MAC Authentication Bypass

This document provides deployment guidance for MAC Authentication Bypass (MAB). MAB is now a core component of Cisco Identity-Based Networking Services (IBNS). Like IBNS, MAB identifies the users or devices logging into an enterprise network. An identity is an indicator of a client in a trusted domain. An identity is typically used as a pointer to a set of rights or permissions to allow for client differentiation. MAB promotes access control by promoting authentication, which is the process of establishing and confirming the identity of the client requesting services. Authentication is crucial for network-based security benefits, and also to establish corresponding authorization.

When identified, 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 access control lists (ACLs) based on the authentication.

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

This document focuses on MAB as a means to achieve clientless authentication in the absence of 802.1X.

This document is intended for field sales engineers and account managers interested in using 802.1X as a model of port-based access control in their customer networks.

Contents

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

Overview

MAB, as described in this document, is intended to provide controlled access to devices based on their MAC address. MAB allows 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.

Today, 802.1X is the recommended port-based authentication method at the access layer in enterprise networks. 802.1X has three primary components: supplicant, authenticator, and the 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). Exception lists on routers or switches are also not scalable for large enterprises. A method is therefore needed for supporting these hosts.

Access control must 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, 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.

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 1.

Figure 1 MAB Overview

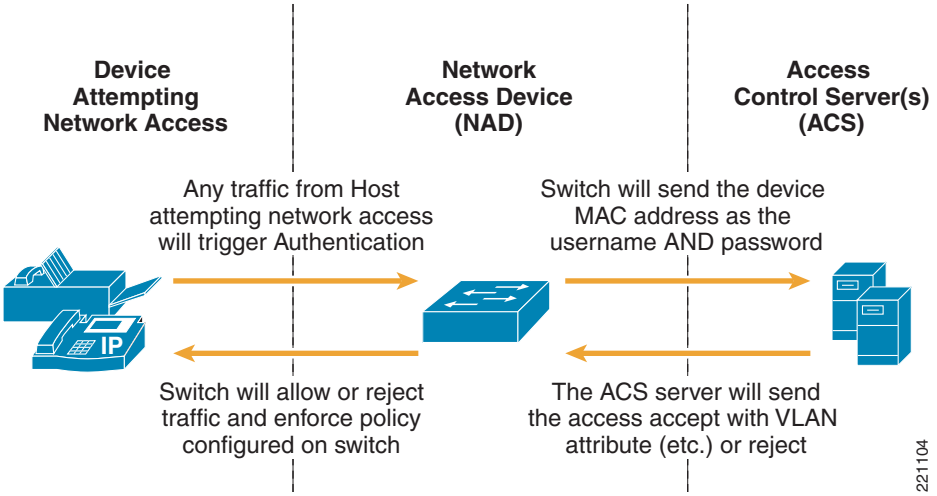


Figure 1 shows a device generating traffic (any traffic, such as DHCP, ARP, and so on), the switch captures the MAC address and forwards this as the username and password to ACS. MAB allows end users to authenticate (without any supplied credentials). As is discussed subsequently in this document, MAB 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.

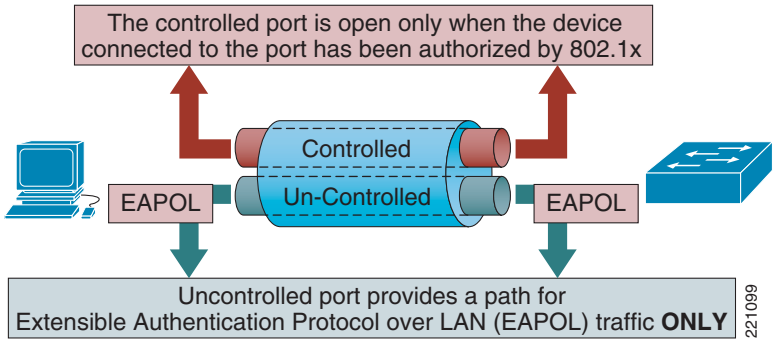
MAC Authentication Bypass Operational Overview

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 2.

