# How To| Use 802.1x Security with AT-WA7400 APs, AT-8624PoE Switches, and Linux's freeRADIUS and Xsupplicant

## Introduction

This How To Note details how to take advantage of 802.1x security to ensure that users who connect to your wireless LAN are authorised first. Additionally, it gets the RADIUS server to pass a WEP key to the supplicant so that wireless conversations to the access points are encrypted.

The example in this Note uses Linux for both the access controller (RADIUS server) and the supplicant (client).

## References

- The sections on freeRADIUS and Xsupplicant were worked out by following the excellent HOWTO written by Lars Strand: 802.1x Port-Based Authentication HOWTO.
- If you want to create a freeRADIUS and Windows supplicant solution, then consult this document for the Linux configuration, and How To Use 802.1x EAP-TLS or PEAP-MS-CHAP v2 with Microsoft Windows Server 2003 to Make a Secure Network for the Windows configuration. This How To Note is available from the Allied Telesis website.

## Equipment

We used the following devices and software to create and test this setup. The instructions are generic enough to cover any Linux platform, not just Mandrake.

- 2 x AT-8624PoE switches
- 1 x AT-8624T/2M switch
- 2 x AT-WA7400 wireless access points
- 1 x Mandrake 10.1 PC with freeRADIUS 1.0.5 (compiled from source)
- 1 x Mandrake 10.1 laptop with xsupplicant-1.0-2mdk (Mandrake package)

### Terminology

User:

A person.

Client:

A user's laptop or PC.

Wireless Node:

A client on a wireless network. A wireless node is not necessarily authenticated or authorised to use the network.

Supplicant:

The intermediary application normally contained within the wireless node, which handles authentication.

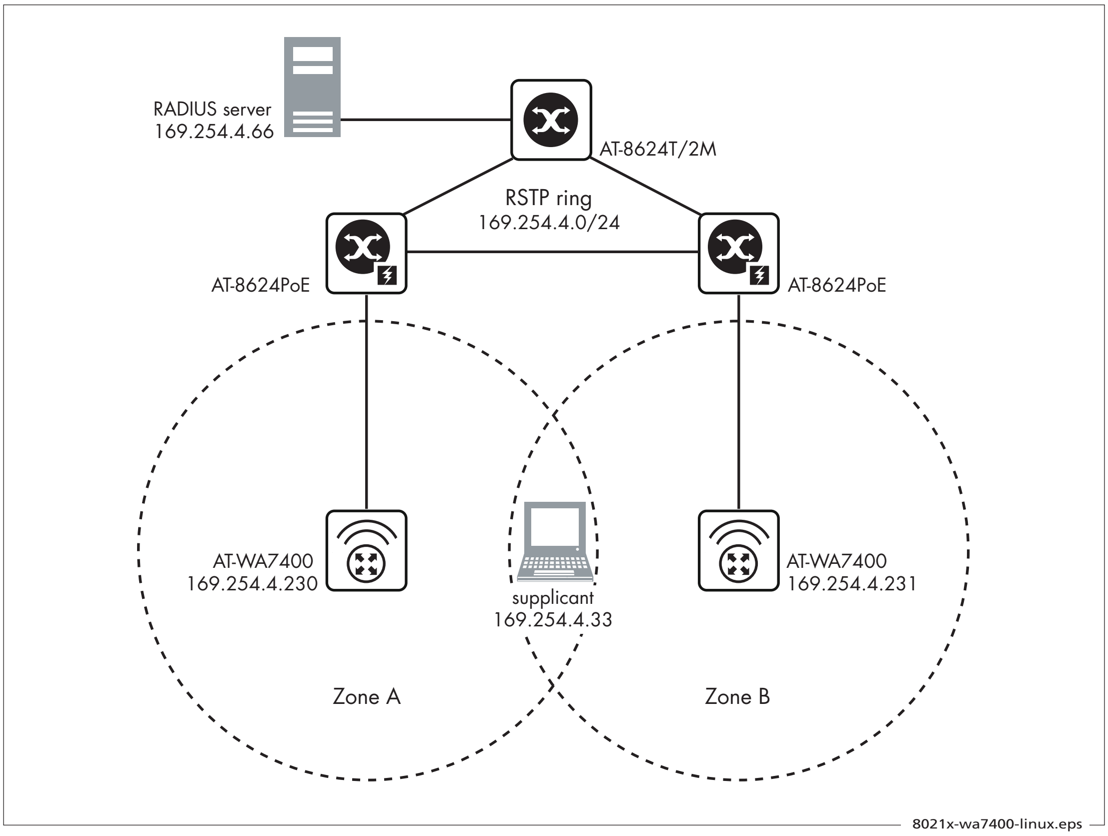
Controlled/uncontrolled port:

