



Local WebAuth Deployment Guide

Last Updated: March 14, 2012



Cisco
Validated
Design



Building Architectures to Solve Business Problems

About Cisco Validated Design (CVD) Program

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

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at <http://www.cisco.com/go/trademarks>. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Local WebAuth Deployment Guide

© 2011 Cisco Systems, Inc. All rights reserved.



Local WebAuth Deployment Guide

WebAuth is a convenient, well-understood method for authenticating end users. This document describes WebAuth network design considerations, outlines a framework for implementation, and provides step-by-step procedures for configuration. This document includes the following sections:

- [WebAuth Overview, page 3](#)
- [WebAuth Solution Components, page 6](#)
- [WebAuth Feature Interaction, page 17](#)
- [Configuring Web Authentication, page 25](#)
- [Configuring the Switch, page 34](#)
- [Monitoring and Troubleshooting Web Authentication, page 43](#)
- [References, page 45](#)
- [Sample Customizable Pages, page 45](#)
- [References, page 47](#)

WebAuth Overview

This section introduces WebAuth and includes the following topics:

- [What is WebAuth?, page 4](#)
- [WebAuth Benefits, page 4](#)
- [WebAuth Limitations, page 5](#)
- [WebAuth Limitations, page 5](#)



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

Copyright © 2011 Cisco Systems, Inc. All rights reserved.

What is WebAuth?

WebAuth enables network administrators to control network access and enforce policy based on the authenticated identity of a user. WebAuth helps prevent unauthorized access yet still enables network access for end hosts that do not support IEEE 802.1X authentication. When combined with other security elements such as infrastructure protection, threat identification and mitigation, and secure connectivity, WebAuth increases the ability of the network to defend itself.

In today's diverse workplaces, partners, consultants, contractors, and even guests need to access network resources over the same LAN connections as regular employees. Although IEEE 802.1X authentication secures the internal network by requiring employees to present valid credentials before accessing the network, provision must be made for users without IEEE 802.1X supplicants.

When used as a fallback mechanism to IEEE 802.1X, web authentication (WebAuth) provides supplemental authentication while maintaining the benefits of an IEEE 802.1X-protected network. IEEE 802.1X is a secure, standards-based, Layer 2 authentication mechanism. Because the switch first attempts IEEE 802.1X authentication, end hosts with IEEE 802.1X supplicants are subjected to a highly secure authentication procedure while also taking advantage of IEEE 802.1X-enabled features such as secure, standards-based authentication, dynamic VLAN assignment, Microsoft Windows machine authentication, and user authentication that is transparent to the user.

When the switch determines that the end host does not have an IEEE 802.1X supplicant or does not have valid credentials, the switch can fall back to WebAuth. WebAuth authenticates the user at the access edge by providing a web-based login page on which the user can enter credentials. After the user is identified, the user's identity can be employed by mapping identities to policies that grant or deny granular network access.

WebAuth Benefits

WebAuth offers the following benefits:

- **Clientless authentication**—WebAuth does not require the end user to have any special client software. Any host with a browser can authenticate with WebAuth.

The ubiquity of browsers helps ensure that most users can use WebAuth. This aspect of WebAuth allows contractors, vendors, or others with unmanaged devices to get access to the network without having to install new software on their PCs.

- **Familiarity**—Because WebAuth is widely deployed (in public hotspots, hotel rooms, and so on), end users are familiar with the process of entering credentials in web-based login pages.
- **Ubiquitous port configuration**—When using WebAuth as a supplement to IEEE 802.1X, administrators can configure every port in the network the same way without having to know in advance the type of device or user that will be connected to that port.

Employees, partners, contractors, and guests can plug into the same wired port and dynamically acquire identity-based granular access through different authentication methods. The ubiquity of the configuration and policy deployment allows device mobility and faster, more efficient rollouts in heterogeneous environments.

- **Visibility**—WebAuth provides greater visibility into the network because the authentication process provides a way to link the user's name with an IP address, MAC address, switch, and port. This visibility is useful for security audits, network forensics, network use statistics, and troubleshooting.
- **Customization**—The current implementation of WebAuth in Cisco Catalyst switches enables network administrators to customize all the web pages (login, success, failure, and expired) needed for the authentication process.

Customizable web pages enable administrators to give these pages the look and feel of their organizations.

WebAuth Limitations

Although WebAuth is a convenient mechanism for user authentication on unmanaged devices, it has a number of limitations that restrict its use, including the following:

- **Security**—IEEE 802.1X is the strongest method for authentication and should be used for managed assets that support an IEEE 802.1X supplicant. IEEE 802.1X acts at Layer 2 in the network. WebAuth is a weaker, password-based form of authentication that works at Layer 3.
- **Transparency**—WebAuth is not transparent to the end user. To access network resources, the user must first launch a web browser.
- **Lack of single sign-on**—WebAuth requires that the end user enter credentials on a web login page. This login is in addition to any other logins that the end user performs (such as Microsoft Windows login). Thus, the end user could end up entering credentials multiple times.

For temporary access (for example, guest, partner, contractor, or new machine), the need for multiple logins is typically well understood. However, for employees and managed assets, the capability to perform single sign-on with strong credentials is often considered essential to productivity. In the latter case, IEEE 802.1X is more appropriate than WebAuth because most IEEE 802.1X supplicants (clients) can be configured to reuse previously entered credentials (such as Microsoft Windows Active Directory–based passwords and X.509 certificates).

- **Device authentication**—Because WebAuth requires a user to enter credentials on a web page, it cannot be used for machine authentication or device-specific authentication. To facilitate the authentication of non-IEEE 802.1X–capable managed devices such as printers, an alternative authentication method such as MAC authentication bypass (MAB) should be used.



Note MAB and WebAuth can both be configured as fallback mechanisms for IEEE 802.1X. In the event that a port is configured for IEEE 802.1X, MAB, and fallback WebAuth, the port first attempts to authenticate the user through IEEE 802.1X. If IEEE 802.1X authentication times out, the switch attempts MAB. If MAB fails, the switch attempts to authenticate with WebAuth. The automatic sequencing of authentication methods allows the network administrator to apply the same configuration to every access port without having to know in advance what kind of device (employee or guest, printer or PC, IEEE 802.1X capable or not, and so on) will be attached to it.

- **Restricted network access**—The switch can be configured to restrict traffic from the port before the WebAuth process is complete. Depending on the type of restrictions that are configured and the device's operating system, WebAuth could interfere with the device's startup sequence (such as Microsoft Windows bootup and group policy download). As long as WebAuth is primarily used for temporary or guest-related access, these types of limitations rarely affect normal operation.
- **Delay**—WebAuth is initiated after IEEE 802.1X times out or fails, which can contribute a significant delay in accessing the network.
- **Access control list (ACL) enforcement only**—WebAuth uses ACLs to restrict access to the network. ACLs provide a quick and well-understood way to provide granular access control, especially in networks with summarized address space, but ACLs do have limitations, especially in the area of scalability. VLAN assignment is not currently supported for WebAuth.

For all of these reasons, Cisco recommends that WebAuth be used only as a means to provide supplemental access in an IEEE 802.1X-enabled network.

WebAuth Solution Components

The following hardware platforms and software releases are the minimum versions required to configure all the features described in this guide:

- Cisco Catalyst Series Switches with Cisco IOS Software Release 12.2(55)SE3
- Cisco Catalyst 3560 Series Switches with Cisco IOS Software Release 12.2(55)SE3
- Cisco Catalyst 3750 Series Switches with Cisco IOS Software Release 12.2(55)SE3
- Cisco Catalyst 4500-E Series Switches with Cisco IOS Software Release 12.2(53)SG5 (requires SUP6E and SUP6L-E)
- Cisco Catalyst 6500 Series Switches with Cisco IOS Software Release 12.2(33)SX17



Note

The SUP32 module has been tested as an access layer component and the SUP720 and SUP720-10G modules have been tested as distribution and core network components.

- Cisco Secure Access Control System (ACS) Version 5.2 + patch 3 (earlier versions of Cisco Secure ACS also support the required functions with the appropriate configuration.)

Although other platforms were not tested as part of this solution, the Cisco Catalyst 4948 Switch is expected to perform similarly with these software releases.

This document does not discuss related technologies, including:

- Authentication proxy (auth-proxy), available in Cisco IOS Firewall Release 12.0(5)T and later
- External WebAuth with Cisco Wireless LAN Controllers Version 4.0 and later

See the [“References” section on page 45](#) for information about these related technologies.

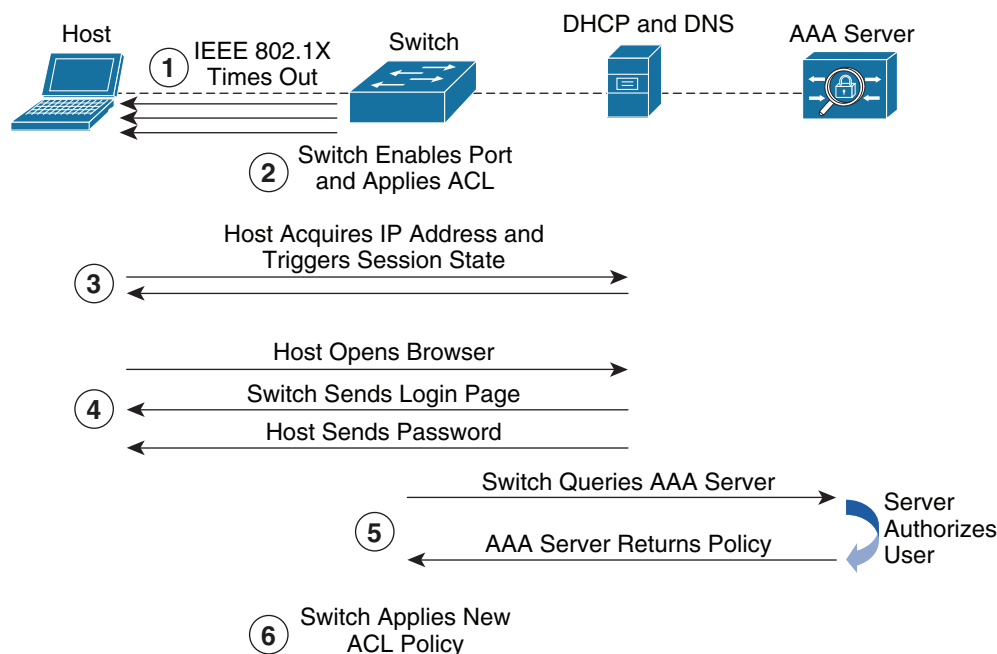
Sequence of Operations

This section describes the sequence of operations that occur during WebAuth, and includes the following topics:

- [Operation Sequence Overview, page 6](#)
- [Operation 1—Before Web Authentication, IEEE 802.1X Times Out or Fails, page 8](#)
- [Operation 2—Switch Opens Port for Limited Access, page 9](#)
- [Operation 3—User Traffic Triggers Web Authentication Session State, page 10](#)
- [Operation 4—User Gets Login Page, page 11](#)
- [Operation 5—Authentication Server Authorizes User, page 13](#)
- [Operation 6—Switch Applies New Policy and Redirects Page, page 15](#)
- [Operation 7—Session Termination, page 15](#)

Operation Sequence Overview

Successful WebAuth is the result of several operations. Although most of these operations are invisible to the end user, a clear understanding of these steps is helpful when deploying and maintaining WebAuth. The high-level sequence of operations is illustrated in [Figure 1](#).

Figure 1 WebAuth Sequence of Operations

1. In an IEEE 802.1X-enabled network, WebAuth can begin only after IEEE 802.1X authentication times out or fails.
2. The Cisco Catalyst switch opens the port for configurable traffic types (for example, Dynamic Host Configuration Protocol [DHCP] and Domain Name System [DNS]) required for WebAuth. Depending on whether the Open Access feature is configured (see the [“Open Access” section on page 21](#)), this process can occur before or after IEEE 802.1X times out or fails.
3. The host requests and receives an IP address, triggering the session state on the port.
4. The host opens a browser. The switch intercepts the host’s HTTP traffic and presents the host with a login page. The user enters credentials on the login page.
5. The switch sends these credentials to the authentication, authorization, and accounting (AAA) server (for example, Cisco Secure ACS). The authentication server validates the credentials and sends back the user-specific policy that should be applied to the port.
6. This switch applies this new policy to the port, and the host can access the network according to the assigned policy. The switch redirects the host to the original web page.

These steps are described in the sections that follow.