Figure 2 Standard 802.1x Operation

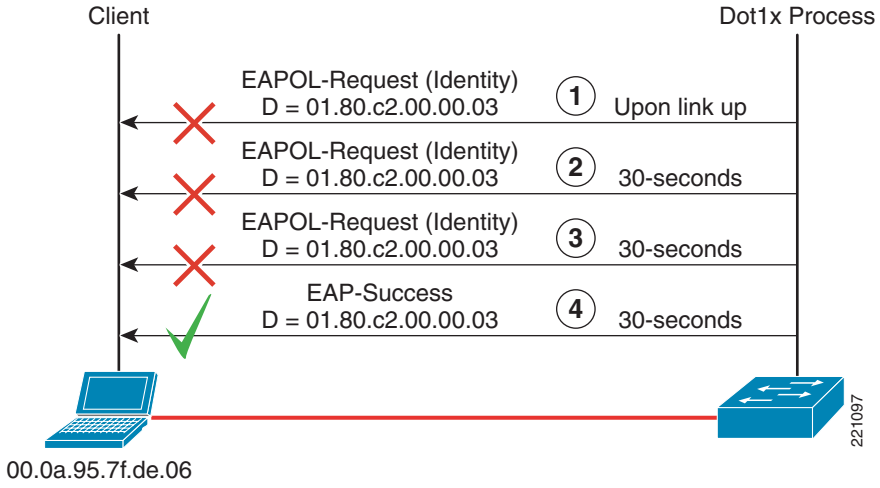


In Figure 2, only EAPOL traffic is typically processed by a switch port while 802.1x is maintained in an operating and active state. Thus, any MAC address from any edge device may not be known until EAPOL frames are processed from it. These are security benefits of 802.1x, and do not change in any way 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 3.

Figure 3 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

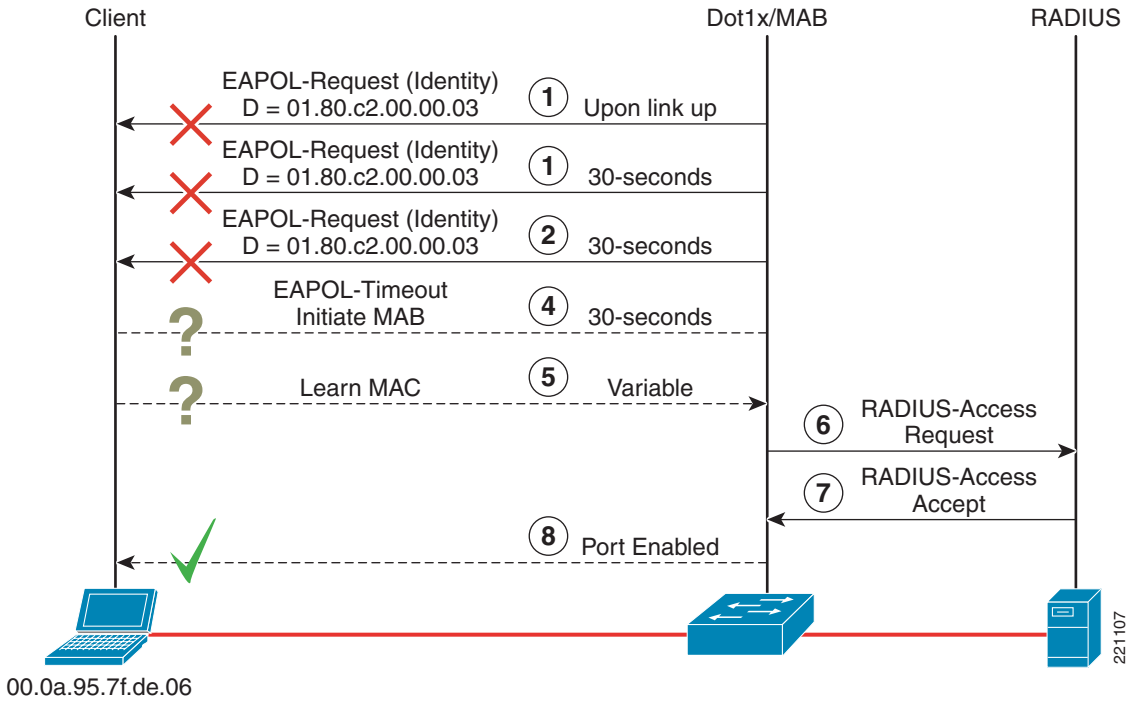
MAC address can be learned by the switch through classic MAC learning techniques. after a MAC address is learned by the switch, it can then be authenticated by RADIUS initiation. The RADIUS call transmits the attributes shown in Table 1 as part of a RADIUS request to AAA.

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 4.

Figure 4 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 4, 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.

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.