These are virtual concepts. The client attempts to gain access to the controlled port by authenticating through the uncontrolled port. However, since these terms are just concepts, they share the same medium (in this case thin air).

The Allied Telesis switches can be any of the following switches:

- Rapier i Series
- AT-8600 Series
- AT-8700XL Series
- AT-8800 Series
- AT-9800 Series
- SwitchBlade
- AT-8948
- AT-9900 Series
- x900 Series

## Network Diagram



# Configure your Switches

The three AT-8600 Series switches can run a very simple configuration if you are installing a layer two segment. Because the switches form a ring for a little redundancy, you must enable RSTP on all three switches.

Enter the following commands on all three switches:

```
enable stp=default

set stp=default mode=rapid
```

# Configure your RADIUS Server

Your RADIUS server needs to have freeRADIUS and OpenSSL installed. The server also requires a valid certificate to issue. If your certificate is self-signed, you need to copy the certificate to the supplicant. This section describes all these elements. Depending on your distribution and the settings you chose when you installed it, you may already have some or all of the elements.

## 1. Install freeRADIUS unless it is already installed

You can use the freeRADIUS version packaged with your distribution, or download it from www.freeradius.org. In this case, we compiled and installed it from source. To install the downloaded source package, we used the following commands:

```
[root@server freeradius-1.0.5]$ ./configure
[root@server freeradius-1.0.5]$ make
[root@server freeradius-1.0.5]# make install
```

## 2. Edit the freeRADIUS configuration files

freeRADIUS configuration files will probably be installed to /usr/local/etc/raddb. In that directory, you need to edit the following files:

- radiusd.conf
- clients.conf
- eap.conf
- users

The following sections show code that the configuration files must include. Your files may also have other configuration options.

## radiusd.conf

```
mschap {
          authtype = MS-CHAP
          use_mppe = yes
          require_encryption = yes
          require_strong = yes
}

authorize {
     preprocess
     mschap
     suffix
     eap
     files
}

authenticate {
     Auth-Type MS-CHAP {
          mschap
     }
     eap
}
```

## clients.conf

```
client 169.254.4.0/24 {
     secret = secret
     shortname = wireless
}
```

## eap.conf

```
eap {
     default_eap_type = peap
     tls {
          private_key_password = whatever
          private_key_file = ${raddbdir}/certs/cakey.pem
          certificate_file = ${raddbdir}/certs/cacert.crt
          CA_file = ${raddbdir}/certs/cacert.pem
          dh_file = ${raddbdir}/certs/dh
          random_file = ${raddbdir}/certs/random
     }
     peap {
          default_eap_type = mschapv2
     }
}
```

## users

```
manager   User-Password == friend
```

## 3. Install OpenSSL unless it is already installed

You can use the OpenSSL RPM that is available on Mandrake CDROMs, or you can download it from www.openssl.org and compile it for your platform. Install it in whichever manner you prefer.

## 4. Generate a self-signed public certificate unless you already have a valid one

This step describes how to generate a self-signed certificate and copy it into the RADIUS directory.

