Common Application Guide (CAG)
Configuring RADIUS Authentication for Port Authentication & Dynamic VLAN Assignment

Configuring RADIUS Authentication for Port Authentication & Dynamic VLAN Assignment

Introduction

The use of AAA services (Authentication, Authorization, and Accounting) allows for several methods of controlling and recording access to AOS-based devices. The two methods of achieving this result involve either RADIUS or TACACS+ servers. This guide will specifically cover the use of controlling switchport access, and by extension network access, using RADIUS authentication.

Port Authentication utilizes the IEEE 802.1X standard for communication with the client device connected to it. Please refer to the IEEE standard for further details.

The AOS implementation of Port Authentication also allows for Dynamic VLAN Assignment of 802.1X clients, which is discussed under the RADIUS server section.

Command Line Configuration

Enabling the Service & Defining a Server

AOS-based devices require that the AAA process be enabled and that at least one RADIUS server is defined. The AOS device and the RADIUS server being accessed must agree on the Pre-Shared Key for the process to operate successfully; the key is used to encrypt the password portion of the RADIUS authentication request message. The configuration is as such:

```
aaa on
!
radius-server host <RADIUS Server IP> key <Pre-Shared Key>
```

NOTE: *Working with multiple AAA servers is covered under a different document.*

Define Authentication Method

The next step is to define the desired Port Authentication method. Currently only authentication to a RADIUS server or the Local-User List is supported.

The primary method will only fail over to the next defined method if the previous method is unavailable. For instance, if the RADIUS server is specified first and then the Local-User List, the Local-User List will only be consulted if the RADIUS server does not respond to RADIUS request messages. If the RADIUS server rejects the user credentials, the user will be denied access and the Local-User List will not be consulted.

If there are no valid methods left in a given list, then the user will be denied access. An example is if only the RADIUS server is defined within the list, and the RADIUS server cannot be contacted for any reason; user login will not be possible because there are no longer any valid methods within the list. For this reason, it is always recommended to end every list with the Local-User List so the device can still be accessed if communication with the RADIUS server is interrupted because the Local-User List will never be unavailable.

The following example will apply the Port Authentication method as RADIUS authentication first and the Local-User List second in the event RADIUS communication fails.

```
aaa authentication port-auth default group radius local
```

NOTE: *In some cases, it is preferable to only define the RADIUS server and not the Local-User List to prevent users from accessing the network and its resources when the RADIUS server cannot be contacted. If this method is preferred, it is suggested that multiple RADIUS servers be defined in the event that the primary RADIUS server has a problem that cannot be readily corrected. Working with multiple servers is covered under a different document.*

Defining the Port-Control Method

Each switchport on the unit has the ability to enter into an *Authorized* or *Unauthorized* state under a Port Authentication configuration. The *Authorized* state allows the port to transmit and receive traffic normally. The *Unauthorized* state only allows for 802.1X authentication frames to enter the port; all other traffic is blocked. By default, an *Unauthorized* port will still transmit network traffic which is typically limited to broadcast and multicast traffic types. This can be modified so that the port will not transmit traffic either through a command-line only option.

Each switchport has three configuration options that will influence the state of the port, and are defined as follows:

- Force-Authorized
  - This will force the port into an *Authorized* state when in an *UP* condition. It is primarily used for connecting to other switches or devices that do not support 802.1X.
- Force-Unauthorized
  - This will force the port into an *Unauthorized* state when in an *UP* condition. It is used when the port is never intended to be utilized but the administrator wants the port to still be active for any reason.
  - By default, this condition will still transmit broadcast and multicast frames. This behavior can be modified on a per-port basis with an optional command.
- Auto
  - This will cause the port to enter an *Unauthorized* state immediately after the port enters an *UP* condition, and allows for a successful 802.1X authentication process to change its state to an *Authorized* state.
  - This mode is required to use 802.1X Port Authentication.

The following provides an example of the different methods of configuring a switchport for the previously defined modes:

```
interface switchport 0/X
  port-auth port-control force-authorized
!
interface switchport 0/X
  port-auth port-control force-unauthorized
!
interface switchport 0/X
  port-auth port-control auto
```

To block traffic in both directions while in an *Unauthorized* state, the configuration is as shown:

```
interface switchport 0/X
  port-auth control-direction both
```

NOTE: *This configuration option is not available on all units.*

Define Port-Authentication Method

Each switchport has the ability to use one of two Port Authentication methods; the method chosen for use on the network primarily is dependant upon the capabilities of the 802.1X client in use on the network. The two Port Authentication methods are:

