Technical Guide

Allied Telesis

Feature Overview and Configuration Guide

AAA and Port Authentication

NETWORK SMARTER | AlliedTelesis.com

C613-22088-00 REV P

Acknowledgements

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. For information about this see www.openssl.org/.

This product includes software licensed under the GNU General Public License available from: www.gnu.org/licenses/gpl2.html

Source code for all GPL licensed software in this product can be obtained from the Allied Telesis GPL Code Download Center at: www.alliedtelesis.com/support/

Allied Telesis is committed to meeting the requirements of the open source licenses including the GNU General Public License (GPL) and will make all required source code available.

If you would like a copy of the GPL source code contained in Allied Telesis products, please send us a request by registered mail including a check for US$15 to cover production and shipping costs and a CD with the GPL code will be mailed to you.

GPL Code Request
Allied Telesis Labs (Ltd)
PO Box 8011
Christchurch
New Zealand

Allied Telesis, Inc. reserves the right to make changes in specifications and other information contained in this document without prior written notice. The information provided herein is subject to change without notice. In no event shall Allied Telesis, Inc. be liable for any incidental, special, indirect, or consequential damages whatsoever, including but not limited to lost profits, arising out of or related to this manual or the information contained herein, even if Allied Telesis, Inc. has been advised of, known, or should have known, the possibility of such damages.

Allied Telesis, AlliedWare Plus, Allied Telesis Management Framework, EPSRing, SwitchBlade, VCStack and VCStack Plus are trademarks or registered trademarks in the United States and elsewhere of Allied Telesis, Inc. Additional brands, names and products mentioned herein may be trademarks of their respective companies.

Contents

Introduction

This guide describes the AlliedWare Plus implementation of Authentication, Accounting and Authorization. Port Authentication commands enable you to specify three different types of device authentication: 802.1X authentication, web authentication, and MAC authentication.

802.1X is an IEEE standard providing a mechanism for authenticating devices attached to a LAN port or wireless device. Web authentication is applicable to devices that have a human user who opens the web browser and types in a user name and password when requested. MAC authentication is used to authenticate devices that have neither a human user nor implement 802.1X supplicant when making a network connection request.

Products and software version that apply to this guide

This guide applies to all AlliedWare Plus™ products, running version **5.4.6** or later.

Feature support may change in later software versions. For the latest information, see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

- MAC authentication available on AR series devices - 5.4.8-2.x
- 802.1X authentication available on AR series device switch ports - 5.4.9-2.x
- Dynamic ACL assignments with port authentication - 5.5.0-1.x
- Authentication priority for tri-authentication - 5.5.0-2.x
- Hardware forwarding on the guest VLAN - 5.5.1-0.x
- Ping-poll type can be ICMP or ARP - 5.5.1-1.x
- RADIUS Change of Authorization (CoA) support - 5.5.1-1.x
- Restrict Port Authenticated Supplicants to One Per VLAN - 5.5.1-2.x
- Port Authentication for Dynamic Multiple VLANs - 5.5.2-0.x
- For security purposes, it is possible to specify a radius server with a named VRF. Placing a radius server within a VRF means that no actor that resides outside of the VRF can contact the radius server. - 5.5.2-1.0
- MAC authentication available on wired Ethernet ports on the TQ-R Series - 5.5.5-0.1

Authentication Methods on Security Appliances

The AR2010V, AR2050V, AR3050S, and AR4050S security appliances support limited port authentication methods. The AR1050V does not support any port authentication methods.

Table 1: Security appliance port authentication support

Device	Port type	Authentication methods	Supported from version
AR1050V	Switch ports	none	-
	Ethernet port	none	-
AR2010V*	Ethernet port	Web authentication	5.4.5-2.1
AR2050V AR3050S AR4050S	Switch ports	MAC authentication 802.1X authentication	5.4.8-2.1 5.4.9-2.1
	Ethernet ports	Web authentication	5.4.5-2.1

*The AR2010V only has Ethernet ports.

Two-step authentication is supported on the devices that have switch ports. When two-step authentication is enabled the sequence is MAC authentication first followed by 802.1X authentication.

Tri-authentication is not supported on these devices as all three authentication methods are not available on the same port type. However, if both MAC authentication and 802.1X authentication are configured on a port and two-step authentication is disabled then the supplicant will be authenticated if either authentication method is successful.

802.1X authentication on security appliances

The following limitations apply to 802.1X authentication on these devices:

- 802.1X authentication is not supported on static channel-groups and dynamic (LACP) channel-groups.
- Dynamic VLAN assignment can only be configured per port and not per MAC address. This means that all supplicants on a port can only be dynamically assigned to the same VLAN. Different VLANS, however, can be assigned on different ports. (See the **auth dynamic-vlan-creation** command for more information on dynamic VLANs.)

Authentication, Authorization and Accounting (AAA)

Introduction

AAA is the collective title for the three related functions of Authentication, Authorization and Accounting. These functions can be applied in a variety of methods with a variety of servers. The purpose of the AAA commands is to map instances of the AAA functions to sets of servers.

The Authentication function can be performed in multiple contexts, such as authentication of users logging in at a console, or 802.1X authentication of devices connecting to Ethernet ports.

For each of these contexts, you may want to use different sets of servers for examining the proffered authentication credentials and deciding if they are valid. AAA Authentication commands enable you to specify which servers will be used for different types of authentication.

Available functions and server types

The three types of servers that can be used for Authentication, Authorization and Accounting are:

- Local user database
- RADIUS servers
- TACACS+ servers

Authentication decides whether the client is allowed access and is performed in the following contexts:

- Login authentication of user shell sessions on the console port, and via Telnet/SSH
- Enable password authentication for user shell sessions on the console port, and via Telnet/SSH (TACACS+ or local user database only)
- 802.1X authentication of devices connecting to switch ports
- MAC authentication of devices connecting to switch ports
- Web-based authentication of devices connecting to switch ports

Authorization decides what level of access a client is allowed, i.e. what services are they allowed to use. In AlliedWare Plus, authorization is performed as part of the login authentication process and there are no separate authorization commands available. Authorization is performed in the following context:

- Login authentication of user shell sessions on the console port, and via Telnet/SSH

Accounting keeps a record of the client's session and collects statistics on their data usages, it is performed in the following contexts:

- Console, Telnet, and SSH login sessions
- Commands executed within user shell sessions (TACACS+ only)
- 802.1X authenticated connections
- MAC authenticated connections
- Web authenticated connections

Different servers might be used for different activities. A network might use RADIUS for 802.1X authentication, but TACACS+ for authenticating users logging into the management interfaces of the device itself.

Server groups

The two protocols most commonly used for Authentication, Authorization, and Accounting are RADIUS and TACACS+. When using these protocols, the device will exchange data with a RADIUS or TACACS+ server.

- For authentication, the device will send user credentials to a RADIUS or TACACS+ server, and listen for the server's response to those credentials.
- For accounting, the device sends accounting messages to the server, and the server uses those to accumulate usage records of network services.

For redundancy purposes, a network will often contain more than one RADIUS or TACACS+ server.

To enable a set of servers to be conveniently referenced from AAA commands, the concept of a **server group** has been introduced to the device command line.

Configuring server groups

A server group is defined by the command **aaa group server.** This command puts you into server group configuration mode. Once in that mode you can add servers to the group by using the command **server ip-address**.

Any number of servers can be added to a group. Typically, you will add servers which have already been configured by the command **radius-server host**. If you add a server that has not yet been configured by the command **radius-server host**, you will receive a warning that the server has not yet been configured, but the command will still be accepted.

There is one server group, named **radius**, that is always present on the device. This group cannot be removed and contains all servers that have been configured using the command **radius-server**

host. As soon as a server is configured by the command **radius-server host**, it is automatically a member of the server group **radius** and cannot be removed from it.

Note: While it is possible to create named server groups for RADIUS servers, there is no equivalent feature for TACACS+ servers. For TACACS+, the only server group is the default group, that contains all the TACACS+ servers configured on the device.

The **show radius server group** command displays information about the RADIUS server groups configured on a device.

Method lists

A method list defines the set of server types that you want to be used for authenticating or accounting a user or device. It also specifies the order in which you want the server types to be used.

You may want to:

- Check the usernames submitted for logging in at the console are in the local user database. You can create a method list that specifies **local**.
- Or, check the TACACS+ servers first, and resort to the local user database if none of the TACACS+ servers respond. You can create a method list that specifies **group TACACS+** first, followed by **local**.
- Or, check the RADIUS servers first, and resort to the local user database if none of the RADIUS servers respond. You can create a method list that specifies **group RADIUS** first, followed by **local**.

A method list defines the servers where authentication requests are sent. The first server listed is contacted; if that server fails to respond then the next authentication server type in the method list is selected. This process continues until there is a successful response or until all server types fail to respond.

In the case of a user logging into the device, the device sends an authentication request to the first authentication server in the method list:

- If the first server in the list is reachable and it contains a username and password matching the authentication request, the user is authenticated and the login succeeds.
- If the authentication server denies the authentication request because of an incorrect username or password, the user login fails.
- If the first server in the method list is unreachable, the device sends the request to the next server in the list, and so on.

For example, if the method list specifies **group tacacs+ local**, and a user attempts to log in with a password that does not match a user entry in the first TACACS+ server, this TACACS+ server denies

the authentication request, then the device does not try any other TACACS+ servers nor the local user database; the user login fails.

If the first server type in the method list is a server group containing multiple servers, then **all** servers in the group are tried before moving on to the next server type in the method list.

The details of how requests are retried progressively through a group of servers are described in "Checking multiple authentication servers" on page 15.

Configuring method lists

Within AlliedWare Plus, it is possible to create method lists for two types of activities:

- authentication
- accounting

The method lists for these two activities can be created for four different contexts:

- 802.1X
- MAC-based authentication
- Web-based authentication
- Device management session login

In addition to the default method list it is possible to create any number of other, named, method lists for all four contexts.

For 802.1X, MAC-auth and web-auth, the method available for authentication is RADIUS and it is necessary to define which RADIUS server group is being used.

Default method lists

For every authentication or accounting type, it is always possible to define a method list called **default**.

As soon as the default method list is defined for a given authentication or accounting type, it is automatically applied as the method list to be used for any instance of that type of authentication or accounting, except for instances to which another named method list has already been specifically applied.

Authentication method lists

The commands to create an authentication method list for 802.1X, MAC-auth, and web-auth are:

```
awplus(config)# aaa authentication dot1x {default|<list-name>} group
{<group-name>|radius}
```

```
awplus(config)# aaa authentication auth-mac {default|<list-name>} group
{<group-name>|radius}
awplus(config)# aaa authentication auth-web {default|<list-name>} group
{<group-name>|radius}
```

Points to note:

- For any one of these authentication types, the authentication will not operate until the either the default or a named method authentication method list has been defined.
- The commands above effectively enable those three authentication types.
- If the server group **radius** is chosen, then all the RADIUS servers configured on the device will be available to the authentication method.

For authentication of the login to management sessions on the device, the **local** method is available, as well as RADIUS and TACACS+.

So, the syntax of the command for creating a login method list is:

```
awplus(config)# aaa authentication login {default|<list-name>}
{[local][group {radius|tacacs+|<group-name>}]}
```

Accounting method lists

The command for creating an accounting method list for 802.1X, auth-MAC, or auth-web is:

```
awplus(config)# aaa accounting <context> {default|<list-name>} {start-
stop|stop-only|none} {group {radius|<group-name>}}
```

where <context> is one of **dot1x**, **auth-mac**, or **auth-web**.

Management login session method lists are configured with the following command:

```
awplus(config)# aaa accounting login {default|<list-name>}{start-stop|
stop-only|none} {group {radius|<group-name>}}
```

The method list definition also defines whether the device will send accounting start and/or stop messages or neither. There is a separate command **aaa accounting update** that controls whether or not RADIUS accounting update messages will be sent. This is a global command, so it controls the action of all accounting sessions, regardless of which method list they are controlled by.

Applying named method lists for port authentication

You apply a named method lists to an interface from that interface's configuration mode. The command to enter the configuration mode for an interface is:

```
awplus# configure terminal
awplus(config)# interface <interface-name>
```

Once in the interface configuration mode apply an authentication method list to the interface with the command:

```
awplus(config-if)# <context> authentication {default|<list-name>}
```

or apply an accounting method list to the interface with the command:

```
awplus(config-if)# <context> accounting {default|<list-name>}
```

where <*context*> is one of **dot1x**, **auth-mac**, or **auth-web**.

The **show aaa server group** command lists the AAA servers and any method lists associated with them.

Applying login method lists

The types of management session to which method lists can be applied are:

- Console sessions on the device's RS-232 port
- Remote CLI sessions via Telnet
- Remote CLI sessions via SSH

The method lists are applied to these session types by configuring the login method on the virtual interfaces via which these sessions access the device.

The virtual interfaces are configured via the **line** command. The command to enter configuration mode for the console virtual interface is:

```
awplus# configure terminal
awplus(config)# line console 0
```

The command to enter configuration mode for the Telnet/SSH virtual interface is:

```
awplus(config)# line vty 0 4
```

Note: Telnet and SSH both use the same set of vty lines.

Within the interface configuration mode for these virtual interfaces, the command to apply an authentication method list is:

```
awplus(config-line)# login authentication <method list name>
```

To configure Telnet/SSH to use a RADIUS group 'trust', then check the local database, configure as

```
awplus(config)# aaa authentication login remote-login group trust local
awplus(config)# line vty 0 4
awplus(config-line)# login authentication remote-login
```

Processing authentication requests

Checking multiple authentication servers

The logic by which a set of servers is checked is as follows:

1. The authentication request is sent to the first server in the list.
2. If the server responds (either to accept or reject the authentication request), no more servers are contacted.
3. If the server does not respond, the device waits for timeout period. The timeout period defaults to 5 seconds, but can be configured, on a per-server basis, to a different value with the commands:

```
awplus(config)# radius-server host <ip-address> timeout <timeout>
awplus(config)# tacacs-server timeout <seconds>
```

In the case of RADIUS, if no response is received within this time, then:

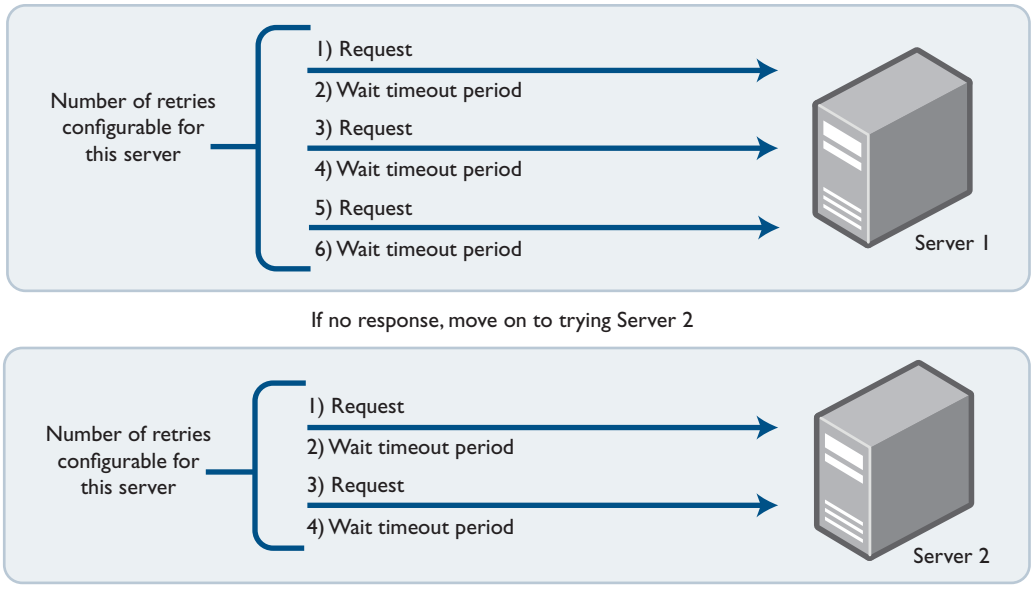
- the authentication request is sent to the server again.
- the device again waits for the timeout period.

This cycle is repeated a number of times. By default, this number is 3, but can be configured, on a per-server basis, to a different value with the command:

```
awplus(config)# radius-server host <ip-address> retransmit <number of retries>
```

4. If a full set of retries has been sent to a server, and still no response has been received, then the device gives up on that server. It moves on to the next server in the group, and sends the request to that server. This process continues until a response has been received, or until all servers have been tried, and none has responded.

Figure 1: Processing authentication requests



In the case of TACACS+, if no response is received from the first attempt, the server is considered dead. This is because TACACS+ uses TCP which is a full connection protocol, if a connection cannot be established there is no purpose in retrying.

It is important to note that if a server's database does not contain a particular username, then it will respond with a reject message. The process of checking a series of servers is not a matter of looking for the server that knows of a user; it is just a matter of looking for a server that responds. A reject response is as valid as an accept response. As soon as the device receives ANY response from a server, it will not check with any more servers in the group.

Configuring AAA login authentication

To configure AAA authentication, create the default method list or a named method list for different authentication types. In the case of login authentication, the named method lists are then applied to consoles or VTY lines.

AAA configuration tasks

To define how a given accounting or authentication type is applied to a given port or line:

- (optionally) create a server group using the **aaa group server** command (RADIUS only),
- create a method list for the authentication or accounting type as required,
- then apply that method list to the port or line as required.

Step 1: **Define a group of RADIUS servers**

Create a RADIUS server group named GROUP1 with hosts 192.168.1.1, 192.168.2.1 and 192.168.3.1, use the commands:

```
awplus(config)# aaa group server radius GROUP1
awplus(config-sg)# server 192.168.1.1 auth-port 1812 acctport 1813
awplus(config-sg)# server 192.168.2.1 auth-port 1812 acctport 1813
awplus(config-sg)# server 192.168.3.1 auth-port 1812 acctport 1813
```

Step 2: **Specify the login authentication or accounting method list**

Create a method list for the authentication (aaa authentication login) or accounting (aaa accounting login) type as required.

To configure a user login authentication method list called USERS to first use the RADIUS servers in the group GROUP1 for user login authentication and then the local user database, use the following commands:

```
awplus# configure terminal
awplus(config)# aaa authentication login USERS group GROUP1 local
```

To configure RADIUS accounting for login shell sessions, use the following commands:

```
awplus# configure terminal
awplus(config)# aaa accounting login USERS group GROUP1
```

Step 3: **Apply method lists to interface port or line**

Apply that method list to the port or line as required:

```
awplus# configure terminal
awplus(config)# line console 0
awplus(config-line)# login authentication USERS
awplus(config-line)# accounting login USERS
```

Sample authentication configurations

Sample 802.1X authentication configuration

The configuration below shows an example configuration for dot1x authentication, using the local RADIUS server.

```
!
radius-server host 127.0.0.1 key awplus-local-radius-server
!
aaa authentication dot1x default group radius
!
radius-server local
server enable
nas 127.0.0.1 key awplus-local-radius-server
user guest password guest!
!
no spanning-tree rstp enable
!
interface port1.0.1
 dot1x port-control auto
!
interface vlan1
ip address 192.168.1.120/24
!
```

The 802.1X authentication feature needs the **aaa authentication dot1x** command configured globally and the **dot1x port-control** command configured on an interface. See the CLI reference for command information to edit this configuration.

The local RADIUS Server has been configured to respond to authentication requests generated by 802.1X authentication in this sample configuration. See the **radius-server local** and **server enable** commands in the CLI reference for command information.

This sample configuration enables 802.1X authentication on interface VLAN1 which has IP address 192.168.1.120. Change the VLAN ID and IP address as required for your configuration.

Sample MAC authentication configuration

The configuration below shows an example configuration for MAC authentication, using the local RADIUS server.

```
!
radius-server host 127.0.0.1 key awplus-local-radius-server
!
aaa authentication auth-mac default group radius
!
radius-server local
 server enable
 nas 127.0.0.1 key awplus-local-radius-server
 user 00-00-5e-00-53-00 password 00-00-5e-00-53-00
!
no spanning-tree rstp enable
!
interface port1.0.1
 auth-mac enable
!
interface vlan1
 ip address 192.168.1.120/24
!
```

The MAC authentication feature needs the **aaa authentication auth-mac** command configured globally and the **auth-mac enable** command configured on an interface.

The local RADIUS server has been configured to use MAC authentication in this sample configuration. For information on the commands used in this configuration, see the CLI reference.

This configuration enables MAC authentication on VLAN1 which has IP address 192.168.1.120. Change the interface VLAN ID, MAC, and IP addresses as needed in your configuration.

Sample web authentication configuration

The configuration below shows an example configuration for web authentication, using the local RADIUS server.

```
!
radius-server host 127.0.0.1 key awplus-local-radius-server
!
aaa authentication auth-web default group radius
!
radius-server local
 server enable
 nas 127.0.0.1 key awplus-local-radius-server
 user guest encrypted password l+lWcLjLm29bCAXwWRPHXK0PFlsA7gNpR+P7wO4kwQQ=
!
no spanning-tree rstp enable
!
interface port1.0.1
 auth-web enable
!
interface vlan1
 ip address 192.168.1.120/24
!
```

The web authentication feature needs the **aaa authentication auth-web** command configured globally and the **auth-web enable** command configured on an interface. See the AAA Commands and Authentication Commands chapters in the CLI reference, for information to edit this sample configuration.

The local RADIUS Server has been configured to use web authentication in this sample configuration. See the **radius-server local** and **server enable** commands in the Local RADIUS Server Commands chapter in the CLI reference, for command information to edit this sample configuration.

Note: The above sample web authentication configuration requires the user name ‘guest’ with password ‘guest!’ on IP address 192.168.1.120 from interface port1.0.1.

Sample configuration using a named method list

The configuration script below is a sample web authentication configuration which makes use of a named method list and server group.

```
!
radius-server host 127.0.0.1 key awplus-local-radius-server
aaa group server radius GROUP1
!
aaa authentication auth-web USERS group GROUP1
!
radius-server local
 server enable
 nas 127.0.0.1 key awplus-local-radius-server
 user guest password guest!
!
no spanning-tree rstp enable
!
interface port1.0.1
 auth-web authentication USERS
 auth-web enable
!
interface vlan1
 ip address 192.168.1.120/24
!
```

The web authentication feature needs the **aaa authentication auth-web** and the **auth-web enable** commands configured on an interface. See the AAA Commands and Authentication Commands chapters in the CLI reference, for information to edit this sample configuration.

The local RADIUS Server has been configured to use web authentication in this sample configuration. See the **radius-server local** and **server enable** commands in the Local RADIUS Server Commands chapter in the CLI reference, for command information to edit this sample configuration.

Sample tri-authentication configuration

The sample configuration script below is a sample tri-authentication configuration that configures 802.1X authentication, MAC authentication, and web authentication on the same interface.

```
!
radius-server host 127.0.0.1 key awplus-local-radius-server
!
aaa authentication dot1x default group radius
aaa authentication auth-mac default group radius
aaa authentication auth-web default group radius
!
radius-server local
server enable
 nas 127.0.0.1 key awplus-local-radius-server
 user guest password guest!
 user 00-00-5e-00-53-00 password 00-00-5e-00-53-00
!
no spanning-tree rstp enable
!
interface port1.0.1
 dot1x port-control auto
 auth-mac enable
 auth-web enable
!
interface vlan1
 ip address 192.168.1.120/24
!
```

The 802.1X authentication feature needs the **aaa authentication dot1x** command configured globally and the **dot1x port-control** command configured on an interface. See the AAA and 802.1X Commands chapters in the CLI reference for command information to edit this configuration.

The MAC authentication feature needs the **aaa authentication auth-mac** command configured globally and the **auth-mac enable** command configured on an interface. See the AAA and Authentication Commands chapters in the CLI reference for command information to edit this configuration.

The web authentication feature needs the **aaa authentication auth-web** command configured globally and the **auth-web enable** command configured on an interface. See the AAA and Authentication Commands chapters in the CLI reference for command information to edit this configuration.

The local RADIUS Server has been configured to use tri-authentication in this sample configuration. See the **radius-server local** and **server enable** commands in the Local RADIUS Server Commands chapter for command information to edit this sample configuration.

This sample tri-authentication configuration requires a user name ‘guest’ with password ‘guest!’ on IP address 192.168.1.120 from port1.0.1. Note this sample also configures 802.1X- and MAC authentication on VLAN1 which has IP address 192.168.1.120. Change the interface VLAN ID, MAC and IP address as needed for your configuration.

Note: When tri-authentication is applied to the same interface, then the order of execution is MAC authentication first, then 802.1X or web authentication, if MAC authentication fails.

802.1X Authentication

Introduction

802.1X is an IEEE standard providing a mechanism for authenticating devices attached to a LAN port or wireless device. Devices wishing to access services behind a port must authenticate themselves before any Ethernet packets are allowed to pass through. The protocol is referred to as 802.1X because it was initially defined in the IEEE standard 802.1X, published in 2001 and revised in 2004 and again as the current 802.1X 2010 standard.

Networks have two important requirements:

- **Security**: Authentication and Authorization
- **Flexibility**: The ability for users to roam

Networks need a device authentication method that is highly secure, but not tied to a port's physical location. Network resources presented to a given user need to be determined from their authentication credentials.

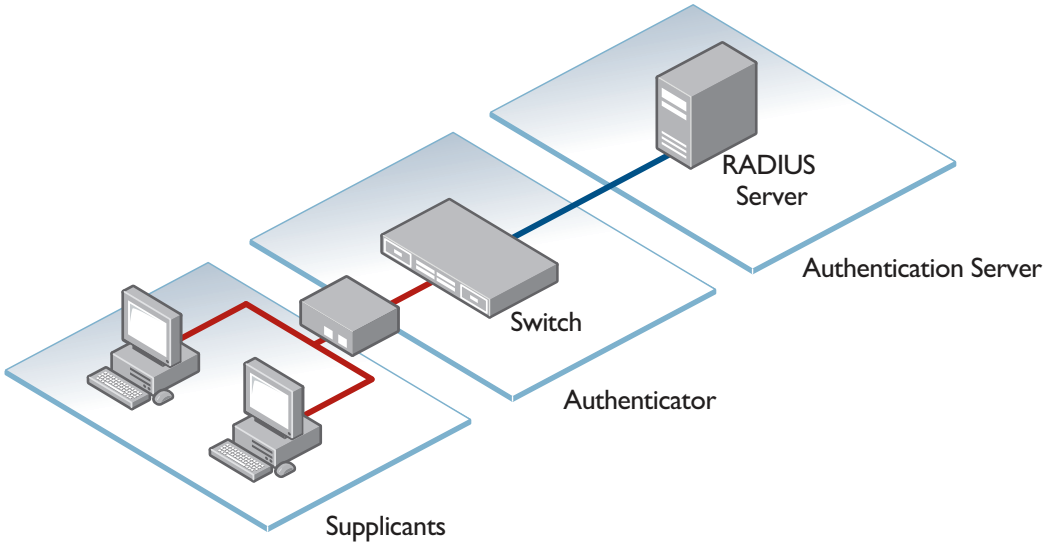
802.1X user authentication satisfies these requirements. It is relatively uncomplicated and has little impact on network performance. It is a protocol that is medium-independent —being equally as effective on wireless connections (802.11i) and wired connections. 802.1X user authentication is rapidly becoming an expected component on networks.

