Understand Web Authentication on Wireless LAN Controllers (WLC)

Contents

Introduction

This document describes the processes for Web Authentication on Wireless LAN Controllers (WLC).

Prerequisites

Requirements

Cisco recommends that you have basic knowledge of WLC configuration.

Components Used

The information in this document is based on all WLC hardware models.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Web Authentication Inner Processes

Web Authentication Position as a Security Feature

Web authentication (WebAuth) is Layer 3 security. It allows for user-friendly security that works on any station that runs a browser.

It can be combined with any pre-shared key (PSK) security (Layer 2 security policy).

Although the combination of WebAuth and PSK reduces the user-friendly portion, it has the advantage to encrypt client traffic.

WebAuth is an authentication method without encryption.

WebAuth cannot be configured with 802.1x/RADIUS (Remote Authentication Dial-In User Service) until the WLC Software Release 7.4 is installed and configured simultaneously.

Clients must go through both dot1x and web authentication. It is intended for the addition of a web portal for employees (who use 802.1x), not guests.

There is not an all-in-one service set identifier (SSID) for dot1x for employees or web portal for guests.

How WebAuth Works

The 802.11 authentication process is open, so you can authenticate and associate without any problems. After that, you are associated, but not in the WLC **RUN** state.

With web authentication enabled, you are kept in **WEBAUTH_REQD** where you cannot access any network resource.

You must receive a DHCP IP address with the address of the DNS server in the options.

Type a valid URL in your browser. The client resolves the URL through the DNS protocol. The client then sends its HTTP request to the IP address of the website.

The WLC intercepts that request and returns the **webauth** login page, which mimics the website IP address. With an external WebAuth, the WLC replies with an HTTP response that includes your website IP address and states that the page has moved.

The page was moved to the external web server used by the WLC. When you are authenticated, you gain access to all of the network resources and are redirected to the originally requested URL by default (unless a forced redirect was configured on the WLC).

In summary, the WLC allows the client to resolve the DNS and get an IP address automatically in **WEBAUTH_REQD** state.

To watch another port instead of port 80, use **config network web-auth-port <port number>** to create a redirect on this port also.

An example is the Access Control Server (ACS) web interface, which is on port 2002 or other similar applications.

> **Note about HTTPS Redirection**: By default, the WLC did not redirect HTTPS traffic. This means that if you type an HTTPS address into your browser, nothing happens. You must type an HTTP address in order to get redirected to the login page which was served in HTTPS.
>
> In Version 8.0 and later, you can enable redirection of HTTPS traffic with the CLI command **config network web-auth https-redirect enable.**
>
> This uses a lot of resources for the WLC in cases where many HTTPS requests are sent. It is not advisable to use this feature before WLC version 8.7 where the scalability of this feature was enhanced. Also note that a certificate warning is unavoidable in this case. If the client requests any URL (such as https://www.cisco.com), the WLC still presents its own certificate issued for the virtual interface IP address. This never matches the URL/IP address requested by client and the certificate is not trusted unless the client forces the exception in their browser.

Indicative performance drop of WLC software release before 8.7 measured :

Webauth	Rate achieved
3 URLs - HTTP	140 / second
1st URL - HTTP 2nd and 3rd URLs - HTTPS	20 / second
3 URLs - HTTPS (large deployment)	< 1 / second
3 URLs - HTTPS (max of 100 clients)	10 / second

In this performance table, the 3 URLs are referred to as:

- The original URL entered by the end-user
- The URL to which the WLC redirects the browser
- The final credentials submission

The performance table gives the WLC performance in case all 3 URLs are HTTP, in case all 3 URLs are HTTPS, or if the client moves from HTTP to HTTPS (typical).

How to Make an Internal (Local) WebAuth Work with an Internal Page

To configure a WLAN with an operational dynamic interface, the clients also receive a DNS server IP address through DHCP.