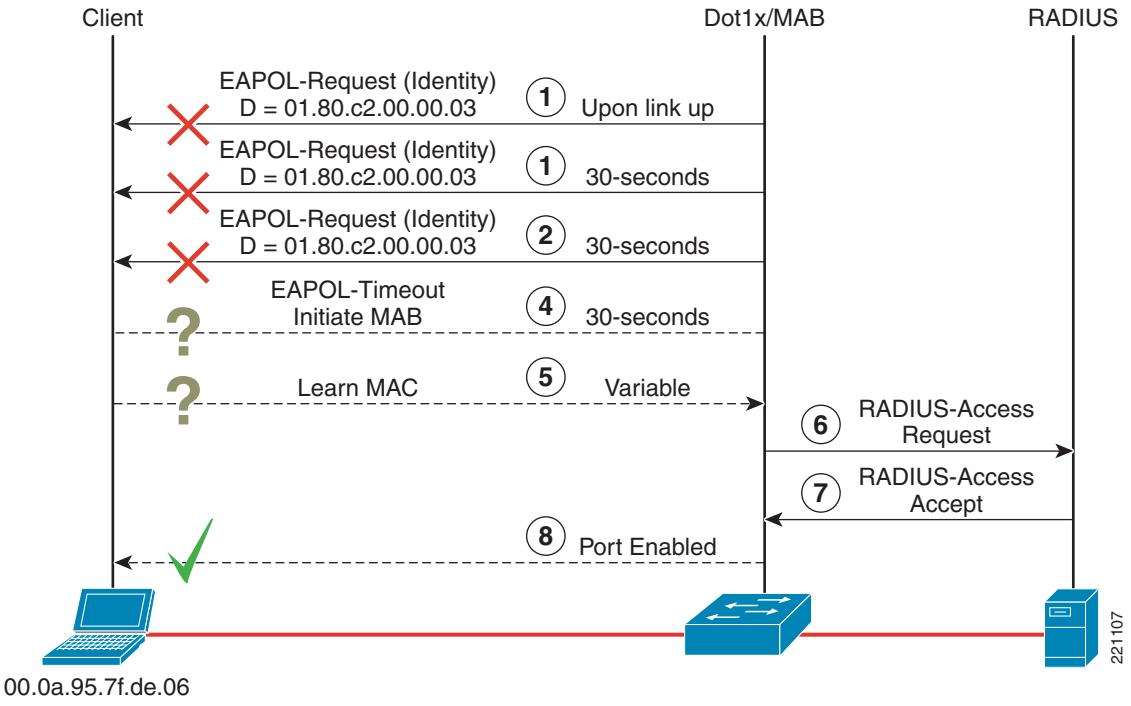
802.1x Timeout

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 5.

Figure 5 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.