- Port-Based
  - This authentication method is compatible with every 802.1X client and will authorize the port to permit traffic flow after a single 802.1X client authenticates.
  - The switchport will immediately send out an *EAP-Identity-Request* message to start the authentication process when the port enters an *UP* condition, as well as on a periodic basis while the port is in a *UP* condition but *Unauthorized* state. Once a successful authentication has taken place, the port will transition to an *Authorized* state and the port will by default discontinue sending EAP messages while in this state.
  - Regardless of the number of devices attached to the port, only one must authenticate for the port to enter an *Authorized* state. By default, only the device that authenticated will be allowed to communicate.
  - An optional parameter can be used to allow multiple hosts to communicate after the first has authenticated.
    - This creates a potential problem if a malicious entity was to use an unmanaged switch or hub between the switchport and the user. Once the first user is authenticated, any device that the malicious entity attaches to the unmanaged switch or hub would now also be allowed to access the network.
- MAC-Based
  - This authentication method requires that the 802.1X client in use supports the use of the *EAP-Start* message, which is an optional function for the client to support according to the IEEE 802.1X standard.
  - The switchport will still enter the *Unauthorized* state when the port enters an *UP* condition, but will wait until the port receives an *EAP-Start* message from the client before sending the *EAP-Identity-Request* message to begin the authentication process. The *EAP-Identity-Request* message will be sent only to the client that sent the *EAP-Start* message instead of the broadcast EAP address. Once a successful authentication has taken place, the port will be transitioned to an *Authorized* state for that device's MAC address only; all other MAC addresses will still be in an *Unauthorized* state, and unable to communicate through that port.
  - This mode also allows for a Distribution-level switch to perform the Port Authentication for devices connected to unmanaged Access-layer switches. This will not protect devices connected to unmanaged Access-layer switches from accessing each other, but would protect network resources, which would presumably be connected at the Core or Distribution-level switches.

The following provides an example of the different methods of configuring a switchport for the previously defined methods:

```
interface switchport 0/X
  port-auth auth-mode port-based
!
interface switchport 0/X
  port-auth auth-mode mac-based
```

To allow multiple devices to communicate across a port-based authenticated switchport after the first device has authenticated, the command is:

```
interface switchport 0/X
  port-auth auth-mode port-based
  port-auth multiple-hosts
```

Securing A Switchport When the Client Does Not Support 802.1X

Not every network device will support 802.1X. These devices are usually printers, faxes, older servers, etc. Thankfully, these devices do not tend to change the port they are connected to without a network administrator aware of the upcoming modification.

Therefore, the solution for these cases is to place the port into the *Force-Authorized* mode, and use another feature referred to as Port Security to only allow the port to accept traffic from the MAC address of the device that will be connecting to it. This does not stop malicious entities that are able to spoof MAC addresses, however.

The following configuration will place the port into a *Force-Authorized*, *Port-Based* mode with a single MAC address statically configured within the Port Security configuration and the maximum addresses allowed by Port Security set to one (1):

```
interface switchport 0/X
  port-auth auth-mode port-based
  port-auth port-control force-authorized
  switchport port-security
  switchport port-security mac-address 01:01:01:01:01:01
  switchport port-security maximum 1
  switchport port-security violation restrict
```

Web Interface Configuration

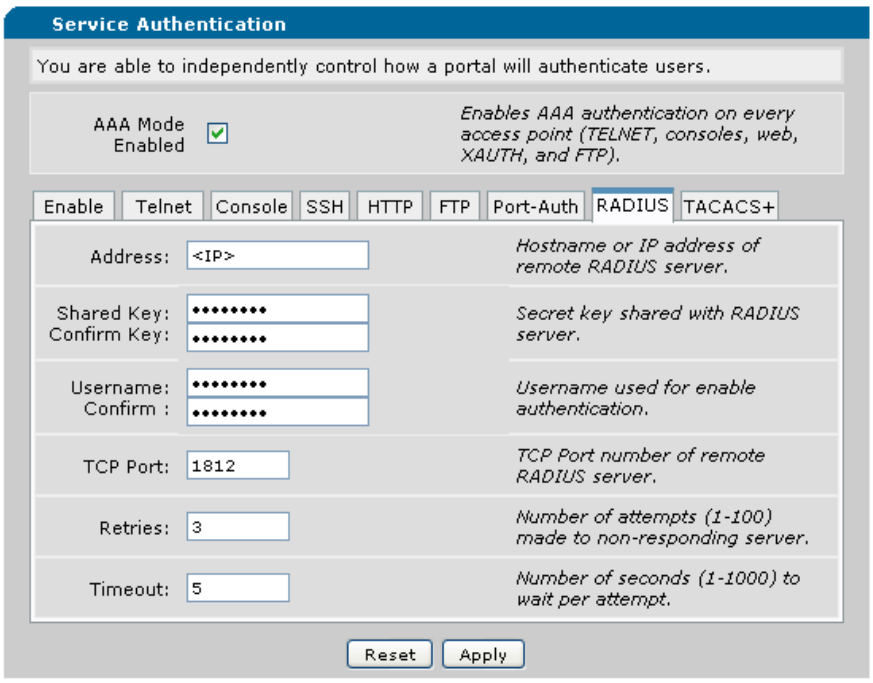
This section will define the methods for configuring this functionality through the GUI. The technology definitions and explanations will not be repeated; please refer to the relevant command line configuration section for more information.

