Whitepaper: MAC Authentication Bypass

Cisco Confidential

Disclaimer

Design Guide: MAC Authentication Bypass

Introduction

This document will provide deployment guidance for MAC Authentication Bypass (MAB). MAB is now a core component of Cisco Identity-Based Networking Services (IBNS) offering. Like IBNS, MAB aims to identify 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: 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, 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.

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 will focus on MAB as a means to achieve clientless authentication, in the absence of 802.1X.

Audience

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.

Overview

MAB, as described in this document, is intended to provide controlled access to devices based on their MAC address. MAB should allow non-802.1X compliant end device 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. It 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, etc. 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). Also, exception lists on routers or switches are not scaleable for large Enterprises. Thus, we need to have some method 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 will typically need the ability to dynamically provision individual MAC addresses (without impacting service availability) for network authentication of non-responsive devices like printers, video conferencing units, satellite receivers, faxes, etc. MAC Authentication Bypass 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. While 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 will reflect updates to changes in recent functionality across the Catalyst switching product line that may impact the related architecture.

1 MAC Authentication Primer

MAC Address Authentication itself is not a new idea. One classic flavor of this is port security. Another flavor is Cisco's VLAN Management Policy Server (VMPS) architecture. With VMPS, a customer can have a text file of MAC addresses and the VLANs they belong to. That file gets loaded into the VMPS server switch via TFTP. All other switches then check with the VMPS server switch to see which VLAN those MAC addresses belong to 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. Along the same lines as VMPS, yet another flavor 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 and/or Controllers. This base functionality for MAC address checking is already in place. For example, Wireless Access Points have the ability to initiate a PAP authentication with a RADIUS server using a client's 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 represent 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, due to the nature of the authentication method used. MAC address 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 represent a consolidation of current efforts toward identity, authentication and security. These are some of the reasons why MAB is suited for Access Control.

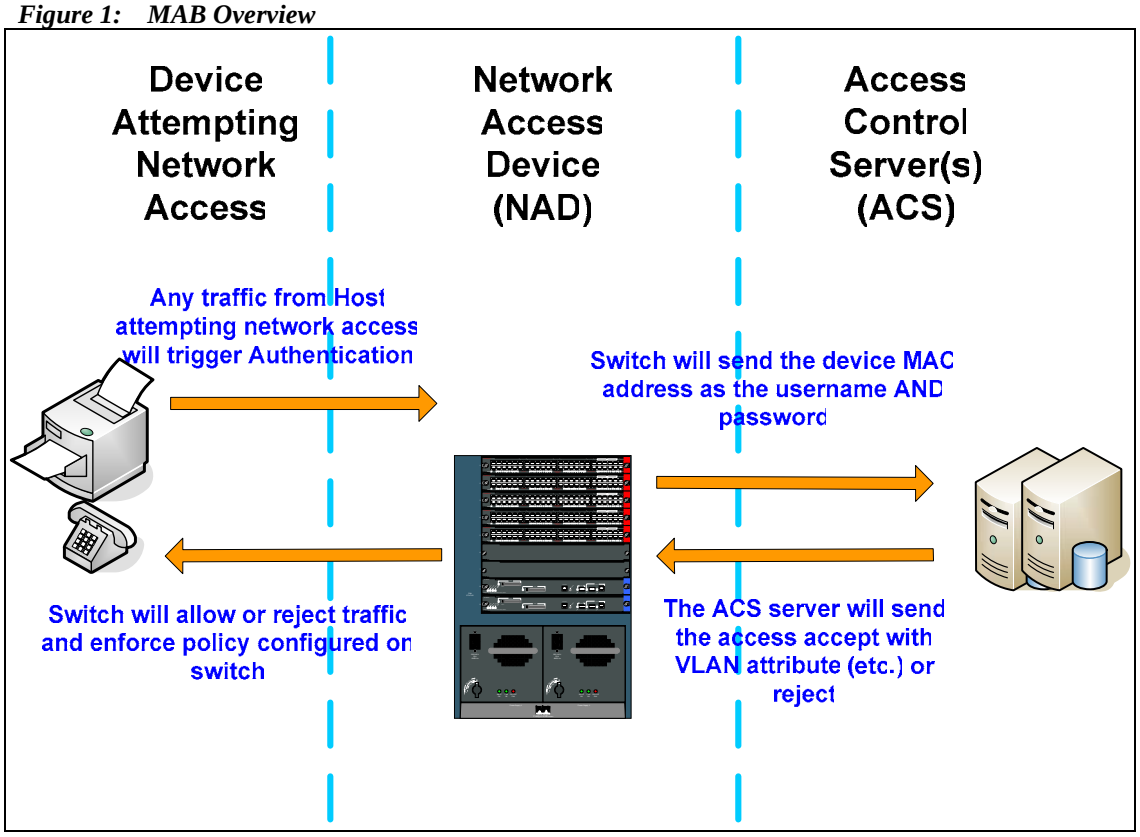
Other reasons to support MAB for Access Control are as follows:

- 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.
- Provides a migration path from port-security.
- Provides a migration from URT and/or VMPS.

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. An overview of MAB is demonstrated in the diagram below:

Figure 1: MAB Overview



The illustration above shows a device generating traffic (any traffic; DHCP, ARP, etc.), 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 will be discussed later 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. While 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 an Access Control strategy. 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 will not be examined further in the clientless context, due primarily to the nature of pervasive client capability in the overall wireless space. Due to 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 like Symbol handhelds with may only support Wired Equivalent Privacy (WEP). For more details surrounding wireless, and MAC authentication capabilities, see the LEAP/MAC Authentication Configuration Guide on CCO; http://www.cisco.com/warp/customer/707/leap-mac-auth.html

Note: Branch router functionality will not be examined further in the clientless context, due primarily to verification and testing resources. Cisco has traditionally provided 802.1X and its set of L3 authorization features on L3 ports popularly referred to as Spouse & Kids (S&K) solution. S&K consists of 802.1X authentication, host-mode support (i.e. 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, please see the Configuring Cisco IOS Easy VPN Remote with 802.1X Authentication whitepaper on CCO; 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 Cisco's position as a leader in that space by providing increased security and semi-automatic provisioning via the authentication of connected network clients.

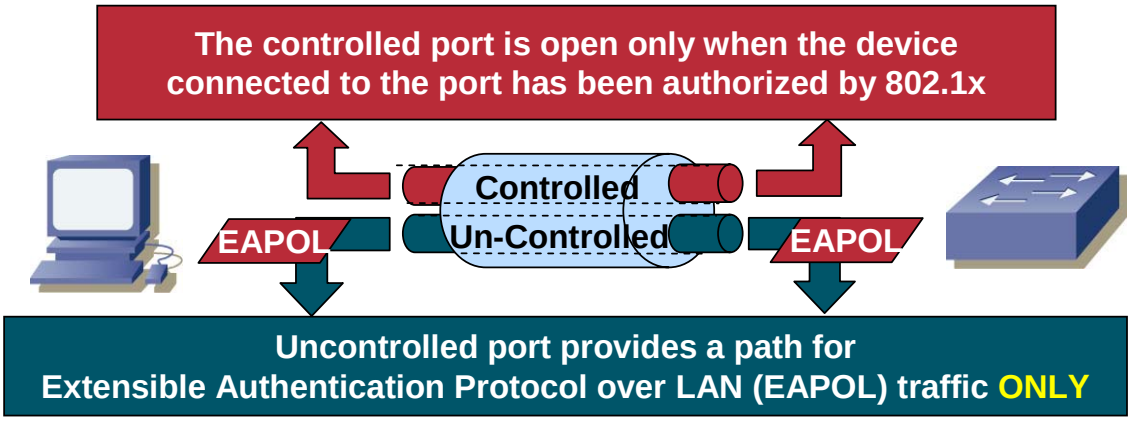
2 MAC Authentication Bypass Operational Overview

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.

2.1 802.1X 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 due to the default operation of 802.1X as depicted in the following diagram:

Figure 2: Standard 802.1X Operation

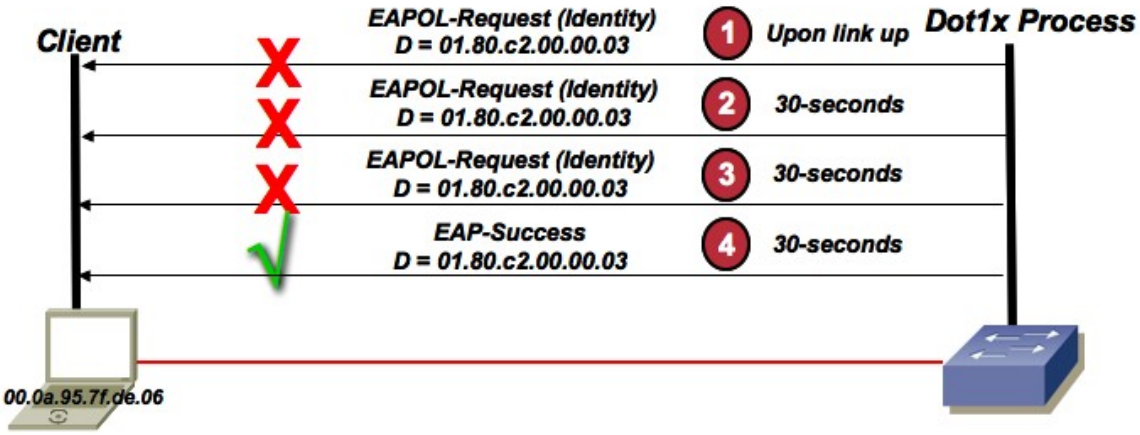


In the diagram above, only 802.1X packets (EAPOL) is typically processed by a switch port while 802.1X is maintained in an operating and active state. Hence, any MAC address from any edge device may not be known until EAPOL frames are processed from it. These are security benefits of 802.1X, and do not change in any way with respect to any MAB implementation. Since 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.

2.2 Guest-VLAN Rehearsal

Prior to MAB, the Guest-VLAN was the only alternative to provide network access to clients that do not speak EAPOL. This process is demonstrated in the figure below:

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. Hence, the ability to employ differentiated services based on

the MAC address alone is advantageous for identification purposes. Upstream, the Guest-VLAN may also only have access to limited resources, as defined by the network administrator. Prior to MAB, a MAC Address can only be known to a switch port after the port is enabled and placed into a Guest-VLAN. Also, once 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). Hence, there are limitations to using the "Guest-VLAN concept" as a solution to provide access for any non-802.1X enabled devices that can be addressed through MAB functionality.

So, 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.

2.3 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. Once a MAC address is learned, it can be authenticated via RADIUS initiation. The RADIUS call transmits the following attribute as part of a RADIUS request to AAA:

Table 1: RADIUS Attributes

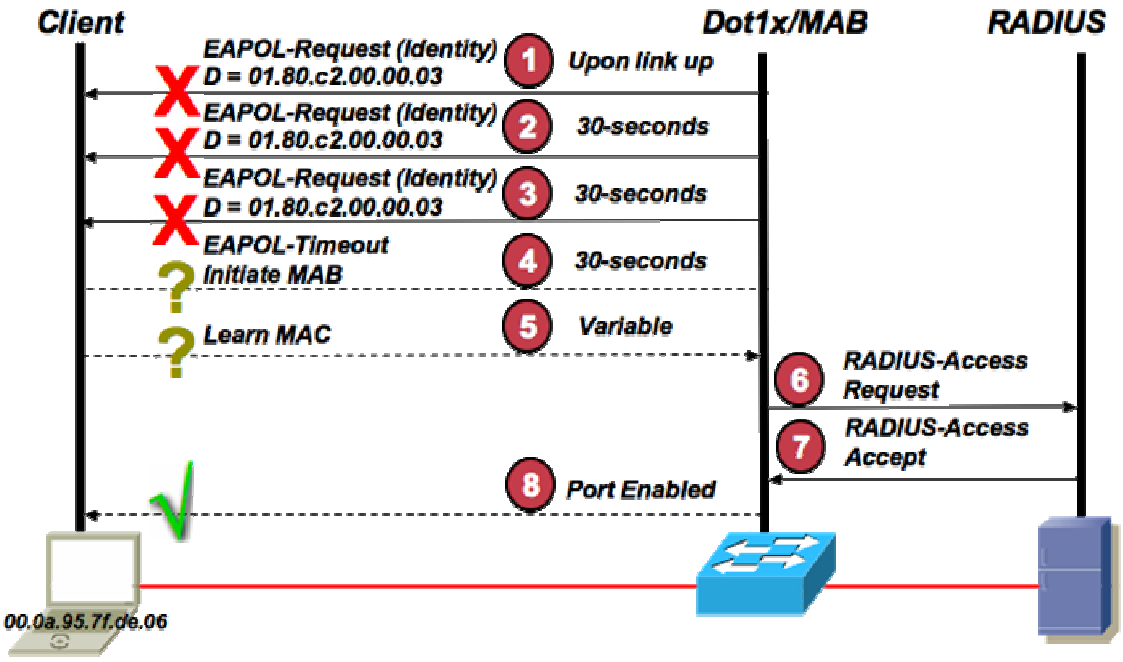
No.	Attribute Name	Description
1	User-Name	MAC Address sent in "hhhhhhhhhhhh" format, all lowercase with no meta characters or white spaces.
2	Password	Same as User-Name, but encrypted per PAP or MD5.
4	NAS-IP-Address	IP address of switch.
5	NAS-Port	Physical port of device acting as the authenticator.
6	Service-Type	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	Indicates MTU to be used by the user. Set to "1500".
30	Called-Station-ID	MAC address of device acting as authenticator, as seen by the peer.
31	Calling-Station-ID	MAC address of client.