802.1X system components

There are three main components to a system using 802.1X port authentication control:

- **Authenticator**: the device that wishes to enforce authentication before allowing access to services that are accessible behind it. An example of this is a switch that has 802.1X port authentication control enabled.
- **Supplicant**: the client that wishes to access services offered by the authenticator's system. An example of this is a Windows XP Professional PC with an 802.1X client.
- **Authentication server**: the device that uses the authentication credentials supplied by the supplicant, to determine if the authenticator should grant access to its services. The AlliedWare Plus implementation of 802.1X supports the use of a RADIUS. authentication server using Extensible Authentication Protocol (EAP) in conjunction with RADIUS.

Figure 1: 802.1X system components



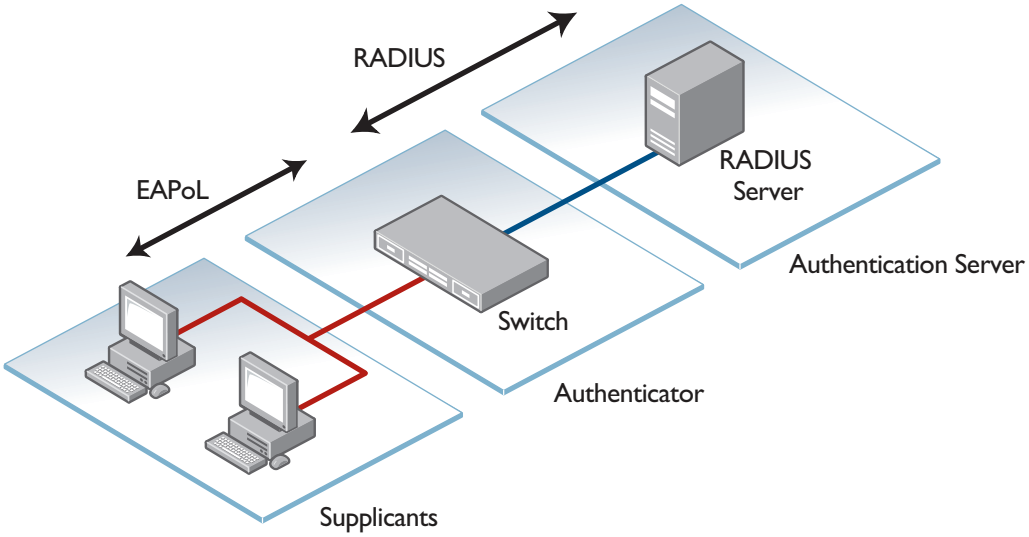
802.1X component protocols

There are two protocols involved in the authentication conversation:

1. EAPoL exchanged between the supplicant and authenticator.
   - EAPoL—Extensible Authentication Protocol over LAN— is the protocol defined in IEEE802.1X.
2. RADIUS exchanged between the authenticator and authentication server.
   - RADIUS has received specific extensions to interoperate with EAPoL.

The diagram below illustrates where EAPoL and RADIUS protocols are used in the authentication conversation:

Figure 2: 801.X component protocols



Basic steps in an 802.1X conversation:

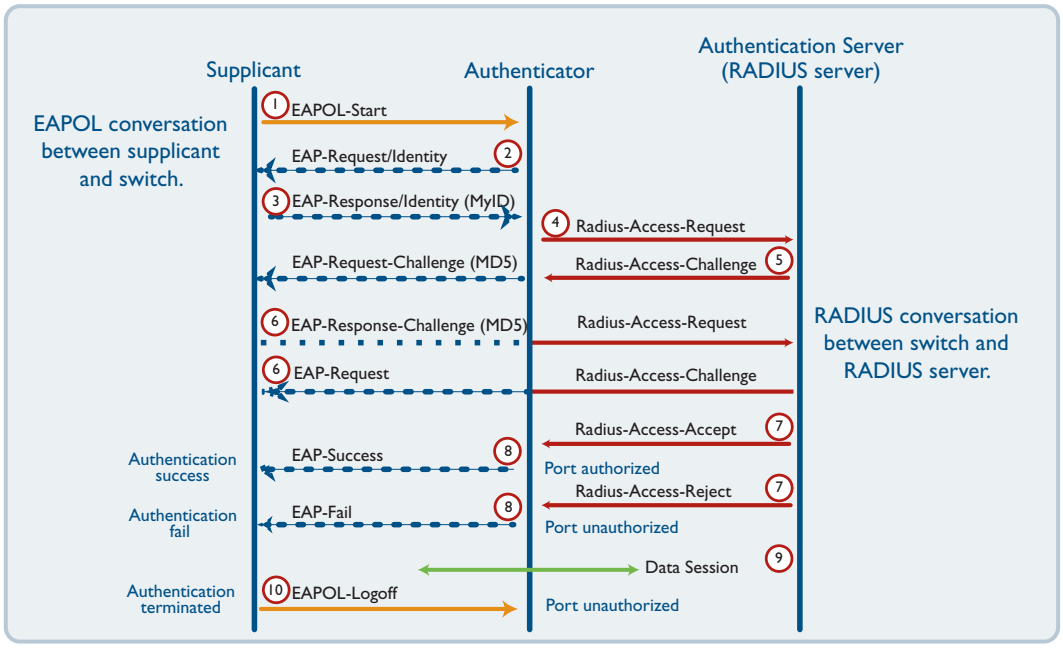
1. The supplicant informs the authenticator that it wants to initiate the conversation.
2. The authenticator requests the supplicant's credentials.
3. The supplicant sends username/password or X.509 certificate.
4. The authenticator wraps the supplicant's reply into a RADIUS packet and sends it to the RADIUS server.
5. The RADIUS server chooses an authentication method, and sends an appropriate request to the supplicant as a ‘challenge’.
6. The RADIUS server and supplicant exchange some messages, ferried by the authenticator.
7. The RADIUS server eventually decides if the supplicant is allowed access and the RADIUS server sends an Access-Accept or Access-Reject message to the Authenticator.
8. The authenticator sends an EAPoL-Success or EAPoL-Fail to the supplicant.
9. The supplicant has a session using the network (if accepted).
10. When the session is over, the supplicant sends a log-off message.

Example message sequence

The diagram below illustrates an exchange using the EAP-MD5 authentication method, which is the simplest authentication method supported by 802.1X.

The EAPoL log-off message, of course, is not sent immediately after the other messages in the diagram, but is sent later on, at the end of the supplicant’s data session, when it wishes to disconnect from the network.

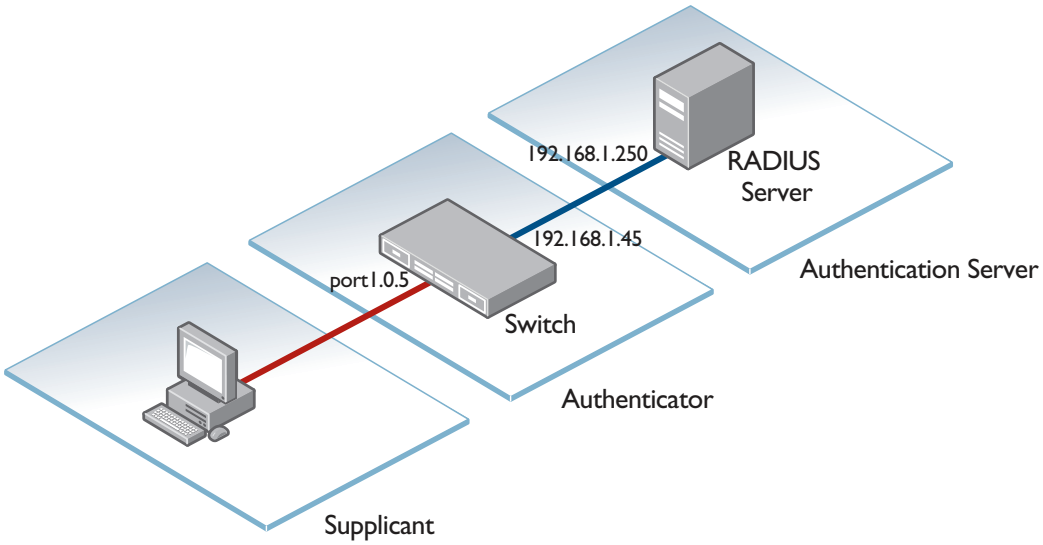
Figure 3: EAPoL message sequence



Basic 802.1X configuration

To configure the switch operating as authenticator, follow the instructions below:

Figure 4: Basic 802.1X configuration



Step 1: **Configure a RADIUS server for the switch to send requests to**

```
awplus(config)# radius-server host 192.168.1.250 key <secret-key>
```

Step 2: **Instruct 802.1X to use the configured RADIUS server**

```
awplus(config)# aaa authentication dot1x default group radius
```

Step 3: **Configure port1.0.5 for 802.1X authentication**

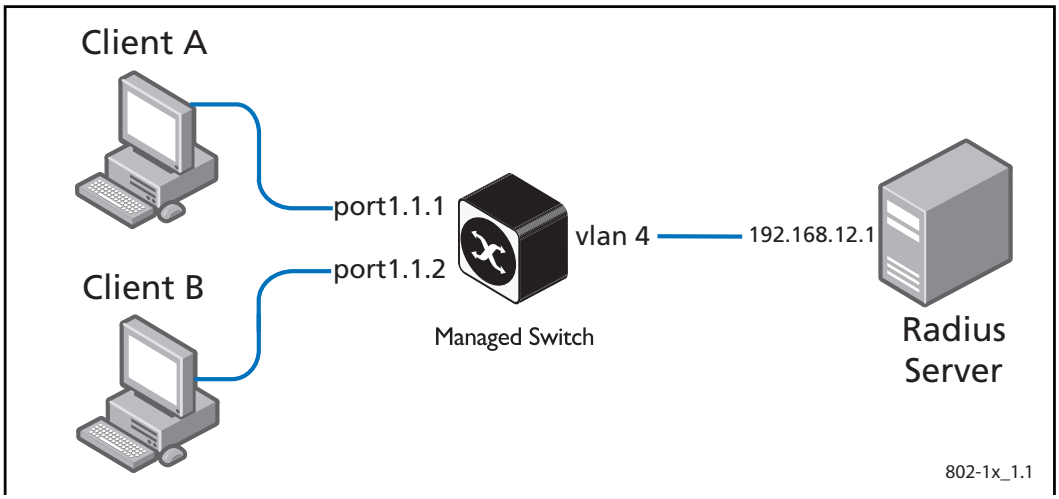
```
awplus(config)# interface port1.0.5
awplus(config-if)# dot1x port-control auto
awplus(config-if)# spanning-tree portfast
```

802.1X configuration example

The following example explains how to configure 802.1X. In this example, the RADIUS Server keeps the Client information, validating the identity of the Client and updating the switch about the authentication status of the client. The switch is the physical access between the two clients and the server. It requests information from the client, relays information to the server and then back to the client.

To configure 802.1X authentication, first enable authentication on port1.0.1 and port1.0.2 and then specify the RADIUS Server IP address and port.

Figure 5: 802.1X configuration example



Step 1: **Enable authentication globally.**

```
awplus# configure terminal
awplus(config)# aaa authentication dot1x default group radius
```

Step 2: **Enable authentication (via RADIUS) on port1.0.1.**

```
awplus(config)# interface port1.0.1
awplus(config-if)# dot1x port-control auto
```

Step 3: **Block traffic in both directions, other than authentication packets, until authentication is complete.**

```
awplus(config-if)# dot1x control-direction both
awplus(config-if)# exit
```

Step 4: **Enable authentication (via RADIUS) on port1.0.2.**

```
awplus(config)# interface port1.0.2
awplus(config-if)# dot1x port-control auto
awplus(config-if)# exit
```

Step 5: **Specify the RADIUS Server address (192.168.12.1) and authentication port.**

```
awplus(config)# radius-server host 192.168.12.1 auth-port 1812
```

Step 6: **Specify the shared key “mysecret” between the RADIUS server and the client.**

```
awplus(config)# radius-server key mysecret
```

Step 7: **Set the IP address on vlan4.**

```
awplus(config)# interface vlan4
awplus(config-if)#ip address 192.168.12.2/24
```

Multi-supplicant modes

AlliedWare Plus can be configured to accept one or more supplicants downstream of a port. Three authentication host-modes are available:

- **single-host**: the default state, only one supplicant allowed per port.
- **multi-host**: once the first host on a port is authenticated, all other downstream hosts are allowed without being authenticated (piggy-back mode).
- **multi-supplicant**: multiple separate supplicants are individually authenticated on one port.

The command (entered in interface configuration mode for a physical port interface) is :

```
awplus(config-if)# auth host-mode {single-host|multi-host| multi-supplicant}
```

This command controls how the switch deals with the situation where multiple authentication supplicants are downstream of a single port. This is possible if an EAP session passes through a Layer 2 switch which has been connected to the port, and the supplicants are attached to that Layer 2 switch.

Single supplicant

The first option that the command can set is single-host. With this option, only one supplicant may be authenticated on the port. Once that host has been authenticated, no other supplicants may be authenticated until the first supplicant's session has closed. This means, of course, that none of the other hosts downstream of the port will be able to send or receive traffic on that port.

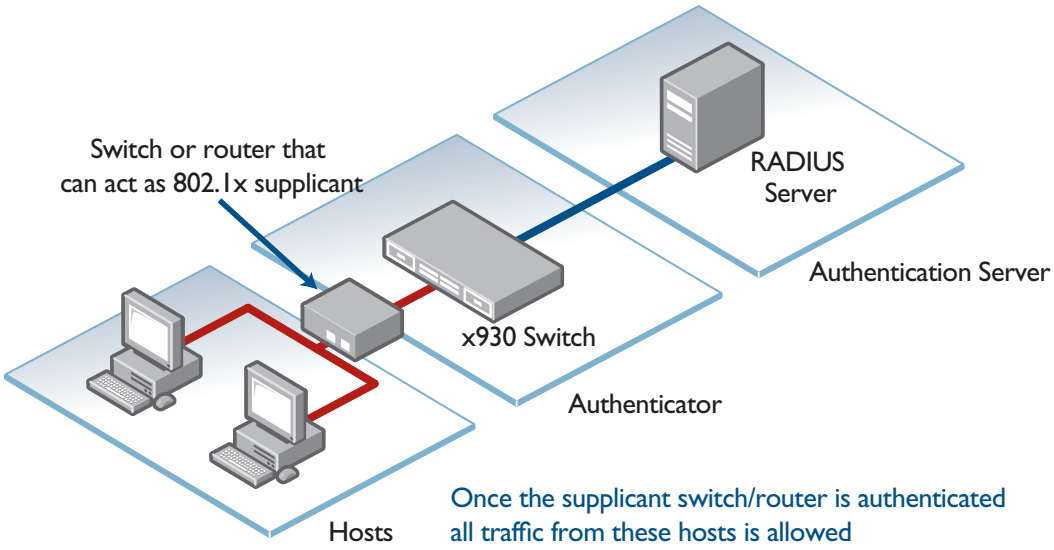
This option is recommended when you know that there should only be one host connected to a port. By limiting the port to a single authenticated host, you guard against the consequences of someone accidentally or maliciously connecting a downstream switch to the port.

Multi-host

The next available host-mode option is multiple host mode (chosen by the parameter value multi-host). With this mode, once the first host has been authenticated on the port, all other downstream hosts are allowed without being authenticated. This is sometimes known as piggy-back mode. It is useful when the downstream switch attached to the authenticating port is an intelligent switch that can act as an authentication supplicant.

If you trust that malicious users cannot be connected to that switch but you do not know the identity of those users, then you can simply authenticate the switch and then allow its attached users to have network access. If the valid switch is disconnected and an invalid one is connected which is not configured with the correct authentication credentials, then the devices connected to the invalid switch will be blocked from accessing the network.

Figure 6: Configuring 802.1X multi-host



802.1X VLAN assignment

Dynamic VLAN assignment

Whilst the authentication of devices attaching to the network is primarily driven by security considerations, it has significant spin-off benefits.

Once a device has been authenticated, the network knows the identity of the device and/or its user. Decisions can be made, based on this identity. In particular, it is possible to decide what network environment, and level of access, to present to this device and its user.

The standard mechanism via which a user's network environment is controlled is VLAN membership. Once a user's packets are classified into a particular VLAN, the user's access to the network will be controlled by the constraints that have been put on that VLAN throughout the network.

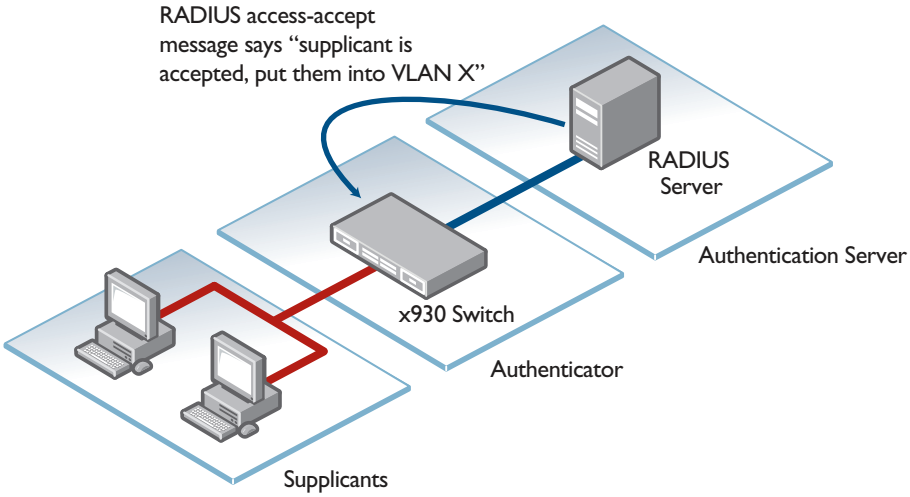
For this reason, it is now common for LAN switches to have the ability to dynamically assign the VLAN into which a device's traffic will be classified, once that device has been authenticated.

Dynamic VLAN assignment is achieved by a collaboration between the authenticator (the LAN switch) and the authentication server (the RADIUS server). When the RADIUS server sends back a RADIUS accept message to the authenticator, it can also include other attributes in that message that identify a VLAN to which the authenticated device should be assigned.

Dynamic VLAN assignment is a powerful extension to 802.1X, as it enables:

- **Identity-based networking**—the user gets the same environment no matter where they connect.
- **Guest Access**—guest users are allowed access to very limited parts of the network.
- **NAC**—level of access is based on a workstation's security status.

Figure 7: Dynamic VLAN assignment



Authenticator configuration

In addition to the basic 802.1X configuration, some further configuration is required to enable Dynamic VLAN creation on the switch. The VLANs that can be dynamically assigned must be present in the VLAN database:

```
awplus(config)# vlan database
awplus(config-vlan)# vlan x
awplus(config-vlan)# vlan y
awplus(config-vlan)# vlan z
awplus(config-vlan)# exit
```

Ports that accept VLAN membership dynamically have to be enabled for dynamic VLAN creation:

```
awplus(config)# interface port1.0.5
awplus(config-if)# auth dynamic-vlan-creation
```

Dynamic VLAN assignment with multiple supplicants

In multi-supplicant mode, what happens if two supplicants downstream of the same port are assigned to different VLANs? The **auth dynamic-vlan-creation** command has two parameters that govern the operation in this situation: rule and type.

The rule parameter

The first parameter is the **rule** parameter.

For switches that don't support the **type** parameter, it is not possible to assign different VLANs to untagged traffic from different supplicants. On these products, dynamic VLAN assignment effectively says 'the one untagged VLAN to be used on the authenticating port is VLAN x'. So, if the first supplicant is authenticated and assigned VLAN 45, then the authenticating port will classify all untagged traffic arriving on the port into VLAN 45. But if a second supplicant downstream of the same port then authenticates, and the RADIUS server assigns VLAN 56 to that supplicant, the

switch then faces a dilemma. It is already using VLAN 45 as the untagged VLAN on that port; it cannot use VLAN 56 as well.

There are two ways that the switch can resolve this situation. It can:

1. Allow the second supplicant to access the network, but assign its data to VLAN 45.
2. Block the second supplicant from having network access.

The **rule** parameter configures which of these choices the switch will opt for. If **rule** is set to **permit**, then option (1) above is chosen. If **rule** is set to **deny**, then option (2) above is chosen.

The type parameter

The second parameter is the **type** parameter.

The **type** parameter applies to devices that support MAC-based VLANs. Refer to your devices datasheet or command reference to see if it supports this feature.

The effect of the **type** parameter is to make use of MAC-based VLAN support to provide a better solution to the case where different supplicants downstream of a single port are dynamically allocated to different VLANs.

If **type** is set to the value **single**, then the MAC-based VLAN capability is not used, and the port's behavior in the different-dynamic-VLANs situation will be controlled by the **rule** parameter.

However, if **type** is set to **multi**, the switch brings the MAC-based VLAN capability into play. This capability enables it to support multiple different untagged VLANs on the same port. This is achieved by associating VLAN membership with the source MAC address of the incoming packets.

So, when different supplicants downstream of a single port are dynamically assigned different VLANs, the switch simply builds a table that maps supplicants' MAC addresses to their dynamically assigned VLANs.

The combination of these parameters results in three options for handling the case where different VLANs are assigned to supplicants on the same ports.

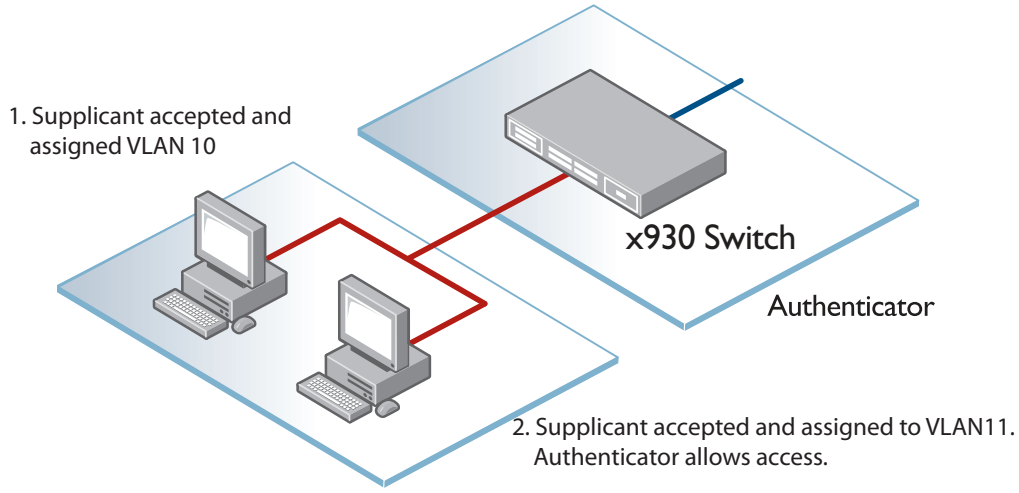
Option1: Deny access to supplicant assigned a different VLAN.

If the first supplicant authenticated on the port is assigned VLAN X, then any supplicants subsequently assigned a different VLAN are denied access. This is the default state when dynamic VLAN creation is enabled.

This is configured with:

```
awplus(config-if)# auth dynamic-vlan-creation rule deny
```

Figure 8: Deny access to supplicant assigned to a different VLAN



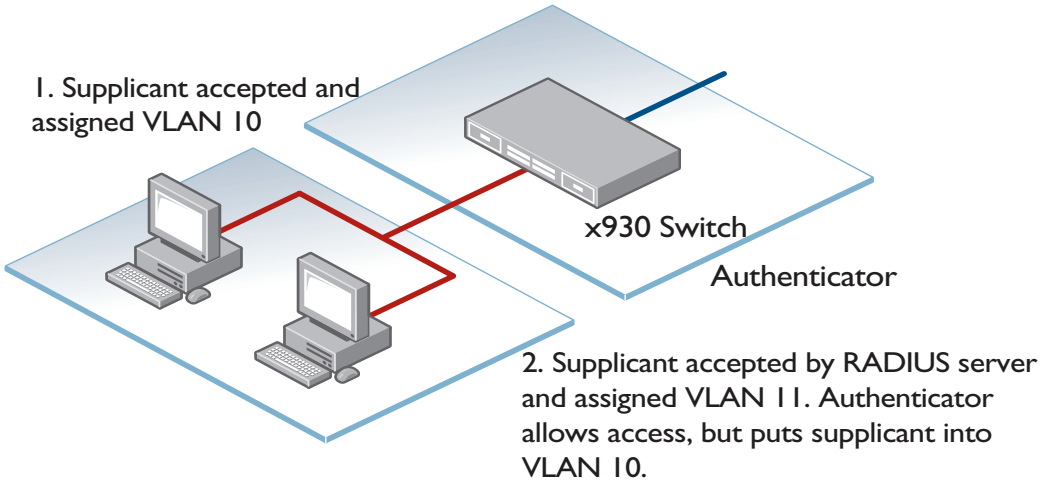
Option2: Force all supplicants into the same VLAN

If the first supplicant authenticated on the port is assigned VLAN X, then any supplicants subsequently assigned a different VLAN are allowed access, but forced into VLAN X

This is configured with:

```
awplus(config-if)# auth dynamic-vlan-creation rule permit
```

Figure 9: Force all supplicants into the same VLAN



Option3: Dynamically assign multiple VLANs to one port

If your device supports MAC-based VLANs, it is possible to assign different VLANs to different supplicants downstream of the same port.

This is configured with:

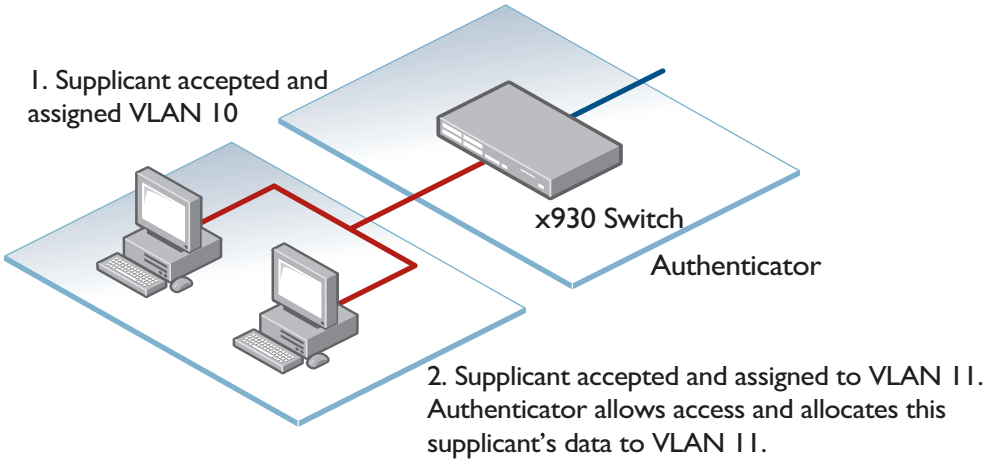
```
awplus(config-if)# auth dynamic-vlan-creation rule permit type multi
```

In addition, configure your RADIUS server to reply in Access-Accept packets with the attributes in the following table.

ATTRIBUTE	VALUE
Tunnel-Type	VLAN (13
Tunnel-Medium-Type	IEEE-802 (6)
Tunnel-Private-Group-ID	The VID or VLAN name*

*The desired VLAN is specified with the Tunnel-Private-Group-ID attribute.

Figure 10: Dynamically assign multiple VLANs to one port



The switch can assign VLAN membership to packets based on source MAC:

- Packets from MAC of supplicant 1 are assigned to VLAN10
- Packets from MAC of supplicant 2 are assigned to VLAN11

Note: The FS980M series switches do not support using this feature:

- with VLAN classifier rules on the same port.
  (VLAN classifier rules enable you to create Protocol-based VLANs.)
- at the same time as IP subnet-based VLANs.