The AAA configuration is accomplished from the "System → Passwords" page, in the bottom section entitled "Service Authentication".

NOTE: *The functionality allowed by the GUI is limited in that it allows for only one method to be defined, and it uses pre-defined names for its authentication lists. This means that it is not possible to have the Local-User List as a fallback position should the RADIUS server be unavailable. For this reason, CLI configuration is recommended.*

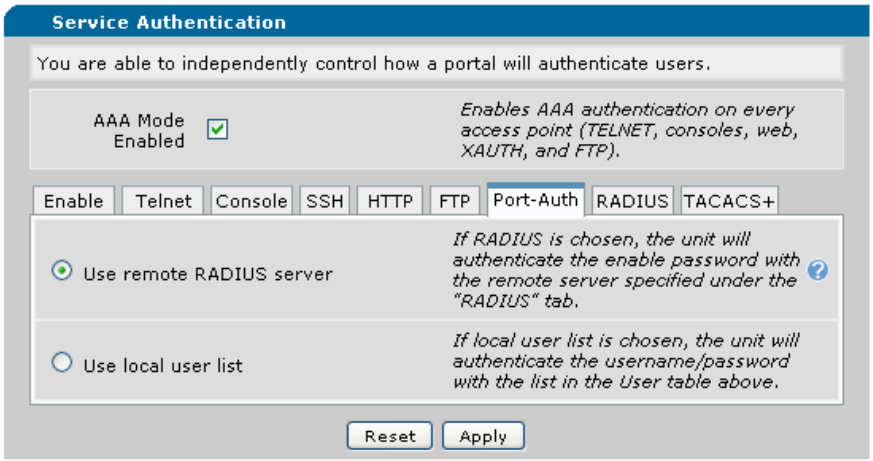
Enabling the Service & Defining a Server

The "AAA Mode Enabled" checkbox must be checked and the RADIUS server defined with a Pre-Shared Key (the Enable Username is not required for Port-Authentication), as shown:



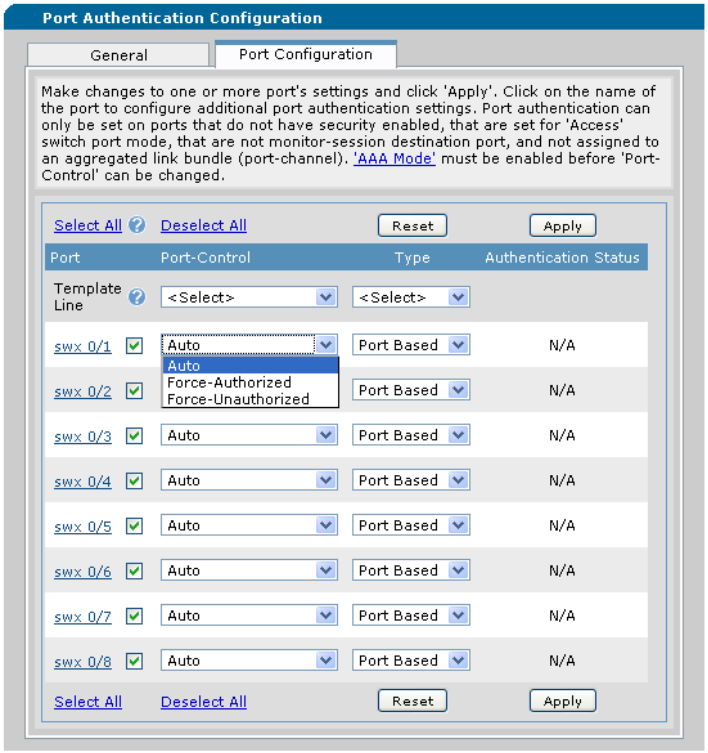
Define & Apply Authentication Method

Select either RADIUS or Local-User List Authentication method, as shown:



Define & Apply the Port-Authentication and Port-Control Methods

To set a switchport for *Force-Authorized*, *Force-Unauthorized*, or *Auto* Port-Control method and *Port-Based* or *MAC-Based* Port-Authentication method, use the "Port Configuration" tab under the "Data → Switch → Port Authentication" page, as shown:



Configuring the RADIUS Server

This section will define the relevant portions of the RADIUS message that the server should be looking for, and use the IAS function of a Windows 2003 Server as an example.