Note

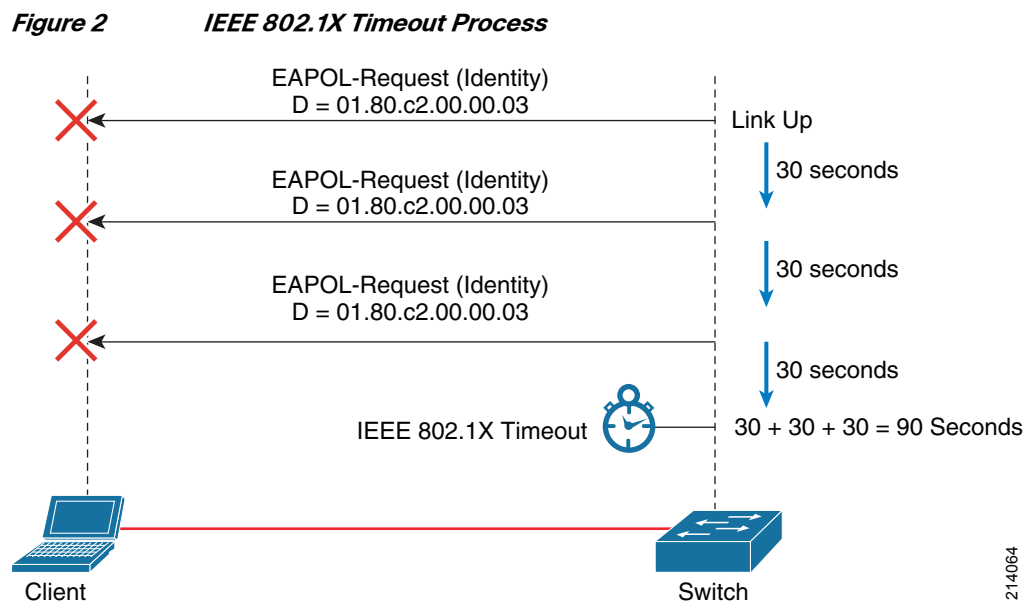
The following sections assume that the port has been configured for default (Closed) access control. For additional considerations when using the open-access feature, see the [“Open Access” section on page 21](#).

Operation 1—Before Web Authentication, IEEE 802.1X Times Out or Fails

WebAuth is a fallback authentication mechanism for IEEE 802.1X-enabled networks. This means that WebAuth is used after the switch has attempted to authenticate the client with IEEE 802.1X. WebAuth is intended to provide only an alternative means of authentication when no IEEE 802.1X supplicant is available, or when an IEEE 802.1X supplicant exists but the authentication fails.

IEEE 802.1X Timeout

Cisco Catalyst switches initiate IEEE 802.1X authentication as soon as a host plugs into a port by sending an Extensible Authentication Protocol (EAP) Identity-Request message. If the host does not respond within a certain amount of time, the switch resends the Identity-Request message. By default, the switch resends the Identity-Request message twice before timing out the IEEE 802.1X authentication. The interval between each message is 30 seconds. In the default configuration, the switch takes a total of 90 seconds to time out. Both the interval and the number of retries can be configured. Only after IEEE 802.1X has timed out does the switch attempt another authentication method such as WebAuth. [Figure 2](#) illustrates the IEEE 802.1X timeout process.



When determining the optimal IEEE 802.1X timer and retry values for your network, you should consider several factors.

In the default configuration, all traffic that is not EAP over LAN (EAPoL) traffic (including DHCP) is dropped until IEEE 802.1X times out. Therefore, the value of the timeout can significantly affect the DHCP client on the end host. Long IEEE 802.1X timeouts may prevent DHCP from functioning correctly after the IEEE 802.1X timeout expires. Without DHCP, a client cannot acquire an IP address and cannot use WebAuth. If an end user opens a browser before IEEE 802.1X times out, the end user experiences a Server Not Found error.

To prevent DHCP clients from timing out, one solution is to use lower IEEE 802.1X timer and retry values to help ensure that IEEE 802.1X times out before the DHCP client times out. Because DHCP timeouts can vary widely, Cisco recommends testing the DHCP clients in your network to discover how long they take to time out and setting the IEEE 802.1X timers accordingly.

After IEEE 802.1X has timed out and the port falls back to WebAuth, the switch no longer initiates IEEE 802.1X authentication by sending EAP Identity-Request messages. If a supplicant later becomes active on the port, the switch does not initiate an IEEE 802.1X session. The supplicant must initiate IEEE 802.1X by sending an EAPoL-Start frame to the switch. Almost all supplicants send (or can be configured to send) EAPoL-Start frames. In the rare instance in which a supplicant does not send an EAPoL-Start frame and the IEEE 802.1X timeout period is so short that the host does not boot up fast enough to launch the supplicant before IEEE 802.1X times out, the IEEE 802.1X-capable client may miss the opportunity to perform IEEE 802.1X authentication and is prompted for WebAuth instead. Therefore, Cisco recommends always deploying a supplicant that sends EAPoL-Start frames in accordance with the IEEE 802.1X specification.

Clearly, setting the IEEE 802.1X timer to an arbitrarily long or short value can have unintended consequences for network operation. Each network has a different optimal value. Therefore, as a best practice, Cisco recommends that you test the IEEE 802.1X timer values in your own network to determine the best value.

IEEE 802.1X Failure

By default, IEEE 802.1X authentication fails if the host has an IEEE 802.1X supplicant but does not have valid credentials. Cisco Catalyst switches can be configured to attempt WebAuth after IEEE 802.1X fails.

When IEEE 802.1X fails, it usually fails quickly, particularly if the supplicant has been configured for single sign-on. Therefore, problems associated with DHCP client timeouts and Server Not Found errors are typically not of concern in this use case.

In most respects, WebAuth works the same way whether it was triggered by IEEE 802.1X timeout or IEEE 802.1X failure. However, the trigger (failure or timeout) does become important if the user attempts (or reattempts) IEEE 802.1X authentication after WebAuth has started. The port “remembers” whether IEEE 802.1X timed out or failed, even after WebAuth has been initiated. If IEEE 802.1X times out, the switch listens for EAPoL-Start messages from the client and restarts IEEE 802.1X.



Note

The default behavior can be changed by using the Flexible Authentication feature set. The switch can be configured to ignore EAPoL messages from the client after an IEEE 802.1X timeout by setting the priority of WebAuth to be higher than IEEE 802.1X. If IEEE 802.1X authentication fails, the switch ignores any additional EAPoL traffic from the end client during and after the WebAuth process (regardless of whether WebAuth succeeded).

Operation 2—Switch Opens Port for Limited Access

By default, IEEE 802.1X requires that, prior to authentication, the port is closed to all traffic except EAP packets. If the port were to remain in this state, the client would not be able to acquire an IP address or use a web browser for login. WebAuth is a Layer 3 authentication method that requires that the end host have an IP address. Therefore, after IEEE 802.1X (or MAB) has timed out or failed, the port must be opened long enough to allow the packets required for WebAuth.

During the WebAuth process, the switch restricts access on the port through a configurable ACL. Before deploying the ACL, however, the switch must open the port in some VLAN. Cisco Catalyst switches running Cisco IOS Software open the port in the default data VLAN that is configured on the port. Therefore, the default data VLAN is used for WebAuth. This is true regardless of whether IEEE 802.1X has timed out or IEEE 802.1X has failed prior to the start of WebAuth.



Note

IEEE 802.1X and MAB endpoints that successfully authenticate but do not receive a dynamic VLAN assignment are assigned to the default data VLAN. Therefore, IEEE 802.1X-authenticated endpoints are in the same VLAN as users that cannot use IEEE 802.1X. In this case, dynamic ACL assignments can be used to differentiate access levels for endpoints authenticated using different methods.

For the highest level of traffic isolation, dynamic VLAN assignment can be used to assign endpoints authenticated by IEEE 802.1X and MAB to a different VLAN.



Note

Cisco switches do not support dynamic VLAN assignment as the result of WebAuth, but it is still possible to dynamically assign VLANs as the result of an IEEE 802.1X or MAB authentication.

By dynamically assigning a VLAN that is different from the default VLAN, the switch can completely isolate traffic from authenticated IEEE 802.1X and MAB endpoints from traffic from WebAuth endpoints. Moreover, the logical isolation provided by separate VLANs can be extended to the routed portion of the network using the path isolation techniques of network virtualization. By creating dedicated logical networks, network virtualization can provide end-to-end solutions for guest access and partner access scenarios. For more information about network virtualization, see <http://www.cisco.com/en/US/netsol/ns658/index.html>.



Note

Before deploying dynamic VLAN assignment for IEEE 802.1X authenticated users, you should understand the design implications of VLAN assignment. In many deployments, most endpoints are authenticated by IEEE 802.1X or MAB. Careful analysis is required to determine whether the cost of dynamically assigning VLANs to the majority of the endpoints is worth the benefit of allowing a minority of users to use WebAuth to access the network in an isolated default data VLAN.

After the port has been opened, the switch enforces a preconfigured ACL. By default, the preconfigured ACL is dynamically applied only when WebAuth is initiated. This initiation is accomplished using a WebAuth fallback profile. The preconfigured ACL in the fallback profile does not apply to ports in the critical VLAN, the auth-fail VLAN, or the guest VLAN.

At a minimum, the preconfigured ACL should allow the traffic required to complete the WebAuth process. In most cases, the ACL should at least allow DHCP (so the client can acquire an address) and DNS (so the client can trigger WebAuth when using fully qualified domain names in URLs). Additional access can be allowed according to the organization's security policy.

Operation 3—User Traffic Triggers Web Authentication Session State

After the switch has opened the port for limited access, it is the responsibility of the end host to trigger the WebAuth session state. Session state is created when the switch sees a DHCP transaction or an Address Resolution Protocol (ARP) packet. The switch depends on the host to send DHCP or ARP traffic to trigger WebAuth. If the host launches a browser before session state has been triggered, the user does not receive the web login page.

When the initial ARP or DHCP packet is received, the device is entered in the IP device tracking table. IP device tracking maintains a table of known devices and periodically sends ARP probes to verify that they are still active. A successful IP device tracking probe initiates session state.

After session state has been triggered, the init-state timer is started. See the init-state timer details in the [“Operation 7—Session Termination” section on page 15](#) for more information about configuring the init-state timer. The end user must enter valid credentials before the init-state timer expires or else the session state is cleared. After the session state is cleared, it must be re-established using one of the methods described in the preceding paragraphs.

Operation 4—User Gets Login Page

After session state has been created and the end user launches a browser, WebAuth is triggered and the switch returns the web login page. The login page contains fields for the user to enter a username and password. The login page can optionally be customized to contain additional information.



Note

When communicating with the end user browser, the switch reuses the IP address of the server that the browser first attempted to contact.

After the user enters credentials at the login page and clicks the Submit button, the switch authenticates the credentials to a back-end authentication server using the RADIUS protocol.

The switch itself acts as the web server and provides the default login page. No external web server is required. Because the switch serves the page, it must be able to switch traffic to the host. This can be accomplished in two ways: using a switched virtual interface (SVI) or a default route. The choice of method depends in part on your existing network design. In either case, the connectivity requirements most likely already exist as part of the basic network design, but it is worthwhile to review them here.

If the switch has an SVI for the host's data VLAN, the switch has a Layer 3 address on the host's subnet and is able to route the login page directly to the host. If your network already supports the use of SVIs for data VLANs at the access edge, no additional configuration is required.

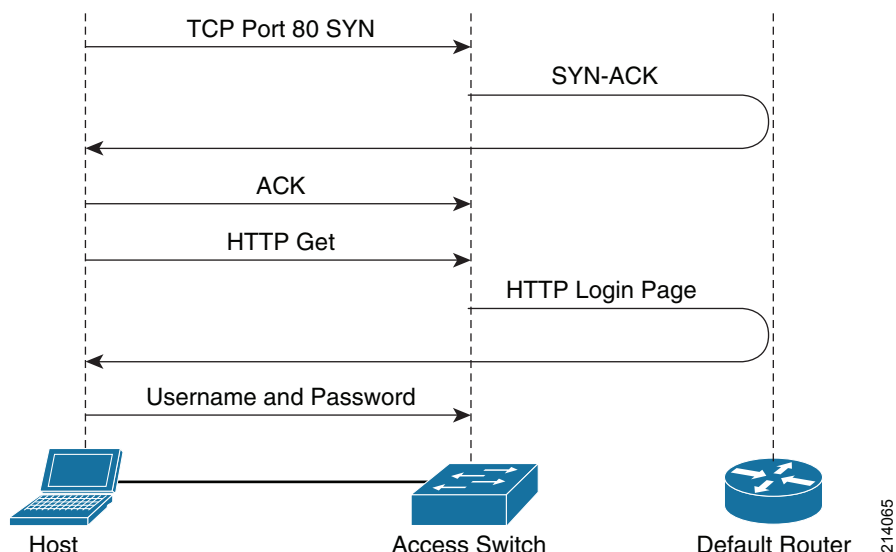


Note

A network that follows the routed access model of campus network design would typically have SVIs at the access device.

If the switch is not configured for SVIs on the data VLANs, the switch can send the login page to the host using a default route. When a default route is used, all traffic from the switch to the host is sent to the default router, which may be one or more hops away. The default router then routes the traffic to the host back through the access switch. [Figure 3](#) shows the initial TCP traffic flow for this situation.

Figure 3 *TCP Traffic Flow for Login Page When No Layer 3 SVI for Host VLAN Exists on Access Switch*



Although this approach introduces additional hops in the return path from the switch to the host, it produces negligible load on the default router and intervening infrastructure because only the WebAuth traffic from the switch to the host follows this path. In campus designs that do not use SVIs on the data VLAN, a default route is typically already configured.



Note

A network that follows the multi-tier campus design model would typically not have SVIs at the access edge.