Dynamic VLANs and auth critical

The **auth critical** command allows a supplicant to be authorized when a RADIUS server is not available. As Dynamic VLANs need information supplied by the RADIUS server they will not work if the RADIUS server cannot be contacted.

In normal circumstances a supplicant will keep retrying if a RADIUS server is unavailable. If, however, **auth critical** is configured then the supplicant can be authorized and assigned to the default VLAN before contact is made between the Authenticator and the RADIUS server.

For this reason we do not recommend configuring **auth critical** on ports that are also configured with Dynamic VLANs.

Using a Guest VLAN

Whilst you need to authenticate the users who will have access to the important services within your network, you might also want to provide some basic level of access to users who fail to authenticate.

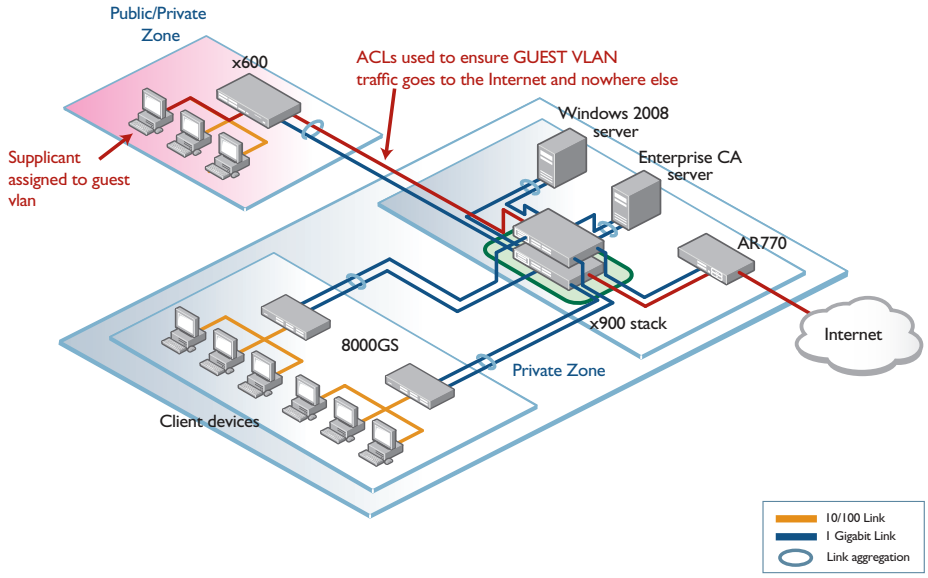
For example, visitors to an enterprise will often need to have Internet access. It would be desirable to have a secure, convenient way to provide this Internet access via the corporate LAN.

By default, 802.1X denies access to users who fail authentication.

Guests are not known to the RADIUS server, so fail authentication. The solution is to provide a Guest VLAN which is configured with:

```
awplus(config)# interface port1.0.5
awplus(config-if)# auth guest-vlan <vlan-id>
```

Figure 11: Using a Guest VLAN



If a supplicant attempts authentication and fails or does not even attempt authentication (no 802.1X client in the PC) then they are dynamically assigned to the Guest VLAN.

Note: Users who failed authentication and are dynamically assigned to the Guest VLAN may see reduced throughput as Guest VLAN traffic is CPU forwarded by default. See Chapter 7, Hardware forwarding on the Guest VLAN for information on configuring the switching chip to forward this traffic.

Verify the operation of 802.1X

When a supplicant has been authenticated on a port the details of the authentication can be seen with:

```
show dot1x supplicant int port1.0.5
Interface port1.0.5
  authenticationMethod: dot1x                    <--- Authenticated by 802.1X
  totalSupplicantNum: 1
    authorizedSupplicantNum: 1
    macBasedAuthenticationSupplicantNum: 0
    dot1xAuthenticationSupplicantNum: 1
    WebBasedAuthenticationSupplicantNum:
    otherAuthenticationSupplicantNum: 0
Supplicant name: Engineer01                      <--- Supplicant name
Supplicant address: 0000.cd28.0880               <---MAC of authenticated device
    authenticationMethod: 802.1x
    portStatus: Authorized - currentId: 9
    abort:F fail:F start:F timeout:F success:T
    PAE: state: Authenticated - portMode: Auto
    PAE: reAuthCount: 0 - rxRespId: 0
    PAE: quietPeriod: 60 - maxReauthReq: 2
    BE: state: Idle - reqCount: 0 - idFromServer: 8
    CD: adminControlledDirections: both - operControlledDirections: both
    CD: bridgeDetected: false
    KR: rxKey: false
    KT: keyAvailable: false - keyTxEnabled: false
    dynamicVlanId: 20                            <--- VLAN assigned, if dynamic VLA
    assignment enabled
```

When a supplicant has been authenticated, and assigned to a VLAN, the port they authenticated on will then be seen to be a member of that VLAN.

```
show vlan 20

VLAN ID  Name      Type State   Member ports
(u)-Untagged, (t)-Tagged
======= ================ ======= ======= ======================
20  Engineering   STATIC ACTIVE  port1.0.5(u)

show vlan 30

VLAN ID  Name      Type State   Member ports
(u)-Untagged, (t)-Tagged
======= ================ ======= ======= ======================
30  Marketing    STATIC ACTIVE  port1.0.5(u)
```

Web Authentication

Introduction

Web authentication, also known as Captive Portal, is a simple way to provide secure guest- user access to a network. It is used in a wide range of environments including WiFi hotspots, hotels, universities, and business centers.

In basic terms, if the switch detects an unauthorized user Web browsing, then irrespective of the IP configuration on their PC, they are re-directed to a web authentication login page. At this point, the user is required to enter a username and password before they can begin to Web browse.

The main benefits of this solution come from not requiring additional customer knowledge, software or special configuration.

Users are able to quickly and easily gain access to the network regardless of the type of device or operating system used.

What is web authentication?

Web authentication is a convenient alternative to 802.1X authentication, it’s commonly used to authenticate users in educational institutions, where regular users’ workstations are not managed by the network administrator. web authentication enables the switch to detect an unauthenticated workstation web browsing into the network, then redirect the user’s web browser to its own authentication web page.

Web authentication works like this:

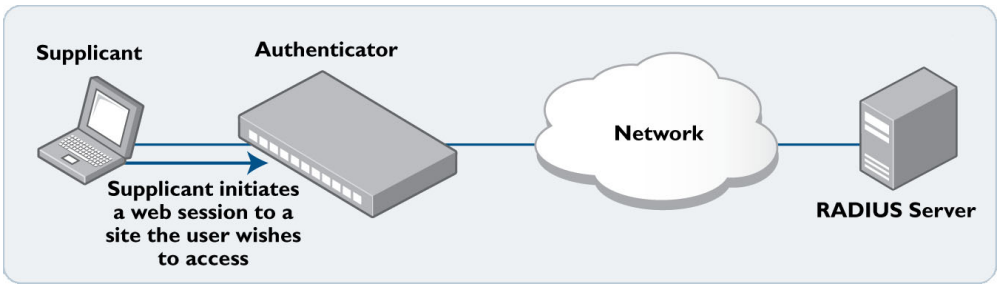
- The authenticating switch hijacks the user's web browsing session, and sends them the auth-web login page.
- The user enters their username and password into the web page, which the switch then sends to a RADIUS server for checking.
- If the RADIUS server accepts the user’s credentials, the switch then allows their traffic into the network.

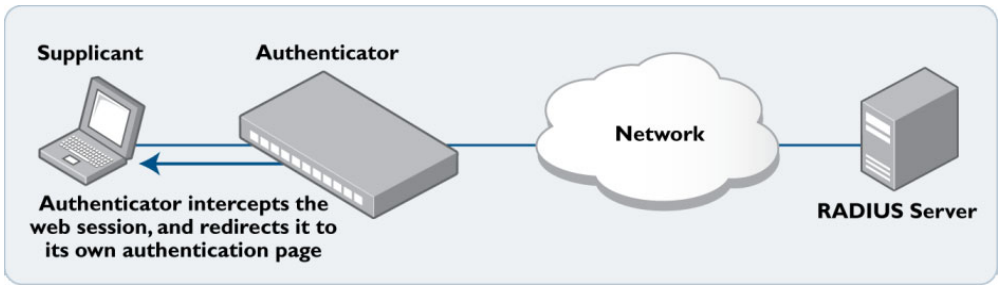
The web-authenticating switch interacts with a RADIUS server in the same way as an 802.1X Authenticator. So the two methods can easily be used together in the same network, using the same RADIUS server.

Web authentication basics

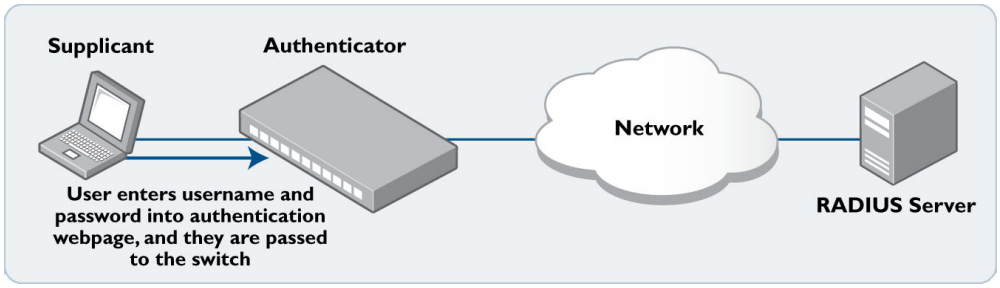
Conceptually, the operation of web authentication is quite simple:

1. The authenticating switch receives HTTP or HTTPS traffic from an unauthenticated supplicant. It intercepts the supplicant's web session, and redirects it to its own internal web server, or specially configured external web servers.

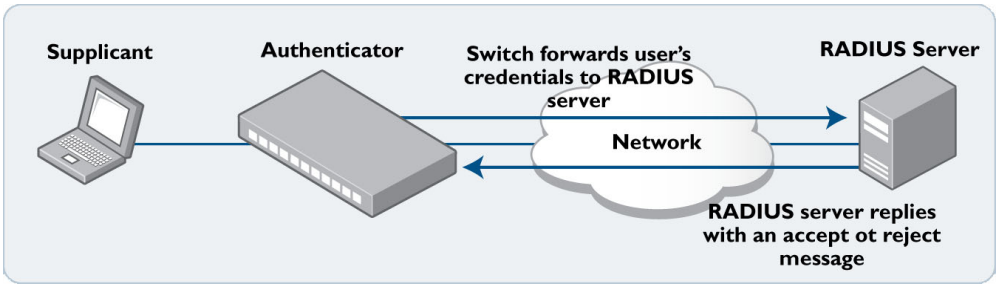




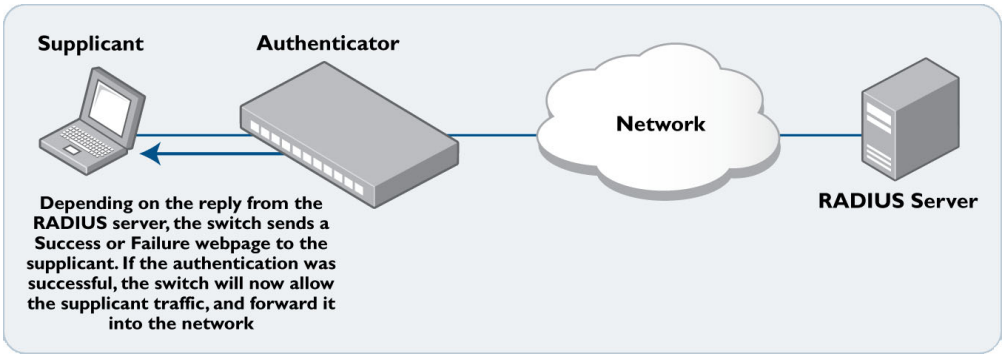
2. The web server serves up an authentication page into which the user may enter their username and password.



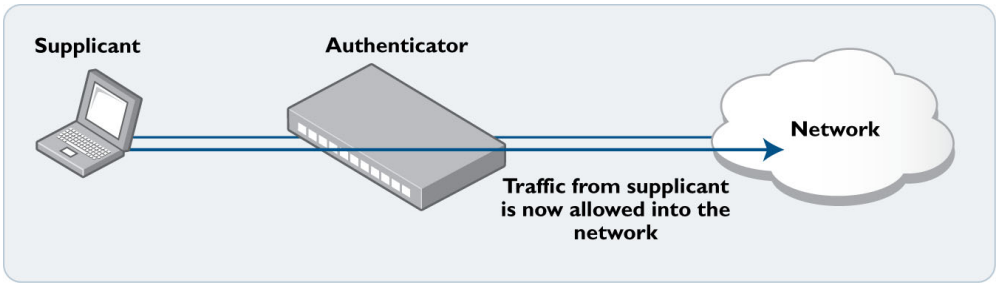
3. The username and password are sent to a RADIUS server, which informs the authenticating switch whether or not the supplicant is authenticated.



4. The user is then informed of the RADIUS server's verdict.



5. If the supplicant has been successfully authenticated, the authenticating switch will give the supplicant workstation access to the network.



Configuring web authentication

Web authentication can be configured on a switch in four simple steps:

Step 1. **Configure a RADIUS server.**

```
awplus(config)# radius-server host <server-ip-address> key <shared secret>
```

Step 2. **Instruct web authentication to use the configured RADIUS server.**

```
awplus(config)# aaa authentication auth-web default group radius
```

Step 3. **Define the IP address the web authentication service will be accessed on.**

```
awplus(config)# auth-web-server ipaddress <ip-address>
```

Step 4. **Configure ports for web authentication.**

```
awplus(config)# interface port1.0.1-1.0.20
awplus(config-if)# auth-web enable
```

Configuring the web authentication server address

When you use the command **auth-web-server ipaddress** you will have to specify an IP address that is not attached to any of the switch's interfaces, i.e. you need to use a virtual IP that belongs just to the web-authenticator.

Because the virtual IP address is not attached to any of the switch's interfaces, you must install a hardware filter to make sure the authorized supplicant can access the "Login Success" web page.

For example, if you configure the web-authenticator to use the arbitrary address 10.0.0.1 as follows:

```
auth-web-server ipaddress 10.0.0.1
```

Then the authorized supplicants can't access the "Login Success" page because the traffic destined to 10.0.0.1 is not necessarily sent to the switch's CPU. So you must also configure a hardware filter to force packets for 10.0.0.1 to the CPU.

Here is an example of a hardware filter used to force packets for 10.0.0.1 to the CPU:

```
access-list hardware acl_webauth
send-to-cpu ip any 10.0.0.1/32
exit
interface port1.0.1
access-group acl_web_auth
```

Configuration example

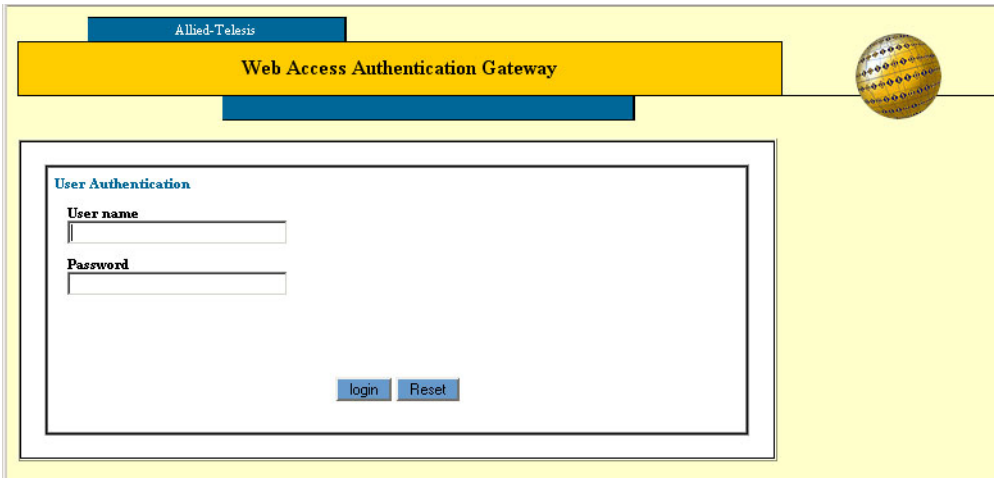
```
VLAN database
  VLAN 10 name edge
  VLAN 30 name core

radius-server host 192.168.30.129 key verysecret
aaa authentication auth-web default group radius
auth-web-server ipaddress 10.0.0.1

access-list hardware acl_webauth
  send-to-cpu ip any 10.0.0.1/32

int vlan10
  ip address 192.168.10.1/24
int vlan30
  ip address 192.168.30.1/24

int port1.0.1-1.0.20
  switchport access vlan 10
  auth-Web enable
  access-group acl_webauth

int port1.0.21-1.0.22
```

Note: You can use a host name to represent the authenticating server, using the command **auth-web-server host-name**. When you use this command please make sure you have already registered the host name on the DNS server that users will access.

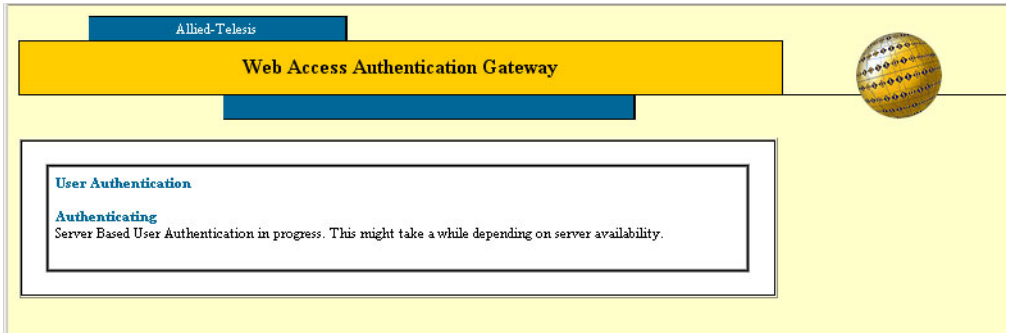
Starting a web authentication session

Let us look at what the user actually sees in a web authentication session:

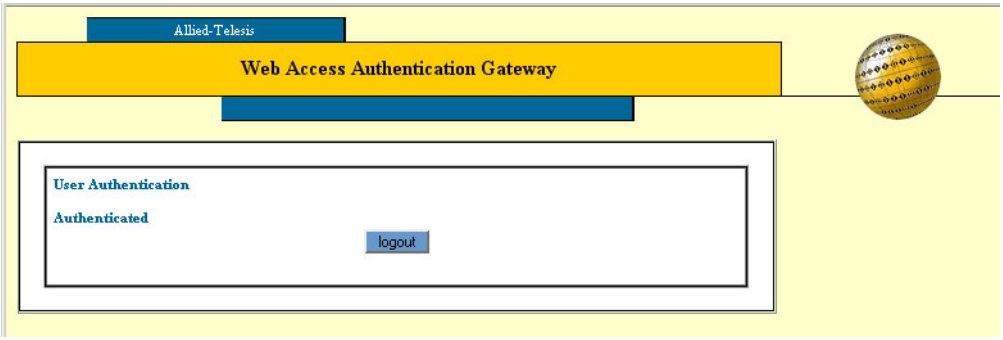
1. The user starts their web browser, and browses to a page they wish to view. Shortly thereafter, the address in the browser's address bar automatically changes to the address of the authenticating switch's authentication page.
2. In the switch's authentication page, the user enters their **user name** and **password**, and clicks **login**.



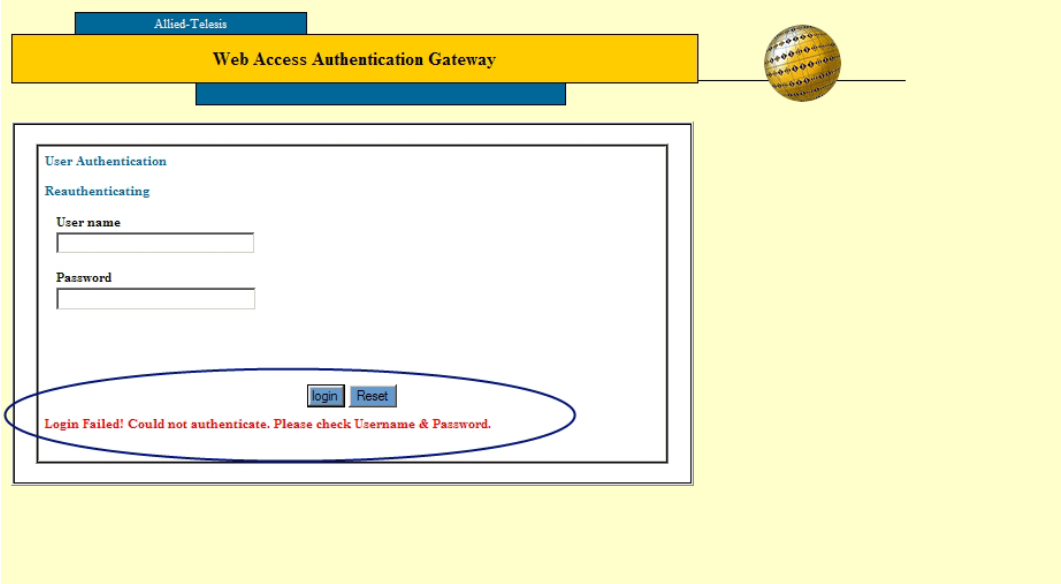
3. The switch displays a page that informs them that authentication is in progress.



4. Once authentication is complete, the authentication result is displayed.



- If the user enters a username/password combination that is not accepted by the RADIUS server, the switch presents an invitation to check the username and password, and try again.
- If the user enters incorrect usernames/passwords several times the authentication has failed. The number of times a user can try to login is configurable, but it is set to 3 by default.



Understanding the web authentication features

While the authentication process, as it has been described so far, is essentially quite simple, there are actually a number of implementation details that it glosses over.

To use web authentication effectively, it is necessary to understand these details – how they work and how to configure them.

We'll take a closer look at:

- Protocol support features
- Secure authentication (SSL)
- Ping-poll monitoring of supplicant presence
- Managing traffic of unauthenticated supplicants

Support for protocols underlying web authentication

Web authentication does not use a dedicated protocol like 802.1X, with a standards-defined set of messages for authentication conversation. When it comes to web authentication, the switch is overlaying the authentication process on top of another process that was not designed for authentication.

The web browser communication process that the authentication overlays, is itself reliant on IP addressing, ARP, and DNS. The authentication needs to occur in a seamless manner for all users, irrespective of their IP and DNS setting, and before they have full access to the network.

To make this possible, the switch needs to provide facilities that enable the user's PC to access the authentication web page.

There are a few different features of web authentication that work together to achieve this:

- ARP/DNS/DHCP packet forwarding is enabled by default
- A built-in DHCP server can be used for web authentication
- Or you can use an external DHCP server for web authentication

ARP/DNS/DHCP packet forwarding enabled by default

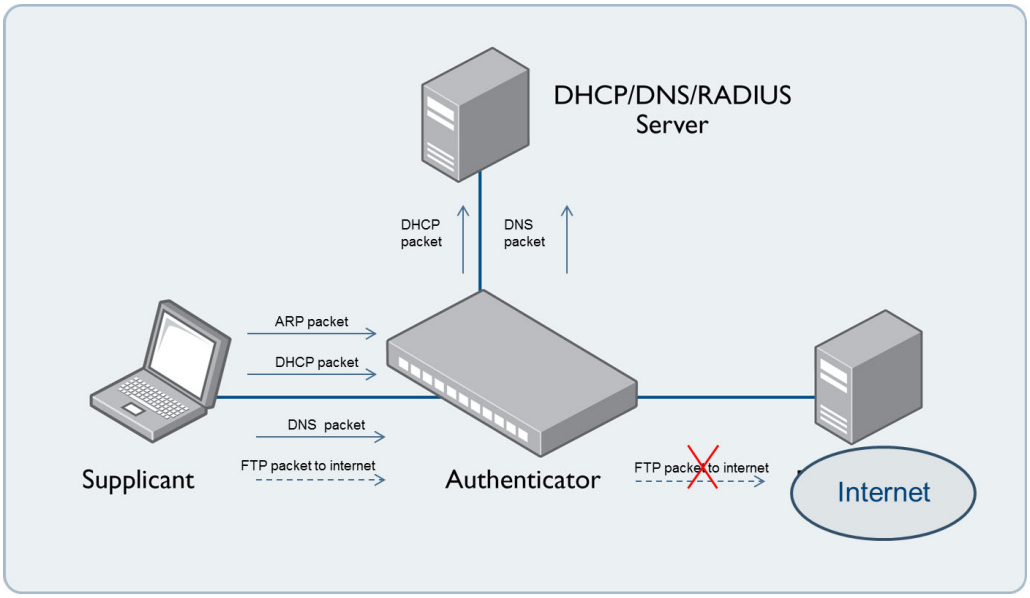
Web authentication differs from MAC authentication and 802.1X authentication, in that it must assign an IP address to the unauthorized supplicant for web access. web authentication must not suppress the unauthorized supplicant's DNS name and address resolution process.

For example, if the user's DNS request receives no reply, the web browser will never progress on to attempting an HTTP/HTTPS session.

Thus web authentication must forward ARP, DHCP and DNS packets.

ARP/DHCP/DNS packet forwarding is enabled by default to facilitate the underlying processes required before an HTTP session is initiated. If you want to disable this feature you can use the command **no auth-web forward** to disable packet forwarding.

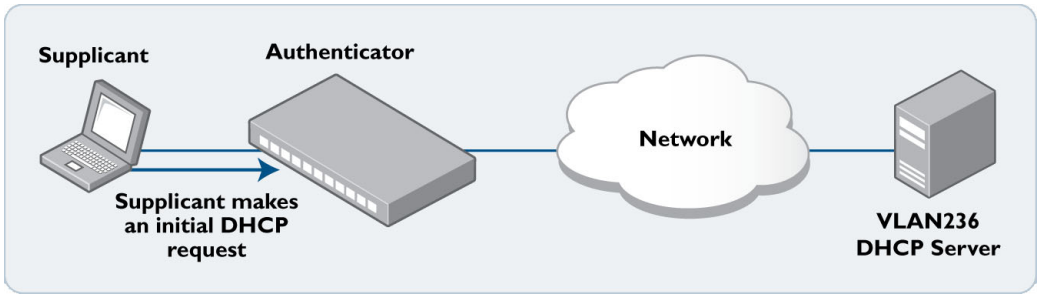
In general supplicants don't know the web authentication login page URL. In fact, supplicants will typically just start trying to browse to somewhere on the Internet. So web authentication must hijack all HTTP/HTTPS packets from unauthorized supplicants and send back the Login page contents instead of allowing the session to the user's intended destination. Therefore, TCP/UDP packet forwarding is disabled. To force TCP/UDP packets to pass through the Authenticator before the supplicant is authorized you need to configure selective TCP/UDP forwarding.

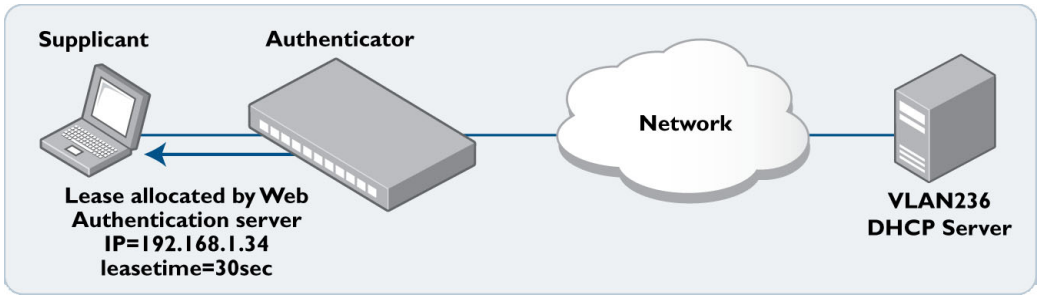


DHCP server for web authentication

To initiate a web browsing session, the supplicant needs an IP address. If the supplicant has been configured to obtain its IP address by DHCP, then the authenticating switch needs to ensure that the supplicant will be served an IP address.