RADIUS Attribute Value Pairs (AVPs)

The RADIUS authentication request will contain several Attribute Value Pairs (AVPs) that facilitate the required functions of authentication. They allow the authentication method defined within the RADIUS server to be specific enough to match only on traffic from this client (or class of clients). If the RADIUS server supports logging at high level of verbosity, they contain information about where the client is originating from for logging purposes. The AVPs that the device will send are:

- Username
  - Contains the unencrypted username attempting to authenticate.
- EAP-Message
  - An encapsulated version of the *EAP-Identity-Response* message from the client.
- Message-Authenticator
  - Used in password authentication.
- NAS-IP-Address
  - Indicates the primary IP of the interface that the RADIUS request packet is sourced from.

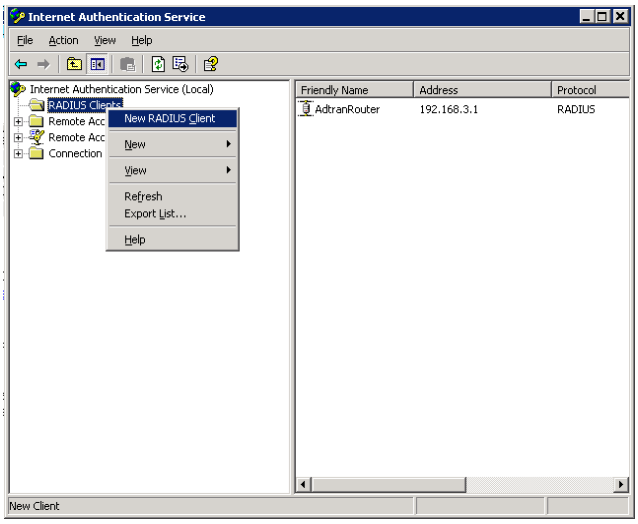
Configuring the RADIUS Server

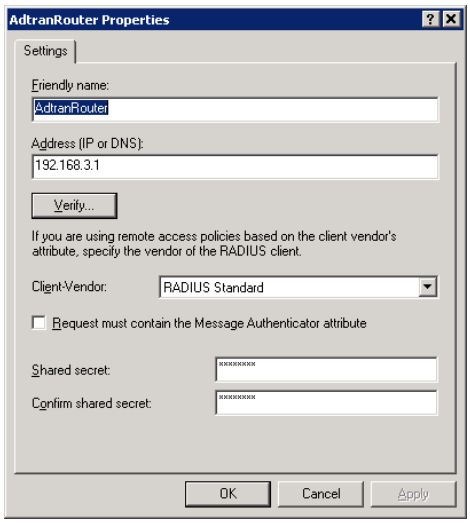
The RADIUS server, using Windows Server 2003's IAS as an example, can specify multiple dependencies that must match before a particular policy is allowed to be used to authenticate a request. It is recommended that these be used to protect the RADIUS server from client authentication requests from unauthorized sources, or to ensure that each RADIUS client has the correct policy applied to it if there are multiple devices sending authentication requests. Examples of such client groups are Device Administrators, Wireless Clients, Port-Authentication, and VPN Clients.

NOTE: *Windows IAS functionality and configuration style may change. The procedures described within this document are only used as an example. ADTRAN is not responsible for configuring the RADIUS server, and will not support the RADIUS server should it be found to be the source of any errors in the authentication process. This article will only cover the Netvanta-specific configuration options within IAS; there may be further configuration on the server required to utilize IAS in this manner.*

For further information, please refer to the Microsoft KB article on IAS, which can be accessed here: http://technet.microsoft.com/en-us/library/cc738432(WS.10).aspx

