500 platforms, Cisco recommends using the minimum software release 8.4(4).

MAC Authentication Primer

MAC address authentication itself is not a new idea. One classic type of this is port security. Another type is the Cisco VLAN Management Policy Server (VMPS) architecture. With VMPS, a customer can have a text file of MAC addresses and the VLANs to which they belong. That file gets loaded into the VMPS server switch via TFTP. All other switches then check with the VMPS server switch to see to which VLAN those MAC addresses belong after being learned by an access switch. Customers can also define actions for the switch to take if the MAC address is not in the MAC address text file. No other security is enforced.

Similar to VMPS, another type is the User-Registration Tool (URT), which uses the VLAN Query Protocol (VQP) and acts like a VMPS. Wireless also has a version of this support available on most access points (APs) and/or controllers. This base functionality for MAC address checking is already in place. For example, wireless access points can initiate a PAP authentication with a RADIUS server using a client MAC address as a username/password. APs can accomplish this based on the fact that initial associations have already been made (and based on that association traffic to/from a wireless NIC is blocked by the AP). No such association exists currently in the wired space. MAB as described in this document represents an attempt to make a wired equivalent of this functionality that is integrated with 802.1X. Similar to the operation examined here, MAB in the wireless space has its own similar security concerns, most notably the granting of network access on a MAC address. This is potentially a security risk for more enterprises, especially for wireless, because of the nature of the authentication method used. MAC addresses can be easily mirrored or spoofed.

With wireless, a MAC address check can even be done before 802.1X, so if a MAC address authentication fails, the user can still get on the network if they then pass 802.1X authentication. Cisco Clean Access (CCA) also provides a way to authorize users based on a MAC address. MAB makes an effort to leverage similar efforts that are already applied to other authentication schemes or mechanisms (802.1X/EAP). This should make deployments easier for customers to deploy and understand. MAB also represents a consolidation of current efforts toward identity, authentication, and security. These are some of the reasons why MAB is suited for network virtualization.

Other reasons to support MAB for access control are as follows:

- To provide a supplemental authentication technique using the EAP standard.
- To provide a supplemental authentication technique to be unified with 802.1X.
- Address the “all or nothing” specter of 802.1X.

- 802.1X + Guest-VLAN alone was not designed for what customers need here.
- There will always be wired devices that do not support 802.1X.
- To provide a migration path from port security.
- To provide a migration from URT and/or VMPS.

MAB, as described in this document, is intended to provide this controlled access to such devices based on their MAC address. MAB should allow non-802.1X compliant end devices to be governed by controlled access to the network in a transparent manner using a pre-populated database technique. The requirement for enabling access for clients that do not support 802.1X supplicant functionality is also applicable to the Network Admission Control (NAC) program, where a need exists to enable network access for all clients who may subsequently carry out a posture assessment. It is critical to network virtualization for MAB to leverage dynamic policy assignment as well. An overview of MAB is shown in Figure 23.

Figure 23 MAB Overview

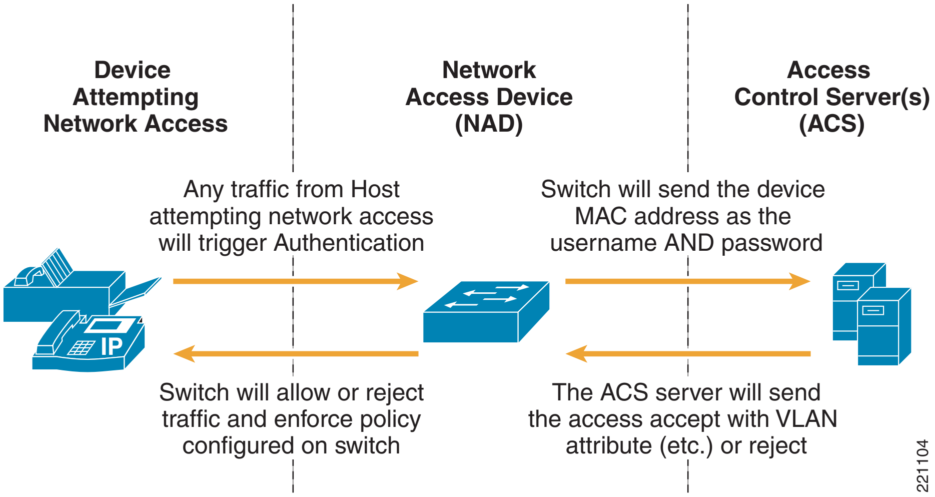

Device Attempting Network Access
Network Access Device (NAD)
Access Control Server(s) (ACS)
Any traffic from Host attempting network access will trigger Authentication
Switch will send the device MAC address as the username AND password
IP
Switch will allow or reject traffic and enforce policy configured on switch
The ACS server will send the access accept with VLAN attribute (etc.) or reject
221104

Figure 23 shows a device generating traffic (any traffic, such as DHCP, ARP, and so on), the switch captures the MAC address and forwards this as the username and password to ACS. MAB allows end users to authenticate (without any supplied credentials). As is discussed subsequently in this document, MAB is not intended to directly provide a MAC address learning mechanism. It is to be provided solely as a means of authentication and enforcement. Although MAB requires some form of a provisioning process, the described functionality is independent of any existing processes. This process alone assumes MAC addresses are already known. MAB should then allow clients that cannot/do not support 802.1X the functionality necessary to integrate into the current access control strategy for network virtualization. Like 802.1X, MAB is designed for the access layer and is supported on the following Cisco Catalyst switches referenced with minimum Cisco CatOS or IOS revisions:

- Catalyst 6500—CatOS 8.5(1)
- Cisco Catalyst 4500/4948—12.2(31)SG
- Cisco Catalyst 3750–2960—12.2(25)SEE
- Cisco Catalyst 2940—12.1(22)EA9

Note Wireless LAN functionality is not examined further in the clientless context, primarily because of the nature of pervasive client capability in the overall wireless space. Because of the nature of the security threat model with the wireless media, MAC authentication is no longer recommended. There may, however, remain some cases to deal with this for wireless, such as Symbol handhelds, which may only support Wired Equivalent Privacy (WEP). For more details about wireless and MAC authentication capabilities, see the *LEAP/MAC Authentication Configuration Guide* at the following URL: http://www.cisco.com/warp/customer/707/leap-mac-auth.html

Note Branch router functionality is not examined further in the clientless context, primarily because of verification and testing resources. Cisco has traditionally provided 802.1X and its set of L3 authorization features on L3 ports popularly referred to as the spouse & kids (S&K) solution. S&K consists of 802.1X authentication, host-mode support (that is, single-host, multi-host, and multi-auth), Cisco IP phone support, guest or authentication failed handling using split-tunneling, and an implicit default behavior of MAB. This behavior is different from the behavior on Catalyst switches examined in this document. On branch routers, locally configured black and while lists based on MAC addresses can be configured as well. For more information, see *Configuring Cisco IOS Easy VPN Remote with 802.1X Authentication* at the following URL;
http://www.cisco.com/en/US/tech/tk583/tk372/technologies_white_paper09186a00801fdef9.shtml#wp1002262

MAB is designed to address the market need for network edge authentication similar in nature and benefits to the functionality provided by the IEEE 802.1X framework, without the requirement for client-side code. It is intended to address a replacement technology for URT/VMPS environments. The target solution space is campus and enterprise switching. The goal of this feature set is to enhance the position of Cisco as a leader in that space by providing increased security and semi-automatic provisioning via the authentication of connected network clients.

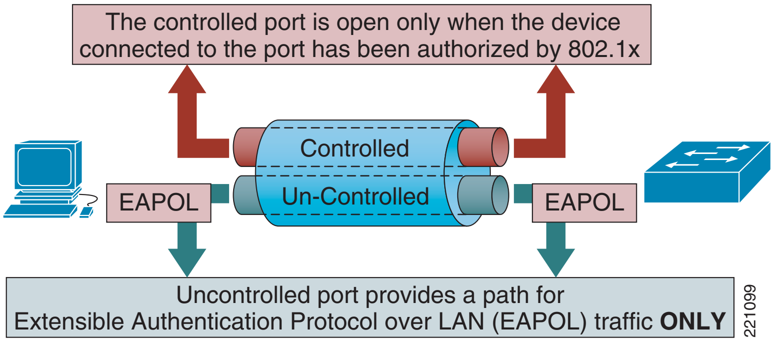
MAC Authentication Bypass Operational Overview

The aspects of MAB operation need to be carefully considered. Before examining MAB, a rehearsal of the operation of 802.1X-enabled ports is provided for context.

802.1X Rehearsal

When 802.1X is enabled on a port, the MAC address of a machine is typically unknown until the port is authorized (or at the very least, until a supplicant sends EAPOL frames). This is because of the default operation of 802.1X, as shown in Figure 24.

Figure 24 Standard 802.1X Operation

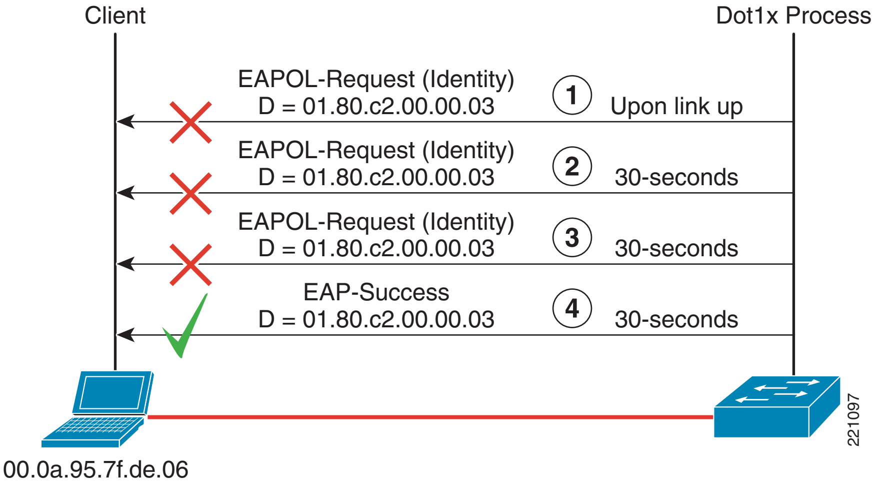

The controlled port is open only when the device connected to the port has been authorized by 802.1x
Controlled
Un-Controlled
EAPOL
EAPOL
Uncontrolled port provides a path for Extensible Authentication Protocol over LAN (EAPOL) traffic **ONLY**
221099

In Figure 24, only EAPOL traffic is typically processed by a switch port while 802.1X is maintained in an operating and active state. Thus, any MAC address from any edge device may not be known until EAPOL frames are processed from it. These are security benefits of 802.1X, and do not change in any way with respect to any MAB implementation. Because it is noteworthy to this discussion, spanning tree is not even in a forwarding state on the port until it is authorized via 802.1X.