Before any **webauth** , is set, verify that WLAN works properly, DNS requests can be resolved (**nslookup**), and web pages can be browsed.

Set the web authentication as Layer 3 security features. Create users in the local database or on an external

RADIUS server.

Refer to the Wireless LAN Controller Web Authentication Configuration Example document.

How to Configure a Custom Local WebAuth with Custom Page

Custom **webauth** can be configured with **redirectUrl** from the **Security** tab. This forces a redirect to a specific web page which you enter.

When the user is authenticated, it overrides the original URL which the client requested and displays the page for which the redirect was assigned.

The custom feature allows you to use a custom HTML page instead of the default login page. Upload your html and image files bundle to the controller.

In the upload page, look for **webauth bundle** in a tar format. PicoZip creates tars that work compatibly with the WLC.

For an example of a WebAuth bundle, refer to the Download Software page for Wireless Controller WebAuth Bundles. Select the appropriate release for your WLC.

It is recommended to customize a bundle that exists; do not create a new bundle.

There are some limitations with **custom webauth** that vary with versions and bugs.

- the .tar file size (no more than 5MB)
- the number of files in the .tar
- the filename length of the files (no more than 30 characters)

If the package does not work, attempt a simple custom package. Individually add files and complexity to reach the package that the user tried to use. This helps to identify the problem.

To configure a custom page, refer to Creating a Customized Web Authentication Login Page, a section within the Cisco Wireless LAN Controller Configuration Guide, Release 7.6.

Override Global Configuration Technique

Configure with the **override global config** command and set a WebAuth type for each WLAN. This permits an internal/default WebAuth with a custom internal/default WebAuth for another WLAN.

This allows configuration of different custom pages for each WLAN.

Combine all pages in the same bundle and upload them to the WLC.

Set your custom page with the **override global config** command on each WLAN and select which file is the login page from all of the files within the bundle.

Choose a different login page inside the bundle for each WLAN.

Redirection Issue

There is a variable within the HTML bundle that allows the redirection. Do not put your forced redirection URL there.

For redirection issues in custom WebAuth, Cisco recommends to check the bundle.

If you enter a redirect URL with += in the WLC GUI, this could overwrite *or* add to the URL defined inside the bundle.

For example, in the WLC GUI, the **redirectURL** field is set to www.cisco.com; however, in the bundle it shows: **redirectURL**+= '(website URL)'. The += redirects users to an invalid URL.

How to Make an External (Local) Web Authentication Work with an External Page

Utilization of an external WebAuth server is just an external repository for the login page. The user credentials are still authenticated by the WLC. The external web server allows only a special or different login page.

Steps performed for an external WebAuth:

1. The client (end user) opens a web browser and enters a URL.
2. If the client is not authenticated and external web authentication is used, the WLC redirects the user to the external web server URL. The WLC sends an HTTP redirect to the client with the imitated IP address and points to the external server IP address. The external web authentication login URL is appended with parameters such as the **AP_Mac_Address**, the **client_url** (**client URL address**), and the **action_URL** needed to contact the switch web server.
3. The external web server URL sends the user to a login page. The user can use a pre-authentication access control list (ACL) to access the server.
4. The login page sends the user credentials request back to the **action_URL**, such as http://192.0.2.1/login.html, of the WLC web server. This is provided as an input parameter to the redirect URL, where 192.0.2.1 is the virtual interface address on the switch.
5. The WLC web server submits the username and password for authentication.
6. The WLC initiates the RADIUS server request or uses the local database on the WLC, and then authenticates the user.
7. If authentication is successful, the WLC web server either forwards the user to the configured redirect URL or to the URL the client entered.
8. If authentication fails, then the WLC web server redirects the user back to the user login URL.

Note : We use 192.0.2.1 as an example of virtual ip in this document. The 192.0.2.x range is advised for use for virtual ip as it is non-routable. Older documentation possibly refers to "1.1.1.x" or is still what is configured in your WLC as this used to be the default setting. However, note that this ip now a valid routable ip address and therefore the 192.0.2.x subnet is advised instead.