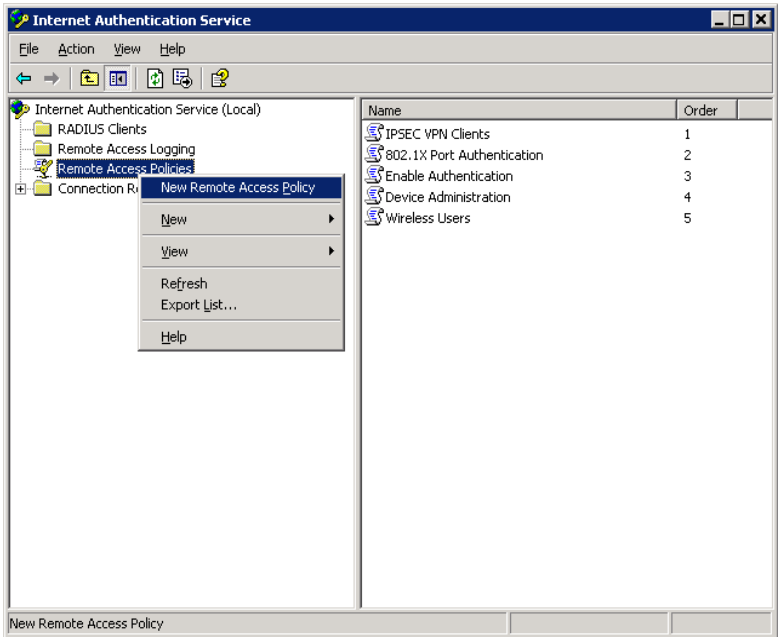
The first step to be completed in most RADIUS servers is to define the RADIUS client device, which involves specifying the Pre-Shared Key and the IP address it will be coming from. This will allow the RADIUS server to receive messages from this RADIUS client. In IAS, it is done in the following manner:

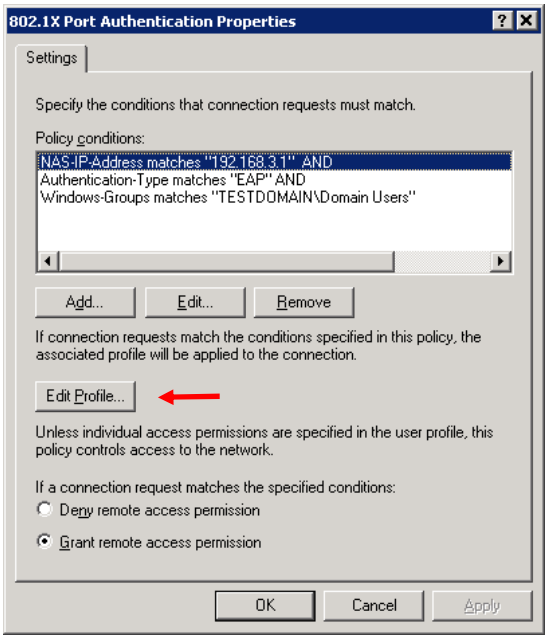


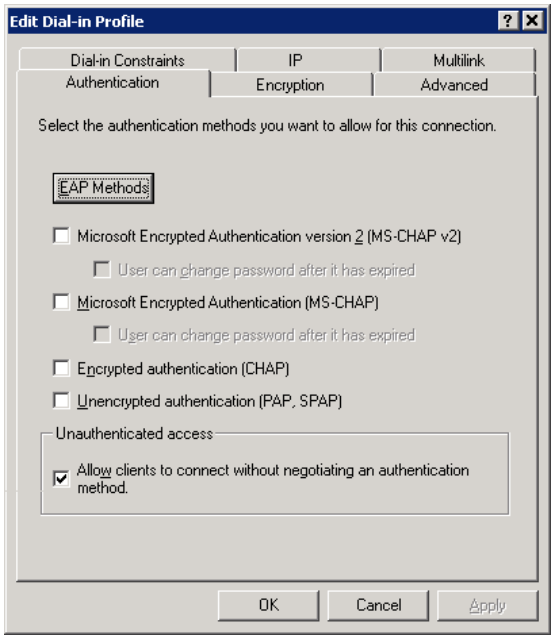


The next step is to create the policy that will process the request, ensure that it matches the required AVPs for this connection type, and permit or deny the request.