The following commands should generate as much as you need to get a valid certificate. You will be prompted to answer a series of questions to put in the certificate. Enter the following commands:

```
# mkdir morecerts

# cd morecerts

# mkdir private

# mkdir backup

# openssl req -config /usr/lib/ssl/openssl.cnf -new -x509 -keyout private/
cakey.pem -out cacert.pem -days 3650

# openssl x509 -in cacert.pem -out cacert.crt

# cp cacert.pem /usr/local/etc/raddb/certs/

# cp cacert.crt /usr/local/etc/raddb/certs/

# cp private/cakey.pem /usr/local/etc/raddb/certs/

# openssl dhparam-check-text -5 512 -out/usr/local/etc/raddb/certs/dh

# openssl rand -out/usr/local/etc/raddb/certs/random 100
```

## 5. Copy the public certificate to the client

Copy the RADIUS server's public certificate to the client so that the client's Xsupplicant will be able to recognise it. This example uses secure copy, which only works if your client and server currently have IP connectivity. Otherwise you need to copy the cacert.crt file by some other means (such as sneakernet).

```
# scp cacert.crt root@<client_ip>:/usr/local/etc/1x/certs/cacert.crt
```

If you are using a Windows PC as the supplicant, you can also import the cacert.crt file into the list of certificates. For more information on configuring a Windows supplicant, please consult the How To Note "How To Use 802.1x EAP-TLS or PEAP-MS-CHAP v2 with Microsoft Windows Server 2003 to Make a Secure Network".

# Install and Configure Xsupplicant on your Client

This section describes how to configure the client. We assume here that you have a wireless card already working in your client. In this example, our NIC was called ath0, so you should replace all instances of ath0 with your own NIC alias.

Once Xsupplicant is installed on your client, there are two or three files to configure, depending on the distribution. If you are using a Mandrake, Fedora or RedHat distribution you should only have to edit the following files:

- /etc/sysconfig/network-scripts/ifcfg-ath0
- /usr/local/etc/1x/xsupplicant.conf

If you are using another distribution, you may need to edit:

- /usr/local/etc/1x/startup.sh
- /usr/local/etc/1x/startup2.sh
- /usr/local/etc/1x/xsupplicant.conf

Do not try to use startup scripts and the ifcfg-ath0 script at the same time.

If you are going to automate the startup of Xsupplicant on boot, do so after the interface has been initialised (brought up) by the operating system. This basically means waiting until networking has started.

## 1. Install Xsupplicant

You can use the Xsupplicant RPM that is available on Mandrake CDROMs, or you can download it from sourceforge.net/projects/open1x/ and compile it for your platform.

## 2. Configure either the ifcfg or startup files

If you are using a Mandrake, Fedora or RedHat distribution, edit the ifcfg script as follows:

### /etc/sysconfig/network-scripts/ifcfg-ath0

```
DEVICE=ath0
BOOTPROTO=static
IPADDR=169.254.4.33
NETMASK=255.255.255.0
NETWORK=169.254.4.0
BROADCAST=169.254.4.255
ONBOOT=yes
METRIC=10
MII_NOT_SUPPORTED=yes
WIRELESS_MODE=Managed
WIRELESS_ESSID=allied
WIRELESS_FREQ=6
WIRELESS_RATE=auto
WIRELESS_ENC_KEY=000000000
WIRELESS_FRAG=2346
WIRELESS_IWCONFIG="key restricted"
```

If you are using a distribution that *does not use* ifcfg files, Xsupplicant lets you write startup files to the wireless NIC as shown in the following lines of code. Note that these files do not define encryption, because Xsupplicant controls that.

### /usr/local/etc/1x/startup.sh

```
#!/bin/sh
echo "Starting $0"
/sbin/ifconfig ath0 down
sleep 1
/sbin/iwconfig ath0 mode managed essid allied channel 6 rate auto enc 0000000000
#iwpriv ath0 authmode 2
/sbin/ifconfig ath0 allmulti up
echo "Finished $0"
exit 0
```

### /usr/local/etc/1x/startup2.sh

```
#!/bin/sh
echo "Starting $0"
iwconfig ath0
/sbin/ifconfig ath0 169.254.4.33 netmask 255.255.255.0 broadcast 255.255.255.0
echo "Finished $0"
iwconfig ath0
exit 0
```

## 3. Configure the xsupplicant.conf file

Edit the xsupplicant.conf file as shown in the following fragment. Take note of these points:

- This was a Mandrake example, so we did not need to use the startup scripts. Therefore we commented them out in our xsupplicant.conf file below (so they are not run).
- The eap-md5 section is irrelevant to the basic configuration, but it shows how simple it is to add a different form of EAP authentication to the supplicant configuration.