In this case, no additional configuration is required to support WebAuth. However, problems may arise in the case in which traffic to the default router is bridged through a stateful firewall. The original SYN packet in the TCP handshake is consumed by the access switch, so the first packet that the firewall sees is the SYN-ACK packet from the access switch. Stateful firewalls typically drop SYN-ACK packets if they have not seen the original SYN packet. In this case, you need to turn off stateful inspection for ports 80 and 443 on the firewall.

When a non-crypto image is used, Cisco IOS Software automatically redirects all HTTP packets (TCP port 80) to itself. URLs that reference HTTPS (TCP port 443) do not trigger redirection. If HTTPS support is required, a Cisco IOS Software crypto image is necessary. Cisco IOS Software crypto images redirect HTTPS traffic and can be configured to redirect HTTP traffic as well. URLs that contain a port other than 80 or 443 (for example, <http://my-acis-server:2002>) do not trigger redirection.



Note

WebAuth can intercept nonstandard ports using an IP port-to-application map (PAM) entry that maps a new port to HTTP (or HTTPS). In addition, the Cisco IOS Software HTTP server needs to be reconfigured to listen on the nonstandard port. However, the Cisco IOS Software HTTP server can run only on a single port. Therefore, support for port 80 and a nonstandard port are mutually exclusive. If PAM is used to remap the port used for HTTP, URLs that reference the default port (80) do not trigger redirection. In addition, if traffic to the default router is bridged through a stateful firewall, that firewall must turn off stateful inspection for the remapped port.

If the crypto image is used and HTTPS is also configured, the switch initiates an SSL session, even if the initial HTTP request was to port 80. The advantage of this approach is that the user's credentials are sent over an encrypted channel to the switch and cannot be snooped. The disadvantage is that the browser prompts the end user to accept the switch's certificate, adding another step to the authentication process. By default, the switch sends a self-signed certificate. Some browsers display a warning or error message when receiving a self-signed certificate. Even if the certificate is signed by a trusted CA, the nature of the redirection process may still cause some browsers to display a warning message.

There are four web pages used in WebAuth that can be customized: login, authentication fail, authentication success, and authentication expired pages. If any one page is customized, the other pages must also be configured as customized pages.

The following design considerations must be addressed when customizing web pages:

- **Management**—Customized web pages are stored on each individual switch and must be managed accordingly.
- **Embedded images**—On most platforms, only a single file can be specified for each of the four customizable web pages (login, success, fail, expired), so any local images you want to display on the login page must be embedded in the `` tag as described in RFC 2397. Be aware that not all browsers support embedded images.



Note As of this writing, Cisco Catalyst 3560, 3750, and 2960 Series Switches running Cisco IOS Software Release 12.2(52)SE and later are the only platforms that support non-embedded images stored on the switch local system directory. See the platform configuration guides for more details.

- **External links**—External links (including links to images) are allowed as long as the preconfigured ACL allows access to the external server. Be aware, however, that the switch intercepts all HTTP and HTTPS requests (even if the preconfigured ACL permits HTTP or HTTPS), so any URLs embedded in the login page must have a scheme other than HTTP or HTTPS or reference a port other than 80 or 443. Sample configurations can be found in the [“Configuring Customized Web Pages \(Optional\)” section on page 41](#).
- **Size**—The maximum size for a customized page, including embedded images, is currently 8 KB.
- **Login page recommendations**—The login form must accept user input for the username and password and must post the data as **uname** and **pwd**. The custom login page should also follow best practices for a web form, such as page timeout, hidden password, and prevention of redundant submissions.

Sample web pages that can be used as the base of customized pages are provided in the [“Sample Customizable Pages” section on page 45](#).

Operation 5—Authentication Server Authorizes User

After the end user enters their username and password, the switch sends a RADIUS Access-Request message to the AAA server. This is a normal Password Authentication Protocol (PAP) authentication request that contains user's name and hashed password.



Note

In the current implementation, WebAuth uses the default login group for AAA authentication (as defined by the command **aaa authentication login default group**). If your existing switch configuration uses this group for other purposes, it may be necessary to reconfigure your AAA authentication.

How the AAA server validates the user's name and password depends on the AAA server. The simplest authentication involves referencing an internal database of usernames and passwords locally configured on the AAA server. Some AAA servers also allow validation of credentials against external databases. The use of external databases often depends on what use case needs to be supported. For example, suppose an employee failed IEEE 802.1X because of an expired certificate and fell back to WebAuth. If the employee enters an Active Directory username and password on the login page, the AAA server must be able to query Active Directory to authenticate those credentials. Another use case might involve a guest without a supplicant or a contractor with a supplicant but without a valid IEEE 802.1X credential. The guest or contractor might then be provided with a temporary username and password that can be used for WebAuth. In those cases, the AAA server would need to query the sponsored guest credential repository (such as the network admission control [NAC] guest server).

Regardless of where the credentials are stored, if the user's credentials are valid, the AAA server sends a RADIUS Access-Accept message to the switch with an authorization policy that determines what level of access the user receives. This authorization policy takes the form of an ACL, which can permit or deny traffic based on IP address and upper-layer application.



Note

The only kind of authorization supported by Cisco IOS Software WebAuth is an ACL (IEEE 802.1X and MAB support VLAN assignment as well as ACL-based authorization). There are many types of dynamic ACLs (filtered ID, proxy ACL, per-user ACL, and so on), but starting in the Cisco IOS Software releases listed in the [“WebAuth Overview” section on page 3](#), the only kind of ACL that can be used for every authentication method (IEEE 802.1X, MAB, and WebAuth) across every switch platform is the downloadable ACL (dACL).

Because dACLs are currently the only ubiquitous dynamic ACL, only dACLs are discussed in this document.

A WebAuth request can be uniquely identified by looking at the Service-Type attribute in the initial Access-Request message from the switch. WebAuth requests from Cisco Catalyst switches running Cisco IOS Software always set the Service-Type attribute to 5. [Table 1](#) shows the value of this attribute for other authentication types.

Table 1 *Service-Type Attribute Values*

Switch OS	Authentication Type	Service-Type Attribute
Cisco IOS Software	IEEE 802.1X	2 (Framed)
Cisco IOS Software	MAB	10 (Call Check)
Cisco IOS Software	WebAuth	5 (Outbound)

By filtering each Access-Request message based on the Service-Type attribute, the AAA server can identify the type of authentication request and use this as another condition or attribute in creating authorization policy rules. In Cisco Secure ACS 5.0, filtering can be accomplished using service selection rules.



Note

In Cisco Secure ACS 4.0, filtering is accomplished using a network access policy (NAP).

Operation 6—Switch Applies New Policy and Redirects Page

When the switch receives an Access-Accept message with the new ACL information from the authentication server, the switch customizes the ACL by adding the IP address of the end host as the source address in each line of the downloaded ACL. This process helps ensure that only the authenticated host can gain access using this ACL. The switch then adds this customized ACL to the preconfigured ACL. By adding the new ACL to the existing ACL, the switch is able to grant the user additional network access beyond the limited access that was allowed for the WebAuth process. The new combined ACL grants granular, per-user access to the network. The customized ACL can specify an IP address or subnet. It can also be configured to permit or deny access to specific applications (through the specification of TCP or User Datagram Protocol [UDP] ports).

Although ACLs are conceptually very simple, they can become very complex in real-world networks. Networks with an addressing scheme that is not summarized or networks in which related resources cannot be logically grouped together (for instance, in the same subnet) can result in long ACLs that are difficult to define, maintain, and troubleshoot. Good network design and well-considered security policy planning can reduce the problems associated with ACL scalability.

Management is not the only scalability concern with ACL-based authorization. Switches use ternary content addressable memory (TCAM) for hardware-based ACL processing. By processing ACLs in hardware, Cisco switches enable secure, high-bandwidth communication within the campus. However, long ACLs applied across numerous ports can overwhelm a switch's TCAM space. When selecting ACLs as an authorization method, be sure that your switches have sufficient TCAM capacity to handle the number and size of ACLs that you plan to deploy. Be aware that the total ACL length includes the port ACL in fallback profile and the dACL.

When designing dynamic ACLs, Cisco recommends using the shortest and simplest possible format. Using a simple ACL optimizes TCAM resources and makes ACL-based authorization more manageable.

After the dACL has been applied, the switch displays a customizable Authentication Success page and then redirect the user's browser to the original URL that the user was trying to access before WebAuth occurred.

Operation 7—Session Termination

This section describes how a WebAuth session can be terminated and includes the following topics:

- [Init-state Timer, page 15](#)
- [Link Down, page 16](#)
- [Absolute Session Timer, page 16](#)
- [Inactivity Timer, page 16](#)
- [Failed Authentications and Denial-of-Service Attacks, page 17](#)

Init-state Timer

If the user fails to enter valid credentials in the login page before the IP admission init-state timer expires, the session state is cleared and the session state must be reestablished by ARP or DHCP traffic from the host before WebAuth can be attempted again. Barring new DHCP or ARP traffic from the host itself, the WebAuth session state is reinstated by the first IP device tracking probe following the init-state timeout. By default, IP device tracking probes are sent every 30 seconds.

Link Down

The most direct way to terminate a WebAuth session is to unplug the host. When the link state of the port goes down, the switch completely clears the session. If the original host (or a new host) plugs in, the switch restarts authentication from the beginning, starting with IEEE 802.1X and MAB if so configured.

Absolute Session Timer

The absolute session timer is optionally returned in attribute 27 by the AAA server during the initial authentication. After this timer expires, the dACL is removed from the port, and the user's access is restricted until WebAuth is successfully completed. The absolute timer applies only to WebAuth; the switch does not restart IEEE 802.1X authentication or MAB when the absolute timer expires. If no absolute timer is specified in the RADIUS Access-Accept message, the session remains active unless it is terminated by a link-down event or the inactivity timer.

Inactivity Timer

The inactivity timer for WebAuth is controlled by the IP device tracking probe interval and retry count.



Note RADIUS attribute 28 [Inactivity Timer] cannot be used to change the inactivity timer for WebAuth.

IP device tracking maintains a table of known devices and periodically probes those devices to verify that they are still active. If no response is received, the switch repeats the probe until the configured count is reached. If all probes go unanswered, the WebAuth session is taken down. The switch does not restart IEEE 802.1X authentication or MAB after an inactivity timeout, but the original host (or any new host) is required to retrigger the session state and re-authenticate using WebAuth. Because the host is removed from the IP device tracking table after the inactivity timeout, no further probes are sent, and the inactive end host must send DHCP or ARP traffic to reinitiate session state.

An important application of the inactivity timer involves indirectly connected devices, such as a laptop connected behind an IP phone or through a hub. Because the switch has no direct knowledge of the link state of the indirectly connected device, it cannot terminate the session when that device unplugs. The best solution is to use the intelligence of the intermediary device to alert the switch when the device unplugs. For example, Cisco IP Phones and switches support the Cisco Discovery Protocol Second Port Status type and length value (TLV) that allows the phone to communicate to the switch when the device behind the phone unplugs, enabling the switch to clear the WebAuth session. For phones that do not support the Cisco Discovery Protocol Second Port Status TLV or for unmanaged devices (such as hubs) that have no way of communicating link-state information, the only solution is to use the IP device tracking-based inactivity timer. Without the inactivity timer, WebAuth sessions on indirectly connected devices would be maintained indefinitely (or until the absolute timer expired).

If you want to increase the inactivity timer for WebAuth, Cisco recommends increasing the IP device tracking probe *count*, not the probe interval, because the length of the IP device tracking probe interval can have a significant effect on other aspects of the end-user experience. As discussed above, IP device tracking is responsible for initializing the session state after an init-state timeout and when the open-access feature is used. Increasing the probe interval also increases the maximum amount of time a user may wait to receive a login page. Increasing the probe retry *count* increases the overall inactivity timer without affecting the amount of time a user may wait to receive a login page.

Failed Authentications and Denial-of-Service Attacks

If the user enters invalid credentials, the authentication server sends an Access-Reject message, and the switch makes no changes to the preconfigured ACL. The user receives a customizable Authentication Failed pop-up message.

During the WebAuth session, all HTTP packets are sent to the CPU for processing. To prevent a malicious user from exploiting this route to launch a denial-of-service (DoS) attack on the switch, an end host is put in a SERVICE_DENIED state after exceeding the maximum number of login attempts. By default, the user is allowed five login attempts before being locked out. By default, users who fail authentication five times are locked out for two minutes. A user who is locked out receives a customizable Authentication Expired pop-up message.

To change the default behavior for failed login attempts (five attempts and a 2-minute lockout), enable the IP admission watch list. If the IP admission watch list is enabled, an end host is added to the watch list if that user fails to authenticate after the maximum number of login attempts. After the host's IP address is on the watch list, the switch does not intercept HTTP packets from that host or perform WebAuth processing until the expiry timer has expired (default is 30 minutes). If the watch list is disabled, the default behavior (five attempts maximum with a 2-minute service denial) is restored.

After a session is terminated, the user must re-authenticate before being granted expanded access to the network again.