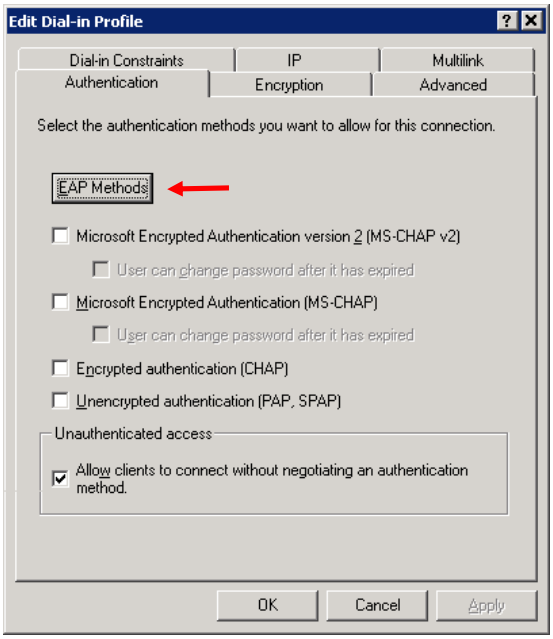
The RADIUS server will need to define the AVPs that will remain static within all authentication attempts from this RADIUS client & change the authentication process to allow PAP authentication without negotiation. In IAS, it is done in the following manner:

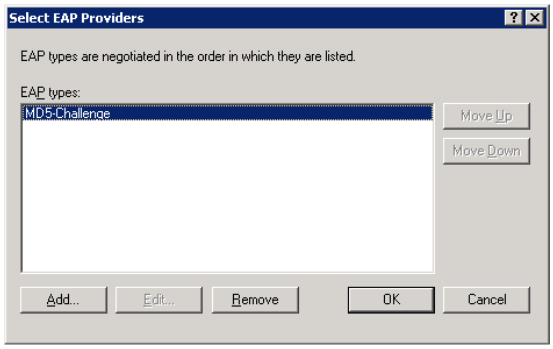






In the case of Port Authentication, an EAP method is used to authenticate. AOS currently only supports the use of the MD5-Challenge EAP method for Port Authentication. In IAS, it is done in the following manner:





Dynamic VLAN Assignment Configuration

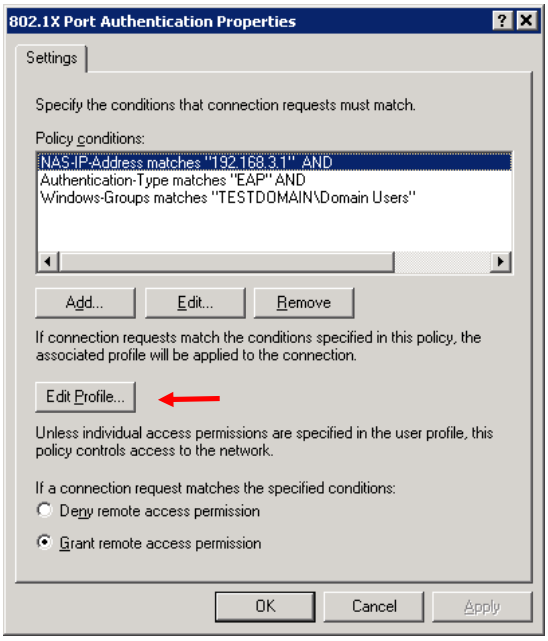
This functionality is on by default in AOS, and cannot be disabled.

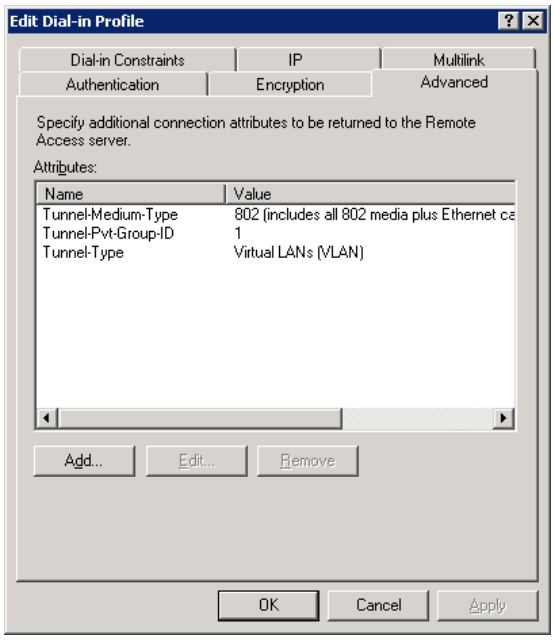
This functionality will not make any permanent changes to the switchport in question, and therefore will not be reflected in the configuration of the unit. It will override the currently configured switchport access VLAN. The temporary configuration will be removed if the user sends an *EAP-Logoff* message (also referred to as Disconnect), or if the port changes to a *DOWN* condition due to administrative shutdown or the cable being unplugged.