If the access points (APs) are in FlexConnect mode, a **preauth** ACL is irrelevant. Flex ACLs can be used to allow access to the web server for clients that have not been authenticated.

Refer to the External Web Authentication with Wireless LAN Controllers Configuration Example.

Web Passthrough

Web Passthrough is a variation of the internal web authentication. It displays a page with a warning or an alert statement, but does not prompt for credentials.

The user then clicks **ok**. Enable email input and the user can enter their email address which becomes their username.

When the user is connected, check your active clients list and verify that user is listed with the email address they entered as the username.

For more information, refer to the Wireless LAN Controller 5760/3850 Web Passthrough Configuration Example.

Conditional Web Redirect

If you enable a conditional web redirect, the user is conditionally redirected to a particular web page after 802.1x authentication has successfully completed.

You can specify the redirect page and the conditions under which the redirect occurs on your RADIUS server.

Conditions can include the password when it reaches the expiration date or when the user needs to pay a bill for continued use/access.

If the RADIUS server returns the Cisco AV-pair **url-redirect**, then the user is redirected to the specified URL when they open a browser.

If the server also returns the Cisco AV-pair **url-redirect-acl**, then the specified ACL is installed as a pre-authentication ACL for this client.

The client is not considered fully authorized at this point and can only pass traffic allowed by the pre-authentication ACL. After the client completes a particular operation at the specified URL (for example, a password change or bill payment), then the client must re-authenticate.

When the RADIUS server does not return a **url-redirect**, the client is considered fully authorized and allowed to pass traffic.

> **Note**: The conditional web redirect feature is available only for WLANs that are configured for 802.1x or WPA+WPA2 Layer 2 security.

After configuration of the RADIUS server, configure the conditional web redirect on the controller with the controller GUI or CLI. Refer to these step-by-step guides: Configuring Web Redirect (GUI) and Configuring Web Redirect (CLI).

Splash Page Web Redirect

If you enable splash page web redirect, the user is redirected to a particular web page after 802.1x authentication has completed successfully. After the redirect, the user has full access to the network.

You can specify the redirect page on your RADIUS server. If the RADIUS server returns the Cisco AV-pair **url-redirect**, then the user is redirected to the specified URL when they open a browser.

The client is considered fully authorized at this point and is allowed to pass traffic, even if the RADIUS server does not return a **url-redirect.**

> **Note**: The splash page redirect feature is available only for WLANs that are configured for 802.1x or WPA+WPA2 Layer 2 security.

After configuration of the the RADIUS server, configure the splash page web redirect on the controller with the controller GUI or CLI.

WebAuth on MAC Filter Failure

A WebAuth on MAC Filter FaFailure quires you to configure MAC filters on the Layer 2 security menu.

If users are successfully validated with their MAC addresses, then they go directly to the **run** state.

If they are not, then they go to the **WEBAUTH_REQD** state and the normal web authentication occurs.

> **Note**: This is not supported with web passthrough.For more information, observe the activity on enhancement request Cisco bug ID CSCtw73512

Central Web Authentication

Central Web Authentication refers to a scenario where the WLC no longer hosts any services. The client is directly sent to the ISE web portal and does not go through 192.0.2.1 on the WLC. The login page and the entire portal are externalized.

Central Web Authentication takes place when you have RADIUS Network Admission Control (NAC) enabled in the advanced settings of the WLAN and MAC filters enabled.

The WLC sends a RADIUS authentication (usually for the MAC filter) to ISE, which replies with the **redirect-url** attribute value (AV) pair.

The user is then put in **POSTURE_REQD** state until ISE gives the authorization with a Change of Authorization (CoA) request. The same scenario happens in Posture or Central WebAuth.