Guest-VLAN Rehearsal

Before MAB, the Guest-VLAN was the only alternative to provide network access to clients that do not speak EAPOL. This process is shown in Figure 25.

Figure 25 Guest-VLAN Feature



There is no differentiation capability for the Guest-VLAN. If the client on the wire cannot speak 802.1X, the Guest-VLAN is enabled. Any device deployed into a Guest-VLAN may be a machine on the network that an administrator does not need or want to be placed in a Guest-VLAN. Thus, the ability to employ differentiated services based on the MAC address alone is advantageous for identification purposes. Upstream, the Guest-VLAN may also have access only to limited resources, as defined by the network administrator. Before MAB, a MAC address can be known to a switch port only after the port is enabled and placed into a Guest-VLAN. In addition, after a port is enabled and placed into a Guest-VLAN, no authentication (other than EAPOL initiation by a supplicant) takes place on the port directly, and the system can learn any number of MAC addresses on the port by default (which inherently does not provide security). Thus, there are limitations to using the Guest-VLAN concept as a solution to provide access for any non-802.1X-enabled devices in the context of network virtualization that can be addressed through MAB functionality.

Therefore, what is needed is a way to update a switch CAM table with a (single) MAC address, while not circumventing the value added from a port-based 802.1X solution to begin with.

MAB Operation

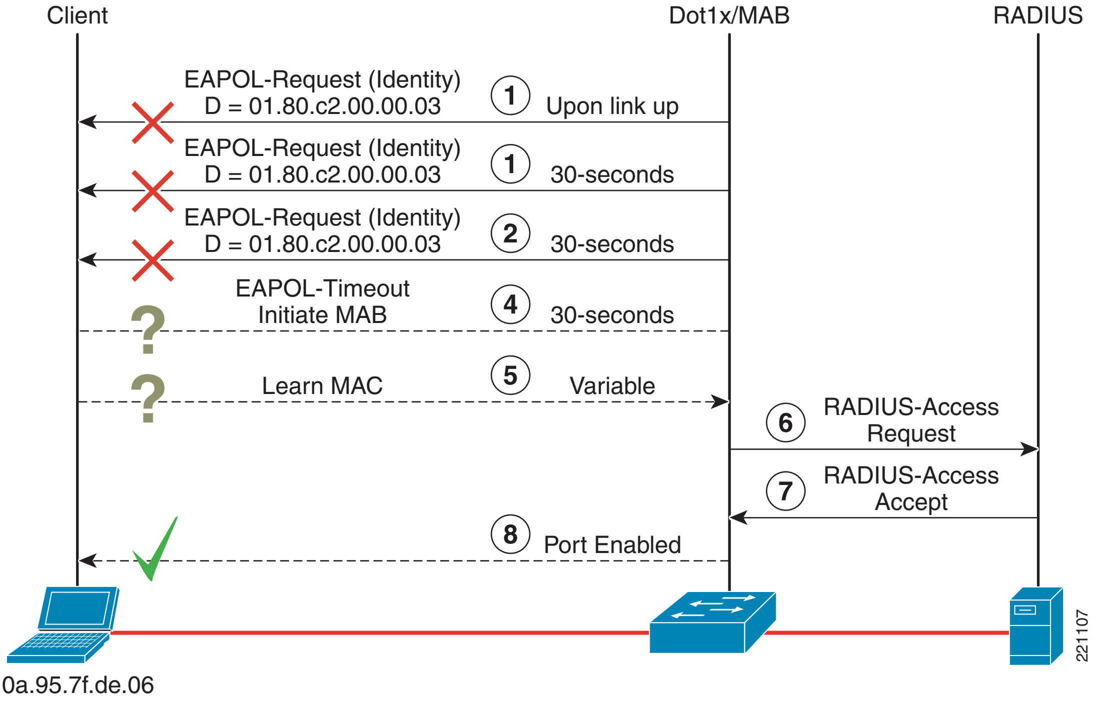
Much like the Guest-VLAN, MAB operates based on an 802.1X timeout condition. After a switch port can ascertain that an 802.1X supplicant is not present on the port, it falls back to checking the MAC address (which is an authentication technique of lesser security). After timing out 802.1X on the port, a MAC address can be learned by the switch through classic MAC learning techniques. after a MAC address is learned by the switch, it can then be authenticated by RADIUS initiation. The RADIUS call transmits the attributes shown in Table 1 as part of a RADIUS request to AAA.

Table 1 RADIUS Attributes

No.	Attribute Name	Request	Accept	Reject	Description
1	User-Name	1	0	0	MAC address sent in “hhhhhhhhhhhh” format, all lowercase with no meta characters or white spaces.
2	Password	1	0	0	Same as User-Name, but encrypted per PAP or MD5.
4	NAS-IP-Address	1	0	0	IP address of switch.
5	NAS-Port	1	0	0	Physical port of device acting as the authenticator.
6	Service-Type	1	0	0	Indicates framing to be used for framed access. This attribute indicates the type of service a user has requested, or the type of service to be provided. It *may* be used in both RADIUS-Access-Request and RADIUS-Access-Accept packets. It has been used on switches in the past to enable RADIUS exec authorization and to launch a user into enable mode. Currently set as Call-Check “10” in Access-Requests, and tracked by ACS in RADIUS Accounting logs.
12	Framed-MTU	1	0	0	Indicates MTU to be used by the user. Set to “1500”.
30	Called-Station-ID	1	0	0	MAC address of device acting as authenticator, as seen by the peer.
31	Calling-Station-ID	1	0	0	MAC address of client.
61	NAS-Port-Type	1	0	0	Indicates type of physical port on the authenticator. Set to “15” for Ethernet.
80	Message Authenticator	1	1	0	HMAC-MD5 to ensure integrity of packet.

A complete operational flow of MAB is shown in Figure 26.

Figure 26 MAB Operation



MAB initiates only after an 802.1X timeout. MAB then requires a variable amount of time for the end station to attempt to send traffic into the network for the MAC to be learned by the switch. After this occurs, RADIUS is initiated to the backend asking whether the MAC should be allowed network access.

As shown in Figure 26, after a host/device fails to supply 802.1X authentication credentials, the network access device takes the learned MAC address and hands it off to the authentication server as both the username and password. If the host/device fails to authenticate at this level, a user can optionally be placed into a pre-determined Guest-VLAN. Alternatively, the Guest-VLAN can be used as a means to support a provisioning process of the MAC address through scanning techniques, or captive portal techniques if end users are applicable to the devices seeking to be authenticated. One example of this is discussed subsequently in this document.

Ultimately, if the host/device passes with MAB credentials, the user can then be placed into the configured VLAN and can acquire an IP address to begin its desired functions. Optionally, dynamic policy can be downloaded from RADIUS the same way this is achieved with 802.1X in the form of VLAN assignment. This allows for consistent processing of authentication features to be applied in a consistent manner. Similarly, if MAB fails, the process continues indefinitely as it does with 802.1X. However, if the Guest-VLAN is also deployed, this serves as the direct failure criteria for MAB. This supports backward compatibility for existing techniques in place to provide network access to the Guest-VLAN solely in the absence of 802.1X.

Dynamic policy downloaded from an authentication server includes any capability currently available with 802.1X on the access switch in question, such as per-user ACLs, VLAN assignment, and so on. In addition, the validity of the authorized session is enforced on the switch much the same way it is enforced with 802.1X. This enforcement is achieved by restricting the traffic originating on the authenticated port to come from only the MAC address that was authorized. With MAB, only one host can be authenticated and locked down per port by default. Any new MAC address that is seen to attempt to pass traffic on a port is treated as a security violation.

Functional Details

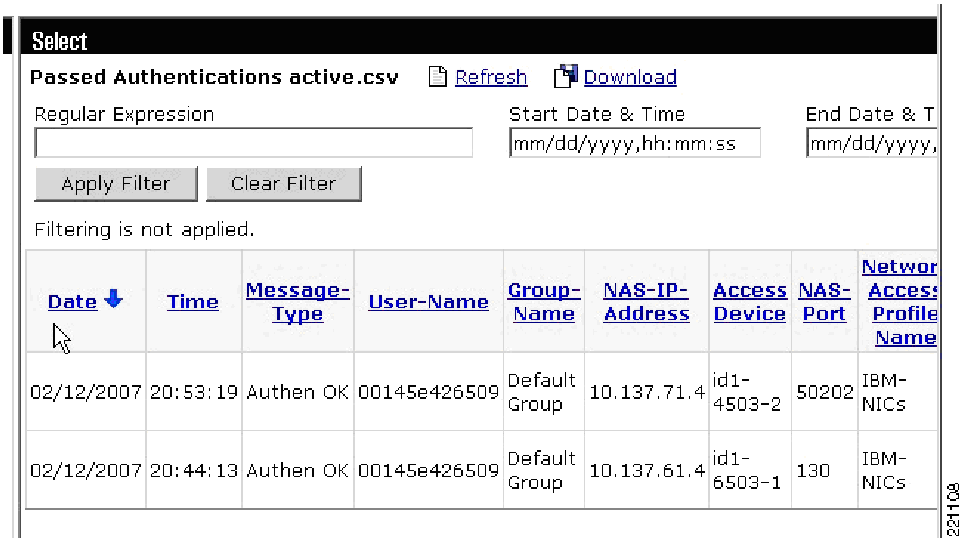
As indicated previously, it is important to understand the format of the MAC address sent in any MAB request when MAB is used by the authentication infrastructure. Any RADIUS requests transmitted by MAB with Cisco Catalyst switches contains the following two RADIUS attributes:

- Attribute [30] (Called-Station-ID)—MAC address of the ingress interfaces of the switch or authenticator
- Attribute [31] (Calling-Station-ID)—MAC address of the 802.1X supplicant or the end-station

Both these attributes are sent in the format of “XX-XX-XX-XX-XX-XX” for all switches. This has recently been updated in switch code base to ensure both compatibility with legacy switch code and also compliance with RFC 3580. 802.1X requests operate the same way. Neither of these attributes, however, is necessarily expected to actually provide the authentication service provided by MAB, as discussed previously. Authentication and authorization are provided from RADIUS Attribute [1] (username) and RADIUS Attribute [2] (password). For IOS-based switches, and recent versions of CatOS, the format for the user-name and password attributes is simply “hhhhhhhhhhhh”; that is, an all lower-case version of “hhhh.hhhh.hhhh” with the punctuation stripped out. Therefore, if an identity infrastructure is to be built to support MAB, it should follow this format.