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
```

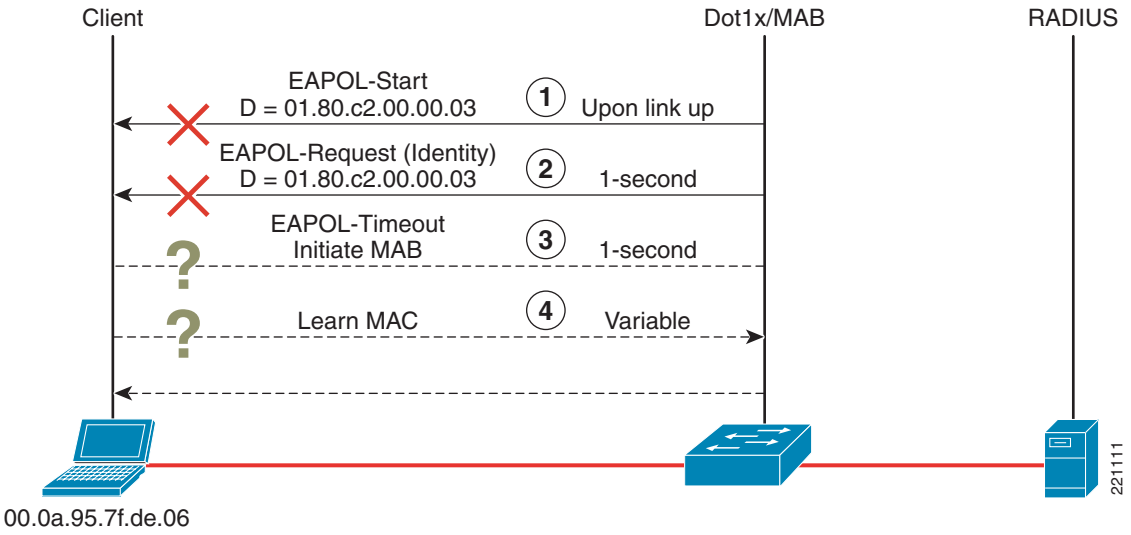
Note In CatOS systems, the values for *max-req* and *tx-period* are set at a global level, and not per port, as they are in Cisco IOS software.

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 6.

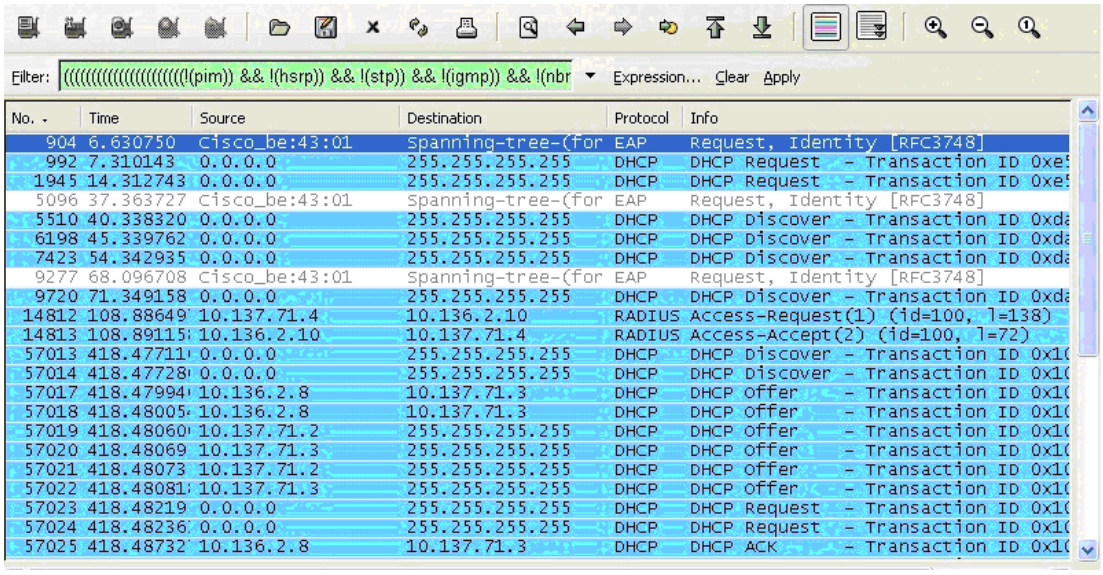
Figure 6 MAB Initiation with Tweaked Timers



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 7.

Figure 7 MAB Impact on DHCP



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.

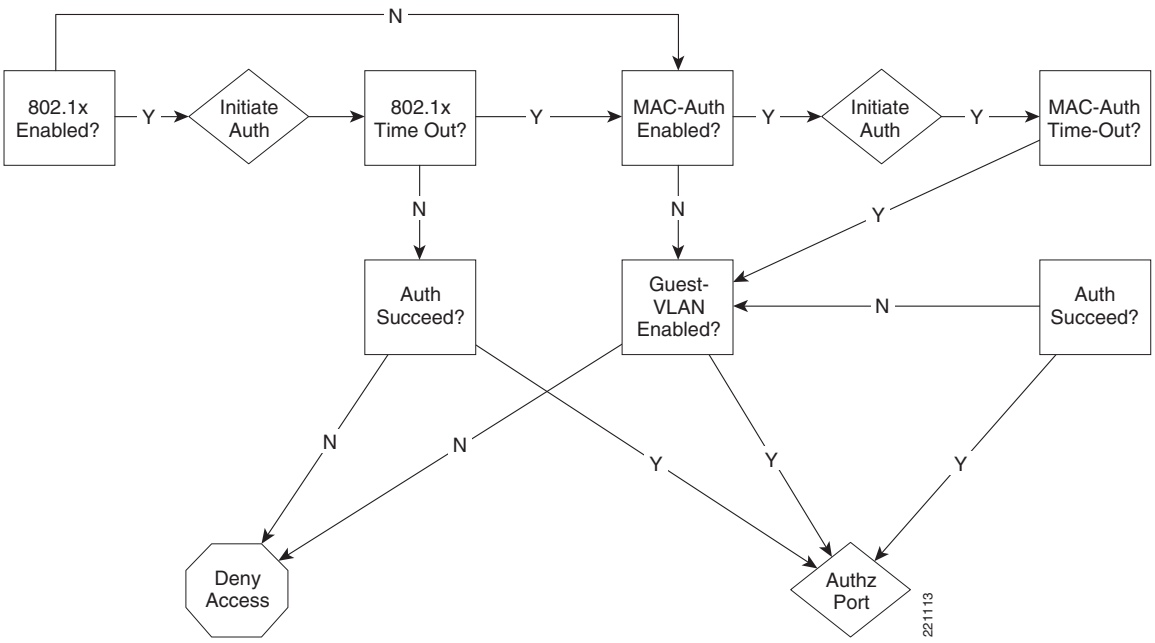
On IOS-based switches, MAB cannot currently be enabled without 802.1x. This also impacts any potential re-authentication scenario. Technically, re-authentication for MAB does not exist on IOS. Therefore, if re-authentication is enabled (for 802.1x) when a switch port is authenticated via MAB, a switch 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 8.

Figure 8 802.1x, MAB and the Guest-VLAN



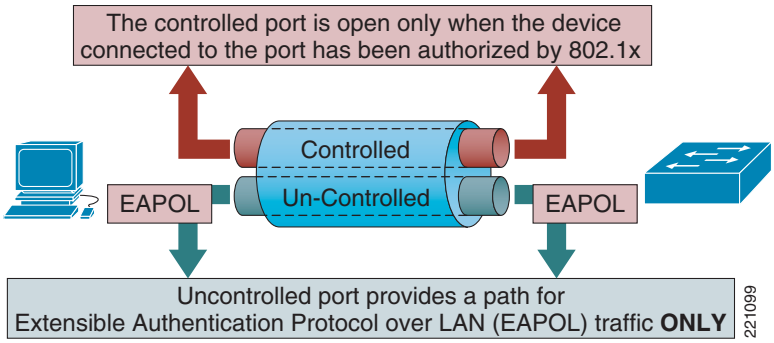
The operational nature of the feature interaction in Figure 8 was designed primarily as part of MAB to support backward compatibility for devices that cannot speak 802.1x and have already deployed the Guest-VLAN.

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.

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 low-energy 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 rehearsed in the figure below:

Figure 9 Standard 802.1X Operation



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 that 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. Therefore, in an effort to interoperate with WoL environments, most Catalyst switches provide unidirectional controlled port functionality as an optional configuration. This configuration is demonstrated 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
```