The simplest way to achieve this is to use the web authentication switch itself to act as a DHCP Server. There is a DHCP server built in to web authentication. This DHCP server is dedicated to serving IP addresses to be used by web authentication clients.





This DHCP service is configured by the command:

`auth-web-server dhcp ip address <ip-address/prefix-length>`

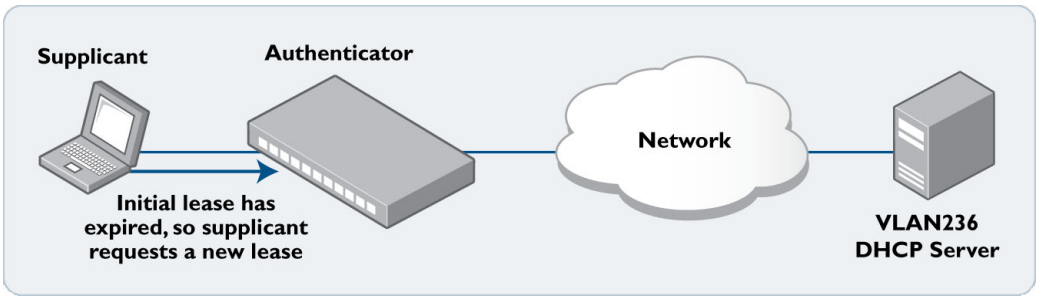
The IP address specified in this command is the IP address of the web authentication service. If the web authentication service's IP address has not already been configured by the command **auth-web-server ip address *<ip-address>***, then this command configures the service's address.

If the web authentication service's IP address has already been configured by the command **auth-web-server ip address *<ip-address>***, then the IP address in the **auth-web-server dhcp** command must be the same as that already configured. By default, this DHCP server serves leases of 30 seconds duration. The lease duration can be changed by the command **auth-web-server dhcp lease *<20-60>***. The short lease is deliberate. It facilitates the transition to a new VLAN/subnet after authentication. The supplicant is unaware that the switch transitions it to another VLAN, with another DHCP server, after authentication succeeds.

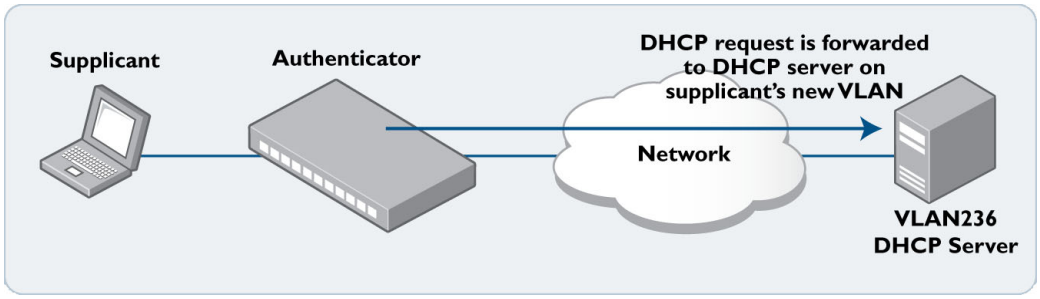
How can we force the supplicant to request a new DHCP lease after the completion of the authentication process? There is no mechanism by which the supplicant's web browser signals down to the DHCP client process to say "I've just completed an authentication session, you need to request a new DHCP lease."

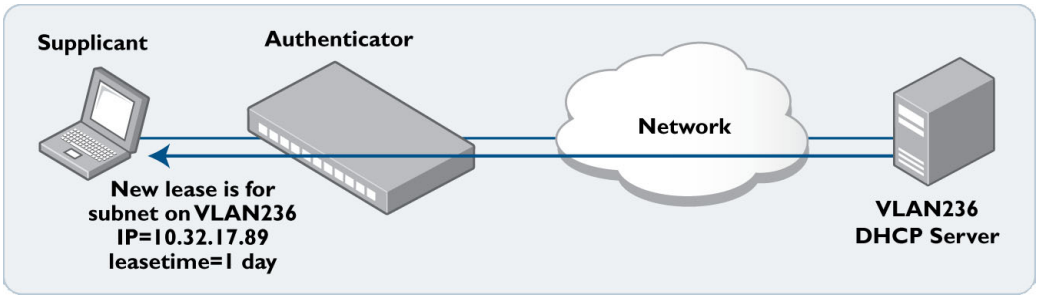
Similarly, there is no mechanism by which the switch signals to the supplicant to say "I have just assigned you to VLAN 236, you now need to obtain a DHCP lease from the DHCP server on that VLAN." How can we force the supplicant to request a new DHCP lease after the completion of the authentication process?

There is no mechanism by which the supplicant's web browser signals down to the DHCP client process to say "I've just completed an authentication session, you need to request a new DHCP lease."



- This new request will now be serviced by the DHCP server on the supplicant's new VLAN.





Note: When the built-in DHCP server is running, ARP/DHCP/DNS/HTTP packets are redirected to the Web-Auth module, and other packets are dropped. Even if packet forwarding (configured by the **auth-web forward** command, including the default setting ARP/DHCP/DNS forwarding) is ignored.

Using an external DHCP server

You can also use a remote DHCP server instead of the built-in DHCP service. In this situation, all supplicant's DHCP packets will be forwarded directly to the remote DHCP server by default, even though the supplicant is not authenticated.

Customising the login page

When users access the login page, you may wish to customize with your company details and/or policy information. There are three ways to customize the login page. You can:

- Create your own login page and serve it from the AlliedWare Plus device (see "Serving your own login page from the AlliedWare Plus device" on page 45), or
- Create your own login page and serve it from an external web server (see "Serving your own login page from an external web server" on page 45), or
- Use the default login page and customise it (see "Customizing the default login page" on page 50)

Serving your own login page from the AlliedWare Plus device

With Version 5.4.6-1.1 and later, you can create your own web authentication login page.

To create your own login page, follow these steps:

Step 1. **Create the page**

Write the page in HTML. Note that it must include the following login form code:

```
<form action="/index.cgi" autocomplete="off" target="_self" name="AUTH"
method="POST">

<div>User name</div>
<div><input size="30" type="text" maxlength="64" name="USERNAME"></div>
<div>Password</div>
<div><input size="30" type="password" maxlength="64" name="PASSWORD"></div>

<div>
<input type="submit" name="ACTION" value="login">
<input type="reset" name="RESET" value="Reset">
</div>

</form>
```

If you do not include the above login form, the page will display in the client browser but will not perform web authentication.

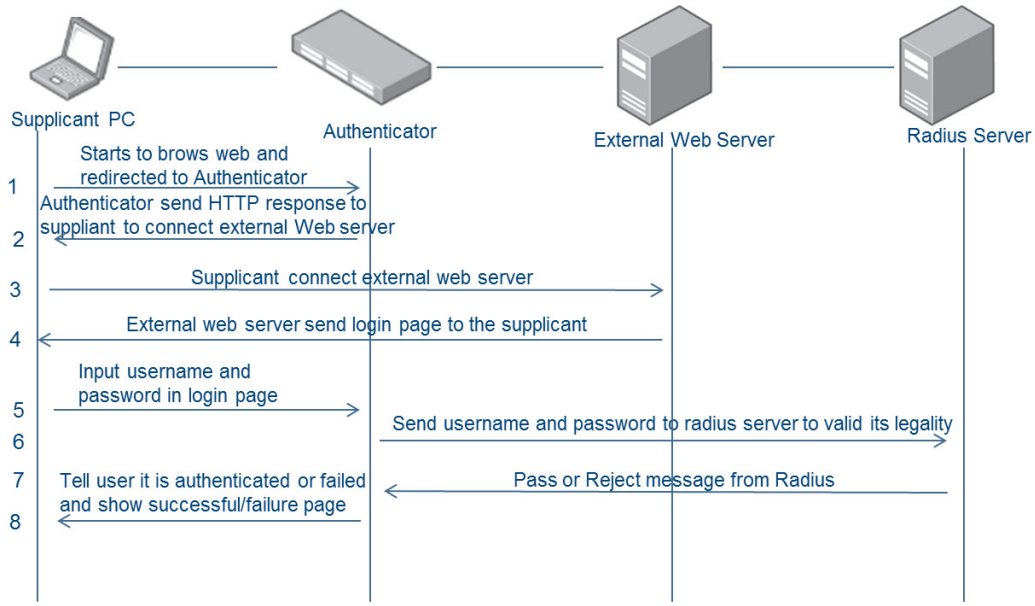
Step 2. **Save the page onto the switch**

Name the file **login_page.html** and save it in the folder **/flash/web-auth/**

Serving your own login page from an external web server

Web authentication supports a method for obtaining a custom login page from an external web server. You can customize this login page fully to give it any appearance you like. See "Customizing the default login page" on page 50 for details.

When web authentication is set up to obtain the login page from an external web server, the sequence of events is as follows:



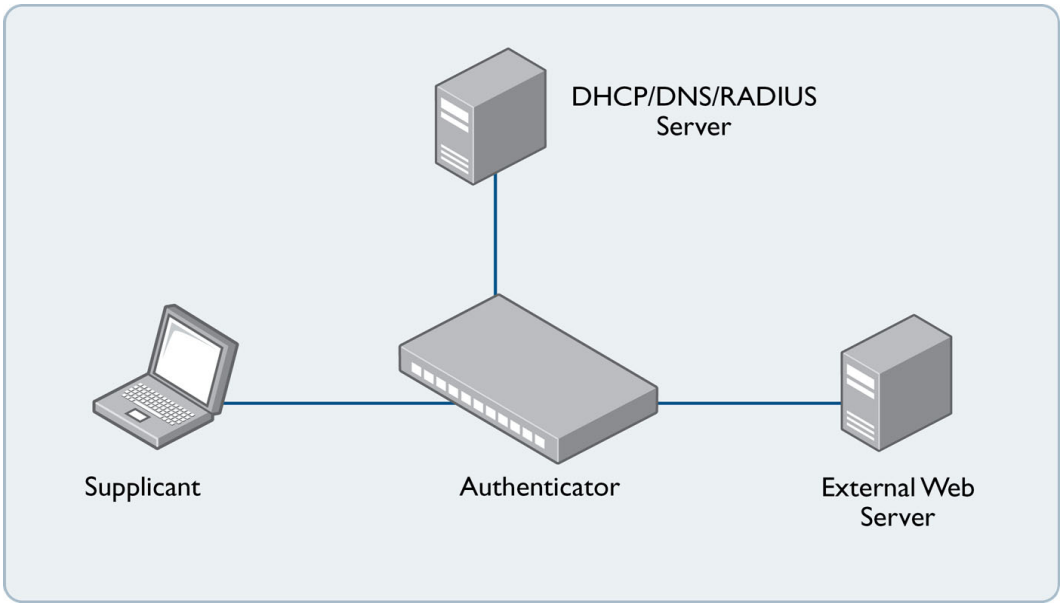
After the supplicant gets an IP address from the DHCP server:

1. The supplicant will start to browse, and the Authenticator will intercept the supplicant HTTP packets.
2. The Authenticator sends an HTTP response packet to the supplicant, and in this packet the Authenticator uses the “refresh” attribute to tell the supplicant to obtain the login page from the external web server.
3. The supplicant sends an HTTP request to external web server requesting the page login.html.
4. The external web server returns the login page. The external server must hold the file that is specified in the command **auth-web-server login-url**.
5. The supplicant then returns to communicating with the Authenticator. When the user enters their username and password, the supplicant sends these to the Authenticator.
6. The Authenticator will pass the user name and password to the RADIUS server for authentication.
7. The RADIUS server sends back the result to the Authenticator.
8. The Authenticator sends the result page to the supplicant. This is not a custom page, but is the standard page built into web authentication.

The role of the external web server is to provide a customized login-page only. web authentication is still performed by the AlliedWare Plus built-in server.

Standard config

Steps 1- 4 below are standard configuration for web authentication, and step 5 explains how to configure the external web server:



Step 1: **Allocate an IP address to the supplicant:**

In this example there is a remote DHCP server. (You can also configure the Authenticator as a local DHCP server if needed).

```
awplus(config)#int vlan1
awplus(config-if)#ip address 192.168.1.2/24
```

Step 2: **Configure the remote DNS server on the Authenticator:**

```
awplus(config)#ip name-server x.x.x.x
awplus(config)#ip dns forwarding
```

Step 3: **Configure auth-web on the Authenticator:**

```
awplus(config)#aaa authentication auth-web default group radius
awplus(config)#int port1.0.1
awplus(config-if)#auth-web enable
```

Step 4: **Configure the remote RADIUS server on the Authenticator :**

(You can also configure the Authenticator as a local RADIUS server if needed.)

```
awplus(config)#radius-server host 192.168.1.254 key 123
```

Step 5: **Now we come to the step that is specific to using the external login server.**

You can use a domain name or the IP address of the external web server (in this example the IP address of the external web server is 192.168.1.1). You also need to allow HTTP packets to the external login server to be forwarded by using the command **auth-web forward**:

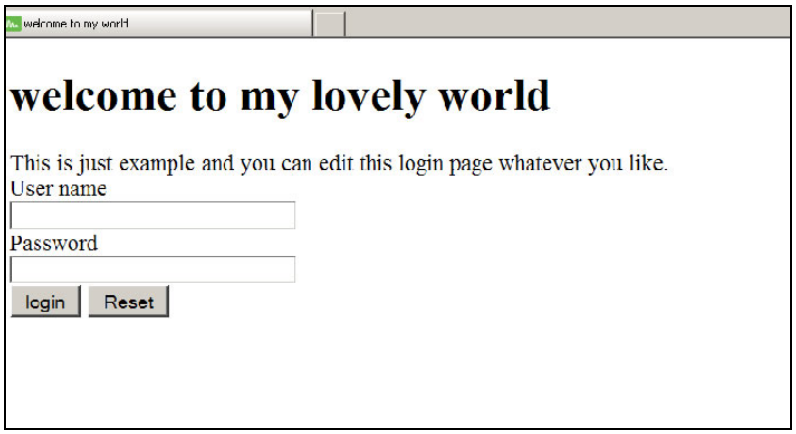
```
awplus(config)#auth-web-server login-url http://192.168.1.1/login.html
awplus (config-if)# auth-web forward 192.168.1.1 tcp 80
```

If the external web server is using another TCP port, then change 80 to the corresponding port number.

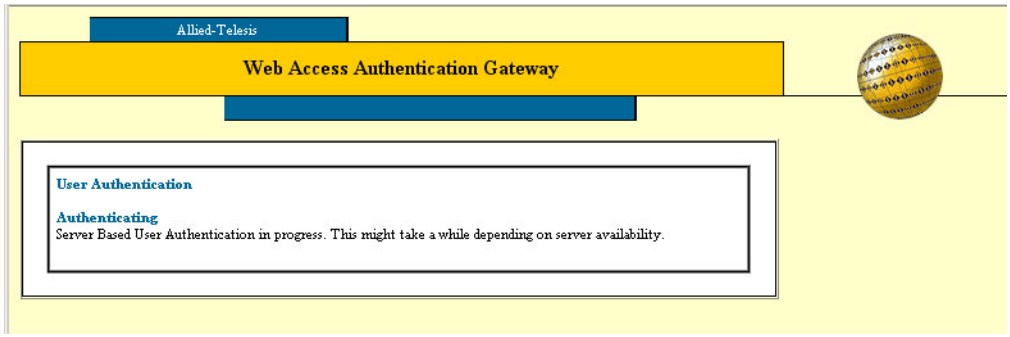
Note: The routing in this network must be set up such that packets can be routed directly between the supplicant and the external web server.

Supplicant login

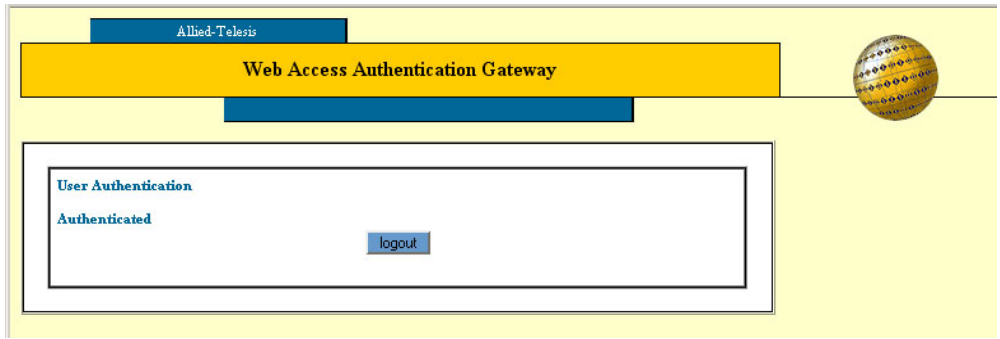
1. When the supplicant tries to access any website, it will be redirected to the external login page for authentication:



2. Enter the username and password. Click **login**. The authenticating page will appear. This is the standard success page from the Authenticator.



Wait for several seconds for the notification of success (or failure).



External login page support

1. A file with the same name as that specified in the command **auth-web-server login-url** must be present on the external web server.

2. The file must contain the following elements:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>HTML 5 complete</title>
  </head>
  <body>‘
    <form action="http://<AW+ IP address>/index.cgi" autocomplete="off"
target="_self" name="AUTH" method="POST">
      <div>User name</div>
      <div><input size="30" type="text" maxlength="64" name="USERNAME"></div>
      <div>Password</div>
      <div><input size="30" type="password" maxlength="64" name="PASSWORD"></div>
      <div>
        <input type="submit" name="ACTION" value="login">
        <input type="reset" name="RESET" value="Reset">
      </div>
    </form>
  <body>
</html>
```

Please note in this feature the Authenticator and the external web server don't communicate with the each other. The role of external web server is just to serve a customized login page to the supplicant.

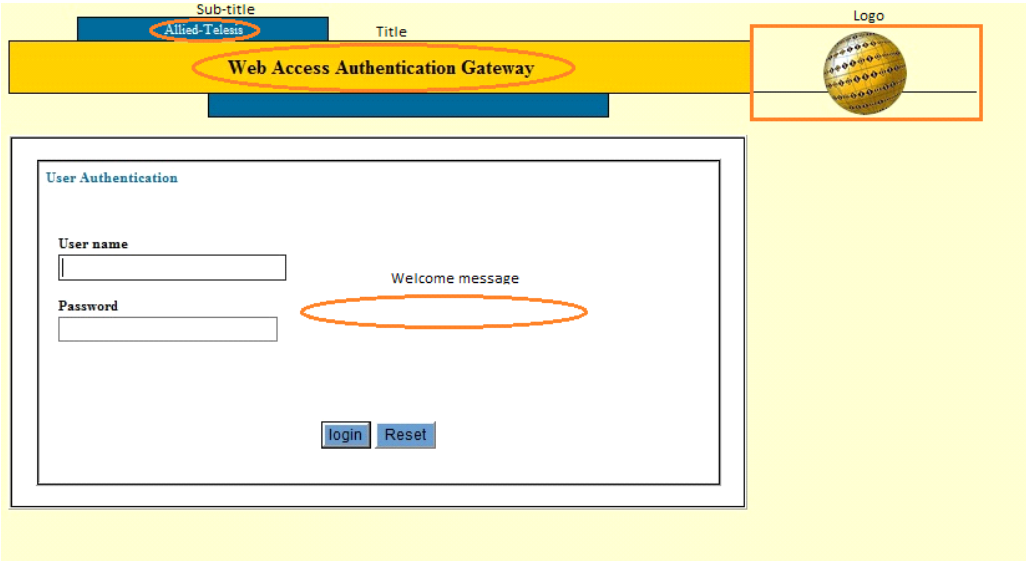
3. Please note that the external login server feature is mutually exclusive with the auth-web DHCP server feature. So it is not possible to configure the commands **auth-web-server login-url** and **auth-web-server dhcp ipaddress** at the same time.

Customizing the default login page

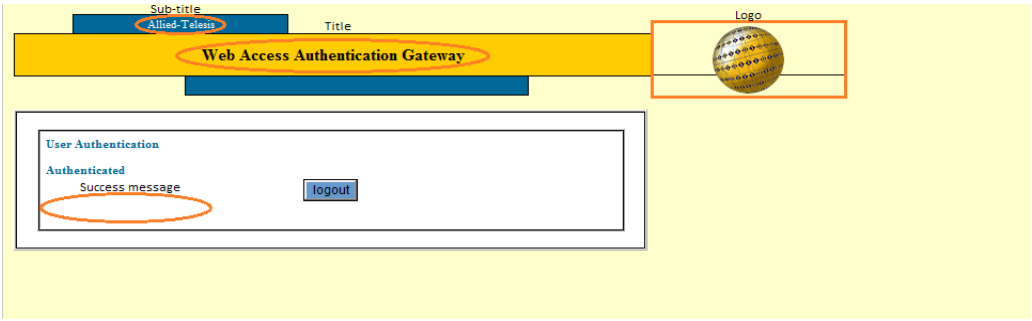
Alternatively, it is possible to do some customization of the pages that web authentication presents to a supplicant.

On the authentication **challenge** page, there are four items that can be customized:

- Title
- Sub-title
- Welcome message
- Logo



The authentication **success** page can also be customized, as can the authentication failure, and any other information web authentication pages.



Customize web-auth pages

To configure customized web authentication pages, use the following steps:

Step 1: **Use the following commands to customize the strings that are present on the pages by default:**

```
auth-web-server page sub-title {hidden|text <sub-title>}
```

```
auth-web-server page title {hidden|text <title>}
```

- The **hidden** option on these commands will, of course, simply remove the string from the page altogether.
- The **no** form of these commands simply takes the string back to its default state.

Step 2: **Use these commands to add the strings that are not present by default:**

```
auth-web-server page success-message text <success-message>
auth-web-server page welcome-message text <welcome-message>
```

- The **no** form of these commands simply removes those strings.

Step 3: **To customise the logo, load your new logo to the location: flash:/logo.gif**

e.g. copy tftp://<tftp server address>/my_logo.gif flash:/logo.gif

By default, the logo displayed on the web authentication pages is:

- The content of flash:/logo.gif if the file exists
- If a flash:/logo.gif does not exist, then the default Allied Telesis globe is used

The following command provides other options:

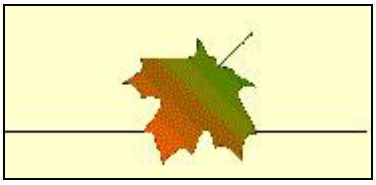
```
auth-web-server page logo {auto|default|hidden}
```

- If **default** is specified, then the logo displayed on the web authentication pages is the default Allied Telesis globe regardless of whether or not flash:/logo.gif exists.
- if **hidden** is specified, then no logo is displayed.
- the **auto** option is the default behaviour described above.

Logo file properties

Format: GIF

Dimensions: The ideal dimensions for the logo file are 185x90 pixels.

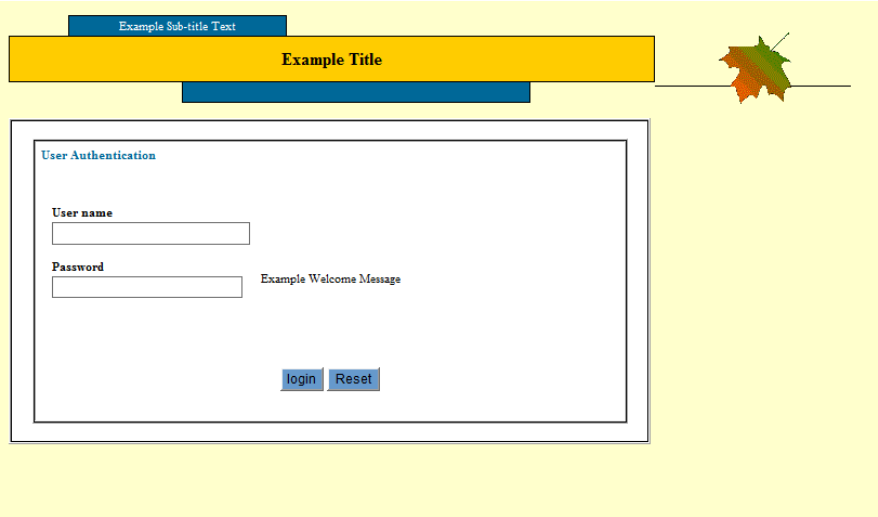


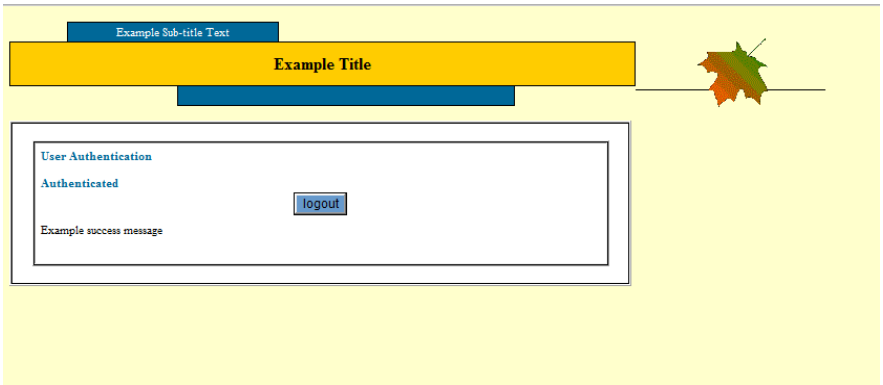
Put it all together example

Using the image above stored as flash:/logo.gif, and the following commands, the authentication challenge page and the success pages appear as shown in the diagrams below:

```
auth-web-server page sub-title text Example Sub-title Text
auth-web-server page title Example Title
auth-web-server page success-message text Example Welcome Message
```

```
auth-web-server page welcome-message text Example Welcome Message
```





Setting the intercept port number

By default web authentication recognizes packets going to TCP port 80 as HTTP packets, and those going to TCP port 443 as HTTPS packets. web authentication redirects HTTP and HTTPS packets received from unauthorized supplicants irrespective of their destination, to its own web authentication server.

If you have Web servers or HTTP proxy servers in your network that are listening on ports other than 80 and 443, then you must register these port numbers as port numbers to intercept. You can configure web authentication to treat particular port numbers as denoting HTTP traffic by using the **auth-web-server intercept-port** command, and other port numbers as denoting HTTPS traffic by using the **auth-web-server ssl intercept-port** command.

For example, If you have an HTTP proxy server listening on TCP port 8080, you must configure the following command:

```
awplus(config)# auth-web-server intercept-port 8080
```

Similarly, if you have an HTTPS server listening on TCP port 900, you must configure the following command:

```
awplus(config)# auth-web-server ssl intercept-port 900
```

If you have proxy server setting in your Intranet, please note that HTTP packets going to the proxy server will be redirected, but web authentication doesn't support the redirecting of HTTPS packets going to a proxy server.

Secure authentication

The web authentication service can be configured to use a secure HTTPS connection. This ensures that the username and password are sent from the supplicant to the switch in encrypted form, and cannot be snooped by anyone eavesdropping on the session.

By default auth-web uses a non-secure HTTP connection to communicate login account information.

You can configure it to use a secure HTTPS instead of HTTP with the command:

```
awplus(config)# auth-web-server ssl
```

You can also use both HTTP and HTTPS using the command:

```
awplus(config)# auth-web-server ssl hybrid
```

When both protocols are used, HTTP packets are redirected to the HTTP server and HTTPS packets are redirected to the HTTPS server respectively.

On startup, if the certificate file **/flash/.web-auth-https/cert.pem** is present on a device, it will be migrated to a PKI trustpoint named ‘web-auth-https’. This trustpoint is then registered for both HTTPS and web authentication traffic. It will persist after a reboot.

To remove this certificate file and the ‘web-auth-https’ trustpoint use the following command:

```
awplus(config)# no crypto pki trustpoint web-auth-https
```

For more information on trustpoints and the AlliedWare Plus PKI implementation, see the Public Key Infrastructure (PKI) Feature Overview and Configuration Guide.

Copying a certificate onto the switch

As well as using the self-created certificate, it is also possible to create a certificate elsewhere, and copy that certificate onto the switch to be the SSL certificate for the web authentication service.

The command to copy the certificate onto the switch is:

```
copy tftp://<tftp server addr>/<certificate file name> web-auth-https-file
```

Note: The file that is copied onto the switch must:

- be in PEM format
- contain both the certificate and the corresponding private key

OpenSSL commands to create a key pair and certificate

Step 1: **Create the self-signed root certificate and key**

(Alternatively you can get a copy of one that already exists.)

- Create the root key:

```
openssl genrsa -out rootCA.key 4096
```

- Create a self-signed root certificate:

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out
rootCA.crt
```

- You will then be prompted for a number of parameters, for example organization name and email address. Enter the relevant details when prompted.
- Verify the root certificate contents:

```
openssl x509 -in rootCA.key -text -noout
```

Step 2: **Create the server certificate and key**

- Create the server private key:

```
openssl genrsa -out privkey.key 2048
```

- Create a certificate signing request for this key:

```
openssl req -new -sha256 -key privkey.key -out server.csr
```

- You will then be prompted for a number of parameters, for example organization name and email address. Enter the relevant details when prompted.
- Verify the certificate signing request's contents:

```
openssl req -in server.csr -noout -text
```

- Generate the server certificate

```
openssl x509 -req -in -server.csr -CA rootCA.key -CAkey rootCA.key
-CAcreateserial -out server.crt -days 1024 -sha256
```

- Verify the certificate's contents:

```
openssl rx509 -in server.crt -text -noout
```

Step 3: **Combine the server private key and certificate into a single file**

- Do this using the cat command:

```
cat server.crt privkey.pem > server.bundle.pem
```

Alternatively this can be done using your favorite text editor.