### /usr/local/etc/1x/xsupplicant.conf

```
### GLOBAL SECTION

network_list = all
default_netname = default
#startup_command = <BEGIN_COMMAND>/usr/local/etc/1x/startup.sh<END_COMMAND>
#first_auth_command = <BEGIN_COMMAND>/usr/local/etc/1x startup2.sh<END_COMMAND>
reauth_command = <BEGIN_COMMAND>echo "authenticated user %i"<END_COMMAND>
logfile = /var/log/xsupplicant.log
allow_interfaces = ath0, wlan0
deny_interfaces = eth0, eth1, sit0

# script continues on next page
```

```
###  NETWORK SECTION

allied
{
     allow_types = all
     identity = <BEGIN_ID>manager<END_ID>
     type = wireless
     wireless_control = yes
     eap-peap {
          root_cert = /usr/local/etc/1x/certs/cacert.crt
          chunk_size = 1398
          random_file = /dev/urandom
          session_resume = yes
          #allow_types = eap_md5
          allow_types = eap_mschapv2
          eap-mschapv2 {
               username = <BEGIN_UNAME>manager<END_UNAME>
               password = <BEGIN_PASS>friend<END_PASS>
          }
          eap-md5 {
               username = <BEGIN_UNAME>manager<END_UNAME>
               password = <BEGIN_PASS>friend<END_PASS>
          }
     }
}
```

# Configure the AT-WA7400

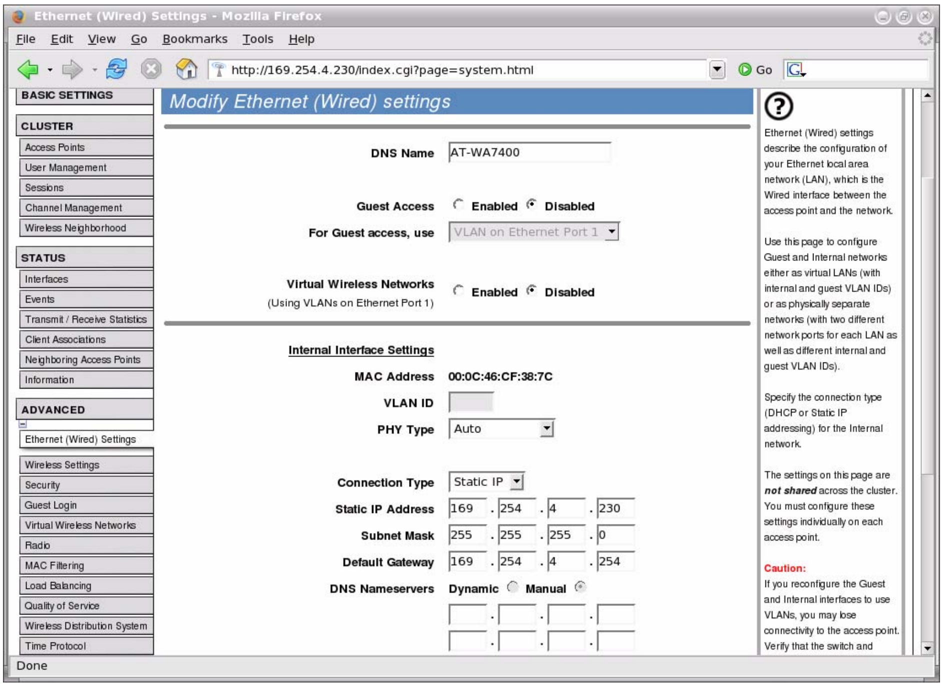
The AT-WA7400 will be powered by the AT-8624PoE switch, so there is no need to use the power supply unit included with the AT-WA7400.

## 1. Connect to the AT-WA7400

If you have not connected to the AT-WA7400 before, you can use your browser to connect to the IP address 192.168.1.230. You connect via the wired port to do this. The default username/password is manager/friend.

## 2. Set the IP address

In the “Wired Settings”, set your IP address.