- 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
```

The configuration above represents a weaker deployment of the technology, but does not necessarily present any security vulnerabilities. This configuration allows only the outgoing traffic on a port, 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. Operationally, the controlled port is now operating only in one direction. A WoL magic packet can now exit the network to wake up a machine if necessary. It is now expected that the machine must then 802.1X authenticate to successfully send traffic into the network. WoL is a per-port feature.

Note A best practice is to enable WoL only on the ports where it is needed.

Thus optionally on a per-port basis, the configuration above represents the notion of a uni-directional controlled port. This means that only EAPOL should come into a switch until authenticated, but anything can now go out of the switch, including a WoL frame, or "magic packet". This represent a successful operation of sending this "magic packet" to a machine to wake it up, even though you have 802.1X configured. It is then the job of the supplicant to perform 802.1X after being woken. Ultimately, this allows for any sort of maintenance, software patching, delivering e-mail to a machine, and so on, that may have been in place on the enterprise LAN before an 802.1X deployment.

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

Note 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.

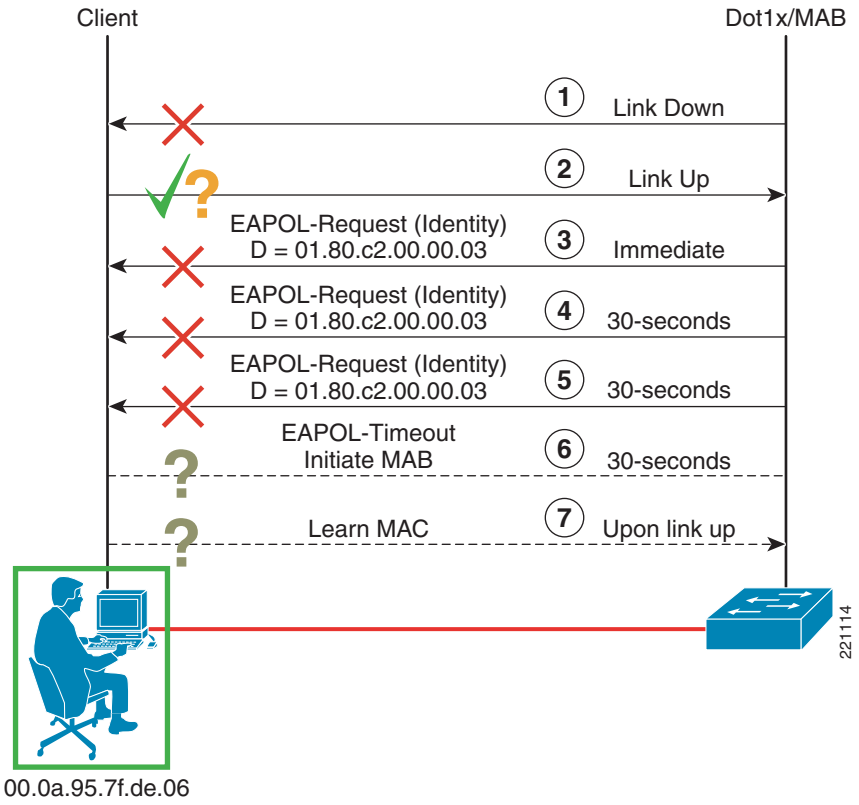
Note If the link goes down, and clients plug directly into the authentication, an EAPOL-Logoff does not represent much value, because 802.1X/MAB state machines are also directly driven by link state.