The order within the file does not matter – the key could be first, or the certificate could be first. Once the file has been copied onto the switch to be the web authentication HTTPS file, the output of the command **show auth-web-server** will look similar to this:

```
awplus#show auth-web-server
Web-authentication server
Server status: enabled
 <SNIP>
Certification: user    <------
<SNIP>
```

If you are not using a certificate that was copied onto the switch, but using one generated by the switch itself, then this is reported as “Certification: Default”. If you wish to remove the certificate that you have copied onto the switch, and go back to using the switch’s self-generated certificate, use the command **erase web-auth-https-file.**

Note: If you have used the command **auth-web-server host-name** to set the host-name of the web authentication server, then the switch will set the certificate's common name (FQDN) to be the same as this configured host name. This is because when a browser verifies a certificate, it will check that the host name and common name of server certificate are the same. If that check fails the Web browser displays an error message indicating peer server may be fake. Also make sure that you have already registered the host name on the DNS server that users will access.

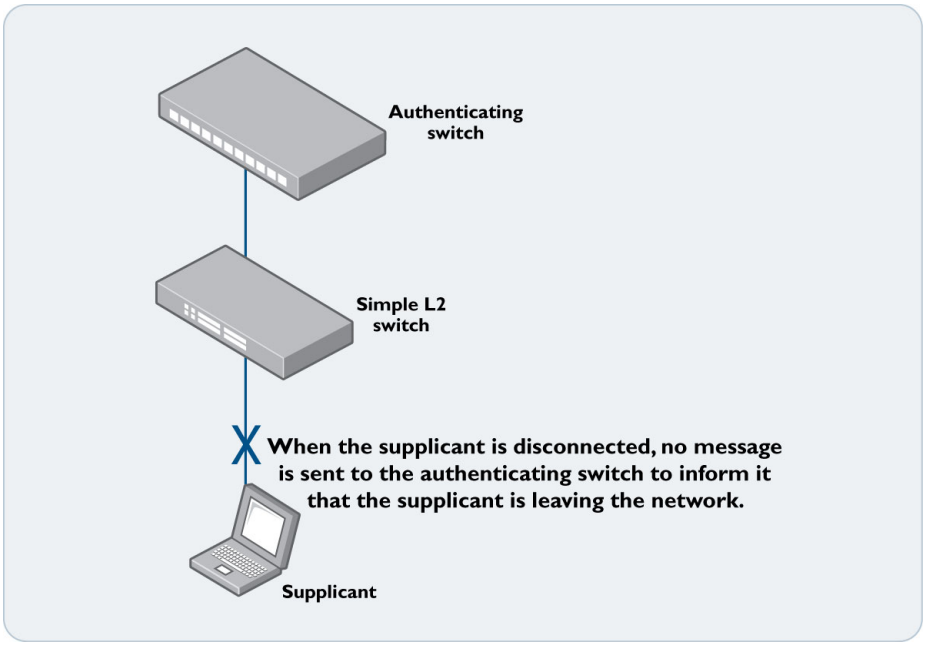
Ping-poll monitoring of supplicant presence

A supplicant’s authenticated session on the network must eventually come to an end. How does the Authenticator decide that a supplicant’s session has ended, and so remove it from the list of authenticated supplicants?

Sometimes it is obvious when the supplicant’s session has ended, if the:

- supplicant unplugs from a port
- the user clicks the **logout** button they were provided with on the “Authentication Success” page, as described in "Starting a web authentication session" on page 39.

Consider the case that a supplicant is not directly connected to the authenticating switch, but is connected to another switch that lies between itself and the authenticating switch, and the user simply disconnects their workstation.



If the network administrator wishes to ensure that the authenticating switch detects the supplicant's disconnection quickly, rather than waiting for the next expiration of the re-authentication period, then they can use **ping polling** to monitor the supplicants.

This feature is enabled by the command:

```
awplus(config)# auth-web-server ping-poll enable
```

Once ping polling has been enabled, the web authentication service will automatically ping-poll every web authentication supplicant once they have been authenticated.

By default the ping-poll has a:

- polling interval of 30 seconds
- timeout of 1 second (i.e. the switch waits 1 second for the ping response before deciding the ping has failed)
- failcount of 5 (i.e. if a given supplicant fails to respond to five pings in a row, its authenticated session is terminated)
- polling type of ping (ICMP)

These default values can be altered by using the commands:

```
auth-web-server ping-poll interval <1-65535>
auth-web-server ping-poll timeout <1-30>
auth-web-server ping-poll failcount <1-100>
auth-web-server ping-poll type {arp|ping}
```

By default the polling type is set to ping (ICMP). If there is a firewall between an authenticating server and a supplicant, it may block ICMP traffic. If this occurs you can change the polling type to ARP with the following command:

```
auth-web-server ping-poll type arp
```

The currently configured values of these parameters can be seen by using the command:

```
show auth-web-server
```

```
awplus#show auth-web-server
Web-authentication server

  PingPolling: enabled
  PingPollingType: Ping
  PingInterval: 30
  Timeout: 1
  FailCount: 5
  ReauthTimerRefresh: disabled
```

Checking the auth-web-server status

To assign a hostname to the web authentication server, use the command:

```
awplus(config)# auth-web-server host-name
```

When you use this command to serve a host-name to represent the web authentication server, the supplicant must be able to resolve that host-name to access the web authentication server. This means the host-name assigned to the web authentication server must be registered to a DNS server. For example, if you enter the command **auth-web-server host-name abcauthenticating**, you also must register a record for **abcauthenticating** to the DNS server. The Authenticator device acts as the default gateway and will register the gateway information when the supplicant is authorized.

Any HTTP Get Request received on an unauthorized interface is redirected to the web authentication server automatically by default. So the command **auth-web-server http-redirect** is hidden.

Checking the IP addresses of the supplicants

To verify the IP addresses of the supplicants that the switch is ping-polling, use the command **show auth supplicant brief**.

```
awplus#show auth supplicant brief
Interface port1.0.19
  authenticationMethod: web

...

Interface   VID  Mode MAC Address    Status        IP Address    Username
=========== ==== ==== ============== ============= ============= ========
port1.0.19  150  W    0000.cd28.0880 Authenticated 192.168.150.9 test
```

Idle time-out

On AR Series devices, web authentication supports an Idle-Time feature. If the authenticator has not seen data from a client for a configurable time period, then the client is automatically set to unauthorized.

To enable the web authentication idle timeout process on eth1, use these commands:

```
awplus#configure terminal
awplus(config)#interface eth1
awplus(config-if)#auth-web enable
awplus(config-if)#auth-web idle-timeout enable
```

To set 30 minutes as the idle-timeout, use these commands:

```
awplus#configure terminal
awplus(config)#interface eth1
awplus(config-if)#auth-web idle-timeout timeout 1800
```

Note that the minimum timeout period is 420 seconds.

Managing unauthenticated supplicant traffic

The forwarding, blocking, and VLAN classification of traffic that arrives at the switch from unauthenticated supplicants is not entirely straightforward, and is subject to configuration.

Before the supplicant tries to authenticate, its packets are managed as shown in the table below, and if a user fails to login 3 times in a row, it will restart the authentication process from the beginning.

Table 1: How web authentication manages different types of traffic

TRAFFIC TYPE	HOW TRAFFIC IS PROCESSED
HTTP packets to web authentication server address	Sent to CPU, processed by web authentication

Table 1: How web authentication manages different types of traffic

TRAFFIC TYPE	HOW TRAFFIC IS PROCESSED
DHCP	By default, traffic is sent to the CPU and passed through to the local subnet (processed by the DHCP server if configured or exists on local subnet). If the interface is configured with the command **no auth-web forward dhcp**, then it is dropped.
DNS	By default, traffic is passed through to the DNS server (by using own routing information). If the interface is configured with the command **no auth-web forward dns,** then it is dropped.
ARP	Traffic is always sent to the CPU, and by default is passed through to the local subnet also. If the interface is configured with the command **no auth-web forward dns**, then it is dropped.
Other packets	If auth-web forwarding is configured with the command **auth-web forward {arp|dhcp|dns|...}**, packets matching the criteria will be forwarded to the native VLAN (not routed to other VLANs). All other packets are dropped.

Monitoring the operation of web authentication

There is no specific debugging available for web authentication. The conversation between web authentication and a RADIUS server can be output by the command:

```
debug RADIUS all
```

An audit trail of web authentication events is kept in the system log. Successful and unsuccessful login attempts, and log offs all generate entries in the system log.

```
2010 Jun4 18:50:54 daemon.notice awplus radiusd[1712]: Login OK: [test] (from
client 127.0.0.1 port 5019 cli 00-00-5e-00-53-01)

2010 Jun4 18:50:56 user.notice awplus 802.1X[1044]: port1.0.19: Web-
authentication successful for test, IP 10.32.4.78, Mac 0000.5e00.5301

2010 Jun 4 18:52:31 daemon.notice awplus radiusd[1712]: Login incorrect: [tester]
(from client 127.0.0.1 port 5019 cli 00-00-5e-00-53-01)

2010 Jun4 18:52:33 user.notice awplus 802.1X[1044]: port1.0.19: Web-
authentication failed for tester, IP 192.168.101.6, Mac 0000.5e00.5301

2010 Jun 15 18:35:00 user.notice awplus 802.1X[1046]: port1.0.19: Supplicant test
unauthorized, Mac 0000.5e00.5301
```

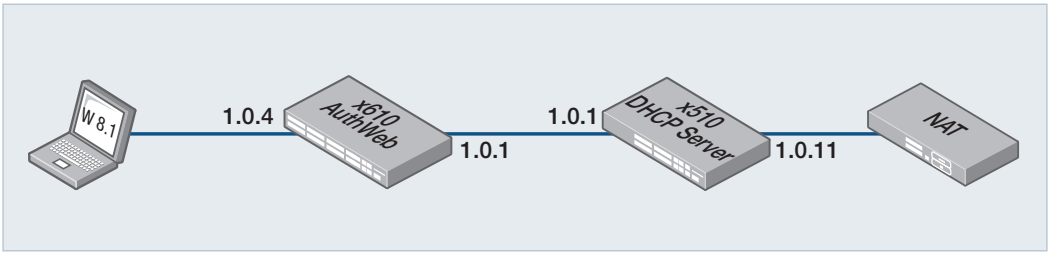
A list of all currently authenticated web authentication supplicants can be seen from the commands:

```
awplus(config)# show auth supplicant
awplus(config)# show auth supplicant brief
```

Configuration example: Guest VLAN and URL redirection

The purpose of this example configuration is to combine Web Authentication with redirection to a specific URL (www.polimi.it) and Guest VLAN. In addition, supplicants use an external DHCP server to assign IPs to authenticated users and also to users in the Guest and temporary VLANs.

The network setup is illustrated in the diagram below:



The configurations are:

x610_AuthWEB

This is the switch that is performing the Web Authentication. It is configured to accept HTTPS connections, redirect users to www.polimi-it, and put failed and guest users into VLAN218.

```
radius-server host 10.168.18.252 timeout 5 retransmit 3 key testing123
!
aaa authentication enable default local
aaa authentication login default local
aaa authentication dot1x default group radius
aaa authentication auth-web default group radius

ip name-server 10.0.0.1
ip domain-lookup

auth-web-server ipaddress 192.168.2.1
auth-web-server ssl
auth-web-server redirect-url http://www.polimi.it
auth-web-server session-keep
!
access-list hardware acl_webauth
 send-to-cpu ip any 192.168.2.1/32

vlan database
 vlan 201 name WR-ED9
 vlan 218 name TESTGUESTWEB

interface port1.0.1
 switchport mode trunk
 switchport trunk allowed vlan add 201,218

[Continued on next page...]
```

```
interface port1.0.2
 switchport access vlan 218

interface port1.0.4
switchport access vlan 201
 access-group acl_webauth
 auth-web enable
 dot1x port-control auto
 dot1x timeout tx-period 5
  218
 auth auth-fail vlan 218
 spanning-tree portfast

interface vlan201
 ip address 10.169.0.201/24

interface vlan218
 ip address 10.168.18.218/24
 ip route 0.0.0.0/0 10.168.18.253
 ip dns forwarding
```

x510_DHVCP_serv:

This is the switch that is operating as the DHCP server.

```
ip dhcp pool dhcp_guest218
 network 10.168.18.0 255.255.255.0
 range 10.168.18.200 10.168.18.235
 dns-server 10.0.0.1
 default-router 10.168.18.254
 lease 0 0 0 20

ip dhcp pool dhcp_201
 network 10.169.0.0 255.255.255.0
 range 10.169.0.200 10.169.0.240
 dns-server 10.0.0.1
 default-router 10.169.0.254
 lease 0 0 60

interface port1.0.1
switchport mode trunk
 switchport trunk allowed vlan add 201,218
 switchport trunk native vlan none

interface port1.0.11
switchport mode trunk
 switchport trunk allowed vlan add 201,218
 switchport trunk native vlan none

interface vlan201
 ip address 10.169.0.253/24

interface vlan218
 ip address 10.168.18.253/24
 ip route 0.0.0.0/0 10.169.0.251
```

What does the user experience?

When a supplicant connects to port1.0.4, they are initially put into the Guest VLAN (VLAN218), and allocated an IP address in the 10.18.168.0/24 subnet.

- If they then try to browse to a website, they are presented with the WebAuth login page.
- If they log in successfully they are:
  - put into VLAN201, and given an IP address in the 10.169.0.0/24 subnet.
  - shown the **Authenticated** page
  - automatically directed to the website www.polimi.it

The terminal monitor on the authorizing switch reports:

```
15:15:14 x610_AuthWEB 802.1X[1431]: Web Authentication successful for
testss@0000.5e00.5300(10.168.18.200) on port1.0.4
15:15:14 x610_AuthWEB NSM[1403]: Interface port1.0.4: authorised (0 --> 1)
15:15:14 x610_AuthWEB NSM[1403]: Notify PM that interface port1.0.4 becomes
authorised by 802.1x
```

DHCP debugging on the x510 shows the change of IP from guest-vlan218 to vlan201.

```
15:18:54 DHCP_serv dhcpd[31660]: DHCPREQUEST for 10.168.18.200 from
00:00:5e:00:53:00 (LAB_D630) via vlan218
15:18:54 DHCP_serv dhcpd[31660]: DHCPACK on 10.168.18.200 to 00:00:5e:00:53:00
(LAB_D630) via vlan218
15:18:44 DHCP_serv dhcpd[31660]: DHCPREQUEST for 10.168.18.200 from
00:00:5e:00:53:00 (LAB_D630) via vlan218
15:18:44 DHCP_serv dhcpd[31660]: DHCPACK on 10.168.18.200 to 00:00:5e:00:53:00
(LAB_D630) via vlan218
15:19:04 DHCP_serv dhcpd[31660]: DHCPNAK on 10.168.18.200 to 00:00:5e:00:53:00 via
vlan201


15:19:04 DHCP_serv dhcpd[31660]: DHCPDISCOVER from 00:00:5e:00:53:00 via vlan201
15:19:05 DHCP_serv dhcpd[31660]: DHCPOFFER on 10.169.0.200 to 00:00:5e:00:53:00
(LAB_D630) via vlan201
15:19:05 DHCP_serv dhcpd[31660]: DHCPREQUEST for 10.169.0.200 (10.169.0.253) from
00:00:5e:00:53:00 (LAB_D630) via vlan201
15:19:05 DHCP_serv dhcpd[31660]: DHCPACK on 10.169.0.200 to 00:00:5e:00:53:00
(LAB_D630) via vlan201
```

MAC Authentication

Why is MAC authentication required?

The authentication mechanisms provided by 802.1X and Web authentication are powerful and effective. But, they are not universally applicable. Web authentication is only applicable to devices that have a human user who opens the web browser and types in a username and password when requested. 802.1X authentication is only possible from devices whose software implements an 802.1X supplicant.

There are plenty of network-connected devices, like printers, scanners, fire-alarm monitors and so on, that have neither a human user nor implement an 802.1X supplicant. In a network that ensures all access is authenticated, there needs to be a mechanism for authenticating these devices.

Fortunately, all Ethernet transceivers have a unique identifier—their MAC address. Hence, even without user input of a username and password, any Ethernet device will automatically identify itself simply by virtue of the source MAC address in the packets it sends. The method that has been developed for authenticating these devices uses the MAC address as the identifier, and so is called MAC-based authentication.

How does MAC authentication work?

In essence, MAC authentication works little differently from 802.1X or web authentication.

Here are the main steps:

1. The supplicant is connected to the switch.
2. The switch (acting as the authenticator) receives an ID from the supplicant.
3. The switch passes the supplicant's ID to a RADIUS server in an Access-Request packet
4. The RADIUS server returns an Access-Accept or an Access-Deny. The Access-Accept can be accompanied with other attributes, for dynamic VLAN assignment.

The unique aspects of MAC authentication are in steps 2 and 3.

MAC authentication does not involve a process whereby the switch sends an ID request to the supplicant. The switch receives the ID from the supplicant by simply looking at the source MAC in the packets being sent from the supplicant.

The MAC address of the supplicant is a single identifier. But a RADIUS access-request requires both a username and a password. The workaround employed by MAC authentication is simply to use the MAC address as both username and password.

The switch extracts the source MAC address from the supplicant's packets and puts it into a string of the form xx-xx-xx-xx-xx-xx, using lower-case letters for any hex digits in the range a-f. This string is then used as both the username and the password in the RADIUS access-request packet. The supplicant MAC address is also sent in the attribute 31 “calling-station-id” as usual.

Configuring MAC authentication on switch ports

Under AlliedWare Plus, there are two steps to setting up MAC authentication.

1. Define the authentication method list that is used for MAC authentication.

There is only one method list that can be created for MAC authentication—the default method list. Moreover, the only authentication server type that can be used is RADIUS.

The command for defining the method list is:

```
awplus(config)# aaa authentication auth-mac default group radius
```

2. Enable MAC authentication on the ports that are to perform this authentication:

The command for defining the method list is:

```
awplus(config)# interface port1.0.2
awplus(config)# auth-mac enable
awplus(config)# spanning-tree edgeport
```

On the RADIUS server, it is necessary to create user entries where both the username and password are the MAC address of the supplicant, in the form xx-xx-xx-xx-xx-xx. For example on the AlliedWare Plus local RADIUS server, the configuration is:

```
awplus(config)# radius-server local
awplus(config-radsrv)# user xx-xx-xx-xx-xx-xx password xx-xx-xx-xx-xx-xx
```

The supplicant, requires no configuration, as the whole purpose of MAC authentication is to authenticate devices that cannot be configured for authentication.

It is also possible to configure the authentication protocol that the switch uses in its interaction with the RADIUS server. There are two choices of protocol: EAP-MD5 and PAP. The default method is PAP, and can be changed by using the command:

```
awplus(config-if)# auth-mac method [eap-md5|pap]
```

MAC authentication on wireless TQ Router Ethernet ports

From AlliedWare Plus version 5.5.5-0.2 onwards, MAC authentication is supported on the Ethernet interfaces of wireless TQ Routers. However, functionality is more limited compared to other AlliedWare Plus products.

MAC Authentication is a simple method of controlling ingress to an interface on a networking device. Multiple supplicants on a port can be individually authenticated, similar to having **auth host-mode multi-supplicant** configured, see "Multi-supplicant modes" on page 27.

The process of configuring MAC authentication on wireless TQ Routers is similar to that on other products. First, define the authentication method list to be used, then enable MAC authentication on the relevant port.

Additionally, you must configure a bridge and add the authenticated Ethernet port to the bridge group. The following is a basic example of bridge configuration:

```
aaa authentication auth-mac default group radius
!
bridge 1
!
interface eth1
 bridge-group 1
 auth-mac enable
```

Authentication commands

These commands are used to configure authentication on TQ Router Ethernet ports:

- **auth-mac username** {hyphen|colon|dot|none} {lower-case|upper-case}

This is similar to the existing ‘auth-mac username’ command, but uses slightly different parameters. It defines the MAC address format used in the username and password sent to the RADIUS server during MAC-based authentication.

The MAC address used as the username and password can be configured in various formats, using either lower-case or upper-case letters. By default, it uses the hyphen-separated, lower-case format.

Format	Example MAC address
hypen	00-15-77-ab-cd-ef
colon	00:15:77:ab:cd:ef
dot	0015.77ab.cdef
none	001577abcdef

For example, to format the MAC address with colons and uppercase letters in the username and password, use the following commands:

```
awplus# configure terminal
awplus(config)# interface eth1
awplus(config-if)# auth-mac username colon upper-case
```

- **auth-mac nas-id** <***name***>

  This command adds a NAS-Identifier attribute to RADIUS authentication requests.

  For example:

  ```
  awplus# configure terminal
  awplus(config)# interface eth1
  awplus(config-if)# auth-mac nas-id NASID100
  ```

- **show auth supplicant**

  This command can be used to show successful and attempted supplicants.

```
awplus# show auth supplicant
Interface    Mode MAC Address    Status
============ ==== ============== =============
eth1         M    0015.77ab.cdef Authenticated
eth1         M    0015.77fe.dcba Failed
```

Note: Wireless TQ-Routers do not support any other AlliedWare Plus authentication commands.

Tri-authentication

Tri-authentication is when multiple port authentication methods (MAC, 802.1X, and/or web-based) are configured on the same interface. With tri-authentication, a supplicant is authorized to use the network as soon as they are successfully authenticated by any of the configured authentication methods.

The default authentication order for tri-authentication is:

- MAC authentication is attempted first.
- If MAC authentication fails and the next packet is:
  - an EAPoL packet, then 802.1X authentication is attempted.
  - HTTP traffic, then web-based authentication is attempted.
- The supplicant is authorized as soon as they are successfully authenticated by any method.

Tri-authentication configuration

The following three steps configure tri-authentication across a range of switch ports:

Step 1: **Define the RADIUS Server:**

Define the RADIUS Server where the switch will send authentication requests by using the commands:

```
awplus# configure terminal
awplus(config)# radius-server host <ip-address> key <key-string>
```

These commands adds the RADIUS Server address and set parameters to the RADIUS server. The key parameter specifies the secret key for the server.

Note: The RADIUS Server, where the switch sends authentication requests, can be the switch's own Local RADIUS Server. For information on how to configure Local RADIUS Server see the Local RADIUS Server Feature Overview and Configuration Guide.

Step 2: **Define the default authentication server lists**

Define the default authentication server lists for 802.1X authentication, web authentication, and MAC authentication:

```
awplus# configure terminal
awplus(config)# aaa authentication dot1x default group radius
awplus(config)# aaa authentication auth-web default group radius
awplus(config)# aaa authentication auth-mac default group radius
```

Step 3: **Enable 802.1X authentication, web authentication, and MAC authentication:**

Follow the instructions below to enable 802.1X authentication, web authentication, and MAC authentication on switch ports to attach supplicant devices. This authenticates the supplicant if any of the three methods that the supplicant tries work.

```
awplus# configure terminal
awplus(config)# interface <interface-range>
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan 1
awplus(config-if)# auth-web enable
awplus(config-if)# auth-mac enable
awplus(config-if)# dot1x port-control auto
awplus(config-if)# auth dynamic-vlan-creation
```

Refer to the other sections of this guide to configure VLANs, IP addresses and other authentication settings for the authentication type you want.

Authentication priority for tri-authentication

From version 5.5.0-2.1 onwards, you can set the order of priority for tri-authentication. By default, no authentication priority is set and a supplicant will try each authentication method until they are successfully authenticated. Once a supplicant is authenticated any future attempts to authenticate are ignored.

When authentication priority is set, however, and a higher priority authentication attempt is made by the supplicant, a new authentication process starts. The supplicant will then be authorized, or unauthorized, based on the result of this new authentication attempt.

Setting the tri-authentication priority:

- Does not change the order of authentication.
- Only affects supplicants that are already authorized.
- Allows an authorized supplicant to initiate another authentication attempt, if the authentication method has a higher priority than the one it was authorized with.
- Applies new information acquired from the RADIUS server during additional authentication attempts, to the supplicant.

Authentication priority example

Giving 802.1X a higher priority than MAC authentication could be useful, for example, in the following scenario.

- A supplicant is authorized on a network using MAC authentication.
- This allows the supplicant to receive the information required to initiate an 802.1X authentication attempt.
- The supplicant is then authorized, or unauthorized, based on the result of this 802.1X authentication attempt.

The following steps describe how to configure 802.1X authentication to have a higher priority than MAC authentication on interface port1.0.1.

Step 1: **Enable 802.1X authentication, and MAC authentication:**

Any authentication methods specified in by authentication priority should be correctly configured.

```
awplus# configure terminal
awplus(config)# interface port1.0.1
awplus(config-if)# switchport mode access
awplus(config-if)# auth-mac enable
awplus(config-if)# dot1x port-control auto
```

Step 2: **Enable 802.1X authentication, and MAC authentication:**

```
awplus(config-if)# auth priority dot1x auth-mac
```

See the relevant sections of this guide to configure VLANs, IP addresses, and other authentication settings for both authorized and unauthorized supplicants.