Figure 27 shows passed authentications on ACS from an IOS-based switch and a CatOS switch running MAB.

Figure 27 MAB from IOS 12.2(31)SG and CatOS 8.5(5)



Passed Authentications active.csv

Regular Expression

Start Date & Time: mm/dd/yyyy,hh:mm:ss

End Date & T: mm/dd/yyyy,

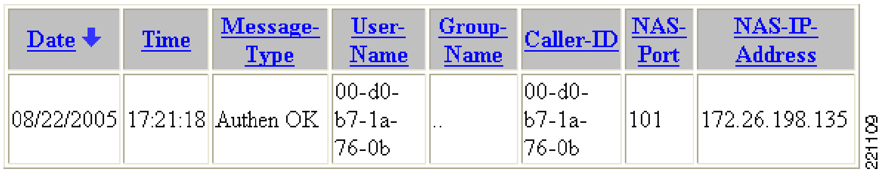
Filtering is not applied.

Date	Time	Message-Type	User-Name	Group-Name	NAS-IP-Address	Access Device	NAS-Port	Networ Access Profile Name
02/12/2007	20:53:19	Authen OK	00145e426509	Default Group	10.137.71.4	id1-4503-2	50202	IBM-NICs
02/12/2007	20:44:13	Authen OK	00145e426509	Default Group	10.137.61.4	id1-6503-1	130	IBM-NICs

Note Although not examined here, WLCs use the username attribute in the same manner reflected above for IOS-based switches.

However, before 8.5(5), CatOS did *not* follow this practice. CatOS transmitted the MAC address information to ACS using an “hh-hh-hh-h-hh-hh” format. ACS could not handle this if the user account is defined like the above for IOS. Figure 28 represents a passed authentication on ACS from CatOS with 8.5(4) demonstrating this condition.

Figure 28 MAB from CatOS 8.5(4) and Before



Date	Time	Message-Type	User-Name	Group-Name	Caller-ID	NAS-Port	NAS-IP-Address
08/22/2005	17:21:18	Authen OK	00-d0-b7-1a-76-0b	..	00-d0-b7-1a-76-0b	101	172.26.198.135

As a result, if the same ACS server is used to authenticate MAB from these various types of switches, the MAC addresses need to be entered into a database twice. This is completely unmanageable, and a recommendation for 8.5(5) is a must in heterogeneous environments. This benefit is now fully realized because a single MAC need only be defined in a single location, while multiple authenticators can use it in the same format.

Timers are important to remember for MAB as well. For IOS-based switches, the standard timers for 802.1X are the same timers for MAB. For example, the timers to decrease the amount of time it takes to enable a port into the Guest-VLAN are *tx-period* and *max-reauth-req*. By default, it should take 90 seconds to enable a port in the Guest-VLAN. In the MAB case, this same timing is used as a signal to the switch platform that it should now open the port to learn the MAC address of an end station to begin the MAB authentication process. More timing details are discussed later in this document.

Re-authentication for MAB is supported the same way 802.1X supports it for IOS-based switches. Any re-authentication configuration that may currently exist on a switch impacts MAB clients. By default, however, if MAB re-authentication is enabled with a specific session-timeout through a port configuration, 802.1X needs to timeout again. After 802.1X times out, MAB then simply uses the MAC address the switch currently thinks is on the wire in its cache as a means to check whether the backend policy may have been disabled or changed for that MAC address. During this period, network access is persistent by default, however. Like 802.1X, MAB also incorporates the support of RADIUS attributes [27] and [29] as well. They can be set to have the switch deny access during the re-authentication event, or for the switch to re-learn a MAC instead of just using the one in the cache.

From a state machine perspective, the 802.1X state machine for an IOS-based switch also goes to an authenticated state after MAB successfully authorizes as MAC address and is updated accordingly.

By default, MAB also operates in single-auth mode like 802.1X. This means that only one MAC is allowed on the port to authenticate, and that any other MAC that appears on a port may be treated as a security violation. In addition, the host mode configured for 802.1X itself also impacts MAB. In other words, if 802.1X is configured to operate in multi-host mode, this allows any number of machines on the port subsequent to the port being authenticated. This is true for MAB as well via the same configuration.

MAC Authentication Bypass Configuration and Verification

Configuration

MAB is a port-based feature and is required to be enabled on ports discretely. The following represents specific port configurations with MAB added.

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 dot1x mac-auth-bypass
 dot1x pae authenticator
 dot1x port-control auto
 spanning-tree portfast
```

```
     spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port mac-auth-bypass 2/1 enable
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

Note The **dot1x pae authenticator** command above is not prevalent or applicable to the 2940. Where it can be applied, this command is used to enable 802.1X and for the specific type of operation under which the port should operate. the **dot1x port-control auto** command is now used as a means to configure the operating mode of 802.1X itself, assuming it has been enabled to begin with.

This is the only additional configuration required on a switch beyond an existing 802.1X configuration that may have already been deployed.

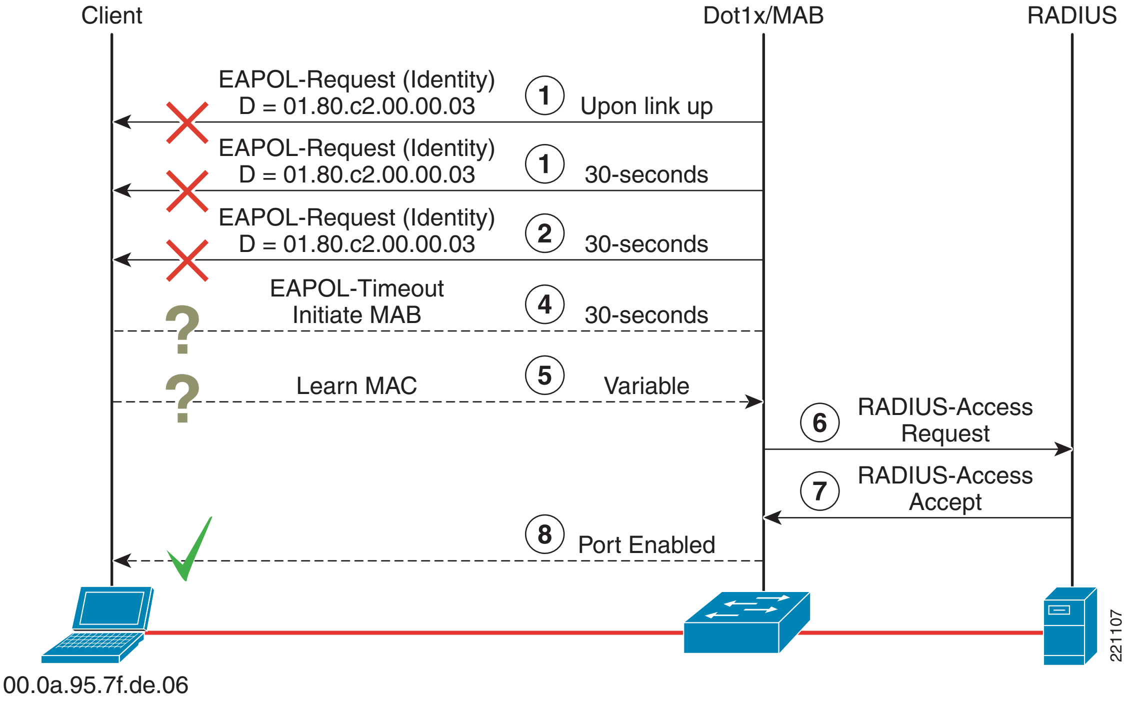
802.1X Timeout

Any port enabled for MAB at present must also timeout on 802.1X before execution of MAB can begin. The default operation of MAB clearly disrupts the boot-up process of standard PCs in an Active Directory environment because of the network delays that are imposed on the port when a machine first boots. By default, this time is over 90 seconds before a machine can begin forwarding traffic on the network successfully. MAB also cannot stand alone today as the only type of authentication configured on a port, so an 802.1X timeout must be employed for any IOS-based switched. For CatOS, however, MAB can be the only type of access control configured on a port, and is an optional configuration.

A best practice recommendation in this regard for an enterprise is to attempt to use MAB for corner cases only, and to allow 802.1X to handle the majority of controlled LAN access.

However, MAB is an ideal option for clients insensitive to delays upon boot-up or login, such as printers. An alternative to the timeout imposed by 802.1X is to reduce the timeout period. As discussed previously, the same timers and values to enable a port into the Guest-VLAN can be used for MAB to reduce the artificial delay imposed by 802.1X, and have MAB execute in a quicker manner if needed. The overall timeout process and MAB is rehearsed in Figure 29.

Figure 29 MAB Operation



The *max-reauth-req* parameter sets the maximum number of times that the switch retransmits an EAP-Identity-Request frame on the wire before receiving a response from the connected client. This value is set to two by default. This is why MAB shows two retries (at Steps 2 and 3) after the initial EAP-Identity-Request frame sent at link-up. The commands used to change this parameter (in CatOS and IOS) are as follows:

- CatOS

```
cat6500> (enable) set dot1x max-reauth-req ?
<max-reauth-req> maximum number of retries to supplicant (1..10)
```

- Cisco IOS

```
cat3750(config-if)#dot1x max-reauth-req ?
<1-10>  Enter a value between 1 and 10
```

The *tx-period* parameter sets the number of seconds that the switch waits for a response to an EAP-Identity-Request frame from the client before retransmitting the request. The responsibility of retransmitting the request unmodified when a response is accepted lies solely with an authentication within the confines of 802.1X. The default value for the tx-period is 30 seconds and is configurable as follows:

- CatOS

```
cat6503> (enable) set dot1x tx-period ?
<tx-period> tx period (1..65535 seconds)
```

- Cisco IOS

```
cat3750(config-if)#dot1x timeout tx-period ?
      <1-65535>  Enter value between 1 and 65535
```

The *max-req* parameter is also part of the configurable 802.1X parameter in Cisco IOS. The *max-req* parameter is different from the *max-reauth-req* parameter. The *max-req* parameter represents the maximum number of retries a switch performs for EAP-Request frames of types other than EAP-Identity-Request. Basically, this parameter refers to EAP-Data frames, which are the EAP frames exchanged after the supplicant has replied to the initial EAP-Identity-Request frame. For this reason, the *max-req* parameter is effective only when there is a valid 802.1X supplicant connected, and it does not apply to any method to deal with the timeout of 802.1X itself on the port.