Central WebAuth is not compatible with WPA-Enterprise/802.1x because the guest portal cannot return session keys for encryption like it does with Extensible Authentication Protocol (EAP).

External User Authentication (RADIUS)

External User Authentication (RADIUS) is only valid for Local WebAuth when WLC handles the credentials, or when a Layer 3 web policy is enabled. Authenticate users locally or on the WLC or externally via RADIUS.

There is an order in which the WLC checks for the credentials of the user.

1. In any case, it first looks in its own database.
2. If it does not find the users there, it goes to the RADIUS server configured in the guest WLAN (if there is one configured).

3. It then checks in the global RADIUS server list against the RADIUS servers where **network user** is checked.

This third point answers the question of those who do not configure RADIUS for that WLAN, but notice that it still checks against the RADIUS when the user is not found on the controller.

This is because **network user** is checked against your RADIUS servers in the global list.

WLC can authenticate users to RADIUS server with Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP) or EAP-MD5 (Message Digest5).

This is a global parameter and is configurable from GUI or CLI:

From GUI: navigate to **Controller** > **Web RADIUS Authentication**

From CLI: enter **config custom-web RADIUSauth <pap|chap|md5chap>**

Note:The NAC guest server only uses PAP.

How to set a Wired Guest WLAN

A Wired Guest WLAN configuration is similar to wireless guest configuration. It can be configured with one or two controllers (only if one is auto-anchor).

Choose a VLAN as the VLAN for wired guest users, for example, on VLAN 50. When a wired guest wants access to the Internet, plug the laptop to a port on a switch configured for VLAN 50.

This VLAN 50 must be allowed and present on the path through the WLC trunk port.

In a case of two WLCs (one anchor and one foreign), this wired guest VLAN must lead to the foreign WLC (named WLC1) and not to the anchor.

WLC1 then takes care of the traffic tunnel to the DMZ WLC (the anchor, named WLC2), which releases the traffic in the routed network.

Here are the five steps to configure wired guest access:

1. **Configure a dynamic interface (VLAN) for wired guest user access.**

   On WLC1, create a dynamic interface VLAN50. In the **interface configuration** page, check the **Guest LAN** box. Then, fields such as **IP address** and **gateway** disappear. The WLC needs to recognize that traffic is routed from VLAN 50. These clients are wired guests.

2. **Create a wired LAN for guest user access.**

   On a controller, an interface is used when associated to a WLAN. Next, create a WLAN on your main office controllers. Navigate to **WLANs** and click **New**. In **WLAN Type**, choose **Guest LAN**.

   In **Profile Name** and **WLAN SSID**, enter a name that identifies this WLAN. These names can be different, but cannot contain spaces. The term WLAN is used, but this network profile is not related to wireless network profile.

   The **General** tab offers two drop-down lists: **Ingress** and **Egress**. Ingress is the VLAN from which users

come (VLAN 50); Egress is the VLAN to which you send them.

For **Ingress**, choose **VLAN50**.

For **Egress**, it is different. If you have only one controller, create another dynamic interface, a **standard** one this time (not a guest LAN), and send wired users to this interface. In this case, send them to the DMZ controller. Therefore, for the **Egress** interface, choose the **Management Interface**.

The **Security** mode for this Guest LAN "WLAN" is WebAuth, which is acceptable. Click **Ok** in order to validate.

3. **Configure the foreign controller (main office).**

   From the **WLAN list**, click **Mobility Anchor** at the end of the **Guest LAN** line, and choose your DMZ controller. It is assumed here that both controllers recognize each other. If they do not, go to **Controller** > **Mobility Management** > **Mobility group**, and add **DMZWLC** on WLC1. Then add **WLC1** on DMZ. Both controllers are not to be in the same mobility group. Otherwise, basic security rules are broken.

4. **Configure the anchor controller (the DMZ controller).**

   The main office controller is ready. Now prepare your DMZ controller. Open a web browser session to your DMZ controller and navigate to **WLANs**. Create a new WLAN. In **WLAN Type**, choose **Guest LAN**.