The RADIUS server has the option of sending its own AVPs in the final message with the affirmative response. The switch is programmed to recognize when three specific AVPs are in that message, and will dynamically configure the port that the authentication attempt was referencing to access the VLAN ID specified in the message. The AVPs that must be sent by the RADIUS server are:

- Tunnel-Type (64)
  - Must be set to VLAN (13).
- Tunnel-Medium-Type (65)
  - Must be set to IEEE-802 (6).
- Tunnel-Private-Group-Id (81)
  - Set to the VLAN ID that the user should access.

If multiple user groups must access different VLANs, then multiple access policies within IAS must be configured. The policy must be modified to reflect the following:





Dynamic VLAN Assignment Verification

The VLAN re-assignment will display in the debug with the following messages using the '*debug port-auth general'* debug command:

```
2010.02.12 10:24:04 PORT_AUTH.GENERAL Int swx 0/1 added to VLAN 1
     by Port-Auth
2010.02.12 10:24:04 PORT_AUTH.GENERAL Valid VlanId 1 rcvd from
     Server for sess 1
```

The following show commands will display the current port-auth VLAN state:

```
3448#show run interface sw 0/1

Building configuration...
!
!
interface switchport 0/1
  spanning-tree edgeport
  no shutdown
  switchport access vlan 2
  port-auth port-control auto
!
End
```

```
3448#show interface sw 0/1
      switchport

Name: swx 0/1
Switchport: enabled
Administrative Mode: access
Negotiation of Trunking: access
Access Mode VLAN (by port-
      auth): 1
Trunking Native Mode VLAN: 1
Trunking VLAN Enabled: 1-4094
Trunking VLAN GVRP Fixed: none
Port Expiration: disabled
Port Security: disabled
Protected: false
```

```
3448#sh vlan id 1

VLAN  Name                    Status    Type     Media    MTU
------------------------------------------------------------------
1    Default                  active    static   enet     1500

      Membership  Ports
      ------------------------------------------------------------
      Configured  swx 0/2, swx 0/3, swx 0/4, swx 0/5,
                  swx 0/6, swx 0/7, swx 0/8
      Port-Auth   swx 0/1
```

Troubleshooting

This section will describe the relevant debug procedures involved when determining any issues with the AAA, RADIUS, or Port-Authentication configuration. The commands that will be used are:

- debug aaa
- debug radius
- debug port-auth

A successful authentication attempt would be similar to the following:

```
2010.02.12 10:23:52 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
     CONNECTING state
2010.02.12 10:23:52 PORT_AUTH.PACKET TX Sent EAP Req/Id for sess
     1 on int swx 0/1
2010.02.12 10:23:52 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:23:52 PORT_AUTH.CTRLDIRSM Int swx 0/1 sess 1 in
     IN_OR_BOTH state
2010.02.12 10:23:52 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state

AAA: New Session on portal 'PORTAUTH'.
AAA: No list mapped to 'PORTAUTH'. Using 'default'.
AAA: Closing Session on portal 'PORTAUTH'.

2010.02.12 10:24:04 PORT_AUTH.PACKET RX Rcvd EAP Resp/Id for sess
     1 on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
     AUTHENTICATING state
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
     RESPONSE state
2010.02.12 10:24:04 PORT_AUTH.GENERAL Init auth with server for
     sess 1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Sent EAP Resp/Id to
     AuthServer for sess 1 on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state

RADIUS AUTHENTICATION: Sending packet to <Server IP> (1812).
RADIUS AUTHENTICATION: Waiting on response from server
RADIUS AUTHENTICATION: Receiving from RADIUS socket
RADIUS AUTHENTICATION: Response received from server (<Server
     IP>)
RADIUS AUTHENTICATION: Received response from <Server IP>.
```

```
AAA: RADIUS authentication requested a challenge.

2010.02.12 10:24:04 PORT_AUTH.GENERAL Rcvd response from server
     for sess 1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
     REQUEST state
2010.02.12 10:24:04 PORT_AUTH.PACKET TX Sent EAP Req for sess 1
     on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:24:04 PORT_AUTH.PACKET RX Rcvd EAP Resp for sess 1
     on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
     RESPONSE state
2010.02.12 10:24:04 PORT_AUTH.GENERAL Init auth with server for
     sess 1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Sent EAP Resp/Id to
     AuthServer for sess 1 on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state

RADIUS AUTHENTICATION: Sending packet to <Server IP> (1812).
RADIUS AUTHENTICATION: Waiting on response from server
RADIUS AUTHENTICATION: Receiving from RADIUS socket
RADIUS AUTHENTICATION: Response received from server (<Server
     IP>)
RADIUS AUTHENTICATION: Received response from <Server IP>.
AAA: RADIUS authentication passed.

2010.02.12 10:24:04 PORT_AUTH.GENERAL Rcvd response from server
     for sess 1
2010.02.12 10:24:04 PORT_AUTH.GENERAL Int swx 0/1 added to VLAN 1
     by Port-Auth
2010.02.12 10:24:04 PORT_AUTH.GENERAL Valid VlanId 1 rcvd from
     Server for sess 1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
     SUCCESS state
2010.02.12 10:24:04 PORT_AUTH.PACKET TX Sent EAP Success for sess
     1 on int swx 0/1
2010.02.12 10:24:04 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in IDLE
     state
2010.02.12 10:24:04 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
     INITIALIZE state
2010.02.12 10:24:04 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
     AUTHENTICATED state; sess is authorized
```