The configurable values for the parameters shown in the preceding configuration example are consistent between the various Catalyst switch platforms, when running the following minimum Cisco IOS software releases (previous releases are characterized by platform-specific configurable values):

- Catalyst 3750—12.2(25)SEE
- Catalyst 4500—12.2(31)SG

For a Catalyst 6500 running CatOS software, the situation is different. In CatOS releases earlier than 8.5, there is no *max-reauth-req* parameter. This implies that the same parameter described above (*max-req*) is used to tune both the number of retries for the EAP-Identity-Request and EAP-Data frames. Note also that the configurable values are consistent with the one detailed for Cisco IOS: *max-reauth-req* (and *max-req*) can vary from 1 to 10, and *tx-period* from 1 to 65535.

The overall configuration of MAB is relatively simple but differs on switches running IOS and CatOS software releases. Complete configurations with tweaked timeouts are as follows:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 dot1x mac-auth-bypass
 dot1x pae authenticator
 dot1x port-control auto
 dot1x timeout tx-period 1
 dot1x max-reauth-req 1
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port mac-auth-bypass 2/1 enable
set dot1x max-reauth-req 1
set dot1x tx-period 1
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

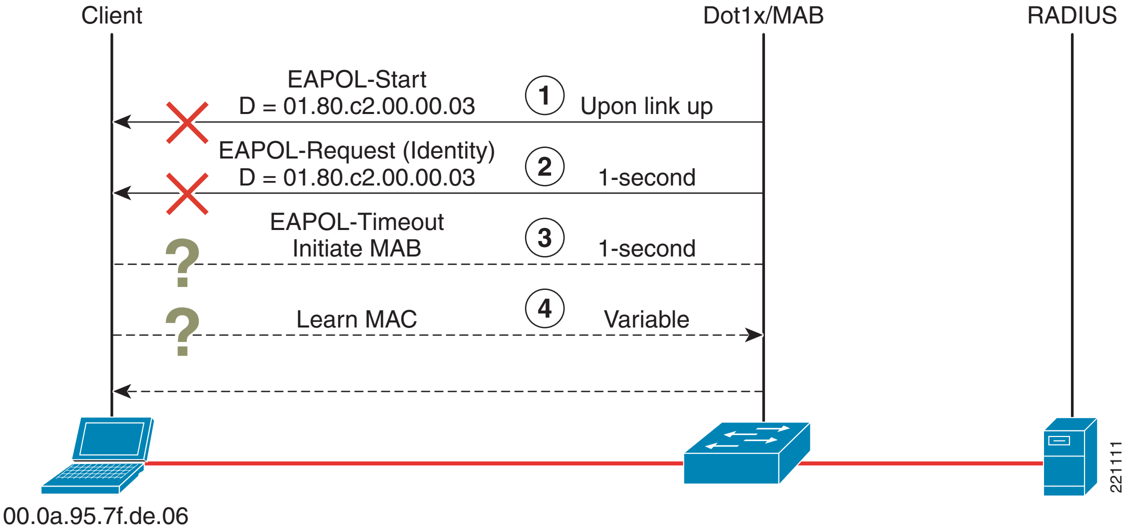
Note In CatOS systems, the values for *max-req* and *tx-period* are set at a global level, and not per port, as they are in Cisco IOS software.

As shown above, the timeout for 802.1X and MAB initiation can be configured as low as 2 seconds. The following formula calculates the time interval before MAB initiates:

[(max-reauth-req + 1) * tx-period]

As stated previously, MAB initiates only at this time. The end station must then attempt to send traffic into the network, so the specific time to ultimately authenticate the end device typically varies. The operation of tweaked timers to timeout 802.1X quickly as indicated above is shown in Figure 30.

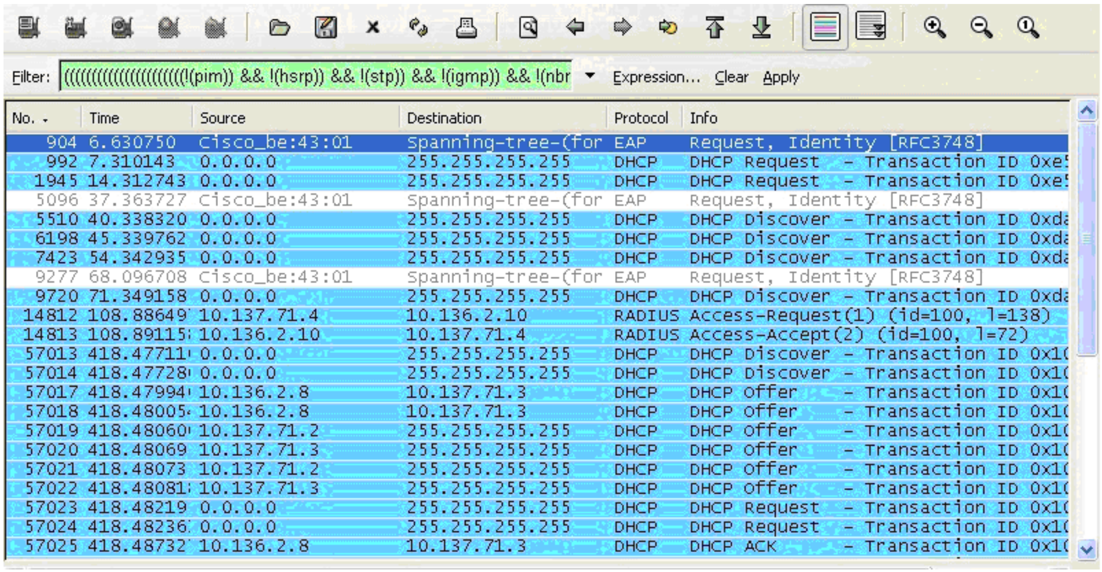
Figure 30 MAB Initiation with Tweaked Timers



This configuration should be attempted only after considering the consequences that this can have on the regular functionality of 802.1X.

Analyzing the integration issues between 802.1X and DHCP at startup time helps in understanding this. MAB was tested with default timers and Windows machines. As indicated, this causes DHCP to timeout entirely upon boot-up and any link-up condition. This process is shown in Figure 31.

Figure 31 MAB Impact on DHCP



Filter: ((((((((((((((((((((!(pim)) && !(hsrp)) && !(stp)) && !(igmp)) && !(nbr

```
No.    Time       Source           Destination          Protocol Info
  904  6.630750   Cisco_be:43:01   Spanning-tree-(for   EAP      Request, Identity [RFC3748]
  992  7.310143   0.0.0.0          255.255.255.255      DHCP     DHCP Request  - Transaction ID 0xe
 1945  14.312743  0.0.0.0          255.255.255.255      DHCP     DHCP Request  - Transaction ID 0xe
 5096  37.363727  Cisco_be:43:01   Spanning-tree-(for   EAP      Request, Identity [RFC3748]
 5510  40.338320  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0xda
 6198  45.339762  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0xda
 7423  54.342935  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0xda
 9277  68.096708  Cisco_be:43:01   Spanning-tree-(for   EAP      Request, Identity [RFC3748]
 9720  71.349158  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0xda
14812  108.88649  10.137.71.4      10.136.2.10          RADIUS   Access-Request(1) (id=100, l=138)
14813  108.89115  10.136.2.10      10.137.71.4          RADIUS   Access-Accept(2) (id=100, l=72)
57013  418.47711  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0x1
57014  418.47728  0.0.0.0          255.255.255.255      DHCP     DHCP Discover - Transaction ID 0x1
57017  418.47994  10.136.2.8       10.137.71.3          DHCP     DHCP Offer    - Transaction ID 0x1
57018  418.48005  10.136.2.8       10.137.71.3          DHCP     DHCP Offer    - Transaction ID 0x1
57019  418.48060  10.137.71.2      255.255.255.255      DHCP     DHCP Offer    - Transaction ID 0x1
57020  418.48069  10.137.71.3      255.255.255.255      DHCP     DHCP Offer    - Transaction ID 0x1
57021  418.48073  10.137.71.2      255.255.255.255      DHCP     DHCP Offer    - Transaction ID 0x1
57022  418.48081  10.137.71.3      255.255.255.255      DHCP     DHCP Offer    - Transaction ID 0x1
57023  418.48219  0.0.0.0          255.255.255.255      DHCP     DHCP Request  - Transaction ID 0x1
57024  418.48236  0.0.0.0          255.255.255.255      DHCP     DHCP Request  - Transaction ID 0x1
57025  418.48732  10.136.2.8       10.137.71.3          DHCP     DHCP ACK      - Transaction ID 0x1
```

As shown in the example above, 802.1X times and MAB is successful ~90 seconds after a link-up event. However, DHCP times out completely after approximately one minute. When Windows reverts to the internal address of 169.254.x.x, however, an IGMP report from this address causes L2 traffic to be learned by the switch, so that MAB can initiate. Therefore, although variable, MAB completes less than 40 seconds after DHCP has timed out in this example. Windows reverts to standard timeout procedures in this case, and does not attempt to renew its address for another five minutes. This is probably unacceptable to any end user experience, so timer tweaking may be needed here to enable this process to operate better for machines sensitive to this timeout condition.

Proceed with caution for tweaking timers, however. If timers are tweaked too low, MAB (or the Guest-VLAN if configured) may execute on the device before 802.1X when the end station may be legitimately configured for 802.1X. An example of this is when a Windows machine boots. 802.1X may not execute on the machine two seconds after the machine starts trying to send traffic. Most 802.1X supplicants are applications themselves, so they also need time to load. This may be an undesired side effect, although nothing may be technically wrong about this operating condition. Security policies may need to dictate this as well. As a result, there may be no optimum for timer recommendations to make in this regard, because mileage varies based on requirements.