   In **Profile Name** and **WLAN SSID**, enter a name that identifies this WLAN. Use the same values as entered on the main office controller.

   The **Ingress** interface here is **None**. It does not matter because the traffic is received through the Ethernet over IP (EoIP) tunnel. There is no need to specify any Ingress interface.

   The **Egress** interface is where the clients are to be sent. For example, the **DMZ VLAN** is VLAN 9. Create a standard dynamic interface for VLAN 9 on your DMZWLC, then choose **VLAN 9** as the Egress interface.

   Configure the end of the Mobility Anchor tunnel. From the **WLAN list**, choose **Mobility Anchor for Guest LAN**. Send the traffic to the local controller, **DMZWLC**. Both ends are now ready.

5. **Fine-tune the guest LAN.**

   You can also fine-tune the WLAN settings on both ends. The settings must be identical on both ends. For example, if you click in the **WLAN Advanced** tab, **Allow AAA override** on WLC1, check the same box on DMZWLC. If there are any differences in the WLAN on either side, the tunnel breaks. DMZWLC refuses the traffic; you can see when you **run debug mobility**.

   Keep in mind that all values are actually obtained from DMZWLC: IP addresses, VLAN values, and so on. Configure the WLC1 side identically, so that it relays the request to the WLCDMZ.

Certificates for the Login Page

This section provides the processes to put your own certificate on the WebAuth page, or to hide the 192.0.2.1 WebAuth URL and display a named URL.

Upload a Certificate for the Controller Web Authentication

Through the GUI (**WebAuth** > **Certificate**) or CLI (transfer type **webauthcert**) you can upload a certificate on the controller.

Whether it is a certificate created with your certificate authority (CA) or a third-party official certificate, it must be in .pem format.

Before you send, you must also enter the key of the certificate.

After the upload, a reboot is required in order for the certificate to be in place. Once rebooted, go to the WebAuth certificate page in the GUI to find the details of the certificate you uploaded (validity and so on).

The important field is the common name (CN), which is the name issued to the certificate. This field is discussed in this document under the section "Certificate Authority and Other Certificates on the Controller".

After you reboot and verify the details of the certificate, you are presented with the new controller certificate on the WebAuth login page. However, there can be two situations.

1. If your certificate has been issued by one of the few main root CAs that every computer trusts, then it is okay. An example is VeriSign, but you are usually signed by a Verisign sub-CA and not the root CA. You can check in your browser certificate store if you see the CA mentioned there as trusted.

2. If you got your certificate from a smaller company/CA, all computers do not trust them. Provide the company/CA certificate to the client as well, and one of the root CAs then issues that certificate. Eventually, you have a chain such as "Certificate has been issued by CA x > CA x certificate has been issued by CA y > CA y certificate has been issued by this trusted root CA". The end goal is to reach a CA that the client does trust.

Certificate Authority and Other Certificates on the Controller

In order to be rid of the warning "this certificate is not trusted", enter the certificate of the CA that issued the controller certificate on the controller.

Then the controller presents both certificates (the controller certificate and its CA certificate). The CA certificate must be a trusted CA or has the resources to verify the CA. You can actually build a chain of CA certificates that lead to a trusted CA on top.

Place the entire chain in the same file. The file then contains content such as this example:

```
BEGIN CERTIFICATE ------ device certificate*   END CERTIFICATE ------ BEGIN
CERTIFICATE ------ intermediate CA certificate*   END CERTIFICATE ------ BEGIN
CERTIFICATE ------ Root CA certificate*   END CERTIFICATE ------
```

How to Cause the Certificate to Match the URL

The WebAuth URL is set to 192.0.2.1 in order to authenticate yourself and the certificate is issued (this is the CN field of the WLC certificate).

To change the WebAuth URL to 'myWLC . com', for example, go into the **virtual interface configuration** (the 192.0.2.1 interface) and there you can enter a **virtual DNS hostname**, such as myWLC . com.