Modify Ethernet (Wired) settings

DNS Name: AT-WA7400

Guest Access: ○ Enabled ● Disabled

For Guest access, use: VLAN on Ethernet Port 1

Virtual Wireless Networks (Using VLANs on Ethernet Port 1): ○ Enabled ● Disabled

Internal Interface Settings

MAC Address: 00:0C:46:CF:38:7C

VLAN ID:

PHY Type: Auto

Connection Type: Static IP

Static IP Address: 169 . 254 . 4 . 230

Subnet Mask: 255 . 255 . 255 . 0

Default Gateway: 169 . 254 . 4 . 254

DNS Nameservers: Dynamic ○ Manual ●

Ethernet (Wired) settings describe the configuration of your Ethernet local area network (LAN), which is the Wired interface between the access point and the network.

Use this page to configure Guest and Internal networks either as virtual LANs (with internal and guest VLAN IDs) or as physically separate networks (with two different network ports for each LAN as well as different internal and guest VLAN IDs).

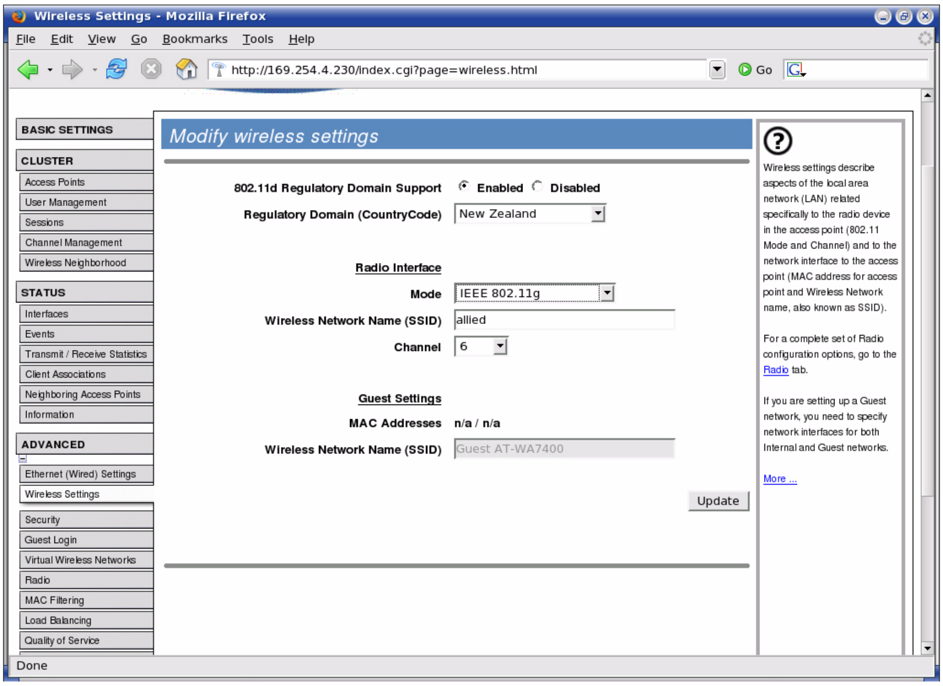
Specify the connection type (DHCP or Static IP addressing) for the Internal network.

The settings on this page are ***not shared*** across the cluster. You must configure these settings individually on each access point.

**Caution:**
If you reconfigure the Guest and Internal interfaces to use VLANs, you may lose connectivity to the access point. Verify that the switch and

## 3. Set the radio details

In the “Wireless Settings”, set your radio details appropriately for your country.



Modify wireless settings

802.11d Regulatory Domain Support: ● Enabled ○ Disabled

Regulatory Domain (CountryCode): New Zealand

Radio Interface

Mode: IEEE 802.11g

Wireless Network Name (SSID): allied

Channel: 6

Guest Settings

MAC Addresses: n/a / n/a

Wireless Network Name (SSID): Guest AT-WA7400

Wireless settings describe aspects of the local area network (LAN) related specifically to the radio device in the access point (802.11 Mode and Channel) and to the network interface to the access point (MAC address for access point and Wireless Network name, also known as SSID).