Verification

Following is an example of MAB working on a port of a CatOS switch:

```
id1-6503-1> (enable) sho port mac-auth-bypass 2/2
Port  Mac-Auth-Bypass State MAC Address       Auth-State         Vlan
----- --------------------- ----------------- -----------------  -----
 2/2  Enabled               00-14-5e-42-65-09 authenticated      601

Port  Termination action Session Timeout Shutdown/Time-Left
----- ------------------ --------------- ------------------
 2/2  initialize         3600            NO       -

Port  PolicyGroups
----- -----------------------------------------------------
 2/2   -
```

Following is an example of MAB working on a port of an IOS-based switch:

```
id1-3750-2#sho dot1x interface g1/0/2 details
Dot1x Info for GigabitEthernet1/0/2
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0
Mac-Auth-Bypass           = Enabled

Dot1x Authenticator Client List
-------------------------------
Supplicant                = 0014.5e42.671b
        Auth SM State     = AUTHENTICATED
        Auth BEND SM Stat = IDLE
Port Status               = AUTHORIZED
Authentication Method     = MAB
Authorized By             = Authentication Server
Vlan Policy               = N/A
```

The verified result from the 2940 implementation differs from the output of the IOS example above:

```
id1-2940-1#sho dot1x interface f0/2
Supplicant MAC 0014.5e42.6523
AuthSM State          = AUTHENTICATED
BendSM State          = IDLE
Posture               = N/A
PortStatus            = AUTHORIZED
MaxReq                = 2
MaxAuthReq            = 2
HostMode              = Single
Port Control          = Auto
ControlDirection      = Both
QuietPeriod           = 60 Seconds
Re-authentication     = Disabled
ReAuthPeriod          = 3600 Seconds
ServerTimeout         = 30 Seconds
SuppTimeout           = 30 Seconds
TxPeriod              = 30 Seconds
Guest-Vlan            = 0
AuthFail-Vlan         = 0
AuthFail-Max-Attempts = 3
Mac-Auth-Bypass       = Enabled
```

MAC Authentication Bypass Feature Interaction

MAB and EAPOL Interaction

As shown above, MAB activates when 802.1X times out waiting for an EAPOL packet on the wire. The 802.1X state machine enters a waiting state and relinquishes control over to MAB to begin device authorization upon this timeout occurring. MAB runs passively and does not transmit any packets to detect devices. Again, the responsibility lies with the attached device to send traffic. If a device sends no traffic, then technically, a port could be listening for packets forever after MAB activates. Packets arriving on a port where MAB is active results in the switch forwarding the packets to the CPU. The source MAC address is gleaned off the packet and forwarded to the MAB process for authorization. The "trigger" packet itself is typically dropped. Before MAB activates, if an EAPOL packet is detected on the wire (such as an EAPOL-Start from an 802.1X supplicant), 802.1X never relinquishes control to MAB. The history of EAPOL packets seen on the wire is maintained as long as the port is physically connected. This "history" is lost upon physical link change.

When MAB activates, a port is typically in an unauthorized state (because 802.1X times out). Therefore, while waiting for a packet to glean a MAC address, if an EAPOL packet is detected, MAB de-activates and relinquishes complete control back to 802.1X entirely. 802.1X then attempts to authenticate the port. From then on, MAB never activates as long as the link is never lost on the port.

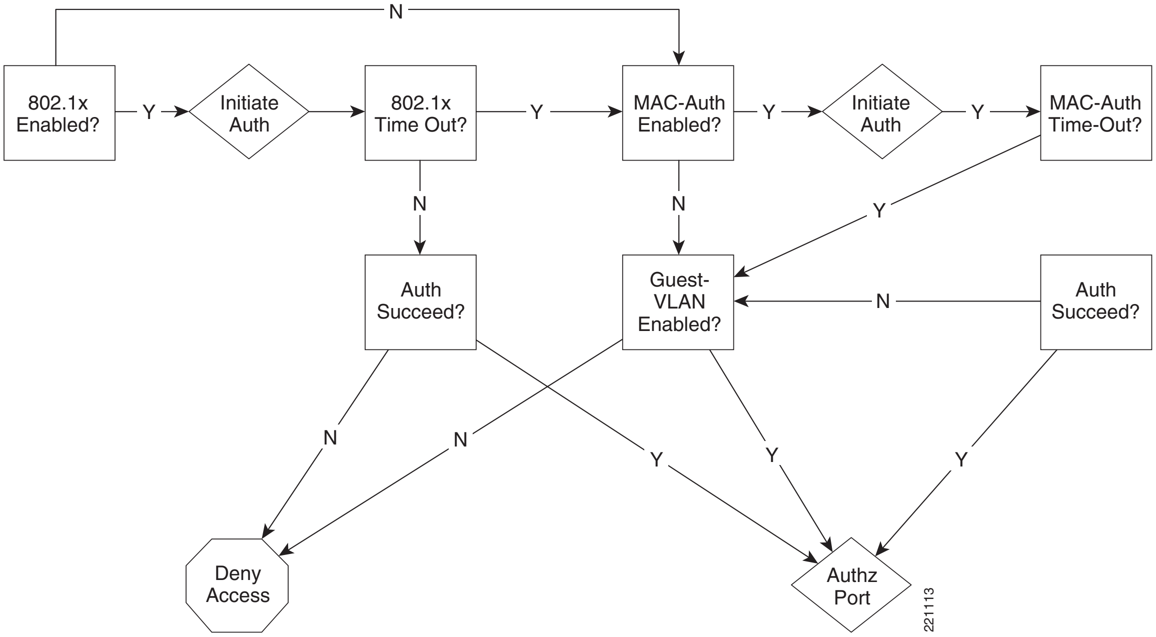
In some cases, MAB may have authorized a port already, and 802.1X is then seen on the wire. An example of this can be a successful MAB attempt before 802.1X has started on the client, or MAB being executed in an effort to assist the end station in downloading 802.1X supplicant software. Typically in this condition, the MAC addresses from both events match. However, if a port is authorized with MAC address A, and an EAPOL packet arrives with a source MAC address of B, this triggers a security violation by the switch.

On IOS-based switches, MAB cannot currently be enabled without 802.1X. This also impacts any potential re-authentication scenario. Technically, re-authentication for MAB does not exist on IOS. Therefore, if re-authentication is enabled (for 802.1X) when a switch port is authenticated via MAB, a switch sends out EAP requests upon re-authentication timer expiry, and 802.1X ultimately relinquishes control back to MAB for authorization only if no response is received. Because 802.1X has to timeout again for MAB re-authentication to occur, MAB re-authentication is not recommended. In addition, any port configuration involving 802.1X re-authentication is also not recommended. For any 802.1X re-authentication use case that may involve MAB, it is recommended that a RADIUS-supplied session-timeout be used to control the behavior for 802.1X devices only, and not for devices that have been authenticated via MAB.

MAB and the Guest-VLAN

The Guest-VLAN serves as a failure condition for MAB if configured on the same port as MAB. Otherwise, the failure process for MAB is to continually try to 802.1X authenticate the port again. For IOS-based switches today, this is primarily because of a MAB failure actually causing the port to go into the HELD state, much the same it would as if an 802.1X supplicant had failed authentication. Therefore, after the HELD state completes, 802.1X is attempted again, times out again, and MAB is attempted again. However, because the Guest-VLAN can serve as the failure criteria for MAB if configured along with MAB, this might provide a systemic value; for example, MAB and the Guest-VLAN could indirectly provide a means to provision credentials in an identity store for MAC addresses that may not be known in advance to the enterprise, as shown in Figure 32.

Figure 32 802.1X, MAB and the Guest-VLAN



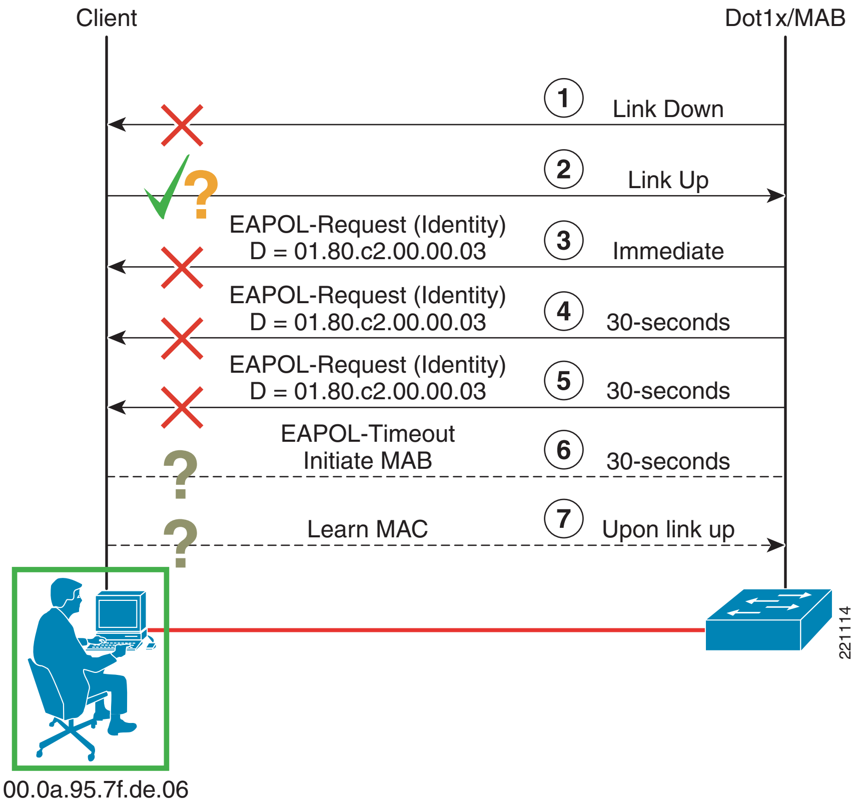
The operational nature of the feature interaction in Figure 32 was designed primarily as part of MAB to support backward compatibility for devices that cannot speak 802.1X and have already deployed the Guest-VLAN.

Note If a port is initially configured for 802.1X with Guest-VLAN, and the port activates in Guest-VLAN, it remains there even though a network administrator enables MAB. The port link status must be flapped to initialize the 802.1X state machine.

MAB and WoL Interaction

If MAB is configured, a port is nailed up into a MAB state of “initiated” soon after the original “go to sleep” event. This process is shown in Figure 33.

Figure 33 Machine Going Into Power Save Mode with MAB



As shown in Figure 33, a machine that goes into power save mode with MAB also enabled bounces link state, and then is nailed up into a state of MAB needing to learn a MAC address to be able to authenticate it. There may be differences between “hibernate” and “standby” settings on end stations, so specific functionality must be examined in detail to evaluate the impact that 802.1X may have on the environment. Also critical to understand is whether an EAPOL-Logoff is, or needs to be, sent by an 802.1X supplicant on the specific implementation when going to sleep.

The operational behavior above exists on ports with the following configurations:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 dot1x mac-auth-bypass
 dot1x pae authenticator
 dot1x port-control auto
 dot1x control-direction in
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port mac-auth-bypass 2/1 enable
set port dot1x 2/2 port-control-direction in
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

For Catalyst switches, any combined deployment of WoL functionality and MAB does not impact the fundamental need to wake up machines from remote management locations. Operationally, when a machine wakes up, it must MAC authenticate because 802.1X has already timed out (while the machine was asleep). However, as discussed above for EAPOL and MAB interaction, a machine may also 802.1X authenticate when it wakes, which tears down all session state for the MAB context and 802.1X access is granted.

A best practice for a combined environment is to support WoL functionality from the statically configured access VLAN, the same way a customer would before 802.1X has been deployed.