This replaces the 192.0.2.1 in your URL bar. This name must also be resolvable. The sniffer trace shows

how it all works, but when WLC sends the login page, WLC shows the myWLC . com address, and the client resolves this name with their DNS.

This name must resolve as 192.0.2.1. This means that if you also use a name for the management of the WLC, use a different name for WebAuth.

If you use myWLC . com mapped to the WLC management IP address, you must use a different name for the WebAuth, such as myWLCwebauth.com.

Troubleshoot Certificate Issues

This section explains how and what to check to troubleshoot certificate issues.

How to Check

Download OpenSSL (for Windows, search for OpenSSL Win32) and install it. Without any configuration, you can go in the bin directory and try **openssl s_client –connect (your web auth URL):443**,

if this URL is the URL where your WebAuth page is linked on your DNS, refer to "What to Check" in the next section of this document.

If your certificates use a private CA, place the Root CA certificate in a directory on a local machine and use the openssl option **-CApath**. If you have an Intermediate CA, put it into the same directory as well.

To obtain general information about the certificate and to check it, use:

<#root>

```
openssl x509 -in certificate.pem -noout -text
```

```
openssl verify certificate.pem
```

It is also useful to convert certificates with the use of openssl:

<#root>

```
openssl x509 -in certificate.der -inform DER -outform PEM -out certificate.pem
```

What to Check

You can see what certificates are sent to the client when it connects. Read the device certificate — the CN must be the URL where the web page is reachable.

Read the “issued by” line of the device certificate. This must match the CN of the second certificate. This second certificate, “issued by”, must match the CN of the next certificate, and so on. Otherwise, it does not make a real chain.

In the OpenSSL output shown here, notice that **openssl** cannot verify the device certificate because its “issued by” does not match the name of the CA certificate provided.

SSL Output

```
Loading 'screen' into random state - done CONNECTED(00000760) depth=0 /O=
<company>.ac.uk/OU=Domain Control Validated/CN=<company>.ac.uk verify error:
num=20:unable to get local issuer certificate verify return:1 depth=0 /O=
<company>.ac.uk/OU=Domain Control Validated/CN=<company>.ac.uk verify error:
num=27:certificate not trusted verify return:1 depth=0 /O=<company>.ac.uk/OU=
Domain Control Validated/CN=<company>.ac.uk verify error:num=21:
unable to verify the first certificate verify return:1 --- Certificate chain
0 s:/O=<company>.ac.uk/OU=
Domain Control Validated/CN=<company>.ac.uki:/C=US/   ST=
Arizona/L=Scottsdale/O=.com/OU=http://certificates.gocompany.com/repository/CN=
Secure Certification Authority/serialNumber=079
692871 s:/C=US/O=Company/OU=Class 2 Certification Authority
i:/C=US/O=Company/OU=Class 2 Certification Authority --- Server certificate

BEGIN CERTIFICATE-----
MIIE/zCCA+egAwIBAgIDRc2iMA0GCSqGSIb3DQEBBQUAMIHKMQswCQYDVQQGEwJV
output cut*
YMaj/NACviEU9J3iot4sfreCQSKkBmjH0kf/Dg1l0kmdSbc=

END CERTIFICATE-----
subject=/O=<company>.ac.uk/OU=Domain Control Validated/CN=<company>c.ac.uk
issuer=/C=US/ST=Arizona/L=Scottsdale/O=.com/OU=http://certificates.
.com/repository/CN=Secure Certification Authority/serialNumber=0
7969287 --- No client certificate CA names sent --- SSL handshake has read
2476 bytes and written 322 bytes --- New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit Compression: NONE Expansion: NONE SSL-Session:

Protocol  : TLSv1
Cipher    : AES256-SHA
Session-ID: A32DB00A7AB7CD1CEF683980F3696C2BBA31A1453324F711F50EF4B86A4A7F03

Session-ID-ctx:Master-Key: C95E1BDAC7B1A964ED7324955C985CAF186B92EA34CD69E10
   5F95D969D557E19
939C6A77C72350AB099B3736D168AB22

   Key-Arg   : None
Start Time: 1220282986
Timeout   : 300 (sec)
Verify return code: 21 (unable to verify the first certificate)
---
```

Another possible issue is that the certificate cannot be uploaded to the controller. In this situation there is no question of validity, CA, and so on.