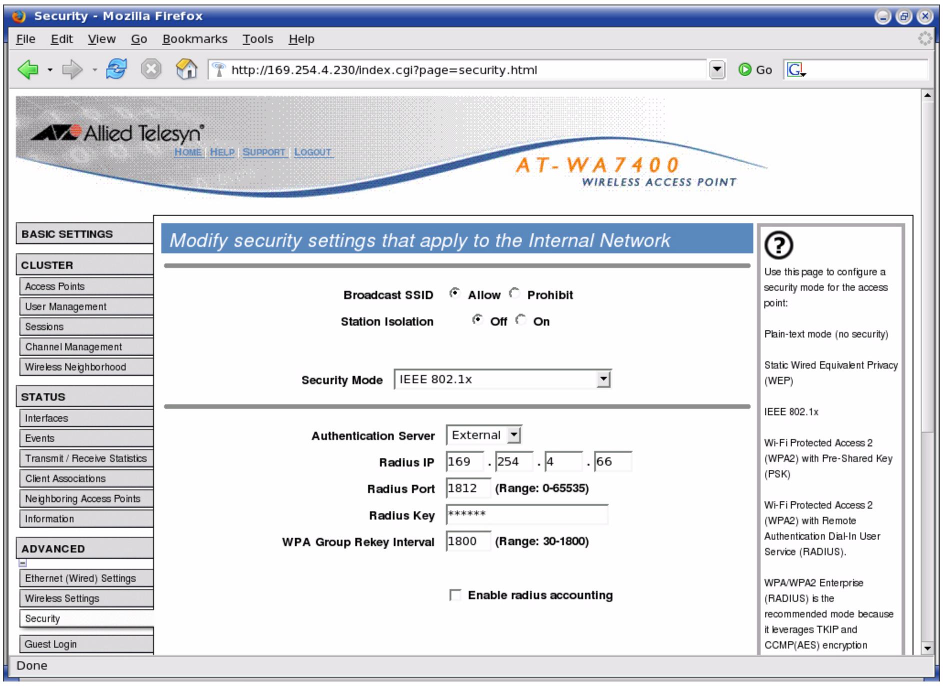
For a complete set of Radio configuration options, go to the Radio tab.

If you are setting up a Guest network, you need to specify network interfaces for both Internal and Guest networks.

More ...

## 4. Set the security settings

In the “Security” tab, set your authentication method. You might want to disable “Broadcast SSID”. Note that the RADIUS key should be the same as the one you entered in clients.conf on your RADIUS server.



Modify security settings that apply to the Internal Network

Broadcast SSID ● Allow ○ Prohibit

Station Isolation ● Off ○ On

Security Mode: IEEE 802.1x

Authentication Server: External

Radius IP: 169 . 254 . 4 . 66

Radius Port: 1812 (Range: 0-65535)

Radius Key: ******

WPA Group Rekey Interval: 1800 (Range: 30-1800)

□ Enable radius accounting

Use this page to configure a security mode for the access point:

Plain-text mode (no security)

Static Wired Equivalent Privacy (WEP)

IEEE 802.1x

Wi-Fi Protected Access 2 (WPA2) with Pre-Shared Key (PSK)

Wi-Fi Protected Access 2 (WPA2) with Remote Authentication Dial-In User Service (RADIUS).

WPA/WPA2 Enterprise (RADIUS) is the recommended mode because it leverages TKIP and CCMP(AES) encryption

# Start freeRADIUS and Xsupplicant

On your RADIUS server, start freeRADIUS with “-X” to show helpful debugging:

```
# radiusd -X
```