WebAuth Feature Interaction

This section includes the following topics:

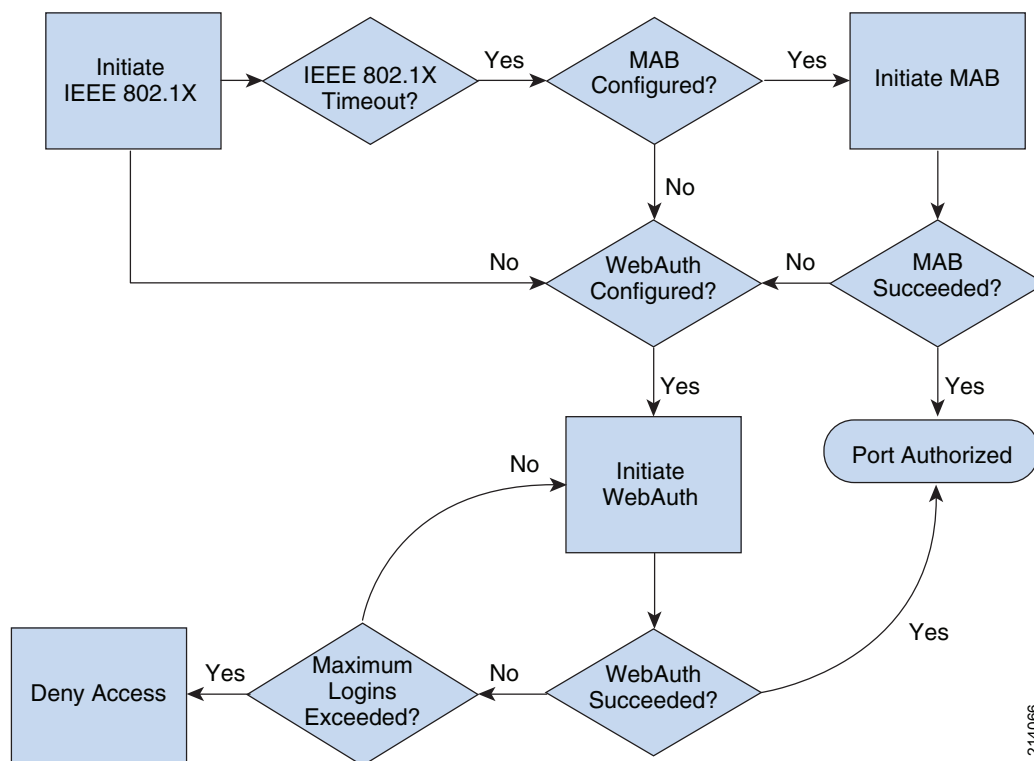
- [MAC Authentication Bypass, page 17](#)
- [Guest VLAN, page 18](#)
- [Auth-Fail VLAN, page 18](#)
- [Inaccessible-Auth Bypass, page 19](#)
- [Port ACLs, page 20](#)
- [Open Access, page 21](#)

MAC Authentication Bypass

MAB can be used for authenticating managed devices that do not have an IEEE 802.1X supplicant. During MAB, the switch learns the MAC address of the connected device and forwards it to the authentication server. The authentication server verifies that the MAC address matches that of a known device and instructs the switch to allow access according to the policy for that device.

WebAuth is fully compatible with MAB. In the event that a port is configured for IEEE 802.1X, MAB, and fallback WebAuth, the port first attempts to authenticate the user through IEEE 802.1X authentication. If IEEE 802.1X authentication times out or fails, the switch attempts MAB. If MAB fails, the switch attempts WebAuth.

The flow chart in [Figure 4](#) shows the interaction between IEEE 802.1X timeout, MAB, and WebAuth.

Figure 4 *Fallback WebAuth and MAB*

Guest VLAN

If both the Guest VLAN and WebAuth are configured on Cisco IOS Software platforms, the guest VLAN configuration is ignored and the switch falls back to WebAuth when IEEE 802.1X times out.

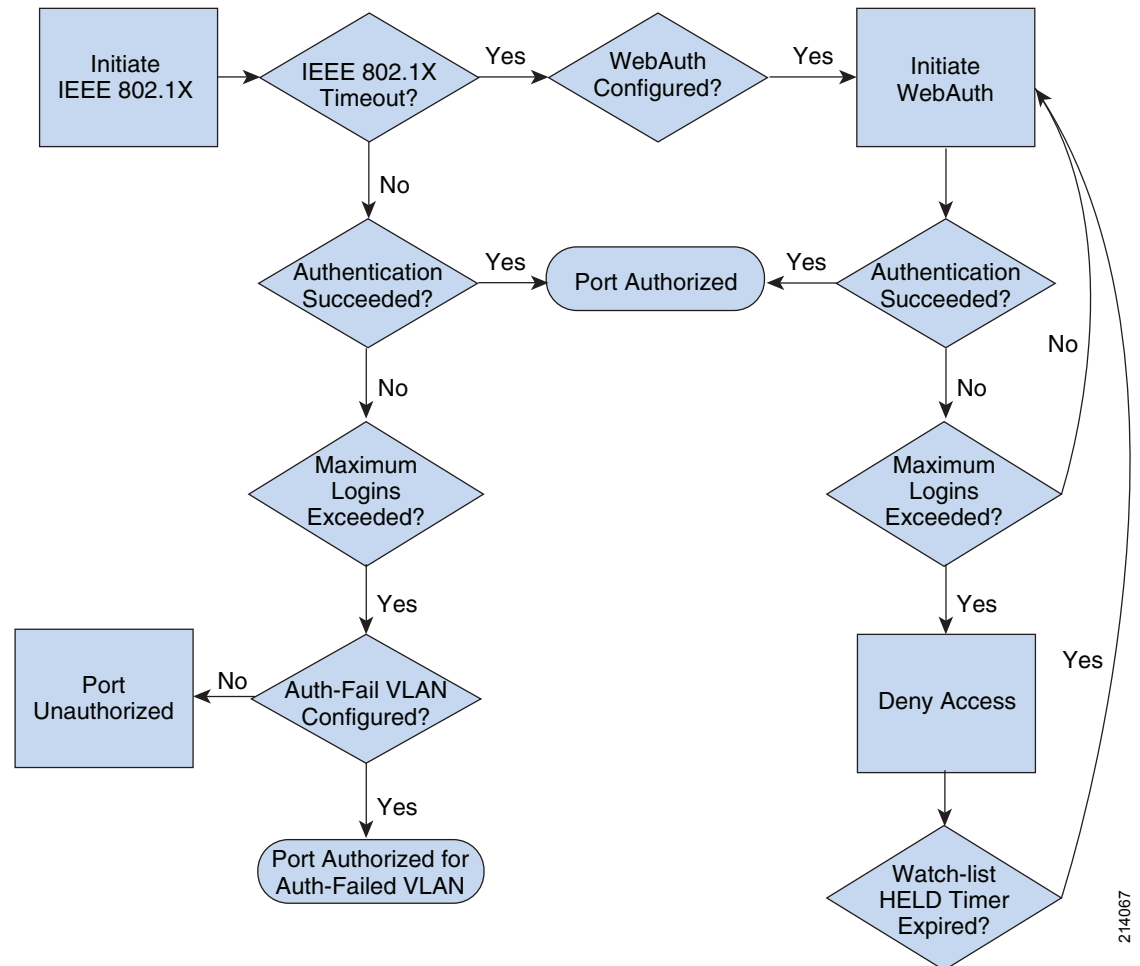
Auth-Fail VLAN

After an IEEE 802.1X authentication failure, the switch can be configured to deploy the auth-fail VLAN or proceed to the next authentication method (MAB or WebAuth). In this sense, auth-fail VLAN and WebAuth are mutually exclusive when IEEE 802.1X fails. However, it is possible to configure the auth-fail VLAN for IEEE 802.1X failures (the client has a supplicant but does not have valid credentials) and still retain WebAuth for IEEE 802.1X timeouts (the client has no supplicant).

The auth-fail VLAN applies only to IEEE 802.1X failures. If WebAuth fails, the auth-fail VLAN is not applied. After a failed WebAuth attempt, the user is allowed to retry WebAuth until the maximum number of login attempts is reached. After that, the user is denied access for a configurable amount of time, as described in the [“Operation 7—Session Termination”](#) section on page 15.

Figure 5 illustrates the interaction of the auth-fail VLAN and WebAuth.

Figure 5 Fallback WebAuth and Auth-Fail VLAN



214067

Inaccessible-Auth Bypass

Fallback WebAuth can be configured with inaccessible authentication (inaccessible-auth) bypass (also known as critical authentication or AAA fail-open). If a port is configured for both inaccessible-auth bypass and fallback WebAuth, the final state of the port when the AAA server is unavailable is determined by the timing of the connectivity loss and when the switch learns that the AAA server has failed.

If the switch already knows that the AAA server has failed, the port is immediately deployed for the critical VLAN as soon as the link comes up. Because the switch has multiple mechanisms for learning that the AAA server has failed, this outcome is the most likely.

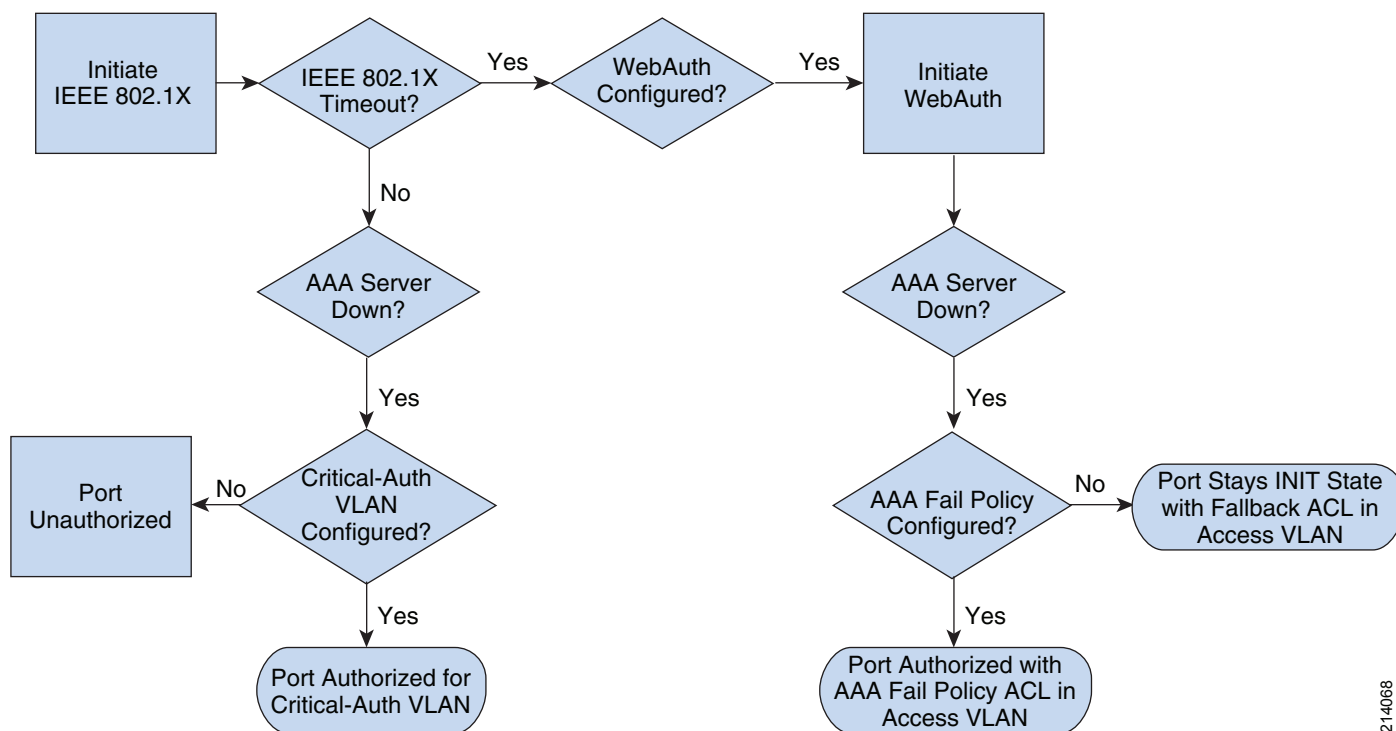
If the switch determines that the AAA server has failed during an IEEE 802.1X or MAB authentication (for example, if this is the first device to connect to the switch after connectivity has been lost), the port is moved to the critical VLAN.

If the switch does not know that the AAA server has failed when the port falls back to WebAuth, the IEEE 802.1X-critical VLAN configuration has no effect. The port is moved to the data VLAN with the preconfigured ACL in the fallback profile. If a user attempts to use WebAuth under these conditions, an authentication failure results, and access to the port continues to be restricted by the default-port ACL.

To change the default behavior, use the WebAuth AAA fail policy. The AAA fail policy (which applies only to WebAuth, not IEEE 802.1X or MAB) enables a user to connect by applying a specified policy (in effect, a new ACL) if the AAA server is not available. When the AAA server returns, the AAA fail policy is removed, and the end user is required to authenticate again.

Figure 6 illustrates the interaction of fallback WebAuth with inaccessible-auth bypass and AAA fail policy when the switch is not already aware that the AAA server has failed.

Figure 6 Interaction of WebAuth, AAA Fail Policy, and Inaccessible-Auth Bypass



Note

AAA fail policy for WebAuth is not supported on the following switches: Cisco Catalyst 2960 Series Switches with Cisco IOS Software Release 12.2(55)SE3, Cisco Catalyst 3560 Series Switches with Cisco IOS Software Release 12.2(55)SE3, and Cisco Catalyst 3750 Series Switches with Cisco IOS Software Release 12.2(55)SE3.