61	NAS-Port-Type	Indicates type of physical port on the authenticator. Set to “15” for Ethernet.
80	Message Authenticator	HMAC-MD5 to ensure integrity of packet.

Note: CSCsh74068 has been filed for Attribute[61] above, as all IOS-based switches incorrectly set this to “5” for Virtual currently. Also, CatOS does NOT send the Framed-MTU attribute as part of a MAB request.

A complete operational flow of MAB is depicted in the following diagram:

Figure 4: MAB Operation



As depicted above, MAB only initiates 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. Once this occurs, RADIUS is initiated to the backend asking if the MAC should be allowed network access.

As depicted above, once a host/device fails to supply 802.1X authentication credentials, the network access device will take the learned MAC address and hand 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 and at this time other authentication methods can be attempted. Alternatively, the Guest-VLAN can be used as a mean to support a provisioning process of 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 will be discussed later 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 achieve 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 like it does with 802.1X. However again, 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, like per-user ACLs, VLAN assignment, etc. Also, 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

It is important to understand the format and location of the MAC address in any MAB request for use in an authentication infrastructure. Any RADIUS request transmitted by 802.1X or MAB on Catalyst switches will contain both RADIUS Attribute [30] (the Called-Station-ID) and Attribute [31] (the Calling-Station-ID). Attribute [30] should be the MAC address of the ingress interface of the switch or authenticator. Attribute [31] is the MAC address of the 802.1X supplicant or the end-station if available. Both of these attributes will be sent in the format of "XX-XX-XX-XX-XX-XX" (e.g. 00-10-A4-23-19-C0). This has recently been updated in switch code base to ensure both compatibility with legacy switch code and also compliance with RFC 3580. Neither of these attributes is expected to actually provide the authentication service provided by MAB though. Authentication and authorization are provided from RADIUS Attribute [1] (the User-Name) and RADIUS Attribute [2] (the password). For MAB, the user-name and password will also contain the MAC address of the peer device. However, for these attributes, the format is simply "hhhhhhhhhhhh" (e.g. 0010a42319c0). This is an all lower case version of "hhhh.hhhh.hhhh" with the punctuation stripped out. So if an identity infrastructure is to be built to support MAB, it should follow this format.

Timers are important to remember to 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-sec to enable a port in the Guest-VLAN. In the MAB case, this same timing is used to as a signal to the switch platform that it should now open the port to learn the MAC of an end-station to begin the MAB authentication process. More details with respect to timing details will be discussed later in this document.

Re-authentication for MAB is supported the same way 802.1X supports it as well for IOS-based switches. Any re-authentication configuration that may currently exist on a switch will impact MAB clients. By default, if MAB re-authentication is enabled with a specific session-timeout through a port configuration, 802.1X will need to timeout again, however. Then, once 802.1X times out, MAB will simply use the MAC address the switch currently thinks is on the wire in its cache as a means to check and see if the backend policy may have been disabled or changed for that MAC address. During this period, network access is persistent by default though. 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 point of view, the 802.1X state machine for 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. Also, 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, then this will allow any amount of machine on the port subsequent to the port being authenticated. This is true for MAB as well via the same configuration.

3 MAC Authentication Bypass Configuration and Verification

The following network infrastructure devices were validated as part of an overall Access Control solution:

- Catalyst 6503 – CatOS 8.5(4)
- Cisco Catalyst 4503 – 12.2(31)SG
- Cisco Catalyst 3750 – 12.2(25)SEE2
- Cisco Catalyst 2940 – 12.1(22)EA9

The following network infrastructure gear was NOT validated through testing as part of the overall solution:

- Any other 12.1E SKU (2950, 3550)

- Any other 12.2S SKU (3560, 2970, 2960, 4948)
- Any IP phone SKU
- 6500 IOS (does not support required feature set)
- Any other IP communications application
- Any branch/teleworker routers, ISRs

3.1 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
```

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
```

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 the port should operate under. "dot1x port-control auto" 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.

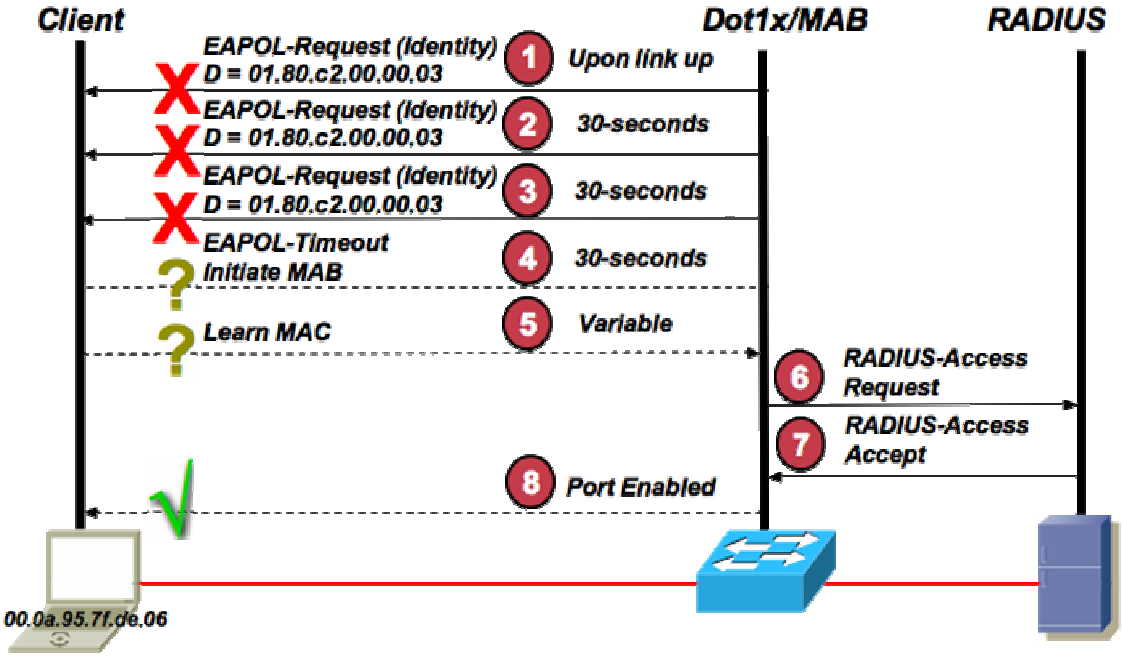
3.2 802.1X Timeout

Any port enabled for MAB today must also timeout on 802.1X before execution of MAB can begin. The default operation of MAB clearly wreaks havoc on the boot-up process of standard PCs in an Active Directory environment due to 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 utilize MAB for corner cases only, and allow 802.1X to handle the majority of controlled LAN access.

MAB should be an ideal option for clients insensitive to delays upon boot-up or login though, like 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 the figure below:

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 way 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; 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 MAB is relatively simple but differs on switches running IOS and CatOS software releases. A complete configuration with tweaked timeouts are demonstrated below:

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
```

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 dot1x max-reauth-req 1
set dot1x tx-period 1
```

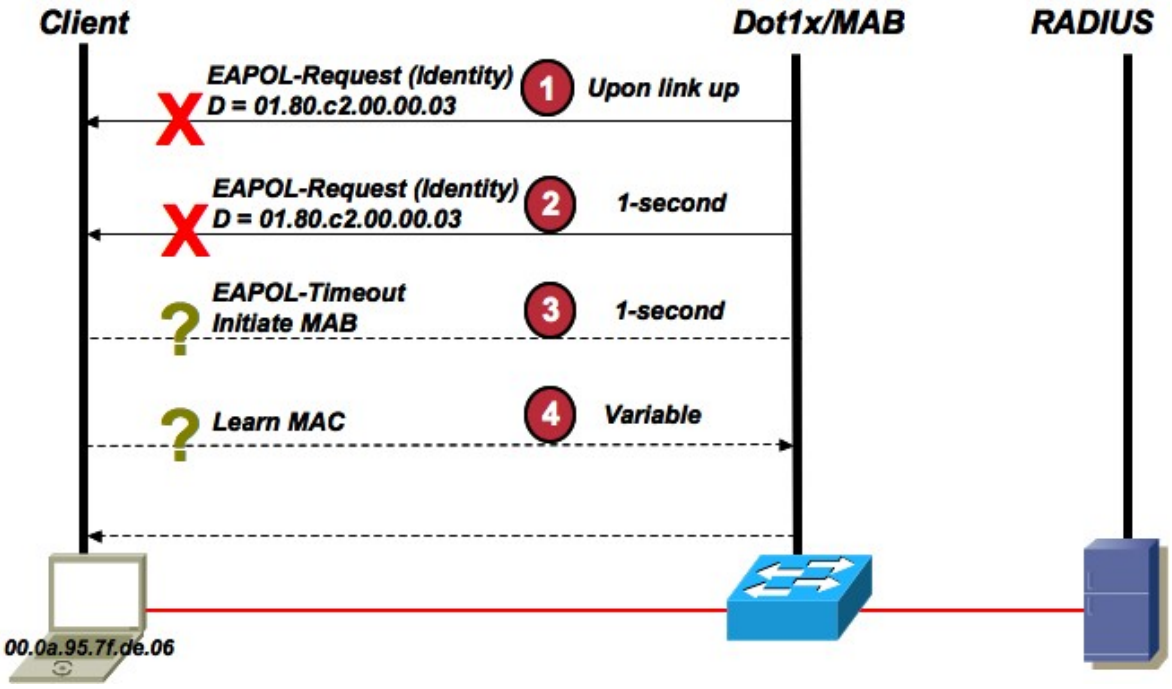
Note: In CatOS systems, the values for max-req and tx-period are set at a global level, and not per port, as they are in Cisco IOS software.

As demonstrated 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 only initiates 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 will typically vary. The operation of tweaked timers to timeout 802.1X quickly as indicated above is demonstrated in the following diagram.