A switch port is down conditionally upon a link-down event being processed by an authenticator, or by an EAPOL-Logoff frame being transmitted by a supplicant before the machine goes to sleep. Link should then come back up on the port immediately. The link-up event is then processed on the port as well.

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 10.

Figure 10 Machine Going Into Power Save Mode with MAB



As shown in Figure 10, 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.

Note 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.

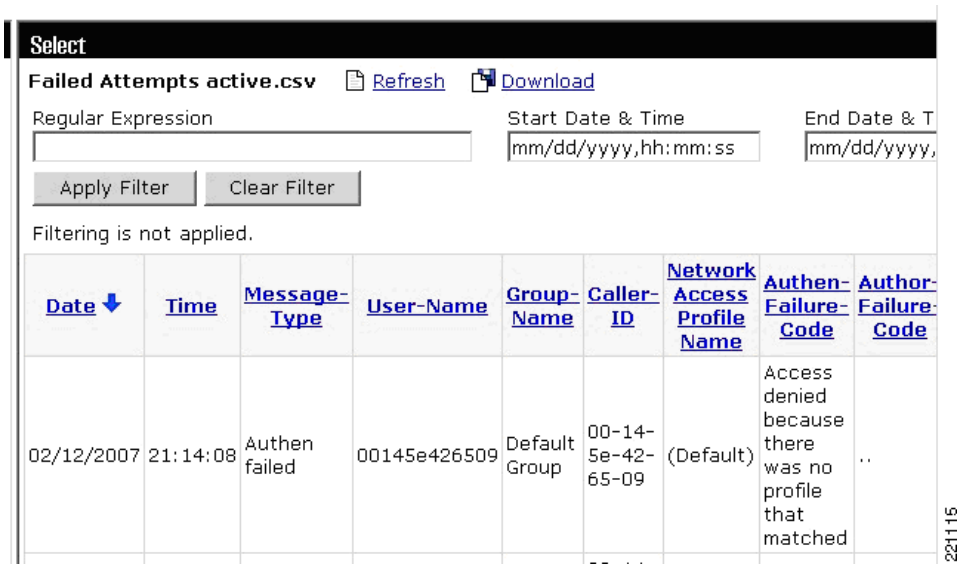
Network virtualization may impact the operation of WoL functionality for PCs, as for example when the access VLAN on a switch serves as entrance criteria to a VRF or VPN separate from the global table, where a WoL server is typically deployed. In that case, you need to be sure that this partition can 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 11.

Figure 11 Deny MAB Request Based on Pre-determined Location



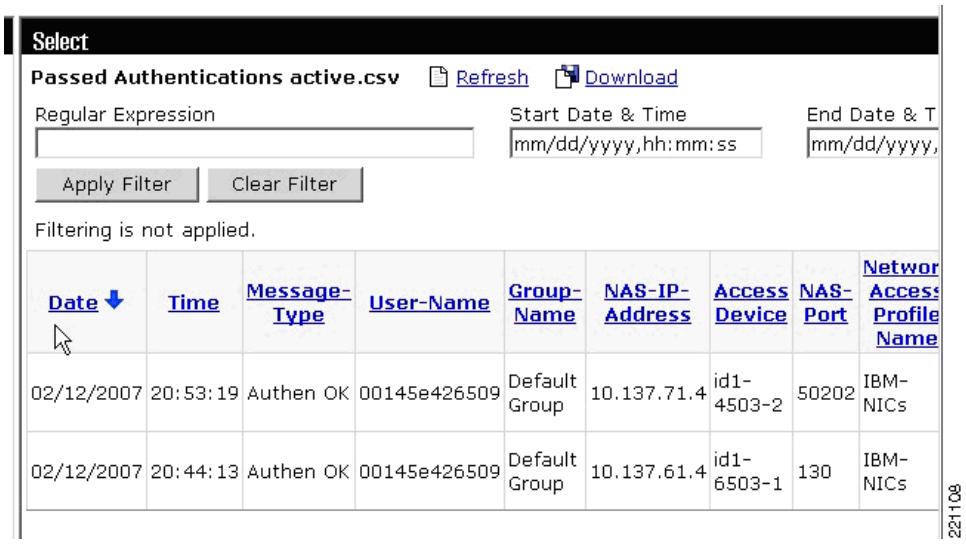
For more information on Network Access Profiles, see the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/csacs4nt/acs40/user/sp.htm

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.

MAB Format on Switches

As indicated previously, the format of the MAC address in any MAB request is important to realize for use by the authentication infrastructure. Any RADIUS requests transmitted by MAB of Cisco Catalyst switches contains both RADIUS Attribute [30] (the Called-Station-ID) and Attribute [31] (the Calling-Station-ID). Attribute [30] is the MAC address of the ingress interfaces of the switch or authenticator. Attribute [31] is the MAC address of the 802.1X supplicant or the end-station. Both of 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] (the User-Name) and RADIUS Attribute [2] (the 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 12 shows passed authentications on ACS from an IOS-based switch and a CatOS switch running MAB.