Port ACLs

When both a statically configured port ACL and a WebAuth fallback profile ACL are configured on the same port, the statically configured port ACL governs access to the port. The ACL in the fallback profile is ignored. Therefore, make sure that the statically configured port ACL allows at least enough access (DNS and DHCP) for the WebAuth process to be performed.

Open Access

WebAuth is compatible with the open-access feature as long as some additional design considerations are addressed.

By default, IEEE 802.1X drops all traffic prior to a successful IEEE 802.1X (or MAB) authentication or WebAuth initialization. This is sometimes referred to as closed mode. Cisco switches can be configured for open-access mode, which allows all traffic prior to successful authentication.

The first design consideration is the value of the IEEE 802.1X timeout that precedes WebAuth. Open-access mode allows configurable traffic types (such as DHCP) before IEEE 802.1X times out, which allows the DHCP client to acquire an IP address before IEEE 802.1X times out. However, even though the client has an IP address, the port does not begin the WebAuth process before IEEE 802.1X times out. Because WebAuth has not yet started redirecting traffic to the login page, the web traffic is subject to the port ACL, which will most likely deny all web traffic. Therefore, if an end user opens a browser before IEEE 802.1X times out, the end user experiences a Server Not Found error. Therefore, in open-access mode, Cisco recommends setting the IEEE 802.1X timers to account for the amount of time it usually takes for an end user to connect to the network and open a browser.

The second design consideration when using WebAuth with open-access mode is how the session state is triggered. The WebAuth session state is triggered when the switch detects DHCP or ARP traffic from the host. In the default closed mode, the host requests an address and ARP for its default gateway only after IEEE 802.1X has timed out or failed and the port has been opened for limited access for WebAuth. Prior to falling back to WebAuth, a port in closed mode drops all traffic, including DHCP and ARP traffic. In open-access mode, however, the port may allow DHCP and ARP traffic while IEEE 802.1X is still running. Thus, a host may have its address and the MAC address of the default gateway long before WebAuth begins. So when the port does fall back to WebAuth, there is no further DHCP or ARP traffic from the host to trigger session state.

The mechanism that helps ensure session state initialization in open-access mode is IP device tracking. IP device tracking maintains a table of known devices and periodically probes those devices using ARP to verify that they are still active. In the absence of DHCP or ARP traffic from the host, WebAuth session state is triggered by the first IP device tracking probe following the fallback to WebAuth. By default, IP device tracking probes are sent every 30 seconds. Therefore, in open-access mode, a user might wait up to 30 seconds (in addition to the time it takes IEEE 802.1X to time out) before the session state is triggered and the switch sends back the login page.

Using Host Modes

The host mode on a port determines the number and type of devices allowed on a port. WebAuth adheres to the host mode configured on the port. With the exception of multihost mode, all host modes are compatible with WebAuth with some design considerations.

Single-Host Mode

WebAuth is compatible with single-host mode. In single-host mode, only a single MAC or IP address can be authenticated (by any method) on a port. If a different MAC address is detected on the port after a host has authenticated with IEEE 802.1X, MAB, or WebAuth, a security violation is triggered on the port.

Multi-Auth Host Mode

WebAuth is compatible with multi-auth host mode. If the port is configured for multi-auth mode, multiple hosts can be authenticated separately using any method, including WebAuth, as long as each device downloads a dACL as a result of its authentication. Any device that does not download a dACL is subject to the port ACL after authentication.

Multidomain-Authentication Host Mode

WebAuth is compatible with multidomain-authentication host mode. For more information, see the [“Multidomain Authentication” section on page 23](#).

Multihost Mode

Because WebAuth uses user-based ACL policies, multihost mode is not compatible with WebAuth.

Preventing Authentication Failure with a Static Port ACL

To prevent authorization failures caused by downloading a dACL when no port ACL is present, a static port ACL must be configured when the port is configured for WebAuth and multi-auth or multidomain host mode.

For example, suppose there are multiple devices on the same port: one that authenticates using WebAuth, and one that authenticates using IEEE 802.1X (or MAB). If the WebAuth device authenticates first, the port ACL in the fallback profile is applied, and the dACL for both the WebAuth device and the IEEE 802.1X device are successfully applied. However, if the IEEE 802.1X device authenticates before the WebAuth fallback profile has been applied, there is no port ACL. This behavior is a problem because the current implementation of dACLs requires a static port ACL on the port before any dACL can be applied. Therefore, in this situation, a dACL applied as a result of an IEEE 802.1X (or MAB) authentication could trigger an authorization failure.

This problem is exacerbated in IP telephony environments because the IP phone is often authenticated by IEEE 802.1X or MAB before the device behind the phone begins the WebAuth process.

To avoid this race condition, statically configure a port ACL on the port (not in the fallback profile). With a static ACL on the port, there is always be a port ACL when any device authenticates by any method. This configuration prevents any authorization failures caused by downloading of a dACL when no port ACL is present.

IP Telephony

This section includes the following topics:

- [Cisco Discovery Protocol Bypass, page 23](#)
- [Multidomain Authentication, page 23](#)
- [IP Telephony and Link State, page 23](#)

Cisco Discovery Protocol Bypass

WebAuth is not compatible with Cisco IP Phones that use Cisco Discovery Protocol Bypass to access the voice infrastructure.

The ACLs that are used to restrict access before and after WebAuth are applied at the port level. This means that the data and voice VLANs are both subject to the ACL. If the phone uses Cisco Discovery Protocol to bypass authentication, it cannot download a dACL to open the port ACL for full access. Therefore, ports that are configured to allow phones using Cisco Discovery Protocol Bypass should not be configured for WebAuth.

Multidomain Authentication

Cisco IOS Software WebAuth is compatible with IP telephony when multidomain authentication host mode is used.

Unlike Cisco Discovery Protocol Bypass, which allows the phone free access to the network without authentication, multidomain authentication requires the phone to authenticate (using IEEE 802.1X or MAB). Because the phone authenticates, it can download its own dACL to access the port.



Note

See the [“Preventing Authentication Failure with a Static Port ACL” section on page 22](#) for an important consideration when using multidomain authentication mode with WebAuth.

IP Telephony and Link State

When WebAuth devices behind IP phones disconnect from the phones, the switch has no direct knowledge of the link state of the WebAuth session. Therefore, the switch does not know that it should take down the WebAuth session.

Cisco IP Phones and switches support a feature called Cisco Discovery Protocol Second Port Status TLV that allows the phone to communicate with the switch when the device behind the phone unplugs, enabling the switch to clear the WebAuth session. For phones that do not support Cisco Discovery Protocol Second Port Status, the only solution is to use the inactivity timer.

RADIUS Accounting

WebAuth supports RADIUS accounting, although the message format is not identical to that generated for a session authenticated by IEEE 802.1X or MAB. For specific details, see the [“Sample Customizable Pages” section on page 45](#). In addition, a WebAuth session has two start records: one when the switch opens the port to begin WebAuth, and another when the user has successfully entered a username and password.

Using Cisco Catalyst Integrated Security Features

Note the following regarding WebAuth and Cisco Catalyst integrated security features:

- **Port Security**—In general, Cisco does not recommend enabling port security when IEEE 802.1X is also enabled. Therefore, port security is not a recommended best practice when deploying WebAuth as a fallback mechanism for IEEE 802.1X.
- **DHCP Snooping**—DHCP snooping is fully compatible with WebAuth and should be enabled as a best practice.
- **Dynamic ARP Inspection**—Dynamic ARP Inspection is fully compatible with WebAuth and should be enabled as a best practice.
- **IP Source Guard**—There are platform-dependent considerations when deploying IP source guard with the dACLs that are used for WebAuth. Check the platform-specific release notes before enabling IP source guard with WebAuth.

Deployment Scenarios

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

- **Monitor mode**

WebAuth is not recommended in monitor mode.

The primary goal of monitor mode is to enable authentication without imposing any form of access control. This approach allows network administrators to see who is on the network and prepare for access control in a later phase without affecting end users in any way.

If WebAuth is enabled, end users that fail or timeout IEEE 802.1X authentication and MAB have their HTTP traffic intercepted and are forced to enter credentials on the web login page (although all other forms of non-HTTP network access are still permitted). This effect on end users contradicts one of the primary goals of monitor mode, so WebAuth should not be enabled in this mode.

- **Low-impact mode**

WebAuth is supported in low-impact mode as long as the following feature interactions are understood:

- Low-impact mode uses the open-access feature, so all the design considerations for open-access mode described in the [“Open Access” section on page 21](#) must be addressed.
- Low-impact mode uses multidomain authentication for IP telephony environments, so all the design considerations described in the [“Multidomain Authentication” section on page 23](#) must be addressed.
- Low-impact mode uses statically configured port ACLs, so all the design considerations for port ACLs described in the [“Port ACLs” section on page 20](#) must be addressed.

- **High-security mode**

WebAuth is supported in high-security mode. However, because high-security mode uses multidomain authentication for IP telephony environments, all the design considerations described in the [“Multidomain Authentication” section on page 23](#) must be addressed.

For more information about scenario-based deployments, see <http://www.cisco.com/go/ibns>.

Configuring Web Authentication

This section includes the following sections:

- [Configuring the Switch in Cisco Secure ACS, page 25](#)
- [Creating a User in Cisco Secure ACS Internal User Database, page 26](#)
- [Creating a Downloadable ACL in Cisco Secure ACS, page 27](#)
- [Creating an Authorization Profile in Cisco Secure ACS, page 28](#)
- [Creating a Web Authentication Access Service, page 30](#)
- [Creating a Web Authentication Service Selection Rule, page 32](#)
- [Configuring the Switch, page 34](#)
- [Monitoring and Troubleshooting Web Authentication, page 43](#)

This section describes how to configure a system based on Cisco IOS Software for IEEE 802.1X with WebAuth fallback. The sample configurations given in this section highlight the following features:

- WebAuth using the internal users database in Cisco Secure ACS
- WebAuth policy enforcement using dACLs
- Customized WebAuth pages
- AAA fail policy
- WebAuth monitoring

All features discussed in this section are required to configure basic WebAuth. Optional features and optimizations are discussed in later sections.

Configuring the Switch in Cisco Secure ACS

To add the switch that performs WebAuth as an AAA client in Cisco Secure ACS, complete the following steps.

Procedure

-
- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, expand **Network Resources**, select **Network Devices and AAA Clients** and click **Create**.

The window shown in [Figure 7](#) appears.

Figure 7 *Network Devices and AAA Clients*

The screenshot shows the Cisco Secure ACS Management interface. The left navigation pane is expanded to 'Network Resources' > 'Network Devices and AAA Clients'. The main pane shows the configuration for a device named 'IDF-S3-S-2-4503'. The configuration includes a description 'Catalyst 4500 Switch', a location 'All Locations: San Jose', and a device type 'All Device Types: Wired Access Switches'. The IP address is set to '10.100.10.4' with the option 'Single IP Address' selected. The authentication options are set to 'RADIUS' with a shared secret of 'cisco123'. The 'Submit' button is visible at the bottom.

Step 3 Specify the name, IP address, and RADIUS shared secret for this switch, and click **Submit**.

Optionally, add Description, Location, and Device Type information.

The RADIUS shared secret must match the key configured on the switch. The IP address must match the IP address of the RADIUS source interface that the switch uses to source RADIUS packets for Cisco Secure ACS. See the [“Configuring the Switch” section on page 34](#) for information about how to configure the key and the RADIUS source interface on the switch.

Creating a User in Cisco Secure ACS Internal User Database

In this section, a WebAuth user is created in the Cisco Secure ACS internal user database. External databases can also be used for WebAuth. See the Cisco Secure ACS product documentation for more information about configuring external databases. To create a WebAuth user in the Cisco Secure ACS internal user database, complete the following steps.

Procedure

Step 1 Open the Cisco Secure ACS Management interface.

Step 2 In the left navigation column, under Users and Identity Stores, expand **Internal Identity Stores** and select **Users**.

Step 3 Click **Create**.

The window shown in [Figure 8](#) appears.

Figure 8 *Users and Identity Stores*

Step 4 Enter a name and password for the WebAuth user and click **Submit**.

Creating a Downloadable ACL in Cisco Secure ACS

To create a downloadable ACL in Cisco Secure ACS to be used in the authorization policy in a subsequent step, complete the following steps.

Procedure

- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, under Policy Elements, expand **Named Permission Objects** and select **Downloadable ACLs**, and click **Create**
- The window shown in [Figure 9](#) appears.

Figure 9 *Downloadable ACLs*

Step 3 Enter a name for this dACL, specify the ACL elements, and click **Submit**

A very simple example (granting full network access) is shown. The syntax for the ACL must conform to the requirements for extended Layer 3 ACLs in Cisco IOS Software with the additional requirement that the source of the ACL must be “any.”



Tip

Type carefully. Cisco Secure ACS does not perform any syntax checking, but the switch fails authorization if the ACL is not properly specified. To the end user, this appears as a failed authentication.

The switch replaces the source “any” in each ACL element with the IP address of the end host when it applies the dACL to the port.