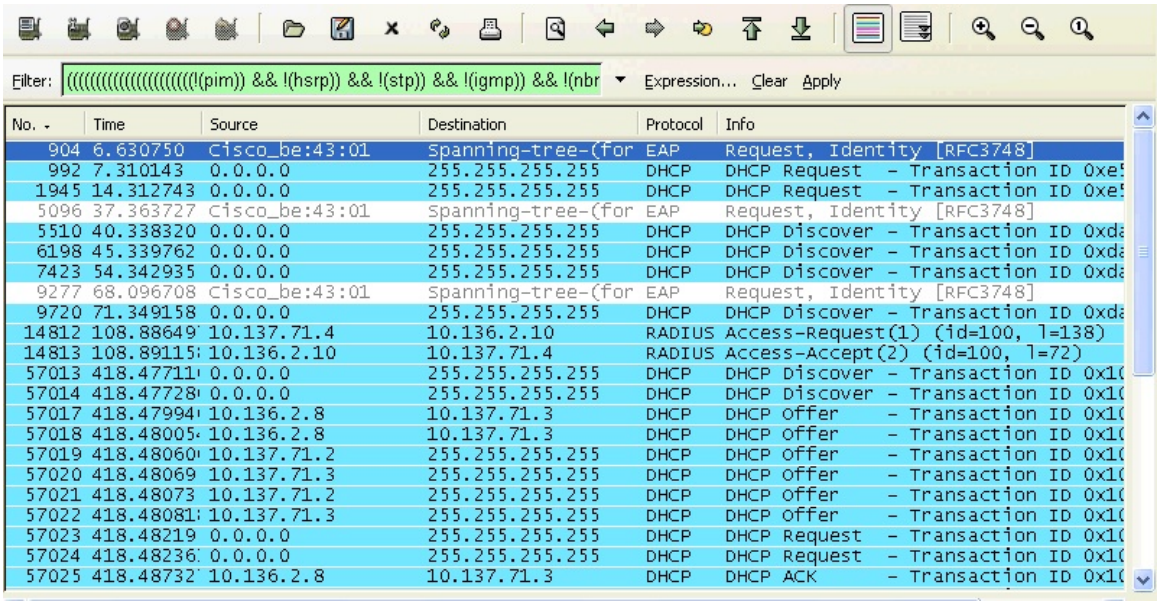
Figure 6: MAB Initiation with Tweaked Timers



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 demonstrated below:

Figure 7: MAB Impact on DHCP



Filter: (((((((((((((((((((!(pim)) && !(hsrp)) && !(stp)) && !(igmp)) && !(nbr Expression... Clear Apply

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 demonstrated in the example above, 802.1X times and MAB is successful ~90 seconds after a link up event. DHCP times out completely after about a minute though. Once Windows reverts to the internal address of 169.254.x.x though, an IGMP report from this address actually causes L2 traffic to be learned by the switch, so that MAB can initiate. So, although variable, MAB completes, in this example, ~40-sec after DHCP has timed out. Windows reverts to standard timeout procedures in this case, and does not attempt to renew its address for another 5 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 machine sensitive to this timeout condition.

Note: Although security feature interaction is not within the scope of this document, MAB was noticed to not learn an address if the packet was a DHCP frame. This may be due to DHCP-Snooping also being enabled and cause even longer delays. CSCsg03626 was filed to account for this.

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 2-seconds after the machine starts trying to send traffic. Most 802.1X supplicants are applications themselves, so 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 “sweet spot” for timer recommendations to make in this regard, as mileage will vary based on requirements.

3.3 Verification

Here 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   -
```

Here 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
```

4 MAC Authentication Bypass Feature Interaction

4.1 MAB and EAPOL Interaction

As demonstrated 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 once MAB activates. When packets arrive on a port where MAB is active, this results in the switch forwarding 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) then 802.1X will never relinquish control over 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.

Once MAB activates, a port is typically in an unauthorized state (since 802.1X times out). So while waiting
for a packet to glean a MAC address, if an EAPOL packet is detected, then MAB de-activates and relinquishes complete control back to 802.1X entirely. 802.1X will then attempt to authenticate the port. From then on, MAB will never activate as long as 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 could 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 will match. However, if a port is authorized with MAC address A, and an EAPOL packet arrives with a source MAC address of B, then this will trigger a security violation by the switch.

On IOS-Based switches, today, MAB cannot be enabled without 802.1X. This also impacts any potential re-authentication scenario. Technically, re-authentication for MAB does not exist on IOS. So, if re-authentication is enabled (for 802.1X) when a switch port is authenticated via MAB, a switch will send out EAP requests upon re-authentication timer expiry, and only if no response is received, then 802.1X will ultimately relinquish control back to MAB for authorization. Since 802.1X has to timeout again for MAB re-authentication to occur, MAB re-authentication is not recommended. Also, any port configuration involving 802.1X re-authentication is not recommended either. 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.

4.2 MAB and the Guest-VLAN

The Guest-VLAN serves as a failure condition for MAB if configured on the same port as MAB. Else, the failure process for MAB is to continually try and 802.1X authenticate the port again. For IOS-based switches today, this is due primarily to a MAB failure actually causing the port to go into the HELD state, much the same it would as it an 802.1X supplicant had failed authentication. So, once the HELD state completes, 802.1X is attempted again, times out again, and MAB is attempted again. However, since the Guest-VLAN can serve as the failure criteria for MAB if configured along with MAB, then this might provide systemic value. An example of value it could provide is for MAB and the Guest-VLAN to indirectly provide a means to provision credentials in an identity store for MAC addresses that may not be known in advance to the Enterprise. An operation of this is depicted in the following flow diagram:

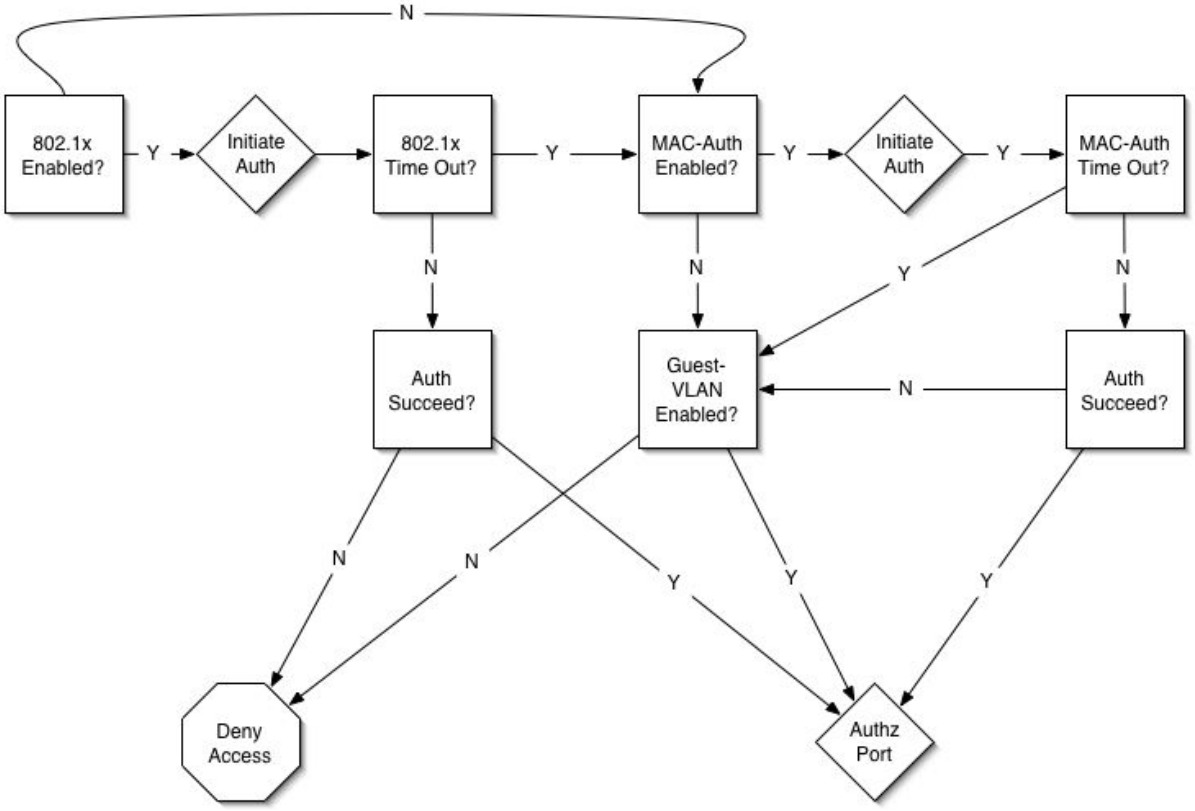


Figure 8: 802.1X, MAB and the Guest-VLAN

The operational nature of feature interaction above 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.

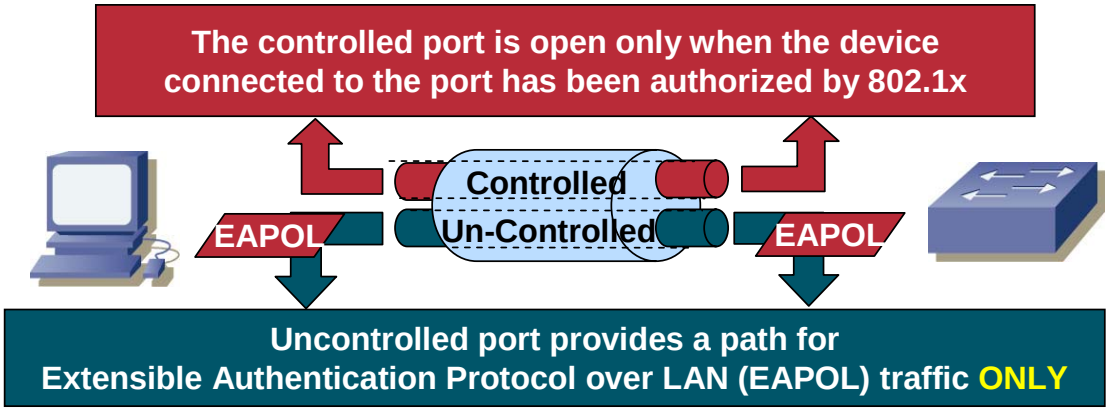
Note: If a port is initially configured for 802.1X with Guest-VLAN, and the port activates in Guest-VLAN, it remains there even though a network administrator enables MAB. The port link status must be flapped to initialize the 802.1X state machine.

4.3 Wake-non-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 forth. By default, 802.1X and

WoL are mutually exclusive. The reason for this is due to 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 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. So, in an effort to interoperate with WoL environments, most Catalyst switches provide unidirectional controlled port functionality as an optional configuration. This configuration is demonstrated below:

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 does represent 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 will now be placed in a forwarding state for any ports that are not yet authorized. Operationally, the controlled port is now only operating in one direction. A WoL magic packet can now exit the network to wake a machine up if need be. It is now the expectation that machine must

then 802.1X authenticate to successfully send traffic into the network. WoL is a per-port feature.

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

So optionally, on a per-port basis, the configuration above represents the notion of a uni-directional controlled port. This means 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 supplicant's job to do 802.1X after getting woken up. Ultimately, this allows for any sort of maintenance, software patching, delivering Email to a machine, etc. 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:

- 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.

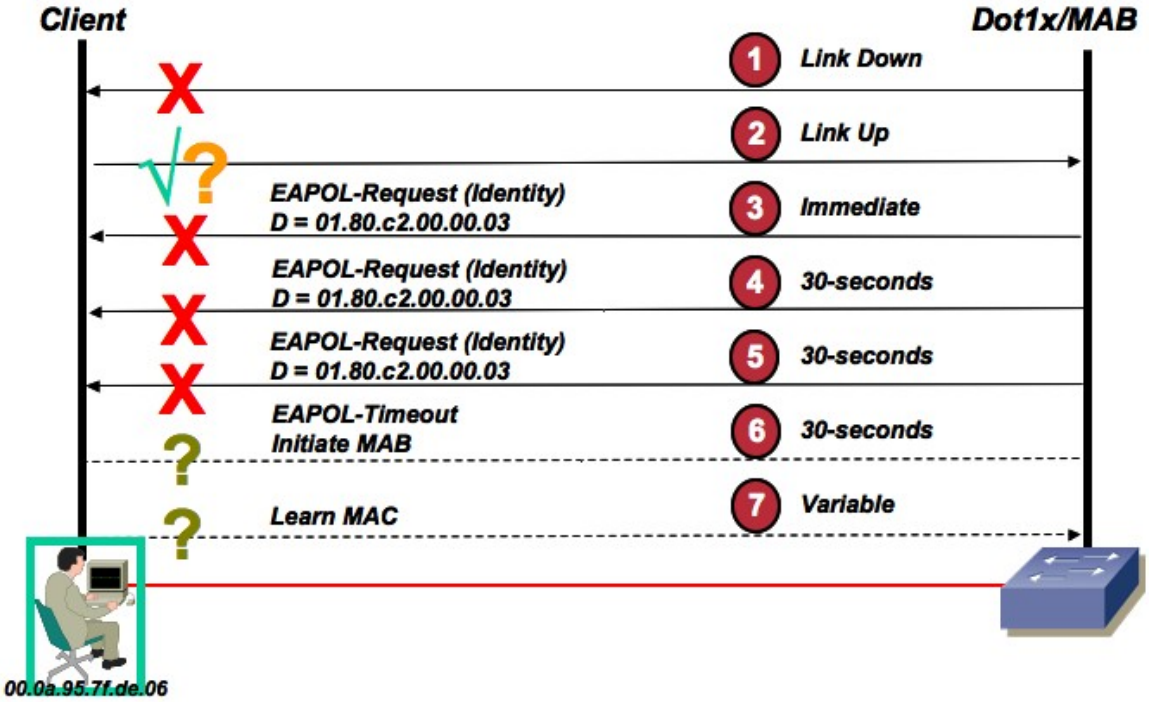
Note: If link goes down, and clients plug directly into the authentication, then an EAPOL-Logoff does not represent much value, since 802.1X/MAB state machines are also directly driven by link state.