Two-step Authentication

The single step authentication methods (either user or device authentication) have a potential security risk:

- an unauthorized user can access the network with an authorized device, or
- an authorized user can access the network with an unauthorized device

Two-step authentication solves this problem by authenticating both the user and the device. The supplicant will only become authenticated if both these steps are successful. If the first authentication step fails, then the second step is not started.

By default the following authentication sequences are supported for two-step authentication:

- MAC authentication followed by 802.1X authentication
- MAC authentication followed by web authentication
- 802.1X authentication followed by web authentication.

This order can be changed with the **auth two-step order** command.

To configure two-step authentication:

1. Configure the first authentication method.
2. Configure the second authentication method.
3. Specify the command **auth two-step enable**.
4. Make sure that both authentication steps require different authentication credentials.

Two-step authentication example

To enable MAC authentication followed by 802.1X authentication, use the following commands:

```
awplus# configure terminal
awplus(config)# interface port1.0.2
awplus(config-if)# switchport mode access
awplus(config-if)# auth-mac enable
awplus(config-if)# dot1x port-control auto
awplus(config-if)# auth dynamic-vlan-creation
awplus(config-if)# auth two-step enable
```

Two-step authentication order example

The default two-step authentication order depends on the combination of the authentication methods configured on an interface:

- If MAC authentication is configured then MAC authentication will be the first method.
- If MAC authentication is not configured then 802.1X authentication will become the first method.
- If only two methods are configured then the remaining method becomes the second method.
- If all three methods are configured then the second method is chosen based on the packet type received (802.1X authentication for an EAPOL packet and web authentication for an HTTP packet).

The **auth two-step order** command allows you to change this default order. The first method must be either **auth-mac** or **dot1x** while the second method can be any of the three methods; **auth-mac**, **dot1x**, or **auth-web**.

If, for example, you would like to configure the two-step authentication order to be 802.1X authentication followed by MAC authentication use the following commands.

```
awplus# configure terminal
awplus(config)# interface port1.0.1
awplus(config-if)# switchport mode access
awplus(config-if)# auth-mac enable
awplus(config-if)# dot1x port-control auto
awplus(config-if)# auth two-step enable
awplus(config-if)# auth two-step order dot1x auth-mac
```

Use the **no** variant of this command to reset the order back to default.

```
awplus# configure terminal
awplus(config)# interface port1.0.1
awplus(config-if)# no auth two-step order
```

Use the **show auth interface** command to see the two-step authentication order configuration.

```
awplus(config)# show auth interface
```

```
awplus# show auth interface
Authentication Info for interface port1.0.1
  portEnabled: true - portControl: Auto
  portStatus: Authorized
...
  dot1x: enabled
    protocolVersion: 1
  authMac: enabled
    method: PAP
    reauthRelearning: disabled
  authWeb: disabled
  twoStepAuthentication:
    configured: enabled
    actual: enabled
    order: dot1x mac
...
```

Use the **show auth supplicant** command to see the a supplicant's two-step authentication status.

```
awplus(config)# show auth supplicant
```

```
awplus# show auth supplicant
Interface port1.0.1
  authenticationMethod: dot1x/mac
  Two-Step Authentication
    firstMethod: dot1x
    secondMethod: mac
  totalSupplicantNum: 1
  authorizedSupplicantNum: 1
    macBasedAuthenticationSupplicantNum: 1
    dot1xAuthenticationSupplicantNum: 0
    webBasedAuthenticationSupplicantNum: 0
    otherAuthenticationSupplicantNum: 0

  Supplicant name: 00-00-5e-00-53-00
  Supplicant address: 0000.5e00.5300
    authenticationMethod: MAC-based Authentication
    Two-Step Authentication:
      firstAuthentication: Pass - Method: dot1x
      secondAuthentication: Pass - Method: mac
    portStatus: Authorized - currentId: 3
...
```

Ensuring Authentication Methods Require Different Usernames and Passwords

If you configure a user or device to use multiple authentication methods, you need to set up your system to avoid a potential vulnerability.

The vulnerability occurs because there is no way for a RADIUS server to determine what authentication method you are using. Authentication simply queries a RADIUS server to see whether a username/password pair is valid.

This means that if you use the same RADIUS server for multiple authentication methods, a user can enter the **same** username/password pair for each of these authentication methods. If that username/password pair is valid for one of the methods, it will work for all of them.

This vulnerability is particularly significant for MAC authentication, because the default username and password is the MAC address of the supplicant device, which is easy to discover.

For example, if you set up two-step authentication of MAC authentication and 802.1X authentication, and both use the same RADIUS server, then an attacker does not need to know the 801.1x username and password. Instead, they can pass the 802.1X authentication step by entering the device's MAC address into the 802.1X username and password fields.

To avoid this vulnerability:

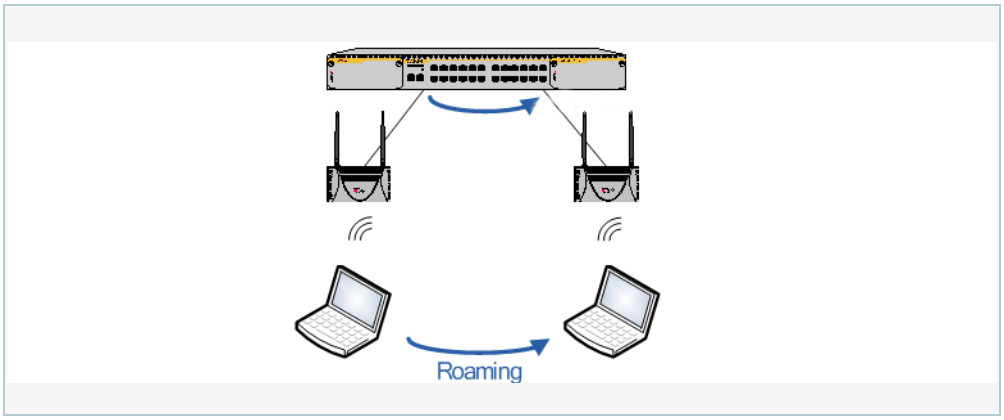
- Use different RADIUS servers for each authentication method, and/or
- Change the default password for MAC authentication, by using the **auth-mac password** command.

Roaming Authentication

When network security is required, the usability of network security must be considered. The Roaming Authentication feature improves the usability of network security by enabling users to move within the network without requiring them to re-authenticate each time they move.

If a supplicant (client device) moves from one wireless access point to another wireless access point, and the wireless access points are connected to different ports, then the switch (authenticator) recognizes that the supplicant has been authenticated and accepts the supplicant without requiring re-authentication.

Figure 1: Diagram showing Roaming Authentication running on a standalone switch

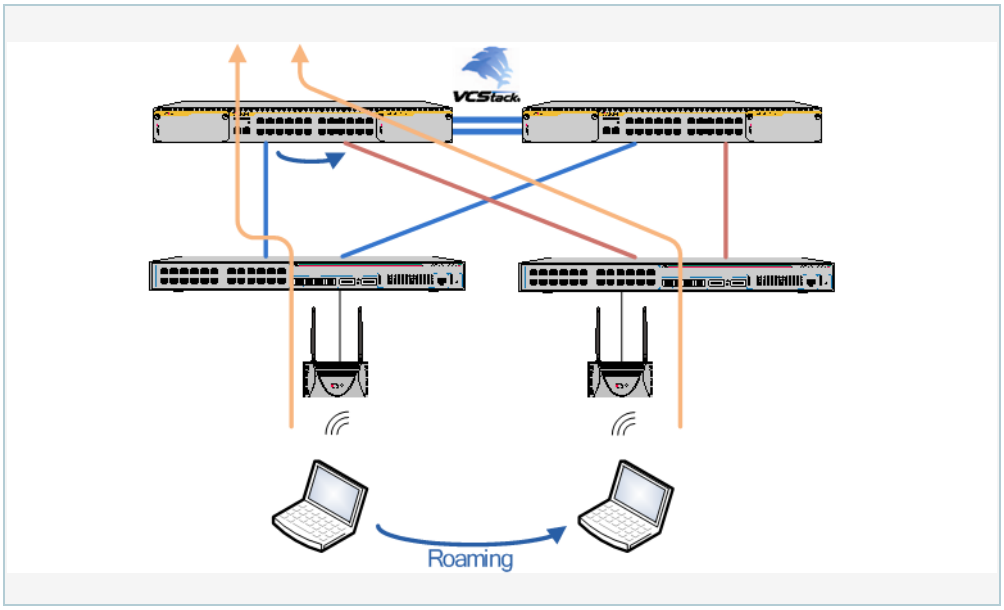


Web authentication and MAC authentication are the authentication methods in a Wireless LAN environment, and 802.1X is the authentication method used for supplicants attached to edge switches.

Roaming Authentication is normally enabled using the **auth roaming enable** command. However, Roaming Authentication has been extended (with the **auth roaming disconnected** command) to work where an interface is link down. This allows you to enable supplicants to move from authenticated interfaces that are link down, without requiring re-authentication.

Roaming Authentication is available for use with the VCStack feature, and is available on static and dynamic (LACP) channel group interfaces.

Figure 2: Diagram showing Roaming Authentication running with VCStack



Roaming authentication overview

Without the Roaming Authentication feature enabled, if a supplicant moves from one switch port to another switch port, the supplicant's authenticated status, authentication, and assigned VLAN is deleted and the supplicant is re-authenticated so the supplicant can access the network, and all traffic from the supplicant is dropped while the supplicant is being re-authenticated.

With the Roaming Authentication feature enabled, a switch port inherits the status of a supplicant from the switch port that the supplicant was moved from. If the Roaming Authentication feature is enabled on a switch, then once a supplicant (client device) is authenticated on the switch it does not have to be re-authenticated if it moves between ports of that switch. Supplicant traffic is not dropped because there is no delay for re-authentication, during which the supplicant cannot access the network.

For example, when the Roaming Authentication feature is used in an wireless LAN environment with wireless access points, then the wireless clients can roam between wireless access points connected to different switch ports without re-authentication.

The Roaming Authentication feature also supports VCStack operation and works on defined static channel group (static aggregators) and dynamic channel group (LACP) interfaces. When VCStack and Roaming Authentication features are used together, the status of a supplicant is inherited from one aggregated interface to another aggregated interface over the stack.

Roaming authentication feature interactions

When the Roaming Authentication feature is disabled, a supplicant must be re-authenticated on the destination interface when it roams. When the Roaming Authentication is enabled, the following supplicant authentication status and information is inherited from the source interface:

- Authentication status
- Authentication method
- Supplicant MAC address
- Supplicant IP address (if an authenticated interface is configured for Web authentication)
- Supplicant name
- Authorized dynamic VLAN ID
- Authorized RADIUS server
- Reauthentication timer (if configured using the auth **timeout reauth-period** command)

Roaming Authentication is only supported between interfaces with the same authentication configuration. If source and destination interfaces have different authentication configuration then the supplicant will be re-authenticated at the destination interface.

When the host mode is set with the **auth host-mode**, a supplicant is not authenticated on a destination interface, and the authentication status is deleted on the source interface.

When a supplicant moves from an interface with authentication configured to an interface without authentication configured, the supplicant's authentication status is deleted.

A supplicant is re-authenticated when it moves to a destination interface that is configured on a different VLAN than the VLAN that is configured for the source interface.

See the following Roaming Authentication feature interactions:

- Multiple Dynamic VLANs are supported when configured with the **auth dynamic-vlan-creation** using the **multi** parameter. Multiple Dynamic VLANs are disabled by default.
- Supplicants are re-authenticated on the destination interface if the VLAN ID changes when Single Dynamic VLANs are configured with the **auth dynamic-vlan-creation** the using the **single** parameter. Single Dynamic VLANs are disabled by default.
- The Roaming Authentication feature is supported on Guest VLANs configured by the **auth guest-vlan** command.
- The Roaming Authentication feature will not function with Dynamic VLANs.

When the Roaming Authentication feature is configured for use on a stack with the VCStack feature, note that supplicants are initialized and re-authenticated if a VCStack failover occurs.

Unauthenticated Supplicant Traffic

When any authentication is configured on a switch port, the question arises as to what the switch does with packets that arrive into the switch port from unauthenticated supplicants.

Unauthenticated supplicants fall into three categories listed below:

- Newly attached supplicants, which are still in the process of their first authentication attempt
- Supplicants that have made an authentication attempt, but have failed authentication
- Supplicants that have been attached, but have not made an authentication attempt. For example, on a port that has only 802.1X authentication enabled, any supplicant that has no 802.1X client software will not be able to attempt 802.1X authentication.

In switches that are running the AlliedWare Plus Operating System, packets from all these three categories of unauthenticated supplicants are treated equally; no distinction is made between these three categories. The treatment of the traffic from unauthenticated supplicants does, however, depend on two factors:

- Whether a Guest VLAN has been configured on the switch port to which the supplicant is attached.
- Whether Web authentication has been configured on the switch port to which the supplicant is attached.

The rules governing the treatment of packets from unauthenticated supplicants are laid out in the table below:

Table 1: Treatment of packets from unauthenticated supplicants

SWITCH PORT CONFIGURATION	NO GUEST VLAN CONFIGURED	NO GUEST VLAN CONFIGURED, AUTH-FAIL VLAN CONFIGURED	GUEST VLAN CONFIGURED
web authentication configured	Packets from unauthenticated supplicants are associated with the Native VLAN of the port. Packets from unauthenticated supplicants are processed according these rules: ■ Packets destined to the WebAuth server IP address/ TCP port are forwarded to the server (which may well be the switch itself). ■ DHCP packets are sent to the CPU, to be processed by a local DHCP server, or relayed to another DHCP server, depending on the configuration of the switch. ■ DNS packets are forwarded to the CPU, and then sent on to a DNS server, if the switch is configured with a DNS server address. ■ ARP packets are forwarded to the CPU, and an ARP entry for the supplicant is learnt. ■ If web-auth forwarding is enabled for particular types of packets, then those packets will be forwarded within the Native VLAN ■ All other packets are dropped.	Packets from unauthenticated supplicants are associated with the Native VLAN of the port. If newly connected supplicants attempt 802.1X port authentication or web authentication and fail, then they are moved to the auth-fail VLAN.	Packets from unauthenticated supplicants are associated with the Guest VLAN of the port. Packets from unauthenticated supplicants are processed according to these rules: ■ Packets destined to the WebAuth server IP address/TCP port are forwarded to the server (which may well be the switch itself). ■ DHCP packets are sent to the CPU, to be processed by a local DHCP server, or relayed to another DHCP server, depending on the configuration of the switch. ■ DNS packets are forwarded to the CPU, and then sent on to a DNS server, if the switch is configured with a DNS server address. ■ ARP packets are forwarded to the CPU, and an ARP entry for the supplicant is learnt. ■ Drop all other packets destined to the IP address of the Guest VLAN. ■ Layer 2 forward packets destined to other addresses within the Guest VLAN. ■ All other packets are dropped.
No web authentication configured	All non-eap packets from unauthenticated supplicants are dropped.	All non-eap packets from unauthenticated supplicants are dropped.	Packets from unauthenticated supplicants are associated with the Guest VLAN of the port. The packets are processed according to these rules: ■ Drop packets destined to the IP address of the Guest VLAN. ■ Layer 2 forward packets destined to other addresses within the Guest VLAN. ■ Drop all other packets.

Deciding when a supplicant fails authentication

Although the treatment of packets from unauthenticated supplicants does not differentiate between the three categories of supplicant, it is still useful to know for sure when the switch decides that a supplicant has failed authentication.

The rules for deciding that a supplicant has failed authentication are listed below for each type of authentication available:

Deciding when a supplicant fails 802.1X authentication

If the supplicant responds to EAP authentication requests, and the supplicant's authentication information is sent to the RADIUS server, and the RADIUS server replies with an Authentication-Reject, then the supplicant is immediately deemed to have failed authentication.

If the supplicant does not respond to EAP authentication requests, then the switch will resend the authentication requests up to a maximum number of attempts set by the command **dot1x max-reauth-req** (the default is 2). The interval between the attempts is set by the command **dot1x timeout tx-period** (the default is 30 seconds). If the supplicant still has not responded after this, it is deemed to have not attempted authentication.

Deciding when a supplicant fails Web authentication

As soon as the supplicant attempts any web-browsing, the switch will intercept the web session, and present the supplicant with an authentication request page. If the user enters a username and password, and clicks the login button, then the switch will send the username and password to the RADIUS server. If the RADIUS server replies with an Authentication-Reject, then the supplicant is immediately deemed to have failed authentication.

Until the supplicant has attempted any web-browsing, or has received the authentication request page, but not yet clicked the login button, the supplicant is deemed to be not yet authenticated (as against not able to authenticate).

Deciding when a supplicant fails MAC authentication

As soon as the supplicant sends any packet, the source MAC address from the packet will be sent to the RADIUS server for authentication. If the RADIUS server replies with an Authentication-Reject, then the supplicant is immediately deemed to have failed authentication.

With MAC-auth there really is no concept of not-yet-attempted authentication, because authentication is attempted as soon as a supplicant sends a packet.

Failed authentication VLAN

The auth-fail VLAN feature allows the network administrator to separate the supplicants who attempted authentication, but failed, from the supplicants who did not attempt authentication.

This feature enables the network administrator to enact a security policy in which the supplicants who fail authentication are given extremely limited access, or are given access to remedial applications.

If the Guest VLAN and auth-fail VLAN are both configured on a switch, then a newly connected supplicant initially belongs to the Guest VLAN. If newly connected supplicants attempt 802.1X port authentication or web authentication and fail, then they are moved from the Guest VLAN to the auth-fail VLAN.

The criteria for how many failed authentication attempts are allowed before the supplicant is moved to the auth-fail VLAN differs, depending on the authentication method used.

If web authentication is used, then the supplicant is moved to the auth-fail VLAN after the first failed attempt. If 802.1X port authentication is used, then the supplicant is moved to the auth-fail VLAN after the number of failed attempts is equal to the value configured by the dot1x max-auth-fail command (by default, three failed 802.1X authentication attempts are allowed).

The MAC authentication feature does not support the max-auth-fail option. If auth-fail VLAN feature is used in conjunction with MAC authentication only one attempt is allowed for a MAC authentication supplicant. If the attempt fails, then the supplicant will be treated as ‘Authenticated’ and the interface will be added to the configured auth-fail VLAN.

Limitations on allowed feature combinations

Table 2: Interoperation of authentication types with Guest VLAN and auth-fail VLAN

AUTHENTICATION TYPE:	GUEST VLAN (WITHOUT ROUTING MODE)	GUEST VLAN (WITH ROUTING MODE)	FAILED AUTHENTICATION VLAN
802.1X authentication	Layer 2 forward packets destined to other addresses within the Guest VLAN.	Unauthorized supplicant can access Guest VLAN. Use ACL for security on the interface.	Failed authentication supplicant can access auth-fail VLAN. See limitations table below for ACL usage limitation.
MAC authentication	Layer 2 forward packets destined to other addresses within the Guest VLAN.	Unauthorized supplicant can access Guest VLAN. Use ACL for security on the interface.	Failed authentication supplicant can access auth-fail VLAN. See limitations table below for ACL usage limitation.
Web authentication	Layer 2 forward packets destined to other addresses within the Guest VLAN.	Unauthorized supplicant can access Guest VLAN. Use ACL for security on the interface.	Failed authentication supplicant can access auth-fail VLAN. See limitations table below for ACL usage limitation.

Table 3: Interactions between Guest VLAN and auth-fail VLAN

AUTHENTICATION FEATURE:	GUEST VLAN (WITHOUT ROUTING MODE)	GUEST VLAN (WITH ROUTING MODE)	FAILED AUTHENTICATION VLAN
Guest VLAN (without routing mode)	(Not Available)	(Not Available)	Cannot configure ACLs on the Guest VLAN when it is not in routing mode. The Guest VLAN without routing mode has reserved ACLs already attached to it.
Guest VLAN (with routing mode)	(Not Available)	(Not Available)	Configuration of ACLs for additional interface security is recommended.
Failed Authentication VLAN	Cannot configure ACLs on the Guest VLAN when it is not in routing mode. The Guest VLAN without routing mode has reserved ACLs already attached to it.	Configuration of ACLs for additional interface security is recommended.	Failed authentication supplicant can access auth-fail VLAN. See limitations table below for ACL usage limitation.

Hardware forwarding on the Guest VLAN

By default all traffic on the guest VLAN is forwarded by the CPU. From AlliedWare Plus version 5.5.1-0.1 you can enable hardware forwarding of guest VLAN traffic. This means that guest VLAN traffic is forwarded by the switching chip, which greatly improves the throughput. This feature is useful if devices are expected to be on the guest VLAN and web authentication or guest VLAN forwarding are not required.

Hardware forwarding cannot be used with features that require the CPU. This means the following are not supported:

- Web-based authentication: web-based authentication is not allowed on a port if hardware forwarding is enabled on that port. If you have already enabled web-based authentication then you must disable it before enabling hardware forwarding.
- Guest VLAN packed forwarding: packet forwarding using the **auth guest-vlan forward** command cannot coexist with hardware forwarding. You must disable it on a port before enabling hardware forwarding.

Configuring hardware forwarding on the guest VLAN

For example, to enable the hardware forwarding on switchport port1.0.2, use the following commands:

```
awplus# configure terminal
awplus(config)# interface port1.0.2
awplus(config-if)# auth guest-vlan hw-forwarding
```

It is disabled using the following command:

```
awplus(config-if)# no auth guest-vlan hw-forwarding
```

Single Supplicant on Multiple VLANs

From 5.4.8-1.1 onwards, AlliedWare Plus supports packet forwarding on multiple VLANs for an authenticated supplicant attached to a trunked (tagged VLAN) port. This feature is available on all AlliedWare Plus switches.

By default, AlliedWare Plus only allows packet forwarding on the VLAN that a device is authenticated on. This enhancement allows packet forwarding to the attached device on any VLAN configured on the switchport. After the device authenticates it will have access to all VLANs configured on the switchport.

Configuring packet forwarding on multiple VLANs

Packet forwarding on multiple VLANs is only available on trunked (tagged VLAN) ports. Once enabled on a port any supplicant authenticating on that port will have access to any VLANs configured on that port.

For example, to enable the feature on switchport port1.0.2, use the following commands:

```
awplus# configure terminal
awplus(config)# interface port1.0.2
awplus(config-if)# auth multi-vlan-session
```

It is disabled using the following command:

```
awplus(config-if)# no auth multi-vlan-session
```

This sample configuration illustrates packet forwarding on vlan10 (the native vlan), vlan20 and vlan30 for authenticated devices on port1.0.2.

```
radius-server host 192.168.1.40 key test

aaa authentication dot1x default group radius

interface port1.0.2
   switchport
   switchport mode trunk
   switchport trunk native vlan 10
   switchport trunk allowed vlan add 20,30
   dot1x port-control auto
   dot1x control-direction in
   auth host-mode multi-supplicant
   auth multi-vlan-session
```

Port Authentication Profiles

What are port authentication profiles

You can manage authentication configurations with port authentication profiles. Authentication profiles are named objects that aggregate authentication configuration commands and are easy to attach to, and detach from, an interface.

Instead of applying authentication commands directly to an interface they are applied to a profile. This profile, which can be thought of as a template, is then attached to an interface. A single profile can be attached to multiple interfaces, alternatively multiple profiles can be created for one or more interfaces and attached as needed.

Using port authentication profiles

A port authentication profile is defined by using the **auth profile *<profile-name>*** command. This command puts you into authentication profile mode. If ***<profile-name>*** does not exist it will be created; if it does exist then any configuration changes you make will be applied to the existing profile.

The following commands configure an authentication profile called ‘student’. If ‘student’ does not already exist it will be created.

```
awplus# config terminal
awplus(config)# auth profile student
awplus(config-auth-profile)#
```

From the port authentication profile mode authentication configuration commands can be entered exactly as they would be entered in interface mode. For example the following commands add authentication configuration settings to the ‘student’ profile. This profile can then be attached to an interface.

```
awplus(config-auth-profile)# auth-mac enable
awplus(config-auth-profile)# auth-web enable
awplus(config-auth-profile)# auth host-mode multi-supplicant
awplus(config-auth-profile)# auth auth-fail vlan 40
awplus(config-auth-profile)# auth dynamic-vlan-creation
awplus(config-auth-profile)# auth-web max-auth-fail 2
awplus(config-auth-profile)# exit
awplus(config)#
```

A profile is attached to an interface using the **auth profile *<profile-name>*** command from the **interface mode.** It can be attached to a static channel, a dynamic (LACP) channel group, or a switch port. Attach the profile ‘student’ to interfaces port1.0.20 and port1.0.21.

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# auth profile student
awplus(config-if)# exit
awplus(config)#
```

Executing **show running-config**, on the configuration above, displays the following authentication information for the profile and interface:

```
awplus#show running-config
...
!
auth profile student
 auth-mac enable
 auth-web enable
 auth host-mode multi-supplicant
 auth auth-fail vlan 40
 auth dynamic-vlan-creation
 auth-web max-auth-fail 2
!
...
!
interface port1.0.20-1.0.21
 switchport
 switchport mode access
 switchport access vlan 10
 auth profile student
!
...
awplus#
```

Any changes made to a profile is immediately applied to each interface to which it is attached. These commands will enable reauthentication on all interfaces the ‘student’ profile is attached to.

```
awplus# config terminal
awplus(config)# auth profile student
awplus(config-auth-profile)# auth reauthentication
awplus(config-auth-profilr)# auth timeout reauth-period 7200
```

Only one profile can be attached to an interface at a time. If you try to attach a second profile an error message will be displayed. Before attempting to attach a new profile to an interface you need to detach the old one with the **no auth profile *<profile-name>*** command.

For example, if you attempt to attach the profile ‘teacher’ to an interface that already has the profile ‘student’ attached an error message is thrown.

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# auth profile student
awplus(config-if)# auth profile teacher
```

Will fail with the message:

```
% port1.0.20: Need to detach old profile first
% port1.0.21: Need to detach old profile first
```

The correct way to attach a new profile to an interface is to first detach the old profile.

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# no auth profile student
awplus(config-if)# auth profile teacher
```

Port authentication interface mode commands and port authentication profiles cannot be used on the same interface. Issuing authentication commands on an interface that already has a profile attached results in an error, and adding a profile to an interface with an existing configuration results in this configuration being overwritten.

In this example the command to enable two-step authentication fails as a profile is already attached to this interface.

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# auth profile student
awplus(config-if)# auth two-step enable
```

Will fail with the message:

```
% Cannot execute individual command while profile is set on port1.0.20
```

Attaching the ‘student’ profile to an interface that already has two-step authentication enabled; disables two-step authentication and displays a warning message.