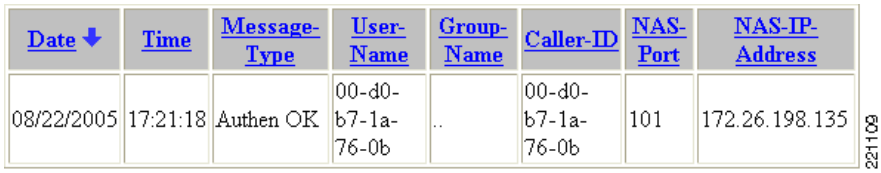
Figure 12 MAB from IOS 12.2(31)SG and CatOS 8.5(5)



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 13 represents a passed authentication on ACS from CatOS with 8.5(4) demonstrating this condition.

Figure 13 MAB from CatOS 8.5(4) and Before



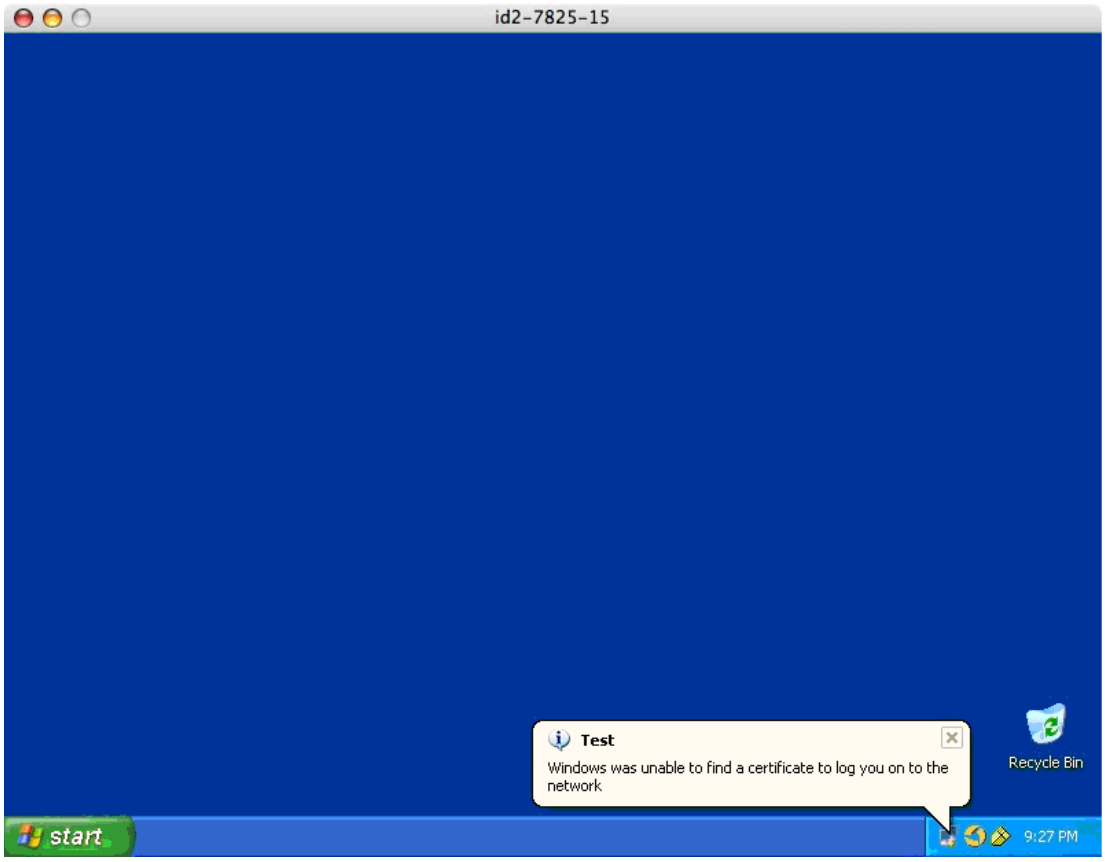
As a result, if the same ACS server is used to authenticate MAB from these various types of switches, the MAC addresses need to be entered into a database twice. This is completely unmanageable, and a recommendation for 8.5(5) is a must in heterogeneous environments. This benefit is now fully realized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

Fallback Technique for New/Re-imaged Machines

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 14.

Figure 14 802.1x Enabled by Default, Although Treated as Clientless



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.

Note 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

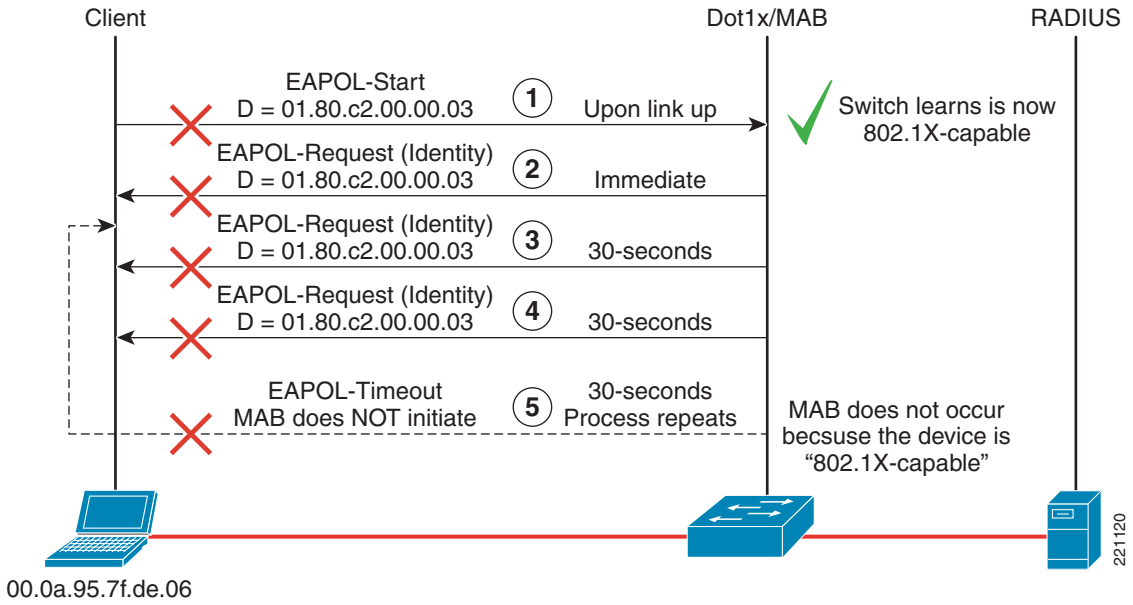
Fallback Technique for Re-imaged Machines

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 15 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 15 802.1x 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.

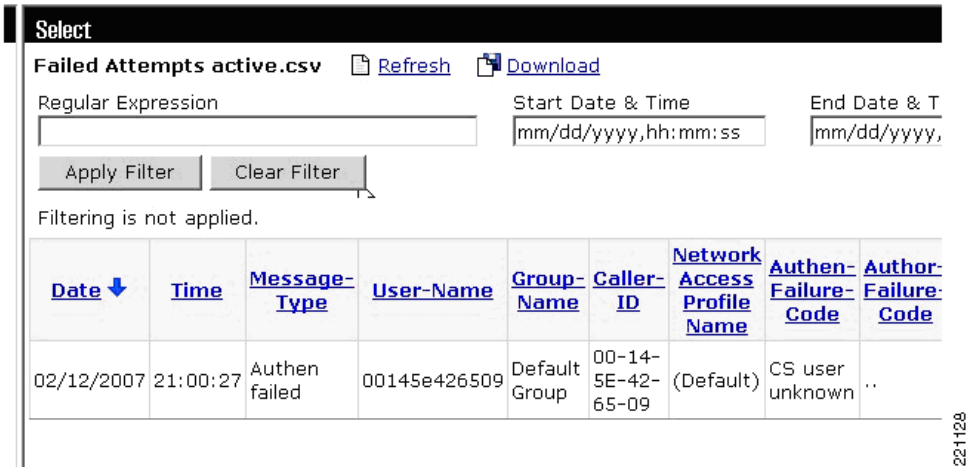
Provisioning

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 16.

Figure 16 MAB Failure



As shown in Figure 16, 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.

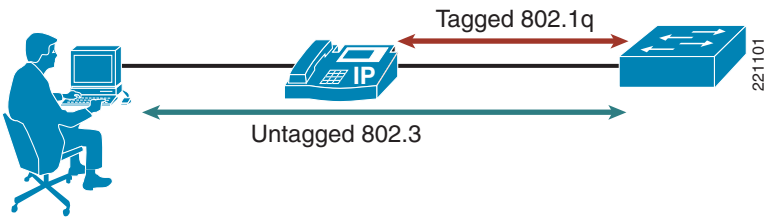
Lack of Existing Identity Store

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 17.

Figure 17 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
```