A switch port will be 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.

4.3.1 MAB and WoL Interaction

If the MAB is configured, then a port will be nailed up into a MAB state of initiated soon after the original "go to sleep" event. This process is depicted below:

Figure 10: Machine Going Into Power Save Mode with MAB



As demonstrated above, a machine that goes into power save mode with MAB also enabled will bounce link state, and then be 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 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
```

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
```

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, once a machine wakes up, it must MAC Authenticate since 802.1X has already timed out (while the machine was asleep). However, as articulated 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.

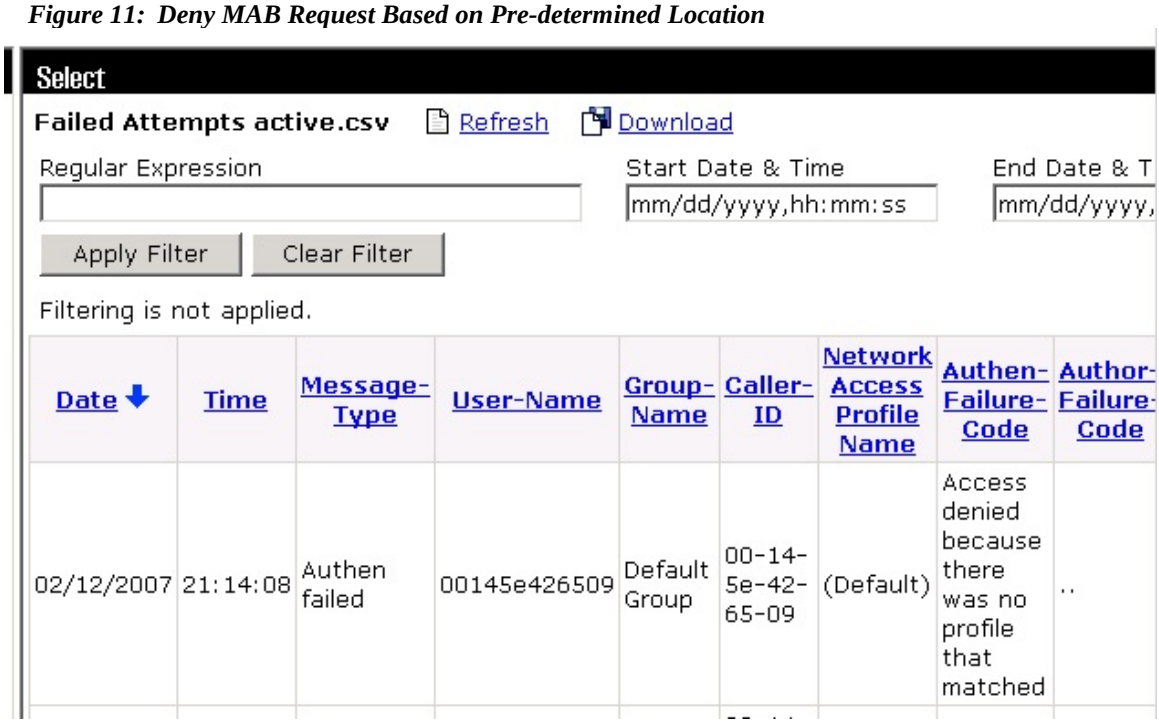
5 MAC Authentication Bypass Opportunities and Benefits

5.1 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 techniques or granular policy enforcement, VLANs can be assigned for granular policy as well. These benefits represent motivations behind the need for MAB. However, there currently is no easy way to have switches authenticate the device and at the same time limit the MAC to a specific location/switch. While this functionality is not currently provided by any turn-key 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 (e.g. robotic are 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 (NAP) can be setup based on the NDG. So, this can cause a MAB request to not match the NAP, since the request may originate from the wrong switch. An end-result of this is demonstrated in the figure below.

Figure 11: Deny MAB Request Based on Pre-determined Location



For more information on Network Access Profiles, please see: 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 complement as a true location-aware based service. However, as 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 demonstrated above due to a specialized need.

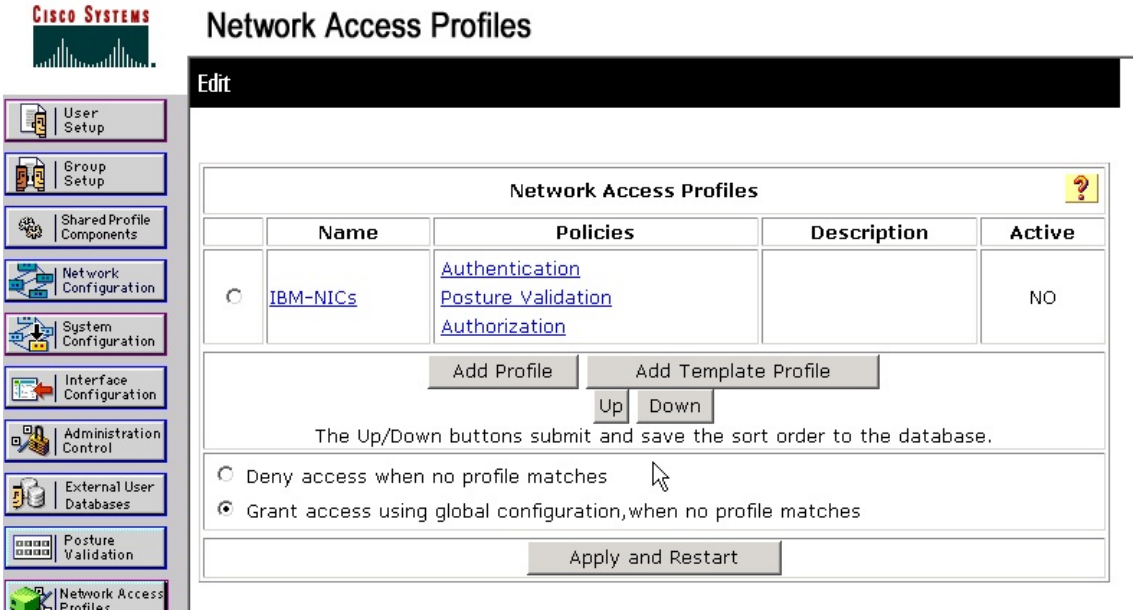
5.2 Network Access Profile Matching and Potential Value

With ACS 4.0, a customer has the ability to match a NAP and also authenticate a MAB request. This technically provides support for the ability to incorporate MAB with a restriction that uses pattern matching to match requests irrespective of the fact that a user request contains a MAC address as the user-name and password. Some customers may not have an existing database of all their MAC addresses. So, as an initial rollout plan, they may be interested in wildcarding MAC addresses based on a vendor code. By

leveraging a NAP, MAB can be made to work on ACS without actually defining a user, by:

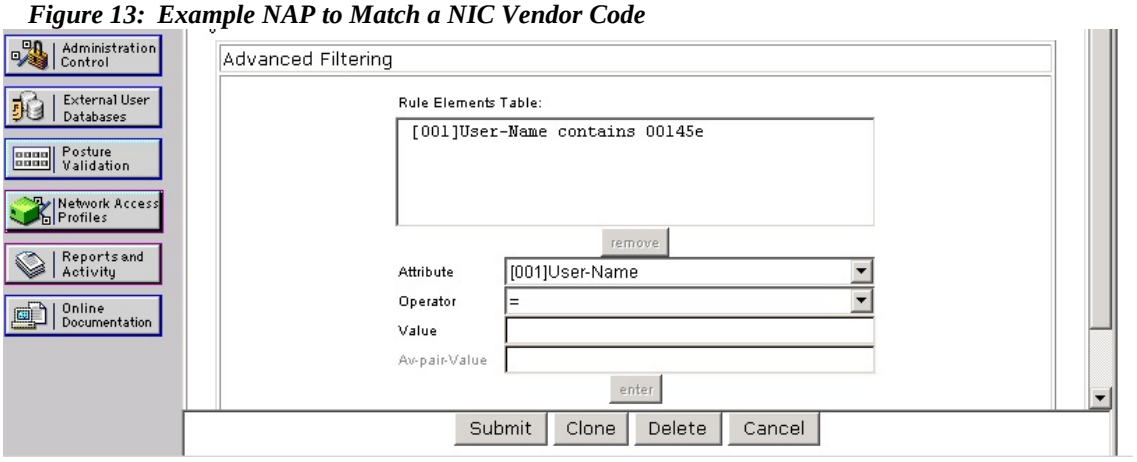
1. Matching RADIUS-Attributes in such a way to distinguish MAB from something else. There are multiple ways to do this. The only key is, by wildcarding, the request can be forced not to go to AD, not enable PEAP, or any other specific authentication settings. An example of this is in the following NAP. First, you can create a NAP to match IBM NICs as an example:

Figure 12: Example NAP to Match a NIC Vendor Code



Then, filtering for the NAP itself should be setup. Here is an example for wildcarding a vendor code in the user-name attribute:

Figure 13: Example NAP to Match a NIC Vendor Code

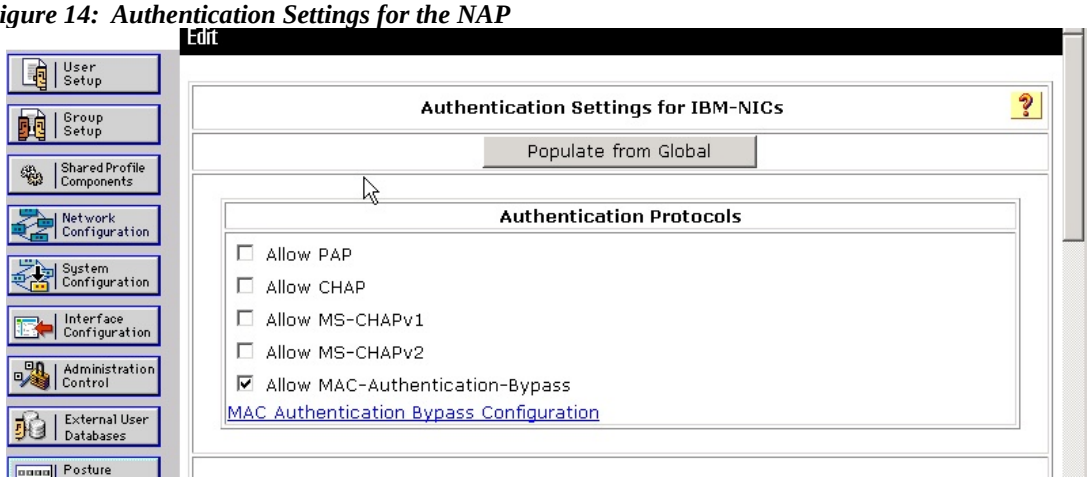


This has now effectively wildcarded the MAB request itself. In other words, only

MAC addresses matching the criteria will be processed further. Else, the MAB request does not match the NAP, and can be implicitly denied or fallback to global authentication settings.