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# auth-mac enable
awplus(config-if)# auth two-step enable
awplus(config-if)# auth profile student
```

Displays the warning message:

```
port1.0.20: Warning: discarded existing Port-auth config before attaching the
profile.
port1.0.21: Warning: discarded existing Port-auth config before attaching the
profile.
```

A profile cannot be deleted if it is still attached to an interface. An error message will be displayed if you try and do this.

The profile ‘student’ cannot be deleted because it is still attached to port1.0.20-1.0.21:

```
awplus(config)# interface port1.0.20-1.0.21
awplus(config-if)# auth profile student
awplus(config-if)# exit
awplus(config)# no auth profile student
```

Will fail with the message:

```
% Need to detach the profile from all interfaces first
```

DHCP Framed IP Lease

Introduction

The DHCP Framed IP Lease feature allows you to automatically configure a DHCP IP lease when a network device is authenticated using 802.1X or MAC authentication. This makes it easier to manage multiple network devices using a central RADIUS server to assign IP addresses and other network settings.

This feature is useful when a network device (e.g. an IP camera) needs to be identified by its IP address. If the device needs replacing it is important that the replacement device has the same IP address and network configuration as the faulty one. DHCP Framed IP Lease allows you to store the DHCP configuration for the device, along with its username and password, on the RADIUS server. When the replacement is installed, the IP address and network configuration will be requested from the RADIUS server by the DHCP server and then applied to the supplicant.

How does DHCP Framed IP Lease work?

The DHCP Framed IP Lease feature makes registering DHCP host configuration easier by automatically feeding per-host DHCP configuration from the RADIUS server to the DHCP server based on a host's username and password.

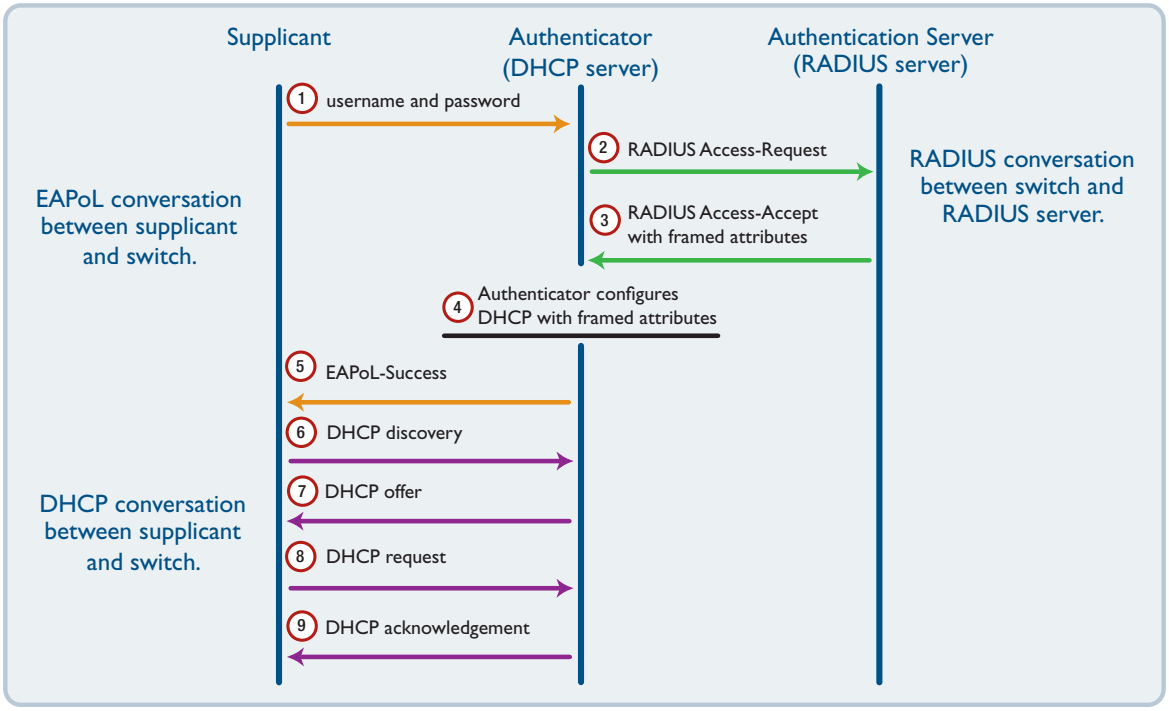
For example, the following shows how the feature works when a supplicant first accesses the network using 802.1X authentication.

1. The supplicant sends a username and password for 802.1X authentication.
2. The authenticator sends an Access-Request message to the RADIUS server.
3. The RADIUS server replies Access-Accept with extra 'framed' RADIUS attributes applicable to that user.
4. The authenticator retrieves the framed RADIUS attributes from the Access-Accept packet and configures DHCP for the supplicant's MAC address.
5. The authenticator sends an EAPoL-Success message to the supplicant.
6. The supplicant (as a DHCP client) starts the DHCP operation by sending a DHCP discovery message to the DHCP server (the authenticator).

7. The DHCP server (the authenticator) sends a DHCP offer message to the supplicant. This offer is based on the configured DHCP information retrieved from the RADIUS server in step 4.
8. The supplicant sends a DHCP request packet for the offered IP address.
9. The DHCP server sends a DHCP acknowledgement to confirm the IP address allocation.

Note: This feature relies on the authenticator being configured with the relevant DHCP server pool for the IP address sent by RADIUS server.

Figure 3: DHCP Framed IP Lease message sequence



Note: If configuring the DHCP server for the supplicant fails in step 4, 802.1X or MAC authentication will continue and an EAPoL-Success packet will still be sent.

Note: If two-step authentication is enabled, and both steps are configured to contact a RADIUS server, the Framed IP Lease configuration is extracted from the second RADIUS response. If, however, the second authentication method is set to force-authorized, which does not require a connection to the RADIUS server, then the configuration is extracted from the first response.

Configuring DHCP Framed IP Lease

You need to complete the following steps to configure the DHCP Framed IP Lease feature on your network.

On the RADIUS server

1. Configure the RADIUS server with the username and password for 802.1X or MAC authentication
2. Configure the following 'framed' RADIUS attributes on the RADIUS server for the user:
   - **Framed-IP-Address (8):** the IPv4 address to be configured for the user
   - **Framed-IP-Netmask (9):** the netmask to be configured for the user
   - **Framed-Route (22):** the default gateway IPv4 address to be configured for the user
   - **Session-Timeout (27):** IP address lease time to be configured for the user

Note: The Frame-IP-Address (8) attribute must be configured for this feature to work. All other attributes are optional.

On the authenticator/DHCP server

3. Configure the RADIUS client
4. Enable 802.1X or MAC authentication on the required interface/s
5. Enable DHCP Framed IP Lease feature on the required interfaces/s
6. Setup a DHCP pool with the network range for the IP address/es registered on the RADIUS server
7. Enable DHCP server

Configuration example

This examples configures the local RADIUS server to provide 'framed' attributes to a network device (such as an IP camera) using 802.1X to authenticate to the network.

On the RADIUS server

Step 1: **Configure the RADIUS server with the username and password for 802.1X authentication:**

This step will depend on your RADIUS implementation. This example snippet is for a local RADIUS server configured on an AlliedWare Plus device.

```
radius-server local
 server enable
 nas 192.168.1.10 key secret
 user camera-user1 encrypted password xxxxx group dhcp-camera-user1
```

Step 2: **Configure the 'framed' RADIUS attributes on the RADIUS server for the user**

This snippet shows the attributes configured as part of the dhcp-camera-user1 group.

```
group dhcp-camera-user1
 attribute Framed-IP-Address 10.1.1.111
 attribute Framed-IP-Netmask 255.255.255.0
 attribute Framed-Route 10.1.1.1
 attribute Session-Timeout 3600
```

On the authenticator/DHCP server

Step 3: **Configure the RADIUS client**

```
awplus# configure terminal
awplus(config)# radius-server host 192.168.1.10 key secret
```

Step 4: **Enable 802.1X on port1.0.1**

```
awplus(config)# aaa authentication dot1x default group radius
awplus(config)# interface port1.0.1
awplus(config-if)# dot1x port-control auto
```

Step 5: **Enable DHCP Framed IP Lease feature on port1.0.1**

```
awplus(config-if)# auth dhcp-framed-ip-lease
awplus(config-if)# exit
```

Step 6: **Setup a DHCP pool**

The network range must include the IP address/es registered on the RADIUS server

```
awplus(config)# ip dhcp pool a
awplus(dhcp-config)# network 10.1.1.0 255.255.255.0
awplus(dhcp-config)# range 10.1.1.101 10.1.1.199
awplus(config)# exit
```

Step 7: **Enable the DHCP server**

```
awplus(config)# service dhcp-server
```

Disable DHCP Framed IP Lease

Use the **no** variant of the command to disable the feature on the required port/s. In the above example you would use the following commands to disable DHCP Framed IP Lease on port1.0.1:

```
awplus(config)# interface port1.0.1
awplus(config-if)# no auth dhcp-framed-ip-lease
```

Monitoring your configuration

The following show commands are useful for monitoring your DHCP Framed IP Lease configuration. These samples assume the configuration in the previous example.

Display the supplicant information using the **show dot1x supplicant** command. This shows the DHCP Framed IP Lease assigned to the supplicant.

```
awplus# show dot1x supplicant
Interface port1.0.1
  authenticationMethod: 802.1X
  totalSupplicantNum: 1
  authorizedSupplicantNum: 1
    macBasedAuthenticationSupplicantNum: 0
    dot1xAuthenticationSupplicantNum: 1
    webBasedAuthenticationSupplicantNum: 0
    otherAuthenticationSupplicantNum: 0

  Supplicant name: 00-00-cd-28-08-80
  Supplicant address: 0000.cd28.0880
    authenticationMethod: 802.1X
    portStatus: Authorized - currentId: 1
    abort:F fail:F start:F timeout:F success:T
    PAE: state: Authenticated - portMode: Auto
    PAE: reAuthCount: 0 - rxRespId: 0
    PAE: quietPeriod: 60 - maxReauthReq: 2
    BE: state: Idle - reqCount: 0 - idFromServer: 0
    CD: adminControlledDirections: in - operControlledDirections: in
    CD: bridgeDetected: false
    KR: rxKey: false
    KT: keyAvailable: false - keyTxEnabled: false
    RADIUS server group (auth): radius
    RADIUS server (auth): 127.0.0.1
    DHCP: IP:10.1.1.111 netmask:255.255.255.0 gateway:10.1.1.1 lease:3600
```

Display the status of the DHCP using the **show ip dhcp pool** command. This shows the supplicant registered as a static host by the DHCP Framed IP Lease feature from AUTHD.

```
awplus# show ip dhcp pool

Pool a :
  network: 10.1.1.0/24
  address ranges:
    addr: 10.1.1.101 to 10.1.1.199
             (static host addr 10.1.1.111 excluded)
  static host addresses:
    addr: 10.1.1.111       MAC addr: 0000.cd28.0880
                           Netmask : 255.255.255.0
                           Gateway : 10.1.1.1
                           Lease   : 3600 seconds
                           Added by AUTHD
  lease <days:hours:minutes:seconds> <1:0:0:0>
  subnet mask: 255.255.255.0 (pool's network mask)
  Probe:                        Default Values
    Status:       Enabled      [Enabled]
    Type:         Ping         [Ping]
    Packets:      5            [5]
    Timeout:      200 msecs    [200]
  Dynamic addresses:
    Total:        98
    Leased:       0
    Utilization:  0.0 %
  Static host addresses:
    Total:        1
    Leased:       1
```

Dynamic ACL Assignments with Port Authentication

Introduction

From version 5.5.0-1.1 onwards, you can configure port authentication to dynamically apply Access Control Lists (ACLs) when a supplicant is authorized. Dynamic ACL rules are defined on the RADIUS server for each supplicant and applied to the port after the supplicant is authorized. When the supplicant is unauthorized these ACLs are removed from the switchport automatically.

Dynamic ACLs gives you greater control over a supplicant's network access by pairing ACLs with 802.1X, MAC, and web-based authentication. Dynamic ACLs offer the following advantages:

- They provide an easy way of applying ACLs per supplicant.
- As they are only installed when a supplicant is connected (and authorized), the number of ACLs is kept low to avoid wasting hardware resources.
- ACLs are configured on the central RADIUS server, this means they can be applied to all switches that use the same RADIUS server.
- Dynamic ACLs support the same filtering rules as static ACLs and will work with IP, IPv6, ICMP, UDP, TCP, IP protocol and MAC matches.
- You can configure a mix of static and dynamic ACLs on a switch.

How do Dynamic ACLs work?

Dynamic ACLs are created and attached to a switchport in the following way:

- A RADIUS server holds the ACL rules for a supplicant, along with that supplicants user name and password.
- These ACL rules can be either:
  - a complete IPv4 or IPv6 hardware rule (stored as a RADIUS NAS-Filter-Rule attribute), or
  - a name or number reference to an existing ACL rule (stored as a RADIUS Filter-Id attribute).
- When the first packet from a supplicant arrives on a switchport the authenticator starts the authentication process with the RADIUS server.
- The RADIUS server returns an Access-Accept packet to the authenticator with the configured ACL rules.
- The authenticator creates and installs these ACLs dynamically on the switchport and authorizes the supplicant.

- These Dynamic ACLs are named IPv4 and IPv6 hardware access-lists. They are internally maintained and cannot be removed or modified from the CLI.
- All traffic on the switchport is filtered using these ACLs.
- When the supplicant is unauthorized, dynamically installed ACLs are removed and detached from the switchport.

Notes:

- On VCS stacks, the dynamic ACLs are updated on all stack members.
- Dynamic ACLs are only supported on switchports and static aggregators.
- Multiple ACLs are allowed, but a mixture of complete ACL rules and numbered/named ACL rules is not supported. This means you cannot have both NAS-Filter-Rule and Filter-Id RADIUS attributes defined.
- Dynamically created ACLs use the following naming convention:
  - For IPv4 rules, a named hardware access-list **dacl-*<port-name>*-*<MAC-address>*** is created. For example **dacl-port1.0.1-0000.5e00.5300**.
  - For IPv6 rules, a named hardware access-list **dacl-*<port-name>*-*<MAC-address>*-ipv6** is created. For example **dacl-port1.0.1-0000.5e00.5300-ipv6**.

Configuring Dynamic ACL assignments.

To use Dynamic ACLs the following configuration steps are required:

Step 1: **Setup your AAA configuration.**

For example, to configure MAC authentication see the “Configuring MAC authentication on switch ports” section.

Step 2: **Enable port authentication on the required ports.**

```
awplus# configure terminal
awplus(config)# interface port1.0.1
awplus(config-if)# auth-mac enable
```

Step 3: **Enable Dynamic ACL on the port**

Dynamic ACLs are disabled by default.

```
awplus# configure terminal
awplus(config-if)# auth dynamic-acl enable
```

Step 4: **Configure dynamic ACL rules on the RADIUS server**

These can be configured as complete IPv4 or IPv6 hardware rules using the NAS-Filter-Rule RADIUS attribute or as references to existing named or numbered ACLs using the Filter-Id RADIUS attribute.

For example, to reject IP traffic from 192.168.1.x to any destination except 192.168.2.x, define these two rules for the user on the RADIUS server using the NAS-Filter-Rule attribute:

```
NAS-Filter-Rule = "ip:permit ip 192.168.1.0/24 192.168.2.0/24"
NAS-Filter-Rule = "ip:deny ip 192.168.1.0/24 any"
```

Alternatively, the format for defining the same ACLs using Filter-Ids could be:

```
Filter-Id = "3000"
Filter-Id = "3001"
```

where ACLs have been defined on the authenticator device as follows:

```
awplus(config)#access-list 3000 permit ip 192.168.1.0/24 any
awplus(config)#access-list 3001 deny ip 192.168.0.0/24 any
```

For more information about configuring ACLs, see the ACL Feature Overview and Configuration Guide.

Considerations when using Dynamic ACLs

Multi-supplicant and multi-host modes

Dynamic ACLs are supported in both multi-supplicant and multi-host modes as well as single-supplicant mode.

When multiple dynamic ACLs are installed on the same switchport, they effect all traffic coming from all supplicants on that port. This means the ACL filter rules must be carefully configured to avoid unexpected behavior in multi-supplicant mode.

Note: In multi-supplicant mode, dynamic ACLs are installed in the order the supplicants are authorized. When a supplicant is unauthorized, the corresponding dynamic ACLs are removed.

In multi-host mode (where authentication from one supplicant is enough to allow other supplicants on the same port), it may be useful to have ACL filter rules by MAC address to reject traffic from certain supplicants.

Roaming Authentication

Roaming authentication allows an authenticated supplicant to move from one interface to another without authentication. In this situation the dynamic ACL filter rules will be installed on the new interface and removed from the old interface.

If, for some reason, the dynamic ACLs fail to install on the new interface then roaming authentication stops and the supplicant starts the reauthentication process on the new interface.

Note: If static ACLs are already configured on the new interface then the cumulative effect of the static and dynamic ACLs may differ from the old interface. It is recommended, therefore, that static ACLs should be configured identically for roaming authentication.

Two-step Authentication

Two-step authentication involves two different authentication methods, for example MAC authentication and 802.1X authentication. Dynamic ACL filter rules are taken from the last RADIUS response to an authentication request. Normally this is the RADIUS response for the second authentication method. If, however, the supplicant is configured with the **skip-second-auth** option, then the RADIUS response for the first authentication method is used.

Configuration example

Configure Dynamic ACLs

Dynamic ACLs are enabled in Interface Configuration mode and work with all port authentication methods (802.1X, MAC and web-based authentication).

Step 1: **Define the RADIUS server to use.**

For example, to use the local RADIUS-server.

```
awplus# configure terminal
awplus(config)# radius-server host 127.0.0.1 key awplus-local-radius-server
```

Step 2: **Define the authentication method list.**

For example, configure MAC authentication.

```
awplus(config)# aaa authentication auth-mac default group radius
```

Step 3: **Enable MAC authentication on the interface.**

```
awplus(config)# interface port1.0.1
awplus(config-if)# auth-mac enable
```

Step 4: **Enable Dynamic ACLs on the interfaces**

```
awplus(config-if)# auth dynamic-acl enable
awplus(config-if)# exit
```

Configure the local RADIUS Server

Dynamic ACLs are defined on the RADIUS server. This can be an external server, like FreeRADIUS server, or your local RADIUS server on the same device.

On the local RADIUS server, dynamic ACLs are defined in a group. This group is then assigned to the users.

Step 5: **Configure the local RADIUS server.**

```
awplus(config)# radius-server local
```

Step 6: **Define a group with the required ACL rules.**

These ACL rules will to reject IP traffic from 192.168.1.x to any destination except 192.168.2.x

```
awplus(config-radsrv)# group dacl-rule
awplus(config-radsrv-group)# attribute repeated NAS-Filter-Rule "ip:permit ip 192.168.1.0/24 192.168.2.0/24"
awplus(config-radsrv-group)# attribute repeated NAS-Filter-Rule "ip:deny ip 192.168.1.0/24 any"
awplus(config-radsrv-group)# exit
```

Step 7: **Add a user with the dynamic ACL rules.**

```
awplus(config-radsrv)# user xx-xx-xx-xx-xx-xx password xx-xx-xx-xx-xx-x
group dacl-rule
```

Step 8: **Enable the local RADIUS server.**

```
awplus(config-radsrv)# server enable
```

Monitoring your configuration

Dynamically created ACLs are displayed using the **show access-list** command. The label '(dynamic)' is added to the end of the access-list name to differentiate them from statically added ACLs.

```
awplus# show access-list
Hardware IP access list 3000
    4 permit ip any any
Hardware IP access list 3001
    4 deny ip 192.168.0.0/24 any
Hardware IP access list dacl-port1.0.49-0000.5e00.5300 (dynamic)
    4 permit ip 192.168.1.0/24 192.168.2.0/24
    8 deny ip 192.168.1.0/24 any
Hardware IPv6 access list dacl-port1.0.49-0000.5e00.5300-ipv6 (dynamic)
    4 deny ipv6 any any
```

The **show interface *<port-list>* access-group** command shows all the ACLs installed on an interface.

```
awplus# show interface port1.0.49 access-group
Interface port1.0.49
  access-group dacl-port1.0.49-0000.5e00.5300
  access-group dacl-port1.0.49-0000.5e00.5300-ipv6
```

ACLs statically configured from the CLI appear in **show running-config** and s**how interface access-group** output. Dynamically installed ACLs, however, only appear in the **show interface access-group** output.

```
awplus# show running-config interface port1.0.49
interface port1.0.49
 switchport
 switchport mode access
 auth-mac enable
 auth host-mode multi-supplicant
 auth dynamic-acl enable
```

Dynamic ACLs are stored in the supplicant configuration and removed when the supplicant is unauthorized. The **show auth supplicant** command has a ‘dynamicACL Rules’ section to show these ACLs

.

```
awplus# show auth supplicant
Interface port1.0.49
  authenticationMethod: mac
  totalSupplicantNum: 1
  authorizedSupplicantNum: 1
    macBasedAuthenticationSupplicantNum: 1
    dot1xAuthenticationSupplicantNum: 0
    webBasedAuthenticationSupplicantNum: 0
    otherAuthenticationSupplicantNum: 0

  Supplicant name: 00-00-53-00-53-00
  Supplicant address: 0000.5e00.5300
    authenticationMethod: MAC-based Authentication
    portStatus: Authorized - currentId: 1
    abort:F fail:F start:F timeout:F success:T
    PAE: state: Authenticated - portMode: Auto
    PAE: reAuthCount: 0 - rxRespId: 0
    PAE: quietPeriod: 60 - maxReauthReq: 2
    BE: state: Idle - reqCount: 0 - idFromServer: 0
    CD: adminControlledDirections: in - operControlledDirections: in
    CD: bridgeDetected: false
    KR: rxKey: false
    KT: keyAvailable: false - keyTxEnabled: false
    RADIUS server group (auth): radius
    RADIUS server (auth): 127.0.0.1
    Session timeout enabled: No
    dynamicACL Rules:
      ip:permit ip 192.168.1.0/24 192.168.2.0/24
      ip:deny ip 192.168.1.0/24 any
      ipv6:deny ipv6 any any
```

RADIUS Change of Authorization

Introduction

From version 5.5.1-1.1 onwards, AlliedWare Plus supports RADIUS Change of Authorization (CoA) to change a supplicant's VLAN or terminate a supplicant's session.

RADIUS CoA provides a mechanism to dynamically change a supplicant's session characteristics after they have been authenticated. RADIUS CoA is an extension to the RADIUS protocol and is defined in RFC5176.

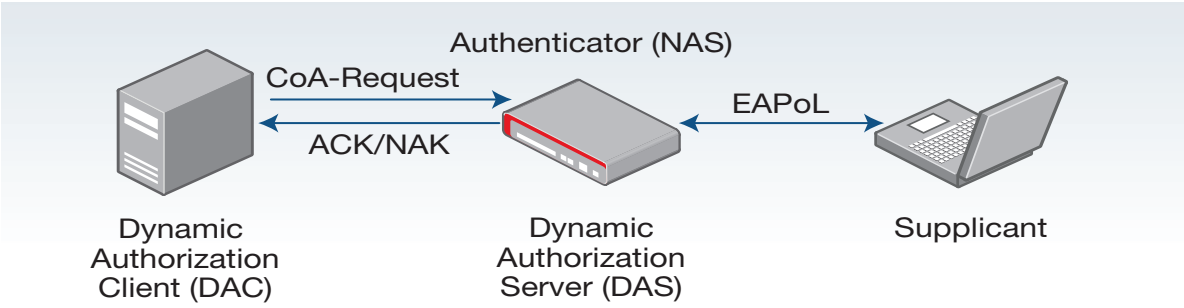
In a typical authorization scenario a supplicant requests permission to join a network using one of the three authentication methods, 802.1X, MAC-based, or web-based. The authenticator, also known as the Network Access Server (NAS), passes the supplicant's credentials to the RADIUS server, which will either accept or reject the request. When accepting the request the RADIUS server can also set certain characteristics of the supplicant's session, such as which VLAN they can access.

The RADIUS protocol does not support unsolicited messages from the RADIUS server to the authenticator. This means, once a supplicant has been authorized their session characteristics cannot be changed unless the RADIUS server is updated and the supplicant re-authenticates. RADIUS CoA allows an administrator to change a supplicant's session characteristics, or terminate a supplicant's session, without the need for the supplicant to re-authenticate.

How does Change of Authorization work?

RADIUS CoA on AlliedWare Plus enables communication between a Dynamic Authorization Client (DAC) and a Dynamic Authorization Server (DAS). This allows an administrator to change the characteristics of a supplicant's session without them needing to re-authenticate. The administrator may choose to move the supplicant to another VLAN or terminate a supplicant's session.

Figure 4: RADIUS Change of Authorization



1. Once a supplicant is authorized, an administrator can make changes to that supplicant's session by initiating a RFC5176 compliant CoA message on a DAC. (The DAC can be, but does not have to be, on the same device as the RADIUS server.)
2. The DAC sends the CoA message to the DAS. This message will either contain a Disconnect-Request or a CoA-Request.
3. On receipt of the message the DAS, which is on the authenticator, does an integrity check of the message. If the message passes the integrity check the authenticator will attempt to process the request and either:
   - return an ACK message to the DAC, if the request is processed successfully,
   - return a NAK message to the DAC, if the request fails.
4. If the request is for a disconnect then the authenticator will move the supplicant into the **Held** state. While in the **Held** state the authenticator will not process any access request messages from that supplicant. After the configurable **auth timeout quiet-period** has elapsed, the authenticator will remove the supplicant from its internal database. This allows subsequent access requests to be processed.
5. The only session change currently supported by AlliedWare Plus is the VLAN attribute. The authenticator will action any CoA-Requests to move a supplicant to another VLAN.
6. For a disconnect message, if the only identification attribute in the CoA message is the NAS-Port then all supplicants connected to that port will be disconnected.

Notes on the AlliedWare Plus implementation of RADIUS CoA.

The AlliedWare Plus implementation of RADIUS CoA has the following limitations:

- Only disconnecting a supplicant or changing a supplicant's VLAN is supported.
- Proxying of CoA messages is not supported.
- An IPsec connection between the DAC and DAS is not supported.

It supports the RADIUS attributes listed below. If the CoA message contains an attribute not on this list, the request will be denied.

At least one of the following is required to identify the authenticator (NAS):

- NAS-IP-Address (4) - this must be the IP address used by the NAS in the original RADIUS authentication transaction
- NAS-Identifier(32)

To match the supplicant(s), at least one of the following is required:

- User-Name (1)
- NAS-Port (5) - for disconnect messages only
- Framed-IP-Address(8) - only use this if the IP address is dynamically allocated using the **auth dhcp-framed-ip-lease** feature
- Calling-Station-Id(31)

It also supports the following attributes:

- State (24)
- Called-Station-Id(30)
- Acct-Session-Id(44)
- Event-Timestamp(55)

The CoA request to change the VLAN is carried in one of the following attributes:

- Egress-VLANID (56)
- Egress-VLAN-Name (58)
- Tunnel-Private-Group-ID (81)

Configuring RADIUS Change of Authorization

Your AlliedWare Plus authenticator (NAS) acts as a Dynamic Authorization Server (DAS). This means it accepts CoA messages from a Dynamic Authorization Client (DAC).

Step 1: Setup the required port authentication and RADIUS server configuration on your device.

See the appropriate section of this document for setting up your configuration.

Step 2: Add a Dynamic Authorization Client (DAC) to your Dynamic Authorization Server (DAS).

All you need to do for your authenticator to accept CoA messages, is to add the DAC to a list of authorized DACs using the following commands:

```
awplus# configure terminal
awplus(config)# radius dynamic-authorization-client <ip-address> key
<key-string>
```

Adding the first DAC enables the Dynamic Authorization Server service.

Step 3: Set your authenticator's NAS-Identifier attribute.

This is required if you wish to use the NAS-Identifier in the CoA requests.

```
awplus(config)# auth radius send nas-identifier [<name>|vlan-id]
```

Step 4: Enable dynamic VLAN assignment.

If you want to use RADIUS CoA to change a supplicant's VLAN then that port must be configured for Dynamic VLANs.