Creating an Authorization Profile in Cisco Secure ACS

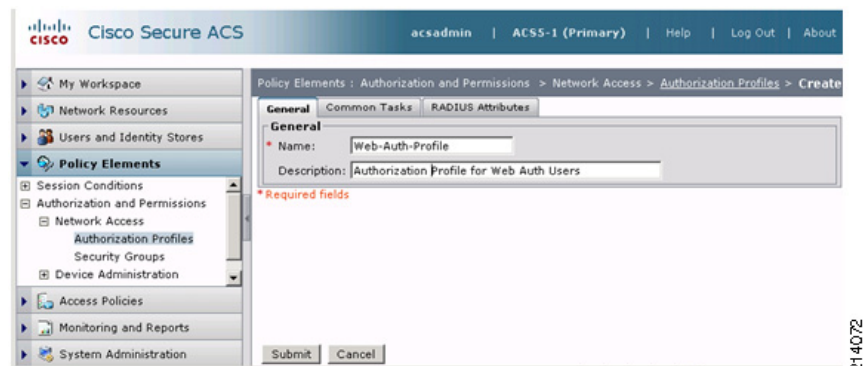
In this section, an authorization profile is created in Cisco Secure ACS. This profile is used in the authorization policy in a subsequent step. The authorization profile has three parts: the profile name, a dACL, and a manually entered RADIUS attribute that enables the switch to apply the dACL. To create an authorization profile, complete the following steps.

Procedure

- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, under Policy Elements, expand **Network Access**, select **Authorization Profiles**, and click **Create**.

The window shown in [Figure 10](#) appears.

Figure 10 *Authorization Profiles*



- Step 3** On the General tab, specify a name for this profile.
- Step 4** Click the **Common Tasks** tab, under ACLs, navigate to Downloadable ACL Name, choose **Static** and then the name of the dACL configured in the previous step.

In this example, the dACL is PERMIT-IP-ANY-ANY (see [Figure 11](#)).

Figure 11 Authorization Profiles—Downloadable ACL Name

Policy Elements : Authorization and Permissions > Network Access > Authorization Profile

General Common Tasks RADIUS Attributes

VLAN ID/Name: Not in Use
 URL for Redirect: Not in Use
 URL Redirect ACL: Not in Use

ACL
 Downloadable ACL Name: Static Value: PERMIT-IP-ANY-ANY
 IOS ACL Filter ID: Not in Use
 Proxy ACL: Not in Use

QoS
 Input Policy Map: Not in Use
 Output Policy Map: Not in Use

Submit Cancel

Step 5 Click the **RADIUS Attributes** tab, and do the following:

- For Dictionary Type, choose **RADIUS-Cisco**
- For RADIUS Attribute, choose **cisco-av-pair**
- For Attribute Value, choose **Static** and type **priv-lvl=15** in the text box (see [Figure 12](#)).
- Click **Add** and then click **Submit**.



Note

The cisco-av-pair attribute *priv-lvl=15* is a special attribute that is required to enable the switch to apply the dACL. Without this attribute, the switch fails authorization, and the WebAuth user does not get access to the network.

Figure 12 RADIUS Attributes

Policy Elements : Authorization and Permissions > Network Access > Authorization Profile

General Common Tasks RADIUS Attributes

Common Tasks Attributes

Attribute	Type	Value
cisco-av-pair	String	priv-lvl=15

Manually Entered

Attribute Type Value

cisco-av-pair String priv-lvl=15

Add Edit Replace Delete

Dictionary Type: RADIUS-Cisco

* RADIUS Attribute: cisco-av-pair Select

* Attribute Type: String

* Attribute Value: Static

* priv-lvl=15

* Required fields

Submit Cancel

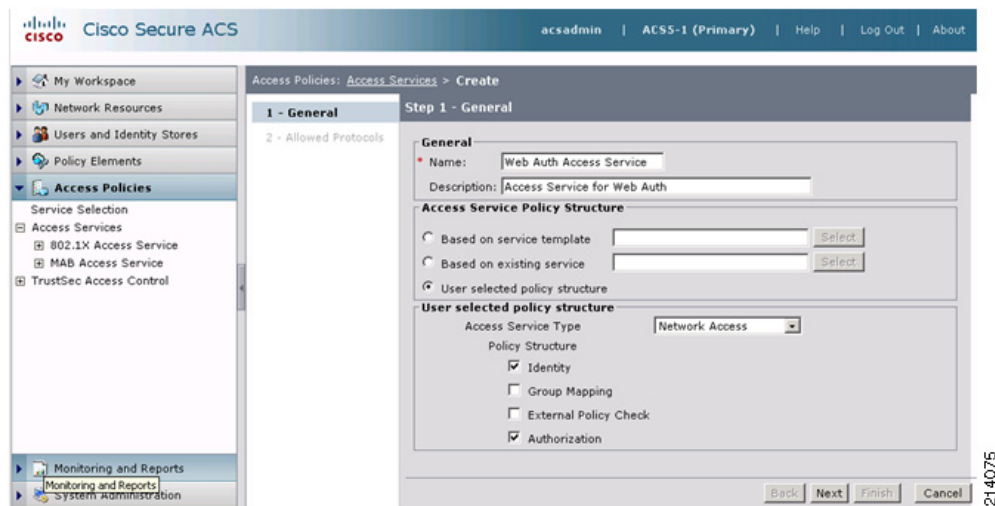
Creating a Web Authentication Access Service

In this section, a WebAuth access service is created in Cisco Secure ACS. This access service is used in the service selection rules in a subsequent step. The access service profile has four parts: the service name, the allowed protocol filter, the identity policy, and the authorization policy. To create a WebAuth access service, complete the following steps.

Procedure

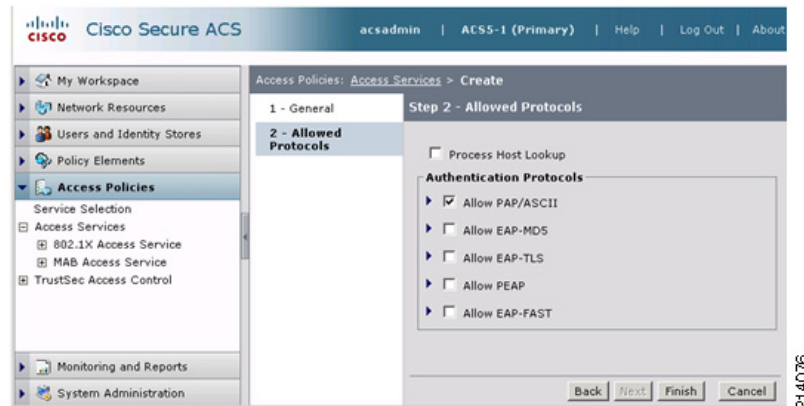
- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, under Access Policies, click **Access Services**.
The list of existing access services appears.
- Step 3** At the bottom of the right window pane, click **Create**.
The window shown in [Figure 13](#) appears.

Figure 13 Access Policies



- Step 4** Complete the General page as follows:
 - a. General: specify a name for this service.
 - b. Access Service Policy Structure: select **User selected policy structure**
 - c. Access Service Type: choose **Network Access**
 - d. Policy Structure: select **Identity** and **Authorization**.
 - e. Click **Next**.

The window shown in [Figure 14](#) appears.

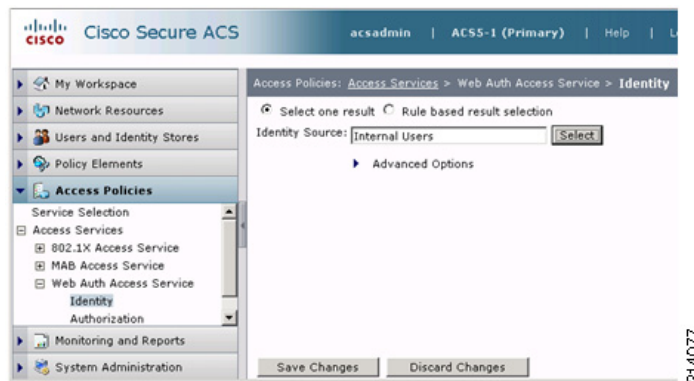
Figure 14 Access Policies—Allowed Protocols

Step 5 Complete the Allowed Protocols page as follows:

- a. Deselect **Process Host Lookup**.
- b. Select **Allow PAP/ASCII**.
- c. Click **Finish**.
- d. When prompted to modify the service selection policy, click **No**.

Step 6 In the left navigation column, expand **Access Policies** to list the access service that was just created, expand **Web Auth Access Service** and click **Identity**.

The window shown in [Figure 15](#) appears.

Figure 15 Access Policies—Access Services

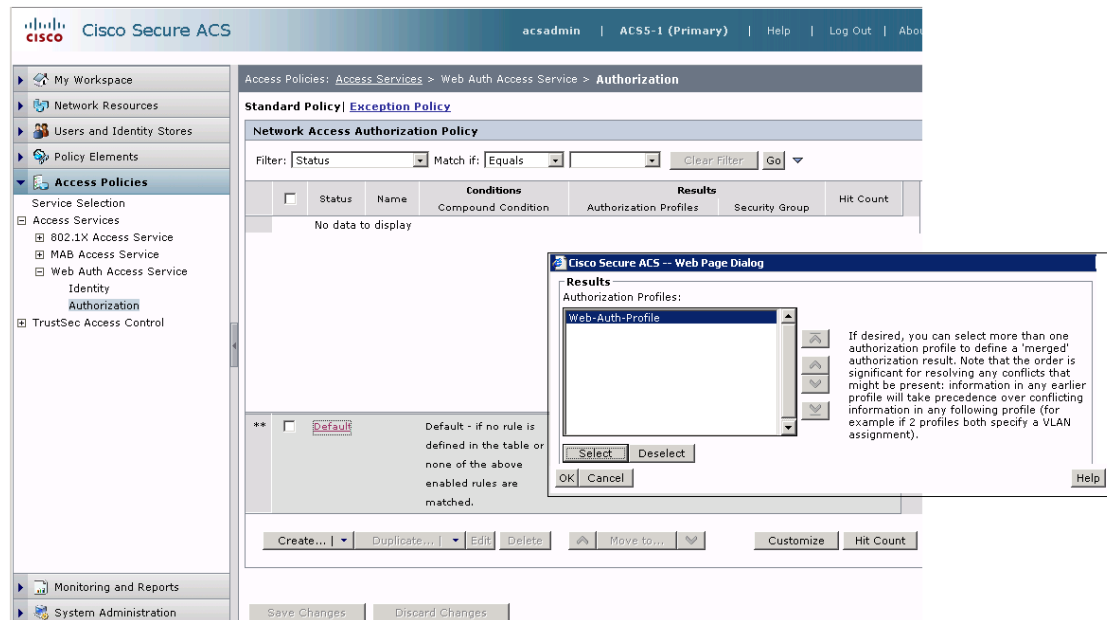
Step 7 In the Web Auth Access Service Identity policy window, select **Select one result**, for Identity Source, choose **Internal Users**, and click **Save Changes**.

Step 8 Select the Default rule for the Web Auth Access Service in the left navigation column:

- a. Expand **Access Policies** to list the access service that was just created.
- b. Expand **Web Auth Access Service** and click **Authorization**.
- c. Scroll to the bottom of the Network Access Authorization Policy rule table and click the **Default** rule.

An Authorization Profiles dialog box appears.

Step 9 Choose the authorization profile that was created earlier and click **OK** (see [Figure 16](#)).

Figure 16 Authorization Profiles

Step 10 Click **Save Changes**.

Creating a Web Authentication Service Selection Rule

This section describes how to create a service selection rule for WebAuth in Cisco Secure ACS. This service selection rule helps to ensure that the policies defined in the WebAuth access service applied to WebAuth requests. To create a service selection rule for WebAuth, complete the following steps.

Procedure

- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, under Access Policies, click **Service Selection**.
The list of existing service selection rules appears.
- Step 3** At the bottom of the right window pane, click **Create**.
The Service Selection Rule dialog box appears (see [Figure 17](#)).

Figure 17 **Service Selection Rule**

Cisco Secure ACS -- Web Page Dialog

General
 Name: Web Auth Service Sele Status: Enabled

The Customize button in the lower right area of the policy rules screen controls which policy conditions and results are available here for use in policy rules.