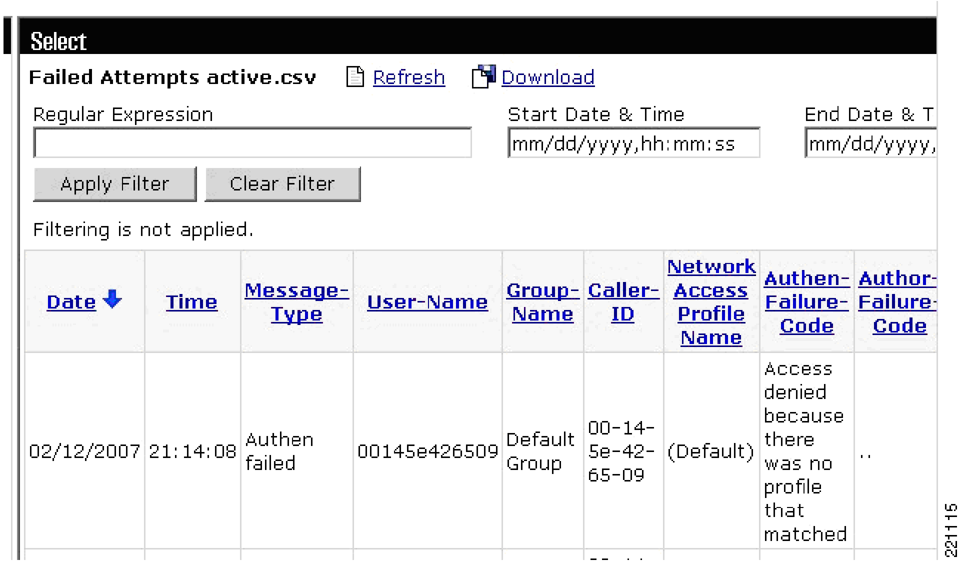
Network virtualization may impact the operation of WoL functionality for PCs, as for example when the access VLAN on a switch serves as entrance criteria to a VRF or VPN separate from the global table, where a WoL server is typically deployed. In that case, you need to be sure that this partition can be reached from the segment on which the WoL server is located. Therefore, this does necessarily need to be planned for as part of the services edge design of a network virtualization architecture. This topology should allow WoL to work, while retaining most forms of separate network partitioning, after devices have been authorized into the networked system.

MAC Authentication Bypass Opportunities and Benefits

Location-Based Awareness

MAB can do a good job of providing MAC-based security, where only known MAC addresses are allowed access to the network, using a central RADIUS server (or identity store) to store the list of MAC addresses. This takes the burden of managing the MAC addresses off of any local switch, and is technically superior to port security in this respect. In support of network virtualization, VLANs can be assigned for granular policy as well. These benefits represent motivations behind the need for MAB. However, there is currently no easy way to have switches authenticate the device and at the same time limit the MAC to a specific location/switch. Although this functionality is not currently provided by any turnkey solution, similar capabilities exist in dial-up or WLAN models. A complete location-based system is not yet integrated into 802.1X or MAB itself for authorization purposes. However, some customer problems based on location can be solved. For example, if a customer has a device that should only be on the machine floor of a production plant (for example, robotic arm device), the authentication system may need to know that this device should only be connected to a single switch. This way, if the device shows up on another switch or location, the authentication system can realize this event and deny the authentication attempt on this basis. One way to technically achieve this is to configure ACS for Network Device Groups (NDG). Then, as part of a Network Access Profile (NAP), Network Access Filter (NAF) can be set up based on the NDG. This can cause a MAB request to not match the NAP, because the request may originate from the wrong switch, as shown in Figure 34.

Figure 34 Deny MAB Request Based on Pre-determined Location



Select

Failed Attempts active.csv Refresh Download

Regular Expression

Start Date & Time

mm/dd/yyyy,hh:mm:ss

End Date & T

mm/dd/yyyy,

Apply Filter Clear Filter

Filtering is not applied.

Date	Time	Message-Type	User-Name	Group-Name	Caller-ID	Network Access Profile Name	Authen-Failure-Code	Author-Failure-Code
02/12/2007	21:14:08	Authen failed	00145e426509	Default Group	00-14-5e-42-65-09	(Default)	Access denied because there was no profile that matched	..

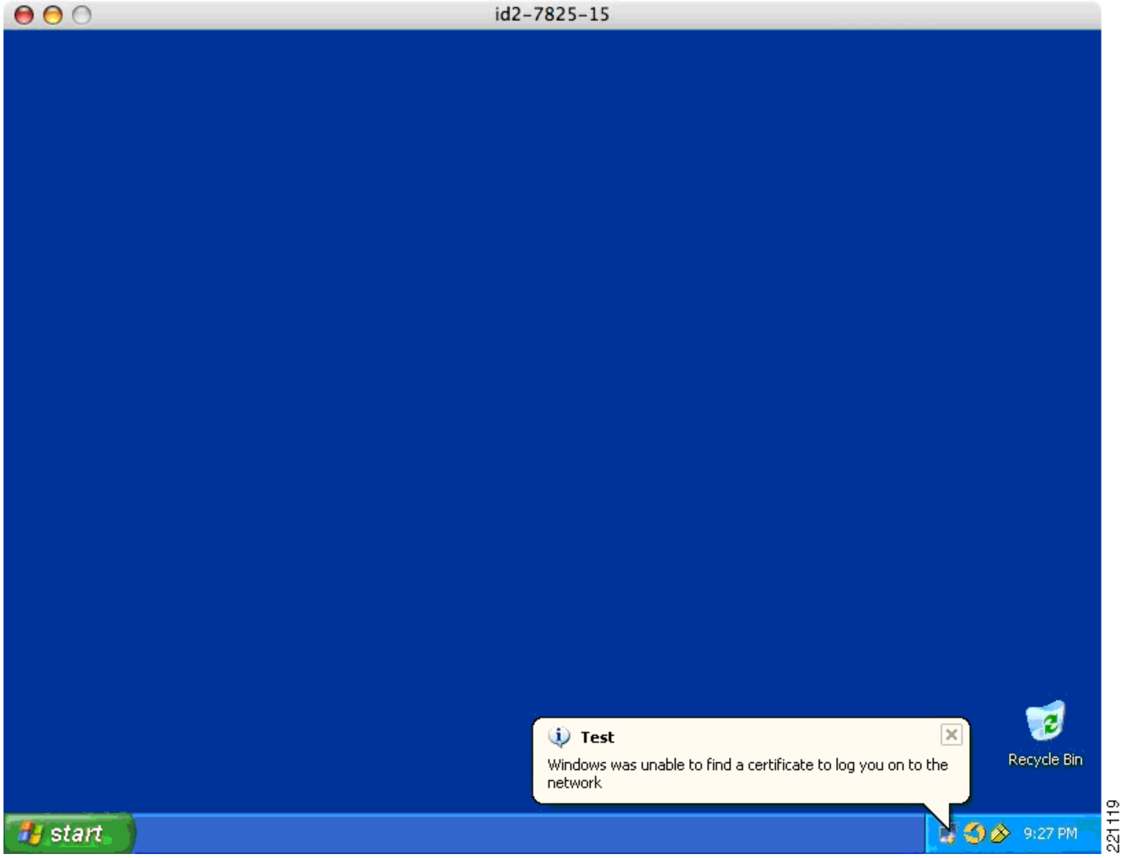
For more information on Network Access Profiles, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/csacs4nt/acs40/user/sp.htm

The above example may not suffice for general use cases. It is not intended to function as a true location-aware based service. However, because location-aware services are not yet prevalent in wired network authentication topologies, this can supplement the service for precedent. In summary, this can be technically achieved as shown above because of a specialized need.

Fallback Technique for New/Re-imaged Machines with WZCSVC

There are systemic challenges associated with MAB and the fallback nature of this supplemental technique in the absence of 802.1X. The first challenge is from Windows XP. A new or re-imaged PC is typically enabled for 802.1X by default. In addition, if the machine is running a default image for Windows XP, the 802.1X supplicant does not send EAPOL-Start frames even though 802.1X is enabled. This means that when the link comes up, the switch begins an 802.1X authentication event by transmitting an EAPOL-Identity-Request packets on the wire. However, although the PC is 802.1X-enabled, the supplicant is also enabled for EAP-TLS and the machine “knows” it does not have a certificate for either the machine or any user that happens to be logged into it. Operationally, a balloon message appears in the system tray at this point with “Windows was not able to find a certificate to log you onto the network”. Because a certificate is not on the device at all, Windows does not speak EAPOL to the switch. In addition, because the supplicant never sent the switch an EAPOL-Start, the switch has no way of knowing the device is actually 802.1X capable. Therefore, this means that a brand-new machine can be initially deployed into the Guest-VLAN, or if the MAC address is known before the connection event, MAB can be used as a means to help deploy 802.1X, or at least provide network access to the device in a fallback method even though 802.1X is technically enabled on the client. An example of this is on the end station is shown in Figure 35.

Figure 35 802.1X Enabled by Default, Although Treated as Clientless



Test

Windows was unable to find a certificate to log you on to the network

The client, having no certificate provisioned before this event, does not reply to this request at all, and demonstrates the message above. The above scenario may hold true for machines that have been re-imaged as well, depending on the operational configuration or characteristics of the image itself.

Recommended best practices for these types of machines are to enable the 802.1X supplicant on the device to send EAPOL-Start frames (through registry setting) only *after* appropriate credentials have been loaded (such as any needed certificates).

MAC Authentication Bypass Limitations and Challenges

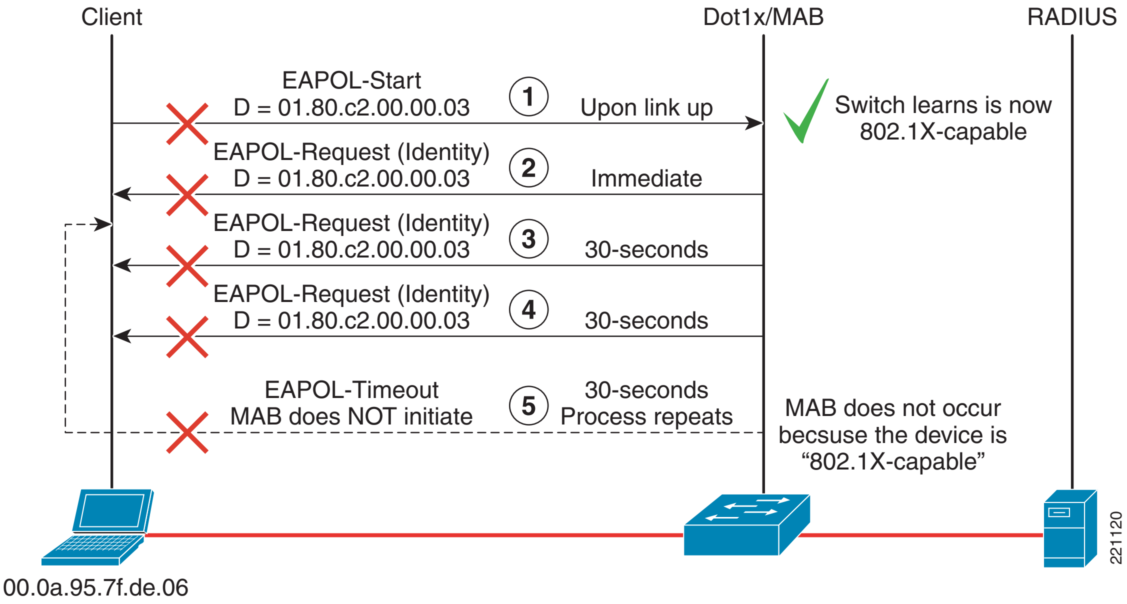
Fallback Technique for Re-imaged Machines with CSSC

A supplicant such as CSSC may be provisioned as part of a standard machine build. If the supplicant then sends an EAPOL-Start frame, with no existing certificate or temporary credential, 802.1X initiates and the 802.1X process times out. In addition, this process continues persistently and MAB never executes. This is because the client sent an EAPOL-Start frame to the switch initially, and a switch uses EAPOL to determine whether the device is supplicant capable (so 802.1X is tried always).