```
awplus(config)# interface <interface-name>
awplus(config-if)# auth dynamic-vlan-creation [rule {deny|permit}] [type
{multi|single}]
```

Step 5: **Remove a Dynamic Authorization Client.**

To remove a DAC from the list of authorized DACs, use the following commands:

```
awplus# configure terminal
awplus(config)# no radius dynamic-authorization-client <ip-address>
```

Removing the last DAC disables the Dynamic Authorization Server service.

Configuration example

The following configuration example will accept CoA messages from a Dynamic Authorization Client with IP address **10.0.2.1** and a shared key of **secret**.

```
hostname x930member
!
radius-server host 10.0.2.1 key secret
!
!
aaa authentication dot1x default group radius
aaa authentication auth-mac default group radius
!
auth radius send nas-identifier guest-nas
radius dynamic-authorization-client 10.0.2.1 key secret
!
vlan database
 vlan 30 name guest_vlan
 vlan 999 name quarantine
 vlan 2,5,10,30,40,60,444,888,999 state enable
!
interface port1.0.3
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 2
 switchport atmf-link
!
interface port2.0.9
 switchport
 switchport mode access
 switchport access vlan 5
 dot1x port-control auto
 dot1x control-direction both
 auth dynamic-vlan-creation
 application-proxy threat-protection quarantine
!
!
interface vlan2
 ip address 10.0.2.67/24
!
```

Monitoring your configuration

To get a list of the Dynamic Authorization Clients currently configured on your device you can use the **show running-config | include dynamic-authorization-client** command.

```
awplus# show running-config | include dynamic-authorization-client
radius dynamic-authorization-client 10.0.2.1 key secret
radius dynamic-authorization-client 10.0.7.1 key secret-key
radius dynamic-authorization-client 10.0.9.1 key super-secret-key
```

Use the **show radius dynamic-authorization counters** command to display the RADIUS CoA message counters. It shows the count of sent and received messages, as well as a count of any error messages.

The counters can be reset to zero using the **clear radius dynamic-authorization counters** command.

```
awplus# show radius dynamic-authorization counters

RADIUS Dynamic Authorization packet counters
--------------------------------------------

Received:
  Disconnect request          : 4
  CoA request                 : 5

Sent:
  Disconnect ACK              : 1
  CoA ACK                     : 1
  Disconnect NAK              : 3
  CoA NAK                     : 1

Dropped:
  Duplicate packet            : 2
  Expired packet              : 0

Error-cause:
  Unsupported attribute       : 0
  Missing attribute           : 0
  Invalid request             : 0
  NAS ID mismatch             : 0
  No session context found    : 3

Errors:
  Unknown message type        : 0
  Unknown client              : 0
  Bad attribute               : 0
  Bad authenticator           : 1
  Malformed packet            : 0
```

Port Authentication for Dynamic Multiple VLAN assignment

Overview

From version 5.5.2-0.1 onwards, AlliedWare Plus supports the dynamic assignment of a supplicant to multiple tagged VLANs. This means an authorized supplicant can access a number of tagged VLANs that have been assigned dynamically by a RADIUS server. The RADIUS server assigns these VLANs by including multiple RADIUS Egress-VLANID(56) and/or Egress-VLAN-Name(58) attributes in the AccessAccept or CoA (Change of Authorization) packets. This feature is available on all platforms that support dynamic VLAN creation and port authentication

How does Dynamic Multiple VLAN assignment work?

The basic process for the dynamic assignment of tagged VLANs is:

1. The normal 802.1x conversation takes place up until an AccessAccept is received from the RADIUS server.
2. If the AccessAccept contains multiple tagged VLAN attributes then these will be processed and if the port has the correct configuration then the VLANs will be added to a list on the port.
3. The tagged VLANs then get assigned to the port in the same way that tagged VLANs are added to a port with the **switchport trunk allowed vlan add** command. However, these VLANs will not appear in the running config.
4. If the port is configured for multi-host, then any other hosts on the same port can join the network on any of the dynamically assigned tagged VLANs.
5. If the primary supplicant becomes unauthorized then all of the dynamically configured tagged VLANs will be removed from the port.

When a CoA request contains multiple tagged VLANs then the following happens:

1. A CoA request containing multiple tagged VLAN attributes is received by the NAS.
2. The multiple VLAN attributes are processed and checked that they exist and the port has the correct configurations.
3. If there are already dynamic VLANs assigned to the port then they will be removed.
4. The new VLANs from the CoA request are added to the port using the same process as tagged VLANs in an AccessAccept.
5. Any supplicants that were authorized on one of the tagged VLANs that have now been removed will be unauthorized from the network.
6. If the primary supplicant becomes unauthorized then the CoA specified VLANs will be removed

from the port.

7. If another supplicant then gets authorized on the port, any dynamically assigned VLANs in the AccessAccept packet will be added to the port again.

Configuring Dynamic Multiple VLAN assignment

This feature requires that the port:

- be in trunk mode with egress filtering enabled (**switchport mode trunk ingress-filter enable**),
- has dynamic VLAN assignment enabled (**auth dynamic-vlan-creation**),
- allows for packet forwarding on multiple VLANs (**auth multi-vlan-session**),
- and is configured with either **auth host-mode single-host** or **auth host-mode multi-host**.

The **auth dynamic-vlan-creation** command allows VLANs to be assigned by the RADIUS server while **auth multi-vlan-session** ensures that more than one VLAN can be used by a single supplicant.

If you wish to have more than one supplicant authorized use the **auth host-mode multi-host** command. This means any subsequent supplicants will share the first (primary) supplicant's authorization. After the primary supplicant is authorized any other supplicants on either dynamically or statically configured VLANs will be force authorized and won't have to go through the authentication process.

If only one supplicant is required to join the network then the **auth host-mode single-host** command should be used.

A RADIUS server is needed to configure this feature. This could be either a local RADIUS server, on an AlliedWare Plus device, or an external RADIUS server, that is able to communicate with the NAS. The RADIUS server should be configured with the user name and password of one supplicant and multiple tagged VLANs.

For multiple supplicants, a switch or TQ wireless access point can act as the primary supplicant. This device should be attached to the auth port on the NAS. It would be this device's credentials that are added to the RADIUS server. It would also need to communicate with the NAS on the native VLAN of the auth port to initiate the authorization process. Other devices would then be connected to the switch and/or TQ. These supplicants would then join the network using primary supplicants authorization.

This feature will not work when:

- a port has been configured to authenticate supplicants individually, i.e. when **auth host-mode multi-supplicant** is enabled on a port.
- authentication roaming is enabled with the **auth roaming enable** command.

Allied Telesis recommend no more than 100 tagged VLANs are specified on a RADIUS server.

Configuration example

This example configures a Network Access Server (NAS) with the local RADIUS server feature. The local RADIUS server sends two VLAN IDs (40 and 50) to an authenticated user via the RADIUS Egress-VLANID(56) attribute. A port on the NAS is then configured to allow a supplicant to dynamically access these VLANs after authentication.

Configuration on the NAS

Step 1: **Configure the local RADIUS server**

```
awplus# configure terminal
awplus(config)# radius-server host 127.0.0.1 key mykey
awplus(config)# radius-server local
awplus(config-radsrv)# nas 127.0.0.1 key awplus-local-radius-server
```

Step 2: **Create a group with tagged VLANs in it**

```
awplus(config-radsrv)# group tagged_vlans
awplus(config-radsrv-group)# egress-vlan-id 40 tagged
awplus(config-radsrv-group)# egress-vlan-id 50 tagged
```

Step 3: **Add the credentials for the supplicant and assign the group to it**

```
awplus(config-radsrv)# user 00-00-f4-27-93-da password 00-00-f4-27-93-da
group tagged_vlans
awplus(config-radsrv)# exit
awplus(config)#
```

Step 4: **Create an authentication method list**

```
awplus(config)# aaa authentication dot1x default group radius
awplus(config)# aaa authentication auth-mac default group radius
```

Step 5: **Configure the port on the NAS**

```
awplus(config)# interface port1.0.1
awplus(config-if)# switchport mode trunk
awplus(config-if)# switchport trunk native vlan 5
awplus(config-if)# auth-mac enable
awplus(config-if)# auth host-mode multi-host
awplus(config-if)# auth dynamic-vlan-creation
awplus(config-if)# auth multi-vlan-session
awplus(config-if)# exit
awplus(config)#
```

Configuration on the supplicant

Step 6: **Configure the port on the supplicant**

```
awplus# configure terminal
awplus(config)# int port1.0.1
awplus(config-if)# switchport mode trunk
awplus(config-if)# switchport trunk native vlan 5
awplus(config-if)# switchport trunk allowed vlan add 40,50
```

Monitoring your configuration

Use the **show auth supp brief** command to check which supplicants have been authorized. It shows which interface the supplicants are authorized on, the ID of the VLAN that the supplicant is using, or was using when it got authorized.

The status tells the user which supplicant was the primary as it will have a status of Authenticated. Those with a status of ForceAuthorized are the supplicants that joined later.

```
awplus#show auth supp brief
Interface port1.0.1
  authenticationMethod: mac
  totalSupplicantNum: 3
  authorizedSupplicantNum: 3
    macBasedAuthenticationSupplicantNum: 1
    dot1xAuthenticationSupplicantNum: 0
    webBasedAuthenticationSupplicantNum: 0
    otherAuthenticationSupplicantNum: 2

Interface   VID  Mode MAC Address    Status            IP Address      Username
=========== ==== ==== ============== ================= =============== ========
port1.0.1   5    M    0000.f427.93da Authenticated     --              00-00...
port1.0.1   50   NONE 0000.0005.0545 ForceAuthorized   --              00-00...
port1.0.1   40   NONE 0000.0005.0540 ForceAuthorized   --              00-00...
```

The **show auth interface** command displays information about one or more port that have been configured to use port authentiction. The dynamically allocated tagged VLAN member count shows how many tagged VLANs have been assigned to the port by the RADIUS server. The line below this shows the tagged VLANs. The **show dot1x interface** command can also be used to show this information.

```
awplus#show auth interface port1.0.1
Authentication Info for interface port1.0.1
  portEnabled: true - portControl: Auto
  portStatus: Authorized
  reAuthenticate: disabled
  reAuthPeriod: 3600
  PAE: quietPeriod: 60 - maxReauthReq: 2 - txPeriod: 30
  PAE: connectTimeout: 30
  BE: suppTimeout: 30 - serverTimeout: 30
  CD: adminControlledDirections: in
  KT: keyTxEnabled: false
  critical: disabled
  guestVlan: disabled
  guestVlanForwarding:
    none
  authFailVlan: disabled
  dynamicVlanCreation: single-dynamic-vlan
    assignFailActionRule: disabled
  multiVlanSession: enabled
  hostMode: multi-host
    maxSupplicant: 1024
  Dynamically allocated tagged vlan member count: 2
    vlan 40,50
  dynamicACLs: disabled
  vlanRestriction: disabled
  dot1x: disabled
  authMac: enabled
    method: PAP
    reauthRelearning: disabled
  authWeb: disabled
  twoStepAuthentication:
    configured: disabled
    actual: disabled
  supplicantMac: none
  supplicantIpv4: none
```

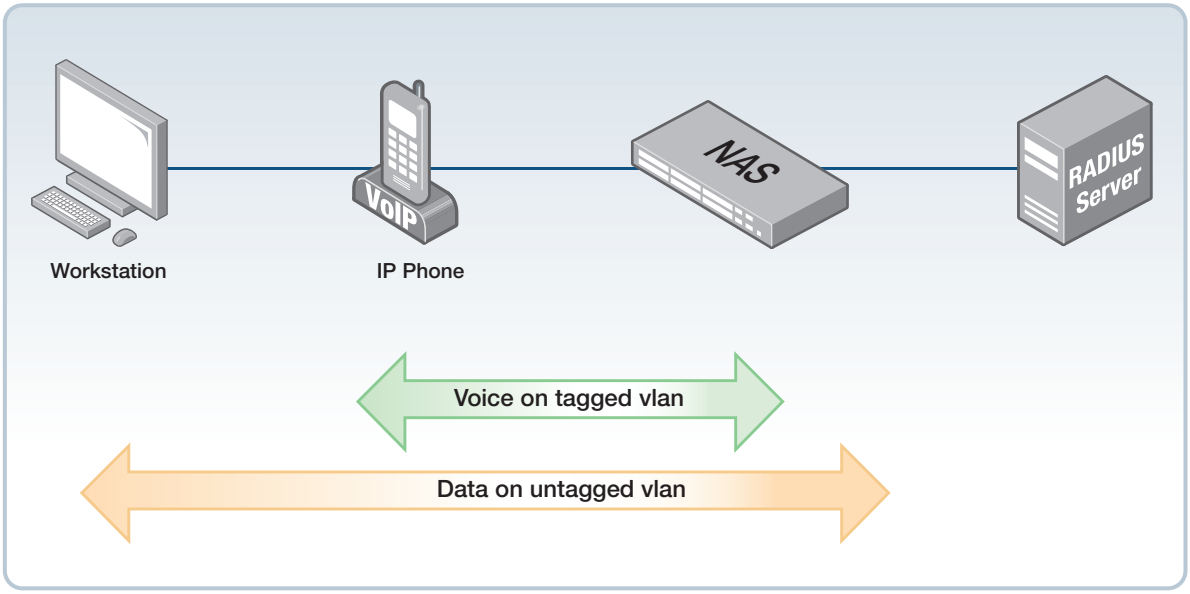
Use the **show vlan** command to check that VLANs have been correctly assigned to a port. In this example, VLAN 5 was statically configured as the native VLAN for port1.0.1, this can be seen by "port1.0.1(u)".

VLAN 50 is a dynamically assigned tagged VLAN and we can see it's been assigned to the port correctly because "port1.0.1(t)" is displayed.

```
awplus#show vlan 5

VLAN ID  Name            Type    State   Member ports
                                         (u)-Untagged, (t)-Tagged
======= ================ ======= ======= ====================================
5       VLAN0005         STATIC  ACTIVE  port1.0.1(u)
awplus#
awplus#show vlan 50

VLAN ID  Name            Type    State   Member ports
                                         (u)-Untagged, (t)-Tagged
======= ================ ======= ======= ====================================
50      VLAN0050         STATIC  ACTIVE  port1.0.1(t)
```

Limit the number of supplicants when connecting via an IP phone

This section describes options to prevent unwanted supplicants from connecting to the network when a host (e.g. a PC) connects to an AlliedWare Plus NAS via an IP phone, as shown in the following figure. In this situation, you mostly want to allow only that host and phone to connect via the port on the NAS.



If you use port authentication, then you can restrict the number of supplicants to 2 using the **auth max-supplicant 2** command. But that does not make sure that the 2 supplicants are the PC and the IP phone. For example, 2 PCs could connect to the network instead.

AlliedWare Plus has 3 mechanisms for ensuring the appropriate supplicants join. Different options are useful in different situations and with different IP phone models. This section describes all 3 options:

- From version 5.5.2-1.1 onwards, you can specify how many tagged and untagged VLANs can authenticate on a port
- From version 5.5.2-1.1 onwards, you can specify that a port can have a single voice and a single data supplicant
- From version 5.5.1-2.1 onwards, you can limit the number of supplicants on an interface to one per VLAN.

Local RADIUS server, authentication method list and port configuration

All 3 examples use the local RADIUS server on the AlliedWare Plus device, and dynamic VLAN assignment. First, follow these steps to configure these:

Step 1: **Configure the local RADIUS server on your device.**

```
awplus# configure terminal
awplus(config)# radius-server host 127.0.0.1 key awplus-local-radius-server
awplus(config)# radius-server local
awplus(config-radsrv)# nas 127.0.0.1 key awplus-local-radius-server
awplus(config-radsrv)# exit
```

For more information, see the Local RADIUS Server Feature Overview and Configuration Guide.

Step 2: **Create the VLANs for the phone (VLAN 50) and the PC (VLAN 5).**

```
awplus(config)# vlan database
awplus(config-vlan)# vlan 5,50
awplus(config-vlan)# exit
```

Step 3: **If using a dynamic voice VLAN then create a group with a tagged VLAN in it.**

```
awplus(config-radsrv)# group phone
awplus(config-radsrv-group)# egress-vlan-id 50 tagged
awplus(config-radsrv-group)# exit
```

Step 4: **Add the credentials for the phone and the PC.**

```
awplus(config-radsrv)# user 00-00-f4-27-93-da password 00-00-f4-27-93-da group phone
awplus(config-radsrv)# user PC1 password example-password
awplus(config-radsrv)# exit
```

Important: this RADIUS configuration is for illustrative purposes only and uses the default password method for the MAC authentication on the phone. This can make the PC user insecure - see "Ensuring Authentication Methods Require Different Usernames and Passwords" for more information. We recommend you change the phone's password, using the **auth-mac password** command.

Step 5: **Create an authentication method list.**

```
awplus(config)# aaa authentication dot1x default group radius
awplus(config)# aaa authentication auth-mac default group radius
```

Step 6: **Configure the port on the NAS.**

```
awplus(config)# interface port1.0.1
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan 5
```

```
awplus(config-if)# auth-mac enable
awplus(config-if)# dot1x port-control auto
awplus(config-if)# auth dynamic-vlan-creation type multi
awplus(config-if)# switchport voice vlan dynamic
```

Option 1: Specify how many tagged and untagged VLANs can authenticate on a port

This option is available from 5.5.2-1.1 onwards. With it, you assign the voice traffic to a tagged VLAN and leave the host's traffic untagged. This option is the most versatile, because you can allow multiple hosts if needed. You can also assign the traffic to the voice VLAN either statically or dynamically.

Note that this option does not work with all IP phones when the voice VLAN is statically assigned, because some phones initially join on the access VLAN, which is untagged. In that case, you need to use another option instead.

After the configuration in "Local RADIUS server, authentication method list and port configuration" on page 113, add the following steps:

Step 1: **Set the host mode to multi-supplicant.**

```
awplus(config-if)# auth host-mode multi-supplicant
```

Step 2: **Set the maximum number of supplicants on tagged and untagged VLANs on the port to 1.**

```
awplus(config-if)# auth max-supplicants tagged-vlan 1
awplus(config-if)# auth max-supplicants untagged-vlan 1
```

Option 2: Specify that a port can have a single voice and a single data supplicant

This option is available from 5.5.2-1.1 onwards. With it, you use the RADIUS attribute 'Cisco-AVPair device-traffic-class=voice' to specify that a supplicant is an IP phone. The NAS considers any supplicant with this RADIUS attribute to be a voice device and any supplicant without this attribute to be a host device. Then you use a host mode option (host-plus-voice) to only let one voice device and one host device join the network.

After the configuration in "Local RADIUS server, authentication method list and port configuration" on page 113, add the following steps:

Step 1: **Use the vendor specific attribute to specify supplicants in this group as voice devices.**

```
awplus(config-if)# exit
```

```
awplus(config-radsrv)# group phone
awplus(config-radsrv-group)# attribute Cisco-AVPair device-traffic-class=voice
awplus(config-radsrv-group)# exit
awplus(config-radsrv# exit
```

Step 2: **Set the host mode to host-plus-voice.**

```
awplus(config)# interface port1.0.1
awplus(config-if)# auth host-mode host-plus-voice
```

Option 3: Limit the number of supplicants to one per VLAN

This option is available from 5.5.1-2.1 onwards. With it, you limit the authorization to a single supplicant per VLAN. While this example shows only one untagged VLAN, with this option, it is possible for other supplicants to join the network on other untagged VLANS. Do not use this option unless that is the behavior you need to have.

This option uses LLDP to identify the IP phone. For more information on configuring IP phones, see the Link Layer Discovery Protocol (LLDP) Feature Overview and Configuration Guide.

After the configuration in "Local RADIUS server, authentication method list and port configuration" on page 113, add the following steps:

Step 1: **Use LLDP to identify the IP phone.**

```
awplus(config-if)# lldp med-tlv-select all
```

Step 2: **Set the host mode to multi-supplicant.**

```
awplus(config-if)# auth host-mode multi-supplicant
```

Step 3: **Enable VLAN restriction.**

```
awplus(config-if)# auth vlan-restriction
```

Monitoring your configuration

Use the command **show auth supplicant brief** to view which supplicants have been authorized or have attempted to be authorized. If a supplicant has not been allowed to authorize because of the restrictions of these features, it will have a status of HELD.

```
awplus#show auth supplicant brief
Interface   VID  Mode MAC Address    Status            IP Address      Username
=========== ==== ==== ============== ================= =============== ========
port1.0.1   5    M    0000.0005.0545 Authenticated     --              PC1
port1.0.1   5    M    0000.0005.0757 HELD              --              PC2
port1.0.1   50   M    0000.f427.93da Authenticated     --              00-00...
```

Use the command **show auth supplicant** to see information about authorized supplicants, including if the supplicant has been specified as a voice device.

```
awplus#show auth supplicant
  Supplicant address: 0000.f427.93da
    authenticationMethod: MAC-based Authentication
    portStatus: Authorized - currentId: 1
    abort:F fail:F start:F timeout:F success:T
    PAE: state: Authenticated - portMode: Auto
    PAE: reAuthCount: 0 - rxRespId: 0
    PAE: quietPeriod: 60 - maxReauthReq: 2
    BE: state: Idle - reqCount: 0 - idFromServer: 0
    CD: adminControlledDirections: in - operControlledDirections: in
    CD: bridgeDetected: false
    KR: rxKey: false
    KT: keyAvailable: false - keyTxEnabled: false
    dynamicVlanId: 0
    dynamicTaggedVlanId: 50
    RADIUS server group (auth): radius
    RADIUS server (auth): 127.0.0.1
    Specified as voice device: Yes
    Session timeout enabled: No
```

Use the command **show auth interface** to see information about the port on the NAS, including the maximum number of supplicants allowed on tagged and untagged VLANS, the host mode, and whether VLAN restriction is on or off. The command **show dot1x interface** also gives this information.

```
awplus#show auth int port1.0.1
Authentication Info for interface port1.0.1
  portEnabled: true - portControl: Auto
  portStatus: Unauthorized
  reAuthenticate: disabled
  reAuthPeriod: 3600
  PAE: quietPeriod: 60 - maxReauthReq: 2 - txPeriod: 30
  PAE: connectTimeout: 30
  BE: suppTimeout: 30 - serverTimeout: 30
  CD: adminControlledDirections: in
  KT: keyTxEnabled: false
  critical: disabled
  guestVlan: disabled
  guestVlanForwarding:
    none
  authFailVlan: disabled
  dynamicVlanCreation: multiple-dynamic-vlan
  multiVlanSession: enabled
  hostMode: multi-supplicant
    maxSupplicant: 1024
    maxSupplicantTagged: 1          <<<< Maximum number of supplicants on tagged VLANs
    maxSupplicantUntagged: 1        <<<< Maximum number of supplicants on untagged VLANs
  Dynamically allocated tagged vlan member count: 0
  dynamicACLs: disabled
  vlanRestriction: disabled
...
```

Specify a RADIUS server that resides in a named VRF

Overview

For security purposes, from version 5.5.2-1.1 onwards, it is possible to specify a RADIUS server with a named VRF. Placing a RADIUS server within a VRF means that no actor that resides outside of the VRF can contact the RADIUS server.

radius-server host

A connection to a RADIUS server is currently configured with the command **radius-server host**:

```
awplus(config-if)# radius-server host {<host-name>|<ip-address>} [auth-port <0-65535>] [acct-port <0-65535>] [timeout <1-1000>] [retransmit <0-100>] [key <string>|key-encrypted <string>]
```

The IP address or hostname specifies the address of the RADIUS server. For correct operation this IP address must be routable from the device.

The connection to the RADIUS server is uniquely defined by hostname and port (auth or acct or both) So 'radius-server host 1.2.3.4 auth-port 1500 key secret' specifies a different connection to 'radius-server host 1.2.3.4 auth-port 1800 key secret'

A radius server host can also be configured with the VRF keyword:

```
awplus(config)# radius-server host {<host-name>|<ip-address>} [vrf <name>] [auth-port <0-65535>] [acct-port <0-65535>] [timeout <1-1000>] [retransmit <0-100>] [key <string>|key-encrypted <string>]
```

```
The named VRF must exist.
```

The IP address specified must now be routable from within the named VRF.

The connection to the RADIUS server is now uniquely defined by hostname, port and VRF name. So 'radius-server host 1.2.3.4 vrf red auth-port 1500 key secret' specifies a different connection to 'radius-server host 1.2.3.4 vrf blue auth-port 1500 key secret'.

Configuring RADIUS server groups

Configuring the switch to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service.

Once again the servers in a group could be uniquely identified by hostname and port. This has been augmented to also include VRF name.

Here we have a RADIUS server specified by IP address:

```
ip vrf red 1

aaa authentication auth-mac default group radius

radius-server host 10.0.0.31 vrf red key secret

vlan database
 vlan 30 name vlan30

interface port1.0.9
! The port under portauth control
 switchport
 switchport mode access
 auth-mac enable

interface port1.0.25
! The physical connection to the RADIUS server
 switchport
 switchport mode access
 switchport access vlan 30

!
interface vlan30
 ip vrf forwarding red
 ip address 10.0.0.30/24
```

Here we have a RADIUS server specified by a hostname. This requires a connection to a DNS server which is also inside a named VRF:

```
ip vrf red 1

aaa authentication auth-mac default group radius

radius-server host example.com2 vrf red key secret

ip name-server vrf red 10.38.34.33
ip domain-lookup via-relay

vlan database
 vlan 30 name vlan30
 vlan 60 name vlan60

interface port1.0.5
! The physical connection to the DNS server
 switchport
 switchport mode access
 switchport access vlan 60
```

```
interface port1.0.9
! The port under portauth control
 switchport
 switchport mode access
 auth-mac enable

!
interface vlan30
 ip vrf forwarding red
 ip address 10.0.0.30/24
!
interface vlan60
 ip vrf forwarding red
 ip address 10.38.34.34/27

ip dns forwarding
```

Monitoring your configuration

Use the command **show radius** to display VRF information if required:

```
Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  VRF                 : red
  Secret Key          : secret
  Timeout             : 80 sec
Server Host : 1.2.3.4
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret
Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret

Server Host/IP       Auth  Acct             Auth           Acct
Address              Port  Port  VRF        Status         Status
-------------------------------------------------------------------------
example.com2         1812  1813  red        Alive          Unknown
1.2.3.4              1812  1813             Unknown        Unknown
example.com2         1812  1813             Unknown        Unknown
x530_NAS#Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  VRF                 : red
  Secret Key          : secret
  Timeout             : 80 sec
Server Host : 1.2.3.4
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret
Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret
```

```
Server Host/IP       Auth  Acct             Auth          Acct
Address              Port  Port  VRF        Status        Status
-------------------------------------------------------------------------
example.com2         1812  1813  red        Alive         Unknown
1.2.3.4              1812  1813             Unknown       Unknown
example.com2         1812  1813             Unknown       Unknown
x530_NAS#Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  VRF                 : red
  Secret Key          : secret
  Timeout             : 80 sec
Server Host : 1.2.3.4
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret
Server Host : example.com2
  Authentication Port : 1812
  Accounting Port     : 1813
  Secret Key          : secret

Server Host/IP       Auth  Acct             Auth          Acct
Address              Port  Port  VRF        Status        Status
-------------------------------------------------------------------------
example.com2         1812  1813  red        Alive         Unknown
1.2.3.4              1812  1813             Unknown       Unknown
example.com2         1812  1813             Unknown       Unknown
```

Use the command **show radius server** group to also display the VRF if configured:

```
awplus#show radius server group admin
RADIUS Group Configuration
  Group Name : admin
    Server Host/IP       Auth  Acct             Auth          Acct
    Address              Port  Port  VRF        Status        Status
    ---------------------------------------------------------------------
    example.com2         1812  1813  red        Alive         Unknown
    1.2.3.4              1812  1813             Unknown       Unknown
```