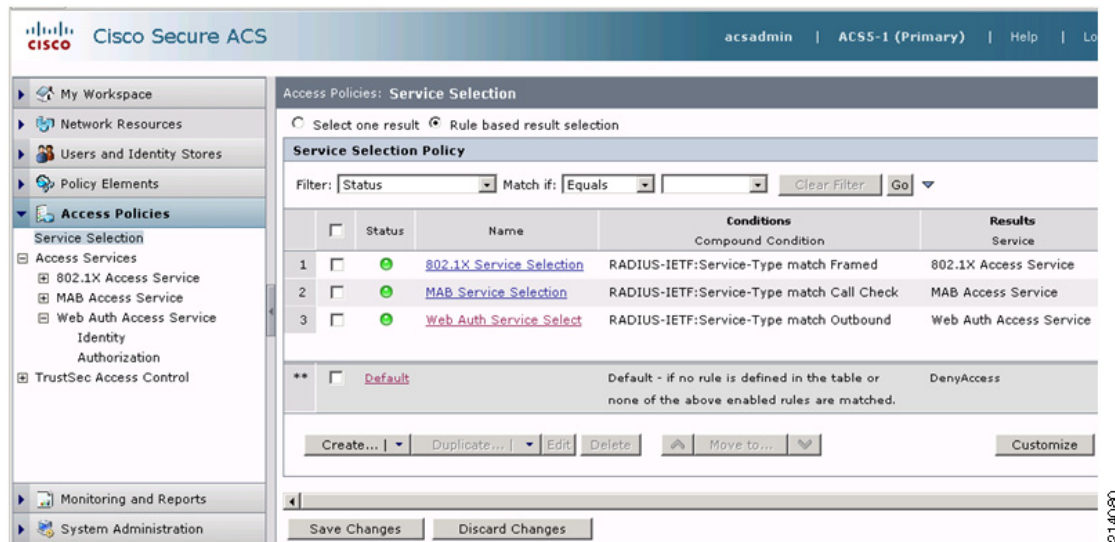
Conditions
☒ Compound Condition:
Condition:
 Dictionary: RADIUS-IETF Attribute: Service-Type Select
 Operator: match Value: Outbound Select
Current Condition Set:
 Add V Edit ^ Replace V
 RADIUS-IETF:Service-Type match Outbound
 And > Or >
 Delete Preview

Results
 Service: Web Auth Access Service

OK Cancel Help

- Step 4** Complete the Service Selection Rule as follows:
- Specify a name for the rule (*Web Auth Service Select* is used here).
 - Under Conditions, select **Compound Condition**.
 - Under Dictionary, choose **RADIUS-IETF**.
 - Under Attribute, select **Service-Type**.
 - Under Operator, choose **match**.
 - Under Value, select **Outbound**.
 - Under Current Condition Set, click **Add**.
 - Under Results, select the access service that was created in the previous step (*Web Auth Access Service*).
 - Click **OK**.

The Service Selection rule summary appears with the new rule, as shown in [Figure 18](#).

Figure 18 **Service Selection Rule Summary**

Step 5 Click **Save Changes**.

Configuring the Switch

In this section, the Cisco IOS Software switch is configured for WebAuth. Everything in this section is required to configure basic WebAuth. Optional features and optimizations are discussed in later sections.

This section includes the following topics:

- [Verifying Existing IEEE 802.1X Configuration, page 35](#)
- [Enabling AAA for Web Authentication, page 36](#)
- [Enabling IP Device Tracking, page 38](#)
- [Enabling HTTP and HTTPS, page 38](#)
- [Creating a Web Authentication Fallback Profile, page 38](#)
- [Assigning the Web Authentication Fallback Profile to an Interface, page 39](#)
- [Reviewing the Configuration, page 39](#)
- [Modifying Web Authentication Timers \(Optional\), page 40](#)
- [Configuring Customized Web Pages \(Optional\), page 41](#)
- [Supporting External Links in Customized Pages \(Optional\), page 42](#)
- [Configuring Web Authentication AAA Fail Policy \(Optional\), page 42](#)
- [Enabling Web Authentication for IEEE 802.1X Failures \(Optional\), page 42](#)
- [Configuring the IP Admission Watch List \(Optional\), page 43](#)

Verifying Existing IEEE 802.1X Configuration

Because WebAuth is being configured as a fallback for IEEE 802.1X authentication, the first step is to verify the existing IEEE 802.1X configuration. A basic configuration of IEEE 802.1X includes global AAA settings, global RADIUS settings, global IEEE 802.1X settings, and interface IEEE 802.1X settings. These settings are summarized in [Table 2](#). An example of a working configuration appears at the end of this section.

Table 2 Cisco IOS Software WebAuth Settings

Cisco IOS Software AAA Settings Prior to Configuring WebAuth	
aaa new-model	Enables the AAA control model
aaa authentication dot1x default group {radius group-name}	Specifies the authentication method for IEEE 802.1X radius: Uses the list of all RADIUS servers configured with the radius-server host command group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument
aaa authorization network default group {radius group-name}	Specifies the authorization method for IEEE 802.1X; this command allows the switch to enforce authorization policies sent by the AAA server radius: Uses the list of all RADIUS servers configured with the radius-server host command group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument
aaa accounting dot1x default start-stop group {radius group-name}	Specifies the accounting method for IEEE 802.1X radius: Uses the list of all RADIUS servers configured with the radius-server host command group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument
Cisco IOS Software RADIUS Settings Prior to Configuring WebAuth	
radius-server host {hostname ip-address} [key string]	Specifies a RADIUS server The value of the key string defined here <i>must</i> match the shared secret configured for this switch on the Cisco Secure ACS.
ip radius source-interface subinterface-name	Specifies a source interface for RADIUS traffic sourced from the switch If there is more than one Layer 3 interface on the switch, use this command to help ensure that the switch sends RADIUS traffic with the same source address used to define the switch in the Cisco Secure ACS configuration.
radius-server vsa send	Enables the switch to recognize and use vendor-specific attributes required for authorization.
Cisco IOS Software IEEE 802.1X Global Settings Prior to Configuring WebAuth	
dot1x system-auth-control	Globally enables IEEE 802.1X port-based access control
Cisco IOS Software IEEE 802.1X Interface Settings Prior to Configuring WebAuth	
authentication port-control auto	Enables port-based authentication and causes the port to begin in the unauthorized state

Table 2 *Cisco IOS Software WebAuth Settings (continued)*

dot1x pae authenticator	Configures the interface to act only as an IEEE 802.1X authenticator and ignore any messages meant for a supplicant
dot1x timeout tx-period <i>seconds</i>	Sets the number of seconds that the switch waits for a response to an EAPoL Identity-Request packet before retransmitting the request; the default is 30 The total value of the IEEE 802.1X timeout is determined by a combination of tx-period and max-reauth-req (see below).
dot1x max-reauth-req <i>count</i>	Specifies the number of times EAPoL Identity-Request packets are retransmitted (if lost or not replied to); the default value is 2 To calculate the total timeout period when there is no IEEE 802.1X supplicant present, use the following formula: tx-period * (max-reauth-req + 1).

The following example shows a basic IEEE 802.1X configuration that should be configured prior to enabling WebAuth. The IEEE 802.1X timeout using this configuration is 15 seconds: tx-period * (max-reauth-req + 1) = 5 * 3 = 15 seconds.

```

aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa accounting dot1x default start-stop group radius
!
dot1x system-auth-control
!
interface Gigabit 1/0/5
 switchport mode access
 switchport access vlan 30
 authentication port-control auto
 dot1x pae-authenticator
 dot1x tx-period 5
!
radius-server host 10.100.10.117 1813 key cisco123
radius-server vsa send

```



Note

For detailed information about configuring IEEE 802.1X on Cisco IOS Software, see the Identity-Based Networking Services (IBNS) configuration guide at <http://www.cisco.com/go/ibns>.

Enabling AAA for Web Authentication

AAA must be enabled for WebAuth (see [Table 3](#)).

Table 3 **Additional Cisco IOS Software WebAuth Settings**

Additional Cisco IOS Software AAA Settings for WebAuth	
aaa authentication login default group {radius <i>group-name</i> }	<p>Specifies the authentication method for WebAuth</p> <p>radius: Uses the list of all RADIUS servers configured with the radius-server host command</p> <p>group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument</p>
aaa authorization auth-proxy default group {radius <i>group-name</i> }	<p>Specifies the authorization method for WebAuth; this command allows the switch to enforce authorization policies (for example, the dACL) sent by the AAA server</p> <p>radius: Uses the list of all RADIUS servers configured with the radius-server host command</p> <p>group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument</p>
aaa accounting auth-proxy default start-stop group {radius <i>group-name</i> }	<p>Specifies the accounting method for WebAuth</p> <p>radius: Uses the list of all RADIUS servers configured with the radius-server host command</p> <p>group-name: Uses a subset of RADIUS servers as defined by the aaa group server radius group-name argument</p>
radius-server vsa send authentication	<p>Enables the use of vendor-specific attributes. This command enables the switch to request downloadable ACLs from the AAA server.</p>

The following example shows the mandatory basic AAA configuration for WebAuth:

```
aaa authentication login default group radius
aaa authorization auth-proxy default group radius
aaa accounting auth-proxy default start-stop group radius
radius-server vsa send authentication
```

It is important to note that the current implementation of WebAuth requires the use of the default login authentication group as RADIUS. As soon as it is configured, the default login group applies to all login attempts for the switch, including virtual teletype terminal (VTY) and console access. Everyone attempting to use Telnet to access the switch or to access the console is required to authenticate through RADIUS. To prevent the default AAA login configuration from applying to the console and VTY sessions, define a nondefault login group and apply this to the VTYs and the console. The following example configures a group named “none” that requires no authentication on VTYs or the console:

```
aaa authentication login LINE-CON none
!
line console 0
 login authentication LINE-CON
line vty 0 4
 login authentication LINE-CON
```

Enabling IP Device Tracking

IP device tracking is required to initiate the WebAuth session state, to maintain inactivity timers, and to correctly apply dACLs for authenticated devices (see [Table 4](#)).

Table 4 *IP Device Tracking Settings for WebAuth*

Global Configuration for IP Device Tracking	
ip device tracking	Enables device tracking; the device tracking feature detects the presence of a host by monitoring DHCP and ARP traffic

The following example shows how to globally enable device tracking:

```
ip device tracking
```

Enabling HTTP and HTTPS

To use WebAuth, the switch must have an HTTP server enabled. To intercept HTTPS requests and receive credentials over an encrypted link, a secure HTTP server (HTTPS) must also be enabled (see [Table 5](#)).

Table 5 *HTTP and HTTPS Settings for WebAuth*

Global Configuration for HTTP and HTTPS	
ip http server	Enables the HTTP server on the switch
ip http secure-server	Enables the HTTPS server on the switch (crypto images only); in images with crypto support, the switch can perform WebAuth processing for HTTPS requests as well as HTTP

The following example shows how to enable both the HTTP and HTTPS servers on the switch:

```
ip http server
ip http secure-server
```

Creating a Web Authentication Fallback Profile

The WebAuth fallback includes an IP admission rule and an access list (see [Table 6](#)).

Table 6 *WebAuth Fallback Profile Settings*

Global Configuration of WebAuth Fallback Profile	
ip admission name <i>admission-name</i> proxy http	Creates an IP admission rule to use WebAuth proxy

Table 6 **WebAuth Fallback Profile Settings**

ip access-list extended <i>access-list-name</i> <i>permit udp any any eq bootps</i> <i>permit udp any any eq domain</i>	Creates a default ACL This ACL controls access to the port before WebAuth completes. At a minimum, DNS and DHCP traffic should be allowed. The ACL elements listed here are only an example. The ACL can be as restrictive or permissive as your security policy allows.
fallback profile <i>fallback-profile</i> ip access-group <i>access-list-name</i> in ip admission <i>admission-name</i>	Creates a fallback profile; this profile must include an IP admission rule and the ACL

The following example shows a WebAuth fallback profile:

```
ip admission name IP_ADMIN_RULE proxy http

ip access-list extended PRE_WEBAUTH_POLICY
 permit udp any any eq bootps
 permit udp any any eq domain

fallback profile WEB_AUTH_PROFILE
 ip access-group PRE_WEBAUTH_POLICY in
 ip admission IP_ADMIN_RULE
```

Assigning the Web Authentication Fallback Profile to an Interface

The fallback profile must be assigned to an interface to take effect. The same profile can be assigned to multiple interfaces (see [Table 7](#)).

Table 7 **Interface Settings for WebAuth Fallback Profile**

Interface Configuration for WebAuth Fallback Profile	
authentication fallback <i>fallback-profile</i>	Enables the specified profile on the interface

The following example shows how to assign a WebAuth fallback profile to an interface:

```
interface Gigabit 1/0/5
 authentication fallback WEB_AUTH_PROFILE
```

Reviewing the Configuration

The following example shows all the required elements of a configuration for IEEE 802.1X with WebAuth fallback in the order they would appear in the command-line interface (CLI):

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

```

aaa authorization auth-proxy default group radius
aaa accounting dot1x default start-stop group radius
aaa accounting auth-proxy default start-stop group radius
!
ip device tracking
ip admission name IP_ADMIN_RULE proxy http
!
fallback profile WEB_AUTH_PROFILE
ip access-group PRE_WEBAUTH_POLICY in
ip admission IP_ADMIN_RULE
!
dot1x system-auth-control
!
interface Gigabit 1/0/5
switchport mode access
switchport access vlan 30
authentication port-control auto
authentication fallback WEB_AUTH_PROFILE
dot1x pae-authenticator
dot1x tx-period 5
!
ip http server
ip http secure-server
!
ip access-list extended PRE_WEBAUTH_POLICY
permit udp any any eq bootps
permit udp any any eq domain
!
radius-server host 10.100.10.117 key cisco123
radius-server vsa send authentication

```

Modifying Web Authentication Timers (Optional)

This section describes how to modify the timers that affect WebAuth.