2. Authentication rules should then be setup. No special authentication configuration is needed, although the tick box for MAC-Auth-Bypass must be enabled. Optionally, ACS can also be configured to place all of the MAC addresses that have matched the NAP per #1 above into a specific CiscoSecure Group. This is demonstrated below:

Figure 14: Authentication Settings for the NAP



3. Posture rules for this NAP can be disabled, since this demonstration has nothing to do with NAC.

4. Authorization rules for the NAP can be configured. The group referenced from #2 above could then also deploy some sort of temporary access, like the dynamic assignment of the Guest-VLAN that may already be on this switch. MAC addresses that do not match the NAP to begin with can be implicitly denied.

The NAP can then be successfully verified much like the example in Figure 11 above.

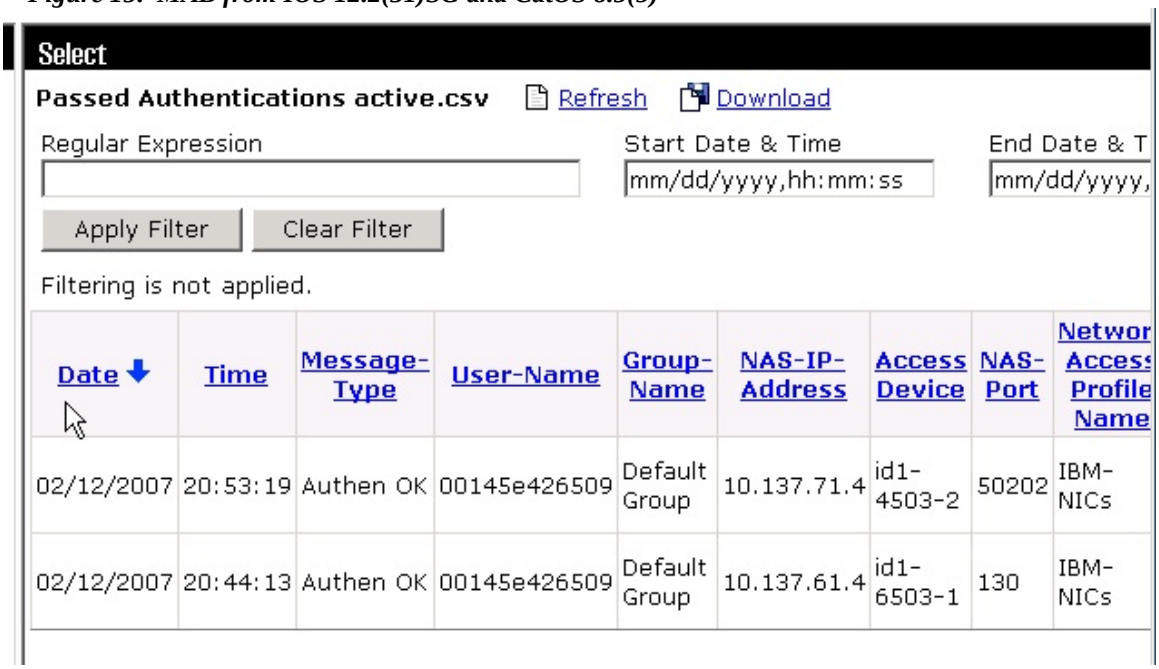
In summary, this is a demonstration of being able to wildcard a MAC address based on a vendor code or specific range of MAC addresses. At the same time, you have granted access (optionally, this access can be differentiated from full access) via MAB, although the MAC address may not have been known previously. This technique could be leveraged to help build a true identity database of MAC addresses. Also, this technique could then be incorporated into a captive portal, for example, that an end-user could interact with for registration purposes. Examples of this include home grown applications a university may use to allow students to register their device for identification purposes onto the network in the absence of endpoint control. This is a demonstration of how MAB can be used to address TCO concerns of Enterprises in endpoint management. This way, a network or server administrator need not play a part in the provisioning mechanism that may already be in place, even in the case of where a MAC address may

not be known previously. Also, the network access ultimately granted to the device follows the device wherever in the network the device plugs in through pervasive path isolation techniques.

5.3 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 will contain 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" i.e. an all lower case version of "hhhh.hhhh.hhhh" with the punctuation stripped out. So if an identity infrastructure is to be built to support MAB, it should follow this format. The following figure represents passed authentications on ACS from an IOS-based switch and a CatOS switch running MAB:

Figure 15: MAB from IOS 12.2(31)SG and CatOS 8.5(5)



Note: Although not examined here, WLCs utilize 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 a "hh-hh-hh-h-hh-hh" format. ACS could not handle this if the user account is defined like the above for IOS. The following figure represents a passed authentication on ACS from CatOS with 8.5(4) demonstrating this condition:

Figure 16: 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 different types of switches, then 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 since a single MAC need only be defined in a single location, while multiple authenticators can use it in the same format.

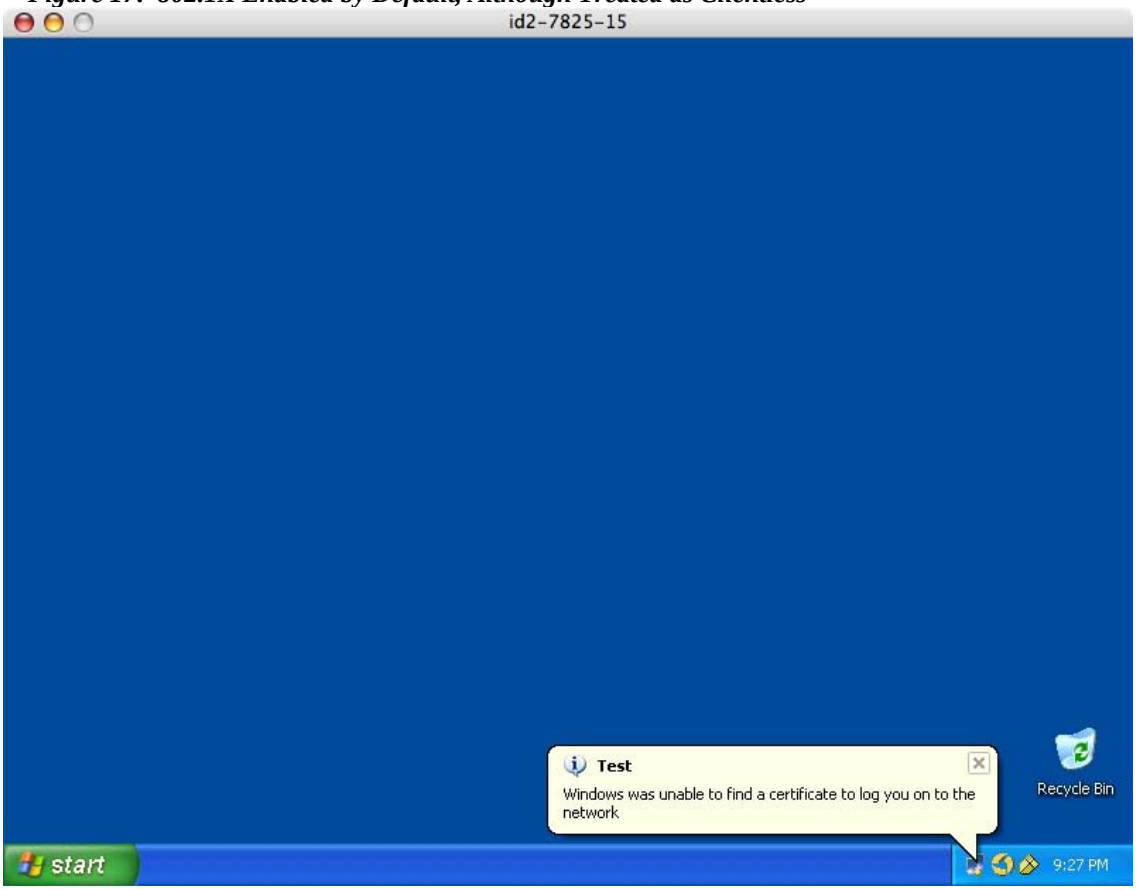
Note: ACS can not currently handle devices defined in a non-case sensitive fashion. Any user account entered must take this into account also; else ACS processing will fail completely. For more details, please reference CSCsd45762.

5.4 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 will typically be enabled for 802.1X by default. Also, 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. What this means is when link comes up, the switch will begin an 802.1X authentication event by transmitting an EAPOL-Identity-Request packets on the wire. However, even though the PC is 802.1X-enabled, the supplicant is also enabled for EAP-TLS and the machine "will know" it does not have a certificate for either the machine or any user that happens to be logged into it. Operationally, a balloon message will appear in the system tray at this point with "Windows was not able to find a certificate to log you onto the network". Since a certificate is not on the device at all, Windows will not speak EAPOL to the switch. Also, since the supplicant never sent the switch an EAPOL-Start, the switch has no way of knowing the device is actually 802.1X capable either. So, this means a brand new machine can be initially deployed into the Guest-VLAN, or if the MAC address is known

prior to 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 fallback method even though 802.1X is technically enabled on the client. An example of this is on the end station is demonstrated below:

Figure 17: 802.1X Enabled by Default, Although Treated as Clientless



The client, having no certificate provisioned prior to this event, will not reply to this request at all, and demonstrate 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.

Best Practice

- 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 (like any needed certificates).

6 MAC Authentication Bypass Limitations and Challenges

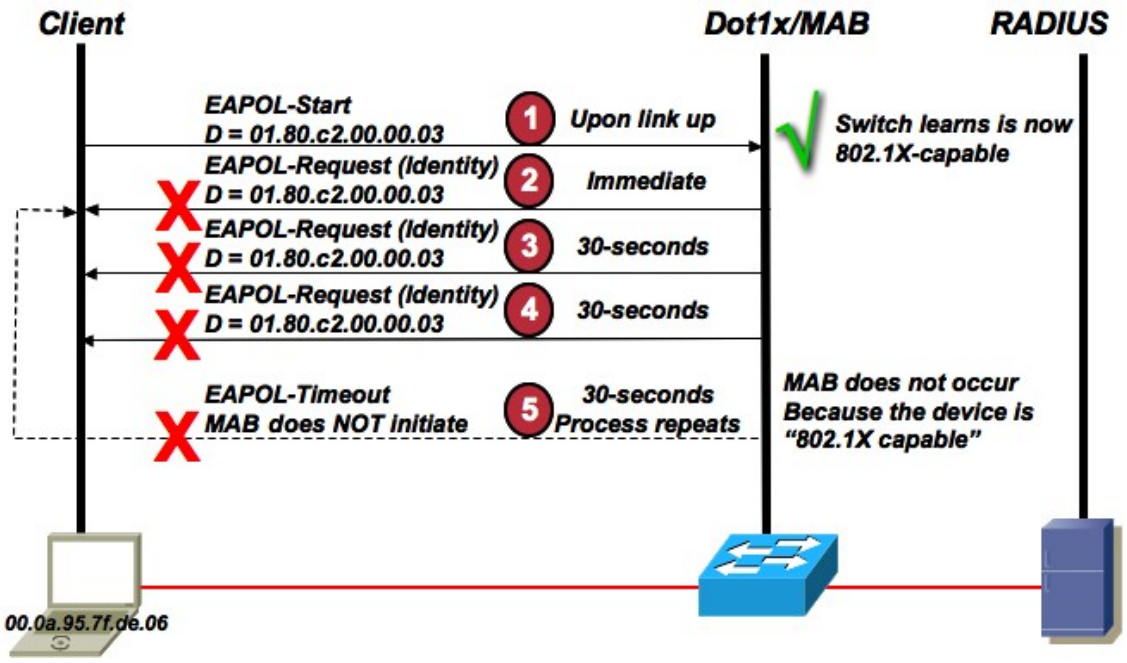
6.1 Fallback Technique for Re-imaged Machines

If a supplicant like 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, then 802.1X will initiate and the 802.1X process will timeout. Also, this process will continue persistently and MAB will never execute. This is because the client sent an EAPOL-Start frame to the switch initially, and a switch uses EAPOL to determine if 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 due to re-imaged machines. 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 is demonstrated in the following figure.