A recommended practice is to integrate with the provisioning process of the re-image of machines to disable 802.1X upon first boot unless 802.1X credentials can be built into the imaging process itself, such as one-time or temporary credentials to 802.1X authenticate, just to be able to attain appropriate network access for the purposes of downloading true user or device credentials.

For some cases, this may not always be the case in how the provisioning process occurs, especially because of re-imaged machines. Figure 36 shows an example of a Windows or Cisco Secure Services Client (CSSC) supplicant enabled for 802.1X that sends EAPOL-Starts and does not have prior certificate credentials.

Figure 36 802.1X and EAPOL-Starts Enabled without Credentials



This would be the same behavior if any other supplicant sent an EAPOL-Start and a screen was displayed to the user to input credentials for a challenge response-based EAP type such as PEAP. If the user does not respond to the credential notification, 802.1X times out and repeats transitively, and MAB is not initiated for these types of cases as well. Again, when a switch knows an 802.1X supplicant is on the wire through the device speaking EAPOL, MAB or the Guest-VLAN can not typically be leveraged. The only exception to the process indicated above is a global configuration available in IOS-based switches. Starting in 12.2(20)SE for Catalyst 3000 Series switches, the command is **dot1x guest-vlan supplicant**. In addition, this command has become hidden starting from the releases 12.2(31)SG for Catalyst 4500 and 12.2(25)SEE for Catalyst 3750. As of 12.2(35)SE, this command is still functional, but remains hidden as well. This command causes the EAPOL history not to be retained by the switch, so that after the above process goes through at least once, the state machine continues to run and eventually the Guest-VLAN or MAB can be enabled on the port if configured. This serves as a workaround if a situation such as the above is encountered. There is no such workaround available in CatOS.

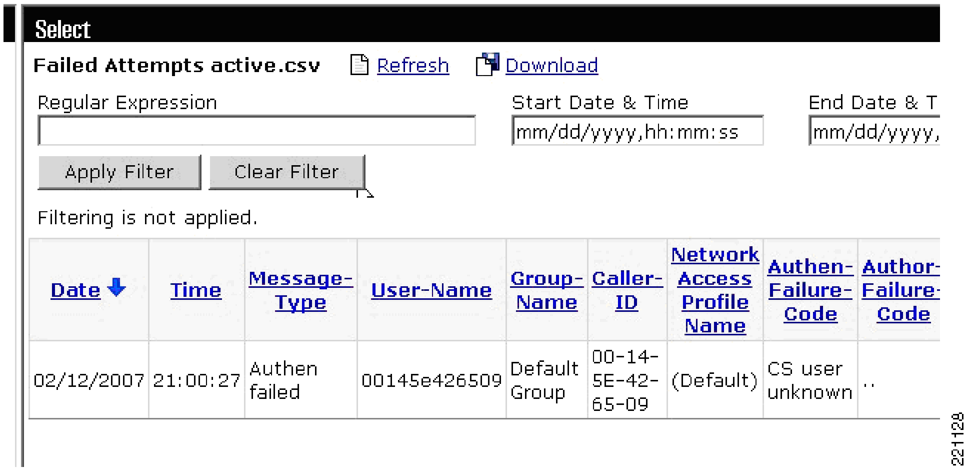
Provisioning

Provisioning is also a service of high concern to customers. A customer may not know what their MAC addresses are in advance. In addition, no turnkey solution is provided by Cisco to fill this void. Third-party products that provide asset management capabilities may help in this regard, such as products from Great Bay Software, Altiris, and so on.

Some customers have attempted to integrate learning techniques with their directory infrastructure. For example, a Cold Fusion front end can be used to force users to authenticate with Active Directory credentials. The front end then pulls the MAC address, host name, and user/machine details and puts them in an ODBC database. This is not only a potential MAC provisioning technique, but also a nice compromise of identity- and machine-based authentication without the complexities of 802.1X if the security model does not call for 802.1X.

However, the deployment of MAB itself can help elicit a provisioning mechanism. In addition, devices can be granted network access as well. An example of this is to use MAB along with the Guest-VLAN. Fundamentally in this scenario, a machine incapable of 802.1X always ends up in the Guest-VLAN. MAB does not necessarily change this, by the Guest-VLAN serving as a failure condition for MAB itself. Therefore, ultimately, a device can get into the Guest-VLAN much the same as it does without MAB, because it is incapable of 802.1X. However, if MAB fails "in the middle", a failure of this event should be recorded on the AAA server. An example from ACS of this failure is shown in Figure 37.

Figure 37 MAB Failure



Failed Attempts active.csv

Regular Expression

Start Date & Time: mm/dd/yyyy,hh:mm:ss

End Date & T: mm/dd/yyyy,

Filtering is not applied.

Date	Time	Message-Type	User-Name	Group-Name	Caller-ID	Network Access Profile Name	Authen-Failure-Code	Author-Failure-Code
02/12/2007	21:00:27	Authen failed	00145e426509	Default Group	00-14-5E-42-65-09	(Default)	CS user unknown	..

As shown in Figure 37, now the MAC address is effectively known to the authentication infrastructure. This MAC can now be potentially inserted into an asset management system or a primary directory infrastructure through various techniques.

Note In-depth guidance on identity management is beyond the scope of this design guide.

However, remember that the gathering of MAC addresses does not extend trust explicitly. LMS from CiscoWorks can also help as a MAC address gathering tool. It also performs device name, IP address, and host name correction. However, none of these techniques necessarily ensure that the entity should be on the corporate network to begin with; they may only prove that it is already there. More work should be done for the verification of network MAC addresses to validate existing identified trusted machines.

Lack of Existing Identity Store

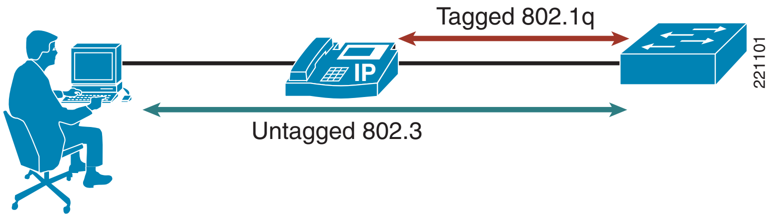
Specific MAC addresses are likely unknown to large enterprises. If they are known, they may not be incorporated directly into an existing directory infrastructure; they may be located only in an asset or inventory management system. For this management system to be used for authentication, it must be able to be interrogated by AAA, or the MAC addresses must be exported to a system that can be interrogated by AAA. For MAB, this means virtually any backend database into which ACS already has access. The identity store can be added onto, however. MAC addresses can be stored as user accounts on Windows

Active Directory. The CiscoSecure ACS database can store MAC addresses as well. The IBM Tivoli agent can add/remove MAC addresses in an ACS NAP. If MAC addresses are being defined as users in ACS, in ACS 4.0, the limit is 300,000 entries.

Lack of Voice Support

The integration of 802.1X, MAB and IP phones is based on the switch configuration of multi-VLAN access ports. Multi-VLAN ports belong to two VLANs: native VLAN (PVID) and auxiliary VLAN (VVID). This allows the separation of voice and data traffic and enables 802.1X and MAB only on the PVID. The type of communication that occurs on these two VLANs is shown in Figure 38.

Figure 38 Multi-VLAN Port



When 802.1X or MAB is enabled on a multi-VLAN access port, a client must complete the authentication process before getting access to the data (native/PVID) VLAN. The IP phone can get access to the voice (auxiliary/VVID) VLAN after sending the appropriate Cisco Discovery Protocol (CDP) packets, regardless of the 802.1X state of the port. The use of CDP with Cisco IP phones may be required, given the lack of pervasive support for an embedded 802.1X supplicant.

The configuration commands for Cisco IOS and CatOS that are required to enable multi-VLAN functionality, in conjunction with 802.1X and MAB, are as follows:

- Cisco IOS

```
interface FastEthernet0/1
 switchport access vlan 2
 switchport mode access
 switchport voice vlan 2
 dot1x mac-auth-bypass
 dot1x pae authenticator
 dot1x port-control auto
 spanning-tree portfast
 spanning-tree bpduguard enable
```

- CatOS

```
set vlan 2 2/1
set port dot1x 2/1 port-control auto
set port mac-auth-bypass 2/1 enable
set port auxiliaryvlan 2/1 2
set spantree portfast 2/1 enable
set spantree bpdu-guard 2/1 enable
```

Therefore, although MAB can be configured on a port and be used to authenticate data device, customers should not use MAB in an attempt to authorize voice devices on a Voice-VLAN. MAB is designed at the moment to authorize devices on data VLANs only and support VLAN assignment. If the MAC address of a phone is provisioned in ACS and it sends out packets, the switch is able to glean the MAC address and begin authorization to grant the phone access into the network on the data VLAN (or VLAN assigned from RADIUS). A switch does not know or pre-suppose the type of device and does not know to put it on the voice VLAN as part of the authentication event, however. Thus, if the customer provisioned the phone to tag its packets on the voice VLAN, it fails as of today, because traffic on voice VLANs for MAB is explicitly ignored. Therefore, a customer cannot use MAB to attempt to authenticate a third-party phone. A potential workaround is to dynamically assign a data VLAN via RADIUS and MAB equal to a voice VLAN without the voice VLAN configured on the switch port. However, this is not recommended because single-auth mode would not allow any other MAC on the wire such as a client plugging into a phone. In essence, MAB shares the same rules in this space that 802.1X does.MAB can be enabled for data devices, and Cisco telephony devices can be ignored with CDP. However, similarly to 802.1X, MAB-authenticated session may disappear from the network without the network knowing about it explicitly. A client disconnecting from the back of an IP phone is not recognized as an event by the switch. The first problem with this behavior is that when a host disconnects from the phone, the host remains authorized on the switch port. In addition, for any new machines that plug into the phone, a security violation may be tripped, because the phone thinks another MAC has appeared on the wire other than the one it has authenticated. Catalyst 3000 switches recently delivered a MAB aging feature to address this in 12.2(35)SE, but could not be verified for this phase of network virtualization.