To verify this, check the Trivial File Transfer Protocol (TFTP) connectivity and try to transfer a configuration file. If you enter the **debug transfer all enable** command, notice that the problem is the installation of the certificate.

This could be due to the wrong key used with the certificate. It could also be that the certificate is in a wrong format or is corrupted.

Cisco recommends that you compare the certificate content to a known, valid certificate. This allows you to see if a **LocalkeyID** attribute shows all 0s (already happened). If so, then the certificate must be reconverted.

There are two commands with OpenSSL that allow you to return from .pem to .p12, and then reissue a .pem with the key of your choice.

If you received a .pem that contains a certificate followed by a key, copy/paste the key part: **----BEGIN KEY ---- until ------- END KEY ------** from the .pem into "key.pem".

1. **openssl pkcs12 -export -in certificate.pem -inkey key.pem -out newcert.p12** ? You are prompted with a key; enter **check123**.

2. **openssl pkcs12 -in newcert.p12 -out workingnewcert.pem -passin pass:check123 -passout pass:check123** This results in an operational .pem with the password **check123.**

Other Situations to Troubleshoot

Although **mobility anchor** has not been discussed in this document, if you are in an **anchored guest** situation, make sure the mobility exchange occurs correctly and that you see the client arrives on the anchor.

Any further WebAuth problems need troubleshoot on the anchor.

Here are some common issues you can troubleshoot:

- **Users cannot associate to the guest WLAN**.

  This is not related to WebAuth. Check the client configuration, the security settings on the WLAN, if it is enabled, and whether radios are active and operative, and so on.

- **Users do not obtain IP address**.

  In a guest anchor situation, this is most often because the foreign and anchor was not configured exactly the same way. Otherwise, check the DHCP configuration, connectivity, and so on.
- Confirm whether or not other WLANs can use the same DHCP server without a problem. This still is not related to WebAuth.

- **User is not redirected to the login page**.

  This is the most common symptom, but is more precise. There are two possible scenarios.

  1. The user is not redirected (user enters a URL and never reaches the WebAuth page). For this situation, check:

     - that a valid DNS server has been assigned to the client via DHCP (**ipconfig /all**),

     - that the DNS is reachable from the client (**nslookup (website URL**),

     - that the user entered a valid URL in order to be redirected,

     - that the user went on an HTTP URL on port 80 (for example, to reach an ACS with http://localhost:2002 does not redirect you since you sent on port 2002 instead of 80).

  2. The user is redirected to 192.0.2.1 correctly, but the page itself does not display.

     This situation is most likely either a WLC problem (bug) or a client-side problem. It could be that the client has some firewall or software or policy block. It also could be that they have

configured a proxy in their web browser.

Recommendation: Take a sniffer trace on the client PC. There is no need for special wireless software, only Wireshark, which runs on the wireless adapter and shows you if the WLC replies and tries to redirect. You have two possibilities: either there is no response from WLC, or something is wrong with the SSL handshake for the WebAuth page. For SSL handshake issue, you can check whether the user browser allows for SSLv3 (some only allow SSLv2), and if it is too aggressive on certificate verification.

It is a common step to manually enter http://192.0.2.1 in order to check if the web page appears without DNS. Actually, you can type http://10.0.0.0 and get the same effect. The WLC redirects any IP address you enter. Therefore, if you enter http://192.0.2.1, it does not make you work around the web redirection. If you enter https://192.0.2.1(secure), this does not work because WLC does not redirect the HTTPS traffic (by default, this is actually possible in Version 8.0 and later). The best way to load the page directly without a redirect is to enter https://192.0.2.1/login.html.