Figure 18: 802.1X and EAPOL-Starts enabled without credentials



This would be the same behavior if any other supplicant sent an EAPOL-Start and a user-screen was displayed to the user to input credentials for a challenge-response based EAP type like PEAP. If the user does not respond to the credential notification, 802.1X will timeout and repeat transitively, and MAB will not be initiated for these types of cases either. Once more, anytime a switch knows an 802.1X supplicant is on the wire through the device speaking EAPOL, MAB or the Guest-VLAN can typically not 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 3K switches, the command is **dot1x guest-vlan supplicant**. Also, 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 like the above is encountered. There is no such workaround available in CatOS.

6.2 Network Admission Control (NAC)

Another systemic challenge with MAB has been caused by the Network Admission Control (NAC) program. The base functionality for MAC address authentication was already in place on ACS. For example, wireless APs and Controllers have the ability to initiate a PAP authentication with a RADIUS server using a client's MAC Address as a username/password.

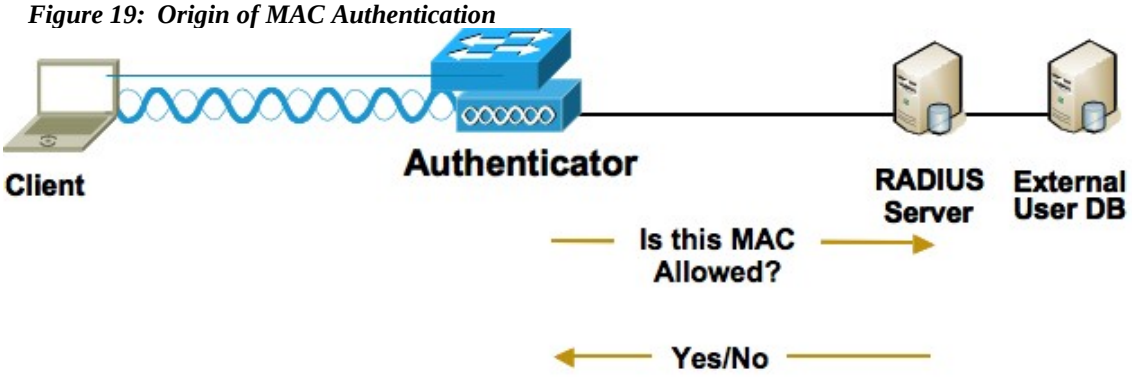
Note: For more information about MAC-Auth for WLAN devices, please see:

http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag/123-04.JA/1100/h_ap_howto_5.htm

The second part of the document is for local authentication (AP as a AAA server). A customer would just need to change from authentication local to ACS server.

Literally, the wireless implementation of MAC address checking achieves the following:

Figure 19: Origin of MAC Authentication

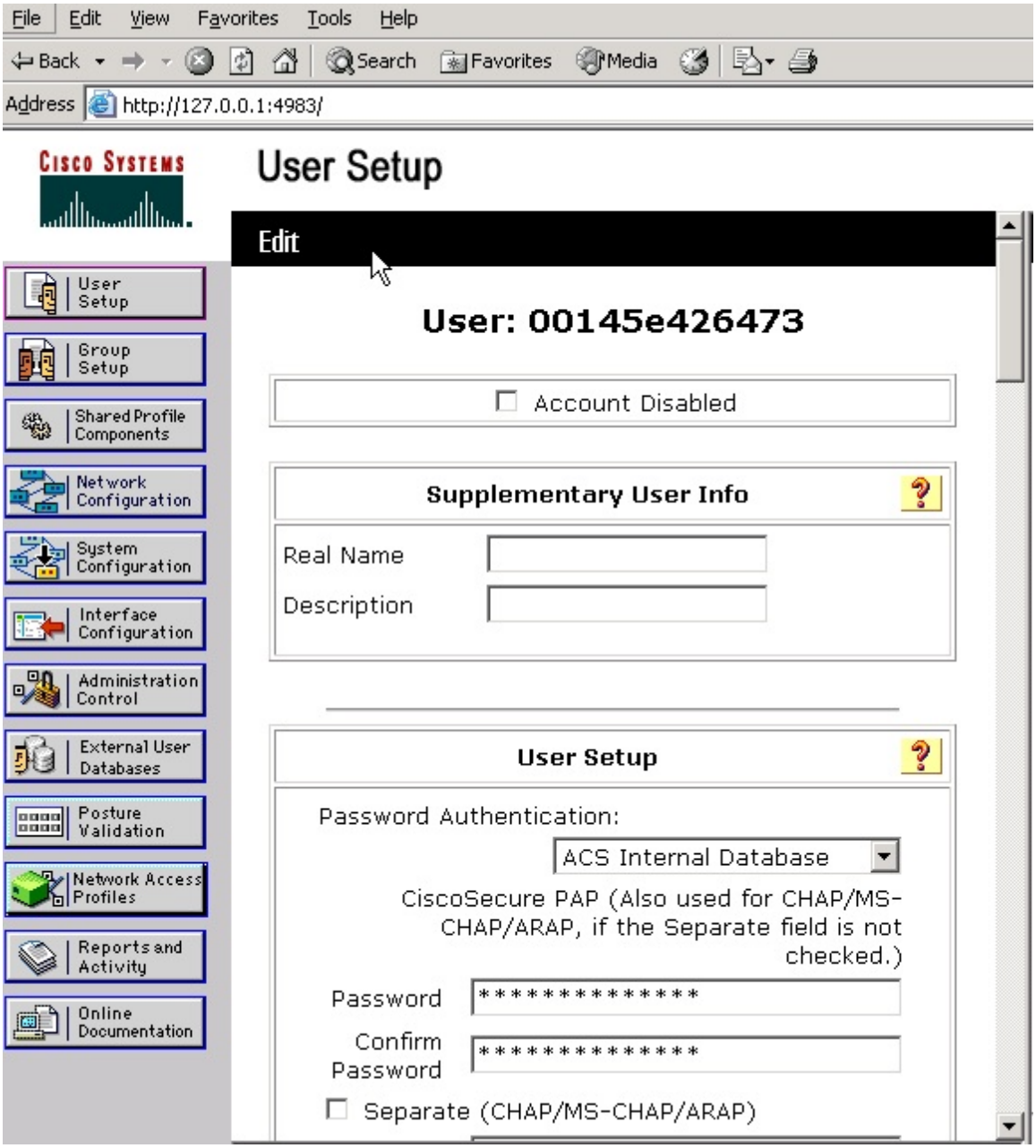


In an open authentication environment (or separate SSID) wireless can at least check the MAC address of an end client as an admission criteria. Due to client ubiquity, these topologies are no longer typically recommended. APs can accomplish this based on the fact that initial associations have already been made (and based on that association traffic to/from a Wireless NIC is blocked by the AP).

No such associations exist currently in the wired space, but MAB on switches provides a wired equivalent to the wireless functionality of checking a MAC address after an 802.1X timeout on a port. MAB on a wired switch is geared to satisfy the same requirement as the previous wireless example. The only difference being that on IOS-based switches, 802.1X must time-out before MAB is attempted as a supplemental authentication method. This means MAB can be used to deal with devices from an exception point of view only. In other words, MAB is used as an authentication technique in the absence of a preferred credential. This is NOT to be used as an exemption criteria, or be ways to deal with failures of the preferred credentials.

MAB is primarily geared toward providing a supplemental authentication technique in the absence of 802.1X. It should be simple by default. A minimum requirement is a database full of MAC addresses. Any database (including the local CiscoSecure database on ACS) supporting authentication requests to RADIUS via PAP or EAP-MD5 can suffice. MAC addresses can be imported/configured, with no punctuation, white spaces, or capitalization in these databases. This really just means the ability to authenticate a MAC as the username/password combo from a RADIUS (PAP or EAP-MD5) exchange. It was designed to be no different than what APs have been doing on this for years. An example of a single user account being added to a local CiscoSecure database as a MAC address is demonstrated below:

Figure 20: User Account being added to the ACS Local Database



Assuming default use cases, this configuration should allow ACS to successfully authenticate devices as switches proxy the authentication requests on behalf of the devices via MAB. No other special configuration is required to enable MAB for ACS other than what may have already been enabled for 802.1X. This functionality should work for ACS revisions back to ACS v2.6. For more information please see the following:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/csacs4nt/csnt26/usergd26/userdb.htm#xtocid171961

The service-type attribute from RADIUS is set in a MAB request from all switches and wireless AP/Controllers as well. The calling-station-ID (CLID) attribute is also set to the MAC address in a MAB request from a switch and WLAN Controller. However, NAC technologies like EAPoUDP (EOU) attempt to use the service-type attribute as a means to differentiate a request to be able to do agentless processing, and in some cases, spawn audit functionality. For more information, see the following NAC documentation for ACS:

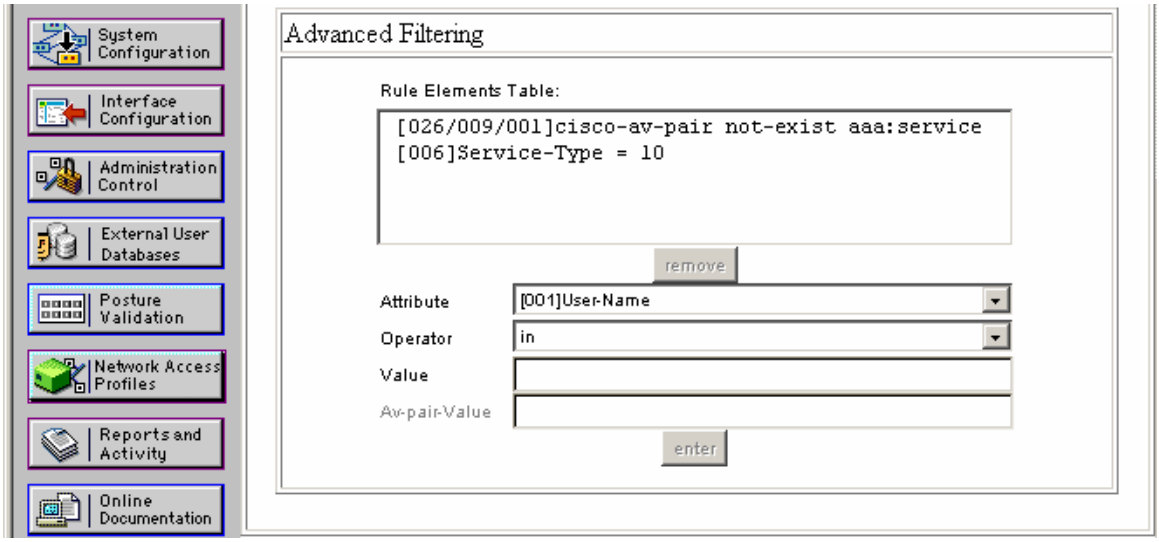
http://www.cisco.com/en/US/partner/products/sw/secursw/ps2086/products_user_guide_chapter09186a00806fe212.html#wp137843

http://www.cisco.com/en/US/partner/products/sw/secursw/ps2086/products_user_guide_chapter09186a008052e984.html#wp1155767

From the documentation above, the service-type attribute is critical to enable Network Access Profiles (NAP) on ACS for NAC, and is critical for service-type processing. This works fine for EOU requests. Also, EOU requests (agentless or otherwise) do not contain user or identity information based on today's operational behavior of EOU on all switches and routers. However, by default, both service-type and CLID attribute can effectively be irrelevant for MAB itself as the authentication event described in this document. The authorization based on the authentication event is rather typical as well, since it is expected to be user-name/password based. The bad news here is that the original ability to deal with a NAC agentless host for EOU was also termed "MAC Authentication Bypass", and the reasoning behind this is unclear.