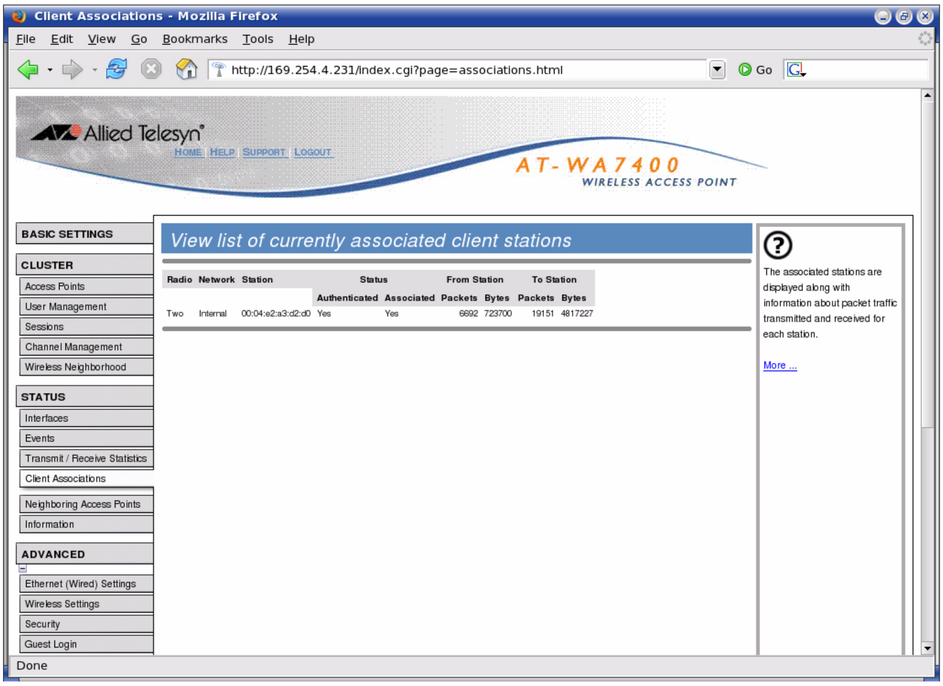
On your client, start Xsupplicant and specify your config file “-c” with debugging options “-d9 f”:

```
# xsupplicant -c /usr/local/etc/1x/xsupplicant.conf -d9 -f
```

When you have both freeRADIUS and Xsupplicant working to your satisfaction, start them without the extra debugging options and they will run quietly in the background.

# Confirm that Authentication Works

On the AT-WA7400, check the “Client Association” tab. When you have successfully used Xsupplicant to authenticate against the RADIUS server, you will see the client’s MAC address there. Beside that, under “Status”, should be “Yes” for “Authenticated” and “Associated”. Until you have two “Yes” entries you have not been authorised to use the controlled port.



View list of currently associated client stations

| Radio | Network | Station | Status | | From Station | | To Station | |
|---|---|---|---|---|---|---|---|---|
| | | | Authenticated | Associated | Packets | Bytes | Packets | Bytes |
| Two | Internal | 00:04:e2:a3:d2:d0 | Yes | Yes | 6692 | 723700 | 19151 | 4817227 |

The associated stations are displayed along with information about packet traffic transmitted and received for each station.

More ...

Notice that in the above picture we have changed the GUI view to a different Access Point (169.254.4.231). This is because the client was actually in an area best covered by the second Access Point. When this happens, the first Access Point (169.254.4.230) does not have a record of a client associated with another Access Point even though, by default, the two Access Points are in a cluster.

# Using a Wired Allied Telesis Switch Instead of the AT-WA7400

All Allied Telesis Rapier, AT8600, AT-8800, AT-9800, Switchblade, AT8900, AT-9900, and x900 Series switches support 802.1x port authentication. You can use any of these switches instead of the AT-WA7400.

For example, if you connected your laptop directly to port 1 on the AT-8624 (see "Network Diagram" on page 2), instead of using wireless, you would apply the following configuration to the AT-8624 switch.

```
# RADIUS configuration
add radius server=169.254.4.66 secret="secret" port=1812 accport=1813

# 802.1X configuration
enable portauth=8021x
enable portauth=8021x port=1 type=authenticator
```

In the above configuration:

- the IP address 169.254.4.66 is the IP address of the RADIUS server
- the RADIUS server is configured to listen on port 1812 for access-requests and 1813 for accounting-requests.

This configuration is the same whether you use Linux's Xsupplicant or Microsoft's supplicant client.

Naturally, the switch needs to have IP connectivity to the RADIUS server, however, the supplicant does not require any IP connectivity before the port authentication process begins. A DHCP server may assign the supplicant an IP after successful authentication.

European Headquarters | Via Motta 24 | 6830 Chiasso | Switzerland | T: +41 91 69769.00 | F: +41 91 69769.11
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
www.alliedtelesis.com