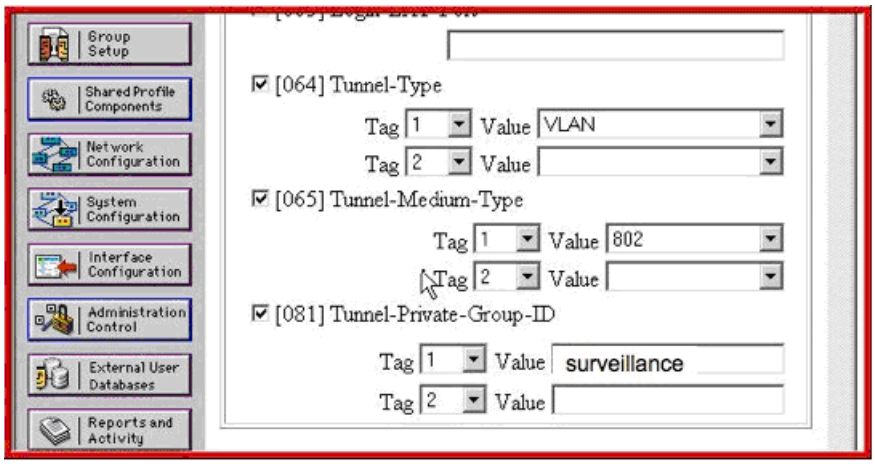
Nothing is needed on the ACS server, outside of what may already be in place for 802.1x as well. The following three standard RADIUS attributes defined by RFC 2868 are required:

```
[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 18 shows an example of configuring a certain group of devices for MAB to be deployed into the "surveillance" VLAN.

Figure 18 VLAN Assignment Configuration on ACS



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.

Summary

With the increasing demands upon modern networks and the need to share information not only within an organization but also 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. However, one of the challenges in implementing IEEE 802.1X is the requirement to support the cutting edge technology of the past, which is now legacy. 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 include network architecture that supports access control. MAC Address Authentication Bypass 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 the network access device request for credentials. The IEEE 802.1X standard and MAC Auth Bypass allows the dynamic configuration of access ports as well as implementing the corporate security policy on the port level. MAC Address Authentication Bypass 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 MAC Address Authentication Bypass feature allows devices without this 802.1X capability to access the network and to perform their desired function while allowing L2 authentication to occur and participate in the dynamic deployment of network policy.

To support the goals above, 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 stand-alone 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 like it 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 also want port security with configured MAC addresses, with more options, and also 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.