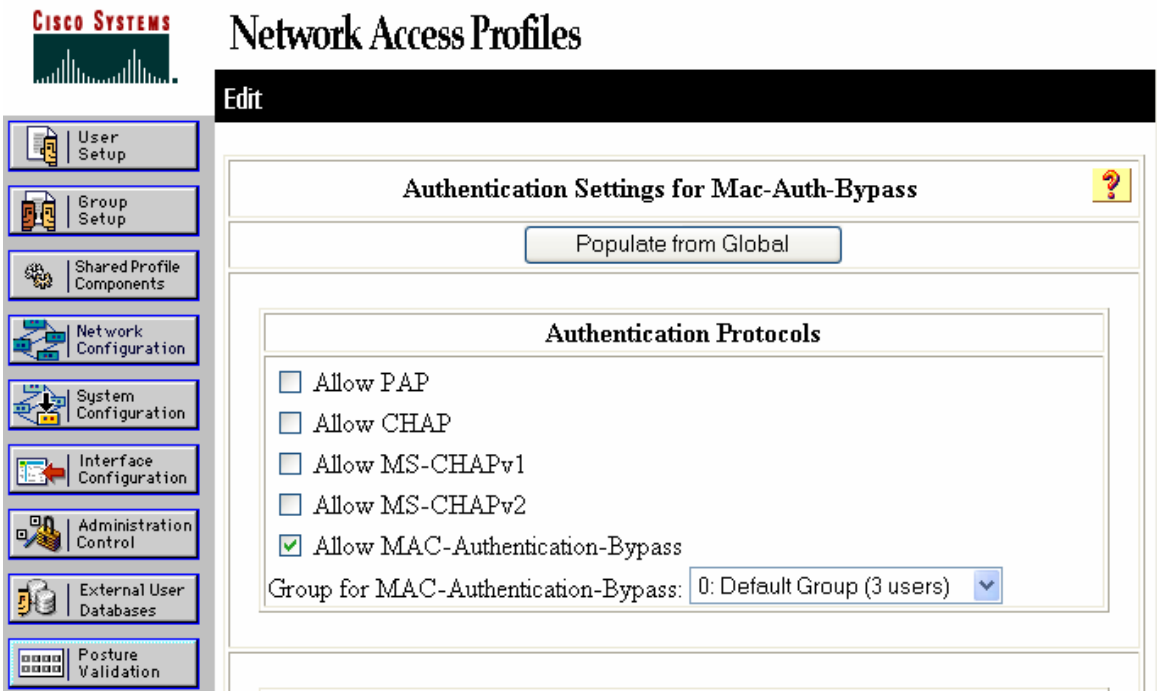
Per the documentation references above, a Network Access Profile that matches a RADIUS request should be setup and configured. It is expected that customers create a new NAP to provide the accurate selection of the service profile for "MAC Authentication Bypass". In the following example, a NAP with the name "MAC-Auth-Bypass" is created as demonstrated in the NAC configuration guide. However, any MAB request from the switch via RADIUS also has a service-type value of 10 just like NAC-L2-IP, but does not have a Cisco Attribute Value Pair (AVP) containing the keywords service (which is a VSA sent by EOU). This means any MAB request will match this NAP, whether intentional or not. Depicted below within the "Advanced Filtering" box are two entries; no match for "aaa:service" in the Cisco AVP and the "Service-Type" equal to 10.

Figure 21: NAP → Advanced Filtering



Once this advanced filtering is defined, the authentication settings of the NAP must be changed (in order to enable "MAC Authentication Bypass" and a group or a pre-configured Network Access Restriction (NAR) for authorization purposes should then be selected. In the Authentication settings for a NAP, the tick box next to "Allow MAC-Authentication-Bypass" should be selected, in addition to the selection of a group that has the correct NAR, in this example, the group default Group.

Figure 22: Assigning "MAC-Authentication-Bypass" to a NAP



No other settings need to be changed in the NAP for "MAC Authentication Bypass" to function properly. However, this functionality **IS NOT** MAB as previously

demonstrated. It is effectively another feature entirely, since it looks only at the Calling-Station-ID field from RADIUS or authorization purposes.

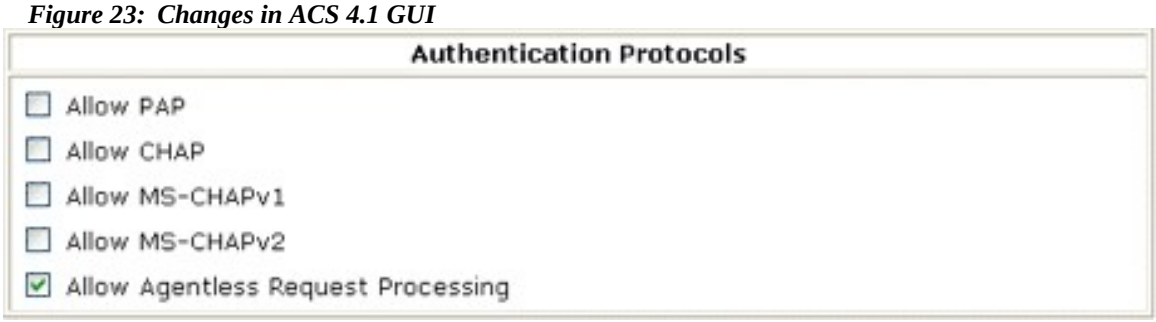
The difference in this processing model is that MAB may be attempted at any time as exemption criteria from any control plane the method is applied to. In other words, MAB could be used as an authentication technique irrespective of the existence of a preferred credential.

This functionality is potentially valuable in NAC use cases to support exemptions for EOU or agentless requests from EOU. An example could be that an EOU request for NAC posture can be completely bypassed even if Cisco trust Agent (CTA) is loaded on a device, simply due to the MAC address of the end-station. Alternatively, once EOU times out on a port in an effort to search for the CTA for NAC, then an agentless request is made with no user information, and the expectation is the CLID field can be interrogated to determine the device access, or to start a NAC audit for the device in question. However, this again, is not the same feature set or functional goal of MAB as described in this document.

For more information, please see the Configuration Guide for NAC:
http://www.cisco.com/application/pdf/en/us/guest/netsol/ns617/c649/cdccont_0900aecd8040bbd8.pdf

Note: Although ACS 4.1 is not within the scope of this document, the user interface was demonstrated above changed in the 4.1 release. It is now appears in the GUI as: "Agentless Request Processing" instead of “Allow MAC-Authentication-Bypass” under the NAP protocols page as see in the following diagram:

Figure 23: Changes in ACS 4.1 GUI



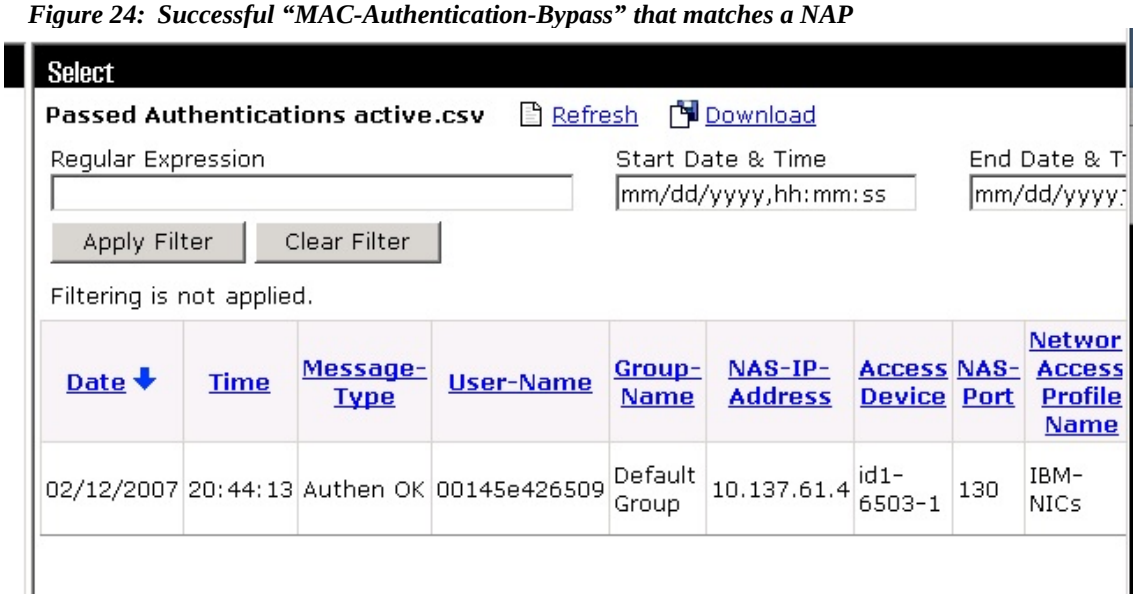
The GUI changes should at least help alleviate confusion in this respect. NAPs are not needed to configure MAB. So, when doing MAC address based exceptions like this for NAC, ACS is looking at the Calling-Station-ID attribute from RADIUS, and not the username in the RADIUS request.

As a reminder, this functionality for NAC has nothing to do with 802.1X either. MAB deals with exceptions as a supplemental authentication technique only. The functionality described here demonstrates exemptions to the control plane, which in most NAC cases is

for EOU. With respect to EAP, RFC 3579 also delivers guidance toward this. A switch may be configured to initiate with a default authentication method. For 802.1X, this usually occurs when the system agrees on what type of EAP method should be used to authenticate the device as part of the beginning of the session. In other words, the first RADIUS-Access-Request from could be sent from the switch with a service-type equal to call-check before EAP is even established to determine the identity through another means. So, it is technically feasible to achieve exemptions for 802.1X today through as a similar method as the one demonstrated here for NAC and EOU, although none of our switches do this today for 802.1X. There are good reasons not to do this either, in the interest of security, since MAC addresses are easily discovered and/or spoofed.

This issue is further complicated by the authentication processing of authentication requests of ACS v4.0. First, with ACS v4.0, if a NAP is NOT matched, a MAB session will still be able to authenticate successfully as long as there is an actual user account in the database as demonstrated before the NAC complications here were introduced. The reason for this is that ACS will revert to “legacy” behavior for any session that does not match a NAP or the specific authentication settings in a NAP. “Legacy” behavior is also attempted if a NAP is not defined at all, of course, although this is available as a configurable condition. However, today with ACS 4.0, a NAP can actually be matched, and both behaviors can actually be made to work. The reason for this is actually due to a bug in ACS processing, since only the CLID field should be interrogated for authorization purposes. This can also be verified in a test such as one demonstrated below:

Figure 24: Successful “MAC-Authentication-Bypass” that matches a NAP



As seen from the figure above, this is working “the legacy way” because the switch sends "00-14-5E-42-65-09” as the format of the Calling-Station-Id while the format of the user-name from the switch is "00145e426509" and is what is in the passed authentications log.

Also, the only other configuration in ACS for this case was a user account equal to “00145e426509".

In summary, there is an inconsistent understanding of what “MAC Authentication Bypass” even is with respect to NAC deployments. Also, unless NAC is needed for EOU, all of the configuration demonstrated in this section may not be needed. It has been demonstrated that a PAP request for MAB can be successful even though a NAP was matched. This is technically a bug in ACS v4.0 since it should not work.

- As a best practice, if customers only need MAB to work (like as a URT replacement, or supplemental authentication technique for 802.1X), and they are not interested in NAC yet, it is not recommended by default to attempt to use a NAP for this in ACS.

6.3 MAB EAP Option on ACS

A wired IOS-based switch also has an EAP option available for MAB. It is configured via the following configuration example:

Cisco IOS

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

With the configuration above, the RADIUS exchange for MAB is not carried out in a single data exchange like it does with PAP. It is carried out in a challenge response mechanism with EAP-MD5 and makes a round trip for RADIUS before succeeding. The username and password for the request remain the MAC address. The MD5 option may be used in conditions where PAP does not meet a security policy, or PAP may have been disabled on the backend database server.