Init-State Timer (Optional)

The default init-state timer is 2 minutes. To change the init-state timer, use the following global configuration (where the timer is specified in *minutes*):

```
ip admission init-state-time 5
```

Inactivity Timeout (Optional)

By default, the switch sends IP device tracking probes at intervals of 30 seconds and declares the host inactive after three unanswered probes. To change these defaults, use the following commands (where the interval is given in *seconds*):

```
ip device tracking probe interval 200
ip device tracking probe count 2
```

Session (Absolute) Timeout (Optional)

The session timeout setting is specified in the RADIUS server configuration. The session timeout setting is controlled by IETF RADIUS attribute 27, Session Timeout. In this example, the WebAuth authorization profile in Cisco Secure ACS 5.0 is modified to send a session timeout of 600 seconds (10 minutes).

Procedure

- Step 1** Open the Cisco Secure ACS Management interface.
- Step 2** In the left navigation column, under Policy Elements, expand **Network Access** and select **Authorization Profiles**.
- Step 3** Click **Web-Auth-Profile**, created earlier and click the **Common Tasks** tab.
- The window shown in [Figure 19](#) appears.

Figure 19 Authorization Profiles—Common Tasks

The screenshot shows the Cisco Secure ACS Management interface. The left navigation pane is expanded to 'Policy Elements' > 'Network Access' > 'Authorization Profiles'. The main pane shows the 'Common Tasks' tab for the 'Web-Auth-Profile'. The 'General' tab is also visible. The 'Common Tasks' tab contains the following fields:

- VLAN ID/Name:** Not in Use
- URL for Redirect:** Not in Use
- URL Redirect ACL:** Not in Use
- ACLs**
 - Downloadable ACL Name:** Static, Value: PERMIT-IP-ANY-ANY
 - IOS ACL Filter ID:** Not in Use
 - Proxy ACL:** Not in Use
- QoS**
 - Input Policy Map:** Not in Use
 - Output Policy Map:** Not in Use
- Voice VLAN**
 - Permission to Join:** Not in Use
- Reauthentication**
 - Reauthentication Timer:** Static, Value: 600 Seconds
 - Maintain Connectivity during Reauthentication:** Yes (Termination-action=radius-request) selected, No (Termination-action=default) unselected

At the bottom, there is a '* Required fields' label and 'Submit' and 'Cancel' buttons.

- Step 4** Under Reauthentication, set Reauthentication Timer to **Static**.
- Step 5** For Value, enter the desired length of the WebAuth session and click **Submit**.



Note

The Termination-Action attribute (RADIUS attribute 29) has no effect on WebAuth.

Configuring Customized Web Pages (Optional)

To specify the use of custom authentication proxy web pages, first store the custom HTML files on the switch's internal disk or flash memory. Four pages are required: login, success, fail, and expired. The filenames are arbitrary. In the following example, the four required pages are stored on disk1, and they are named login.htm, success.htm, fail.htm, and expired.htm:

```
ip admission proxy http login page file disk1:login.htm
ip admission proxy http success page file disk1:success.htm
ip admission proxy http fail page file disk1:fail.htm
ip admission proxy http login expired page file disk1:expired.htm
```

Sample web pages are provided in the [“Sample Customizable Pages”](#) section on page 45.

Supporting External Links in Customized Pages (Optional)

Any customized pages can include external links. A common application of an external link is to reference an image stored on a different server.

Suppose your login page includes a link to a company logo stored on another server:

```
switch# more login.html | include img src
img src="http://10.100.10.119:8080/logo.jpg" alt="Company Logo">
```

As discussed in the [“Operation 4—User Gets Login Page”](#) section on page 11, the destination port in the URL must be something other than TCP port 80 or 443. In addition, the ACL in the fallback profile must allow traffic to the URL. In the preceding example, the URL of the external image has been specified as port 8080 (assuming that the HTTP server on 10.100.10.119 is listening on that port). Therefore, the preconfigured ACL must be modified to allow access to 10.100.10.119 port 8080:

```
ip access-list extended PRE_WEBAUTH_POLICY
permit udp any any eq bootps
permit udp any any eq domain
permit tcp any host 10.100.10.119 eq 8080
```

Configuring Web Authentication AAA Fail Policy (Optional)

The AAA fail policy consists of an ACL that is applied to the port when a user attempts web authentication when the AAA server is unavailable. There are three steps to configuring an AAA fail policy:

-
- Step 1** Create an access list that should be applied when the AAA server is unavailable. In the following example, the ACL grants complete access to the network:


```
ip access-list extended PERMIT
permit ip any any
```
 - Step 2** Create an identity policy that contains that ACL:


```
identity policy FAILOPEN
access-group PERMIT
```
 - Step 3** Modify the IP admission rule to call this identity policy when AAA is down:


```
ip admission name IP_ADMIN_RULE proxy http event timeout aaa policy identity FAILOPEN
```
-

Enabling Web Authentication for IEEE 802.1X Failures (Optional)

By default, WebAuth applies only when IEEE 802.1X times out because there is no supplicant on the end host. To also enable WebAuth when IEEE 802.1X fails, add the following to the switch *interface* configuration:

```
authentication event fail action next-method
```

Configuring the IP Admission Watch List (Optional)

To enable the watch list, use the following command:

```
ip admission watch-list enable
```

To change the amount of time that users on the watch list are denied access to WebAuth, use the following command (where the expiry time is specified in *minutes*):

```
ip admission watch-list expiry-time 60
```

To manually add or delete addresses from the watch list, use the following commands:

```
ip admission watch-list add-item 66.66.66.66
no ip admission watch-list add-item 10.100.10.4
```

To manually change the number of times that a user is allowed to attempt authentication before being added to the watch list, use the following command:

```
ip admission max-login-attempts 3
```

To see active entries in the watch list, use the following command:

```
switch# show ip admission watch-list
Authentication Proxy Watch-list is enabled
Watch-list expiry timeout is 30 minutes
Total number of watch-list entries: 2
  Source IP      Type      Violation-count
  66.66.66.66    CFGED      N/A
  10.100.10.56   MAX_RETRY  MAX_LIMIT
Total number of black-listed users: 2
```

Monitoring and Troubleshooting Web Authentication

This section describes how to monitor WebAuth.

- Step 1** Verify that IEEE 802.1X authentication has timed out and that the port has been authorized for the default data VLAN with the following commands:

```
switch# show authentication sessions interface G1/13
  Interface: GigabitEthernet1/13
  MAC Address: 0014.5e95.d6cc
  IP Address: 10.100.60.201
  Status: Authz Success
  Domain: DATA
  Oper host mode: multi-domain
  Oper control dir: both
  Authorized By: Authentication Server
  Vlan Policy: N/A
  Session timeout: N/A
  Idle timeout: N/A
  Common Session ID: 0A640A050000001705BD5664
  Acct Session ID: 0x00000019
  Handle: 0x20000017

Runnable methods list:
  Method      State
  dot1x       Failed over
```

webauth Authc Success

**Note**

The runnable method list should show **webauth Authc Success**. This means that IEEE 802.1X has passed control to WebAuth. It does not indicate the state of WebAuth. The **show ip admission cache** command (that follows) indicates the state of WebAuth.

Step 2 Verify that device tracking is enabled and an entry for the host exists in the device-tracking table.

This entry indicates that the switch has detected ARP or DHCP traffic from the host. The host should be in the ACTIVE state.

```
switch# show IP Device Tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
-----
  IP Address      MAC Address      Interface          STATE
-----
10.100.60.200    0014.5e95.d6cc  GigabitEthernet1/13  ACTIVE
```

Step 3 Verify that initial IP admission session state exists for the end host.

The INIT state indicates that the switch is ready to receive credentials from the host.

```
switch# show ip admission cache
Authentication Proxy Cache
Total Sessions: 1 Init Sessions: 0
Client IP 10.100.60.200 Port 0, timeout 60, state INIT
```

Step 4 After the end host enters valid credentials, verify that the IP admission state transitions to established (ESTAB).

```
switch# show ip admission cache
Authentication Proxy Cache
Client IP 10.100.60.200 Port 4884, timeout 60, state ESTAB
```

Use the reporting capabilities on Cisco Secure ACS to verify the session details, as shown in [Figure 20](#).

Figure 20 Session Details

✓=Pass ✗=Fail 🔍=Click for details										
Logged At	Status	Details	Username	Calling Station ID	Authentication Method	Selected Authorization Profiles	NAS IP Address	NAS Port	Access Service	Identity Store
9:43:17.276 PM	✓	🔍	WebAuthUser	10.100.60.200	PAP_ASCII	Web-Auth-Profile	10.100.10.5	50104	Web Auth Access Service	Internal Users

214082

[Table 8](#) summarizes some common problems encountered when configuring WebAuth.

Table 8 WebAuth Troubleshooting

Symptom	Possible Root Causes	Resolution
End user does not get IP address	Preconfigured ACL in fallback profile does not permit DHCP. End user's DHCP client times out before IEEE 802.1X falls back to WebAuth.	Permit DHCP in the fallback profile. Decrease the dot1x timeout tx-period so that fallback occurs before DHCP times out.

Table 8 **WebAuth Troubleshooting**

End user has IP address but does not receive login page	IP device tracking is not enabled. IP device tracking interval is too long.	Enable IP device tracking. Decrease the IP device tracking interval.
End user submits valid credentials but does not gain network access	Cisco Secure ACS does not send priv-lvl=15 attribute in Access-Accept message. No preconfigured ACL exists in the fallback profile. Incorrect syntax is used in the dACL.	Add priv-lvl=15 to the WebAuth authorization profile. Add an ACL to the fallback profile. Correct the dACL.

References

This section provides a list of references.

- Configuring WebAuth on the Cisco Catalyst 3750 Series Switches—
http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/sw8021x.html
- Configuring WebAuth on the Cisco Catalyst 4500 Series Switches—
<http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst4500/12.2/53SG/configuration/webauth.html>
- Configuring WebAuth on the Cisco Catalyst 6500 Series Switches—
<http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/webauth.html>
- Scenario-based IEEE 802.1X design guide—
http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/whitepaper_C11-530469.html
- Scenario-based IEEE 802.1X configuration guide—
http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/Whitepaper_c11-532065.html
- Cisco IOS Firewall authentication proxy—
http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a0080094eb0.shtml
- WebAuth with Cisco Wireless LAN Controllers—
http://www.cisco.com/en/US/partner/tech/tk722/tk809/technologies_configuration_example09186a008076f974.shtml#external-process

Sample Customizable Pages

The following section contains sample web pages that can be used to begin customization.

Login Page

```
<HTML><HEAD><TITLE>Authentication Proxy Login Page</TITLE>
<script language="JavaScript"><!-- Begin
var pxypromptwindow1;
var pxysubmitted = false;
```

```

function doreload() {
if(pxypromptwindow1.closed)
{window.location.reload(true);
} else {reloadtimeout=setTimeout("doreload()", 300);}
}
function submitreload() {
if(pxysubmitted == false)
{pxypromptwindow1=window.open('',
'pxywindow1','resizable=no,width=300,height=300,scrollbars=yes');reloadtimeout=setTimeout(
"doreload()", 1000);pxysubmitted = true;return true;
} else {
alert("This page can not be submitted twice.");
return false;
}
} // -->
</script> </HEAD>
<BODY BGCOLOR="#FFFFFF" LINK="#ffcc00" ALINK="#ffffff" VLINK="#ffcc00" >
<H1> <BR><BR>
<FORM method=post action="/" target="pxywindow1">
<input type=hidden name=au_pxytimetag value="612790020">Username: <input type=text
name=uname>
<BR><BR>
Password: <input type=password name=pwd>
<BR><BR><input type=submit name=ok value=OK onClick="return submitreload()">
</H1></FORM></script></BODY></HTML>
Expired Page
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
        <title>Authentication Proxy Login Failure Page</title>
    </head>
    <body>
        <noscript>
            <h1 id="noScript">Please enable Javascript</h1>
        </noscript>
        <h1 id="expire">Expired</h1>
    </body>
</html>
Login Failed Page
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
        <title>Authentication Proxy Login Failure Page</title>
    </head>
    <body>
        <noscript>
            <h1 id="noScript">Please enable Javascript</h1>
        </noscript>
        <h1 id="failure">Failed</h1>
    </body>
</html>
Login Success Page
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
        <title>Authentication Proxy Login Page</title>
    </head>
    <body>
        <script type="text/javascript">
            var q = window.location.search;
            q = q.replace('?', '');
            q = q.split('&');
            for(var i in q) {
                var aux = q[i].split('=');

```

```

        if (aux[0] == 'redirect_url') {
            var site = aux[1];
            break;
        }
    }
    if (site) {
        window.location.replace(unescape(site));
    }
}
</script>
<h1>You are now logged in</h1>
</body>
</html>

```

References

TrustSec 1.99 Documents

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

Related Documents

- Configuring WebAuth on the Cisco Catalyst 3750 Series Switches—
http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/sw8021x.html

- Configuring WebAuth on the Cisco Catalyst 4500 Series Switches—
<http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst4500/12.2/53SG/configuration/webauth.html>
- Configuring WebAuth on the Cisco Catalyst 6500 Series Switches—
<http://www.cisco.com/en/US/partner/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/webauth.html>
- Cisco IOS Firewall authentication proxy—
http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a0080094eb0.shtml
- WebAuth with Cisco Wireless LAN Controllers—
http://www.cisco.com/en/US/partner/tech/tk722/tk809/technologies_configuration_example09186a008076f974.shtml#external-process