- **Users cannot authenticate**.

  See the section of this document that discusses authentication. Check credentials locally on the RADIUS.

- **Users can successfully authenticate through WebAuth, but they do not have internet access afterwards**.

  You can remove WebAuth from the security of the WLAN, and then you have an open WLAN. You can then try to access the web, the DNS and so on. If you experience issues there as well, remove WebAuth settings altogether and check your interfaces configuration.

For more information, refer to: Troubleshooting Web Authentication on a Wireless LAN Controller (WLC).

HTTP Proxy Server and How it Works

You can use an HTTP proxy server. If you need the client to add an exception in its browser that 192.0.2.1 is not to go through the proxy server, you can make the WLC listen for HTTP traffic on the port of the proxy server (usually 8080).

In order to understand this scenario, you need to know what an HTTP proxy does. It is something you configure on the client side (IP address and port) in the browser.

The usual scenario when a user visits a website is to resolve the name to IP with DNS, and then it asks the web page to the web server. The process always sends the HTTP request for the page to the proxy.

The proxy processes the DNS, if required, and forwards to the web server (if the page is not already cached on the proxy). The discussion is client-to-proxy only. Whether or not the proxy obtains the real web page is irrelevant to the client.

Here is the web authentication process:

- User types in a URL.
- Client PC sends to the Proxy server.

- WLC intercepts and imitations Proxy server IP; it replies to the PC with a redirect to 192.0.2.1

At this stage, if the PC is not configured for it, it asks for the 192.0.2.1 WebAuth page to the proxy so it does not work. The PC must make an exception for 192.0.2.1; then it sends an HTTP request to 192.0.2.1 and proceeds with WebAuth.

When authenticated, all communications go through proxy again. An exception configuration is usually in the browser close to the configuration of the proxy server. You then see the message: "Do not use proxy for those IP addresses".

With WLC Release 7.0 and later, the feature **webauth proxy redirect** can be enabled in the global WLC configuration options.

When enabled, the WLC checks if the clients are configured to manually use a proxy. In that case, they redirect the client to a page that shows them how to modify their proxy settings to make everything work.

The WebAuth proxy redirect can be configured to work on a variety of ports and is compatible with Central Web Authentication.

For an example on WebAuth proxy redirection, refer to Web Authentication Proxy on a Wireless LAN Controller Configuration Example.

Web Authentication on HTTP Instead of HTTPS

You can login on web authentication on HTTP instead of HTTPS. If you login on HTTP, you do not receive certificate alerts.

For earlier than WLC Release 7.2 code, you must disable HTTPS management of the WLC and leave HTTP management. However, this only allows the web management of the WLC over HTTP.

For WLC Release 7.2 code, use the **config network web-auth secureweb disable** command to disable. This only disables HTTPS for the web authentication and not the management. Note that this requires a reboot of the controller!

On WLC Release 7.3 and later code, you can enable/disable HTTPS for WebAuth only via GUI and CLI.

Related Information

- **Wireless LAN Controller Web Authentication Configuration Example**
- **Download Software for Wireless Controller WebAuth Bundles**
- **Creating a Customized Web Authentication Login Page**
- **External Web Authentication with Wireless LAN Controllers Configuration Example**
- **Wireless LAN Controller 5760/3850 Web Passthrough Configuration Example**
- **Configuring Web Redirect (GUI)**
- **Configuring Web Redirect (CLI)**
- **Troubleshooting Web Authentication on a Wireless LAN Controller (WLC)**
- **Web Authentication Proxy on a Wireless LAN Controller Configuration Example**
- **Requests for Comments (RFCs)**
- **Technical Support & Documentation - Cisco Systems**