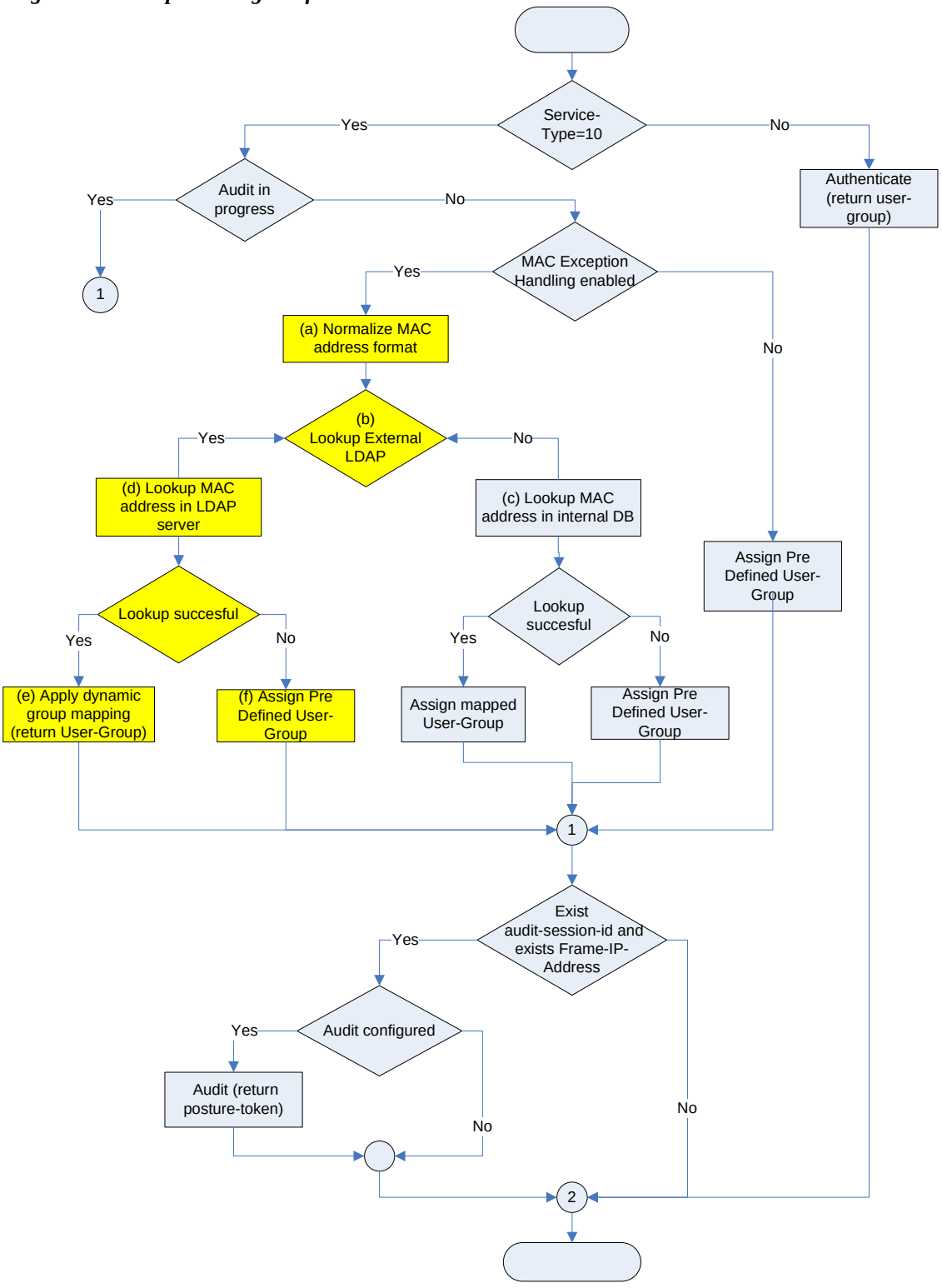
However, if you configure the same process as indicated in the prior section for ACS and a NAP, this does not work. In this condition, MAB neither passes or fails, and is the anatomy of CSCse12673. Hence, if EAP-MD5 is needed on switches for MAB for any reason, this ACS configuration and the one for NAC are not recommended.

6.4 ACS 4.1

No form of MAB as explained in this document can currently be used with ACS 4.1. This includes all wired switches that run MAB as a supplemental authentication technique

described here, or for any MAC Authentication done by wireless APs/Controllers. The reason for this is due to a change in processing for ACS 4.1 to better handle NAC and auditing capability. Please refer to the following diagram:

Figure 25: ACS processing as of ACS 4.1



As you can see above, authentication cannot be performed if the service-type attribute is set. It is unclear what the purpose of an “Audit in Progress” even means, especially if the customer is not running NAC. There are more considerations not included in this document related to these changes for ACS 4.1. Although ACS 4.1 is not within the scope of this document, MAB cannot be used for this ACS revision at all, and is not recommended. CSCsh62641 and CSCsh63236 have been filed to account for this, however. The current plan is to fix these issues in ACS v4.1.3 (FCS by end of April 2007).

6.5 Provisioning

Provisioning is also a service of high concern to customers. A customer may not know what there MAC addresses are in advance. No turn-key solution is provided by Cisco to fill this void either. Third-party products that provide asset management capabilities may help in this regard. These products include those from Great Bay Software, Altiris, etc. Some customers have attempted to integrate learning techniques with their directory infrastructure though. 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, 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. Also, 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 winds up in the Guest-VLAN. MAB does not necessarily change this, by the Guest-VLAN serving as a failure condition for MAB itself. So ultimately, a device can get into the Guest-VLAN much the same as it does without MAB, since it’s incapable of 802.1X. However, if MAB fails “in the middle” then a failure of this event should be recorded on the AAA server. An example from ACS on this failure is indicated below:

Figure 26: MAB Failure

As you can see above, 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: More in-depth guidance on Identity Management should ideally be provided here, but is not within the scope of this paper.

As a reminder, the gathering of MAC addresses, however, does not extend trust explicitly. LMS from CiscoWorks can also help as a MAC address gathering tool. It also does device name, IP address and host name correction. However, none of these techniques necessarily insure the entity should be on the corporate network to begin with. It 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.

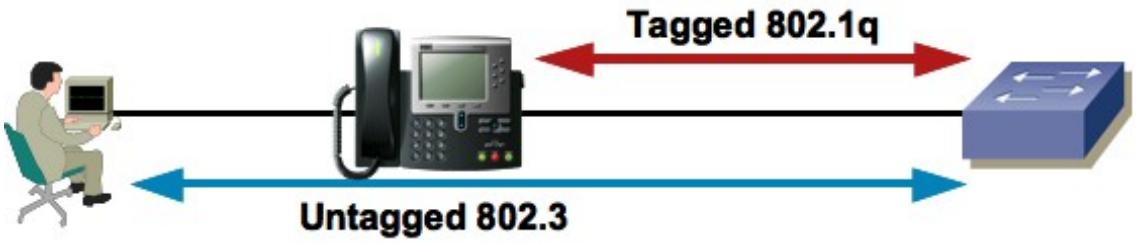
6.6 Lack of Existing Identity Store

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 only be located 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 ACS can already hook into. The identity store can be added onto though. 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.

6.7 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 below.

Figure 27: 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
 panning-tree portfast
```

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
```

So, while 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 a phone's MAC is provisioned in ACS and it sends out packets, the switch will be 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 will not know or pre-suppose the type of device and will not know to put it on the voice VLAN as part of the authentication event though. So, if the customer provisioned the phone to tag its packets on the voice VLAN, it will fail as of today, because we explicitly ignore traffic on voice VLANs for MAB. Therefore, a customer cannot use MAB to attempt to authenticate a third-party phone either. 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 since single-auth mode would not allow any other MAC on the wire like a client plugging into a phone. In essence, MAB shares the same rules in this space that 802.1X does. For more details on how 802.1X interoperates with voice, today, please see the following:

http://wwwin-eng.cisco.com/pcb/edcs/getedcs.cgi?SEARCHTYPE=CONTENT&KEY=DOCNO&VALUE=EDCS-394895

As alluded to in the documentation reference above, MAB can be enabled for data devices and Cisco telephony devices can be ignored with CDP. However, in similar nature to 802.1X, any MAB authenticated session may disappear from the network, and the network may not know 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. Also, for any new machines that plug into the phone, a security violation may be tripped, since the phone will think another MAC has appeared on the wire other than the one it has authenticated. Catalyst 3K switches recently delivered a MAB aging feature to address this in 12.2(35)SE, but could not be verified for this document.

Further integration with IP Communications is planned for Multi-Domain-Auth (MDA) and MAB aging. MDA is a new solution-based feature set which 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 3K switches in 12.2(35)SE, and 4500 switches in 12.2(37)SG.

6.8 MAC Movement

Like 802.1X, a MAC address authenticated by MAB is not allowed to move on a switch unless the port the device moved from is unauthorized. This issue is exacerbated by the MAB aging issue introduced in the previous section with respect to IP telephony. So, if a device is authenticated via MAB behind a phone and then moves to another port on the same switch, the port the user moved to will go down. This will render 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 will not help in this case. This violation behavior handling would only help 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 since the move will drop 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 recommended design. Also, in CatOS today, a port must be manually reset when this event occurs. There is no auto reset after a configurable interval.

7 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. There is no special configuration on a switch needed to achieve dynamic VLAN assignment.

Standard recommendations for 802.1X with VLAN assignment remain with MAB. It 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
```

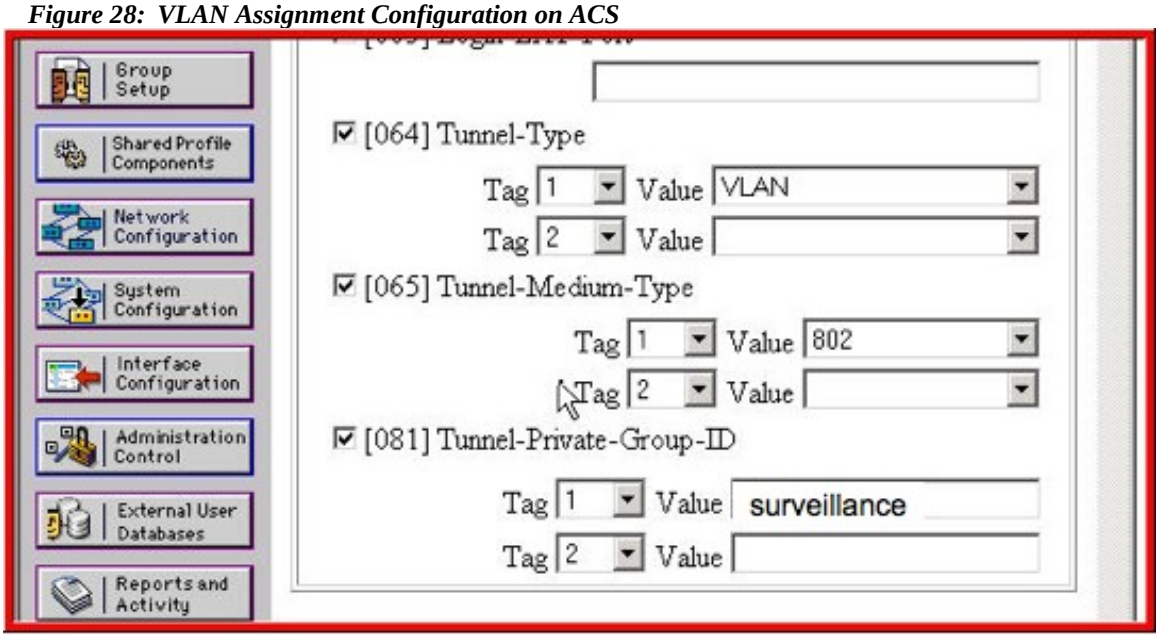
Nothing is needed on the ACS server, outside of what may already be in place for 802.1X as well. What is required are threee standard RADIUS attributes defined by RFC 2868:

```
[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 checkboxes 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.

The following diagram represents an example of configuring a certain group of devices for MAB to be deployed into the "surveillance" VLAN:

Figure 28: VLAN Assignment Configuration on ACS



This will enable 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 will deny 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, etc.) 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 will utilize separate VLANs as entrance criteria into each separate network partition. With wireless, you may also optionally insure 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, IBNS techniques were incapable of leveraging this differentiation capability. IBNS 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 in the arsenal, this can now be expanded to include differentiated services between robotic arms on a factory floor vs. X-Ray machines in a hospital vs. IP-enabled surveillance devices vs. standard corporate PCs. This increases the end-to-end impact IBNS provides with this additional, fine-grained access-control.

8 Summary

With the increasing demands upon today's 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. One of the challenges in implementing IEEE 802.1X, however, is the requirement to support yesterday's cutting edge which is now today's 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 included network architecture with 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 devices' 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 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 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 will not / cannot do 802.1X, but who have also bought into port-security with configured MAC addresses more options, and provides customer running URT or VMPS technologies, a migration path. MAB also works with any standard RADIUS server, with a default timeout of 30-seconds with three retries. This means 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.