Further integration with IP Communications is planned for a later phase of network virtualization, which will examine Multi-Domain-Auth (MDA) and MAB aging. MDA is a new solution-based feature set that allows any phone to authenticate via 802.1X or MAB, and is also able to authenticate a client plugging in behind an IP phone via 802.1X or MAB starting with Catalyst 3000 switches in 12.2(35)SE, and 4500 switches in 12.2(37)SG.

MAC Movement

Like 802.1X, a MAC address authenticated by MAB is not allowed to move on a switch unless the port from which the device moved is unauthorized. This issue is exacerbated by the MAB aging issue introduced in the previous section with respect to IP telephony. Therefore, if a device is authenticated via MAB behind a phone and then moves to another port on the same switch, the port to which the user moved is err-disabled. This renders the phone on that port inoperable as well. With CatOS, there is a configurable nature for security violation behavior handling to restrict traffic from an offending MAC instead of shutting the port down. However, even this does not help in this case. This violation behavior handling would help only for the appearance of a second MAC address on the original port, not for the movement of the MAC address to begin with. This is typically not an issue for a MAB port with no IP telephony because the move drops link on the port and clear the binding of the address to MAB. This issue may persist in a hub-based topology, though this is not a supported design. In addition, in CatOS today, a port must be manually reset when this event occurs. There is no auto reset after a configurable interval.

MAC Authentication Bypass Policy Assignment

Based on the consistent architecture MAB promotes along with 802.1X, MAB can automatically leverage any specialized policy enforcement techniques that may already be available to 802.1X. Especially important to network virtualization is dynamic VLAN assignment via RADIUS. No special configuration on a switch is needed to achieve dynamic VLAN assignment.

Standard recommendations for 802.1X with VLAN assignment remain with MAB. It is highly recommended to plan and build out any supporting VLAN architecture in advance. VLAN assignment is done by name with MAB like it is with 802.1X. This can support flexible VLAN management techniques for various L2 or L3 VTP architectures, allowing for independence between separate L2 domains. The architecture also allows for policies to be applied to groups or down to a per-device level. Depending on the specialized need, MAB may be managed on a per-host basis like this in some cases.

Remember on IOS-based switches to make sure you enable AAA and specify the authentication and authorization methods:

```
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
```

Note RADIUS attributes received in CatOS are automatically implemented if 802.1X is enabled. However, this is *not* the case for IOS. This is why you need the last configuration statement above, for the switch to accept configuration commands via RADIUS.

As mentioned above, none of the above applies to CatOS platforms, and these configuration steps are not needed by default. However, VLAN assignment, can be optionally disabled via the following configuration:

```
id1-6503-1> (enable) set dot1x radius-vlan-assignment ?
  disable                    Disable dot1x Radius Vlan Assignment on the system
  enable                     Enable dot1x Radius Vlan Assignment on the system
```

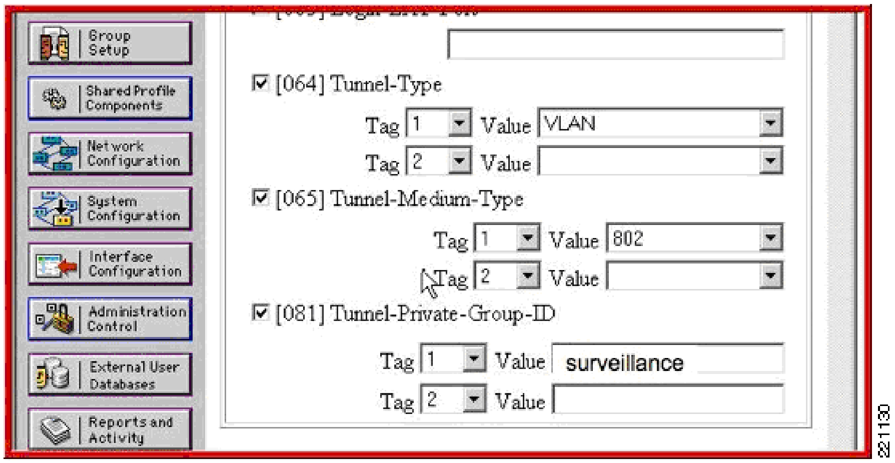
Nothing is needed on the ACS server, outside of what may already be in place for 802.1X as well. The following three standard RADIUS attributes defined by RFC 2868 are required:

```
[64] Tunnel-Type – “VLAN” (13)
[65] Tunnel-Medium-Type – “802” (6)
[81] Tunnel-Private-Group-ID - <VLAN name>
```

Note Before ACS 4.0, these features were viewable by default. To enable group-level viewing, they needed to be viewed under the “RADIUS (IETF)” link under the Interface Configuration configuration button. There are check boxes for each attribute. With ACS 4.0, however, this configuration step is not needed, and the attributes are enabled by default via per-user, or a per-group deployment scenario.

Figure 39 shows an example of configuring a certain group of devices for MAB to be deployed into the “surveillance” VLAN.

Figure 39 VLAN Assignment Configuration on ACS



☑ [064] Tunnel-Type

Tag 1 Value VLAN

Tag 2 Value

☑ [065] Tunnel-Medium-Type

Tag 1 Value 802

Tag 2 Value

☑ [081] Tunnel-Private-Group-ID

Tag 1 Value surveillance

Tag 2 Value

This enables any user members of the group configured for VLAN assignment to be assigned into the named VLAN. The VLAN name must be present on the switch, and be the identical name of the configuration in ACS. This includes white spaces and capitalization. The VLAN must exist on the switch as well. If any of these are not valid, a switch denies authorization. The user may provide a credential authorizing the user to access the network on a VLAN. However, if the switch cannot verify the information about the VLAN itself (though any sort of VLAN name mismatch, type-o, and so on), a switch treats this as a user not in fact providing valid credentials.

The VLAN name is mapped to a VLAN number. Upstream, path isolation uses separate VLANs as entrance criteria into each separate network partition. With wireless, you may also optionally ensure the original request originated on the correct SSID to ultimately map a session into the correct VLAN.

By leveraging dynamic policy enforcement, this completes the ability of an enterprise to differentiate between clientless sessions on the network. Previously, network virtualization was incapable of leveraging this differentiation capability. Network virtualization could differentiate between client contexts with 802.1X, but could only default to providing a de facto level of access if 802.1X was not resident on an end device. By having MAB and policy enforcement available, network virtualization can now be expanded to included differentiated services among robotic arms on a factory floor, x-ray machines in a hospital, IP-enabled surveillance devices, or standard corporate PCs. This increases the end-to-end impact network virtualization provides with this additional, fine-grained access control.

MAC Authentication Bypass Summary

In summary, MAB functions as a port-based feature. It is primarily used as a fallback mechanism to 802.1X, although it is optionally available as a standalone authentication method with CatOS. There is no de facto ability to support more than one MAC per port. MAB is single host in nature just like 802.1X, and there is no multi-auth for MAB. A MAB port can be optionally enabled for multi-host mode just as is done with 802.1X. MAB cannot be used as a means to deal with failed 802.1X authentication attempts. MAB provides customers who do not or cannot do 802.1X, but who have also bought into port security with configured MAC addresses more options, and provides a migration path to customers running URT or VMPS technologies. MAB also works with any standard RADIUS server, with a default timeout of 30 seconds with three retries. This means that the total timeout period is at least 90 seconds by default, which is the same minimum default timeout of the Guest-VLAN. A device must also send traffic into a switch for the MAC to be learned after the 802.1X timeout. If MAB fails, network access is implicitly denied. If MAB fails and the Guest-VLAN is also configured, the Guest-VLAN is enabled (for backward compatibility). Additional network policy can be downloaded as well. This supports dynamic virtualization, and the least common denominator is what 802.1X can currently do for the switch in question. A provisioning mechanism is not called for by MAB, although the Guest-VLAN can be used to assist in this process.

Overall Summary

With the increasing demands upon modern networks and the need to share information not only within an organization but as well with vendors and customers, security along with network access have become the top priority. The IEEE 802.1X specification for port-based network control has become the standard method for Layer 2 authentication access, not only with wireless but with the wired ports as well. 802.1X is a core technology component in support of access-control to promote end-to-end network virtualization.

One challenge in wired topologies is how to support failed authentications, especially for the 802.1X-enabled guest or partner. The Auth-Fail-VLAN is a way to grant access to 802.1X-enabled guest or partners, while providing differentiated access from internal machines to promote network virtualization. Additional challenges in implementing IEEE 802.1X are the requirements to support the cutting edge of yesterday, which is now the legacy of today. Most legacy devices, such as printers, VoIP phones, and new emerging devices such as IP security cameras, do not have the ability to support an 802.1X supplicant but must be included in any pervasive virtualized network architecture. MAC Address

Authentication Bypass (MAB) is not meant to replace 802.1X; rather it is meant to allow an alternate means of authentication when a host or device does not respond to requests for credentials by network access devices.

The IEEE 802.1X standard and MAB allows the dynamic configuration of access ports as well as implementing the corporate security policy on the port level. MAB addresses the difficulty of deploying an 802.1X infrastructure throughout an enterprise network. An 802.1X supplicant is required to authenticate to an authentication server via a network access device. The MAB feature allows devices without this 802.1X capability to access the network and perform their desired function while allowing L2 authentication to occur and participate in the dynamic deployment of network policy to promote a virtualized network architecture.

The Guest-VLAN is also an option for devices incapable of 802.1X. By combining MAB and the Guest-VLAN, an enterprise can now differentiate between clientless stations in support of device-specific access control as an application of network virtualization. In addition, the access control methods described in this document provide multiple levels of user access, making it the first element of network security. In addition, by supporting end-to-end network virtualization, these levels of access can take on more of a matrix model, with departmental and divisional roles dictating where access can be applied in the future.

Network virtualization can help reduce overall risk, add value, and remove operational cost from the business because of its logical network overlays while promoting security. Enterprises should develop corporate strategies now to control network access. Future challenges such as closed user groups, dynamic project creation, or even more on-demand device mobility can then be evaluated with the network as a direct enabler of business services.