An unsuccessful authentication attempt would be similar to the following:

```
2010.02.12 10:46:52 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
      DISCONNECTED state; sess is Unauthorized
2010.02.12 10:46:52 PORT_AUTH.PACKET TX Sent EAP Failure for sess
      1 on int swx 0/1
2010.02.12 10:46:52 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
      CONNECTING state
2010.02.12 10:46:52 PORT_AUTH.PACKET TX Sent EAP Req/Id for sess
      1 on int swx 0/1
2010.02.12 10:46:52 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:52 PORT_AUTH.CTRLDIRSM Int swx 0/1 sess 1 in
      IN_OR_BOTH state
2010.02.12 10:46:52 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.PACKET RX Rcvd EAP Resp/Id for sess
      1 on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in
      AUTHENTICATING state
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
      RESPONSE state
2010.02.12 10:46:58 PORT_AUTH.GENERAL Init auth with server for
      sess 1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Sent EAP Resp/Id to
      AuthServer for sess 1 on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state

RADIUS AUTHENTICATION: Sending packet to <Server IP> (1812).
RADIUS AUTHENTICATION: Waiting on response from server
RADIUS AUTHENTICATION: Receiving from RADIUS socket
RADIUS AUTHENTICATION: Response received from server (<Server
      IP>)
RADIUS AUTHENTICATION: Received response from <Server IP>.
AAA: RADIUS authentication requested a challenge.

2010.02.12 10:46:58 PORT_AUTH.GENERAL Rcvd response from server
      for sess 1
2010.02.12 10:46:58 PORT_AUTH.GENERAL Int swx 0/1 added to VLAN 1
      by Port-Auth
2010.02.12 10:46:58 PORT_AUTH.GENERAL Valid VlanId 1 rcvd from
      Server for sess 1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
      REQUEST state
2010.02.12 10:46:58 PORT_AUTH.PACKET TX Sent EAP Req for sess 1
      on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
```

```
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.PACKET RX Rcvd EAP Resp for sess 1
      on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in
      RESPONSE state
2010.02.12 10:46:58 PORT_AUTH.GENERAL Init auth with server for
      sess 1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Sent EAP Resp/Id to
      AuthServer for sess 1 on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state

RADIUS AUTHENTICATION: Sending packet to <Server IP> (1812).
RADIUS AUTHENTICATION: Waiting on response from server
RADIUS AUTHENTICATION: Receiving from RADIUS socket
RADIUS AUTHENTICATION: Response received from server (<Server
      IP>)
RADIUS AUTHENTICATION: Received response from <Server IP>.
AAA: RADIUS authentication failed.

2010.02.12 10:46:58 PORT_AUTH.GENERAL Rcvd response from server
      for sess 1
2010.02.12 10:46:58 PORT_AUTH.GENERAL Int swx 0/1 added to VLAN 1
      by Port-Auth
2010.02.12 10:46:58 PORT_AUTH.GENERAL Valid VlanId 1 rcvd from
      Server for sess 1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in FAIL
      state
2010.02.12 10:46:58 PORT_AUTH.PACKET TX Sent EAP Failure for sess
      1 on int swx 0/1
2010.02.12 10:46:58 PORT_AUTH.BKENDSM Int swx 0/1 sess 1 in IDLE
      state
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
2010.02.12 10:46:58 PORT_AUTH.AUTHSM Int swx 0/1 sess 1 in HELD
      state; sess is Unauthorized
2010.02.12 10:46:58 PORT_AUTH.REAUTHSM Int swx 0/1 sess 1 in
      INITIALIZE state
```

DISCLAIMER

ADTRAN provides the foregoing application description solely for the reader's consideration and study, and without any representation or suggestion that the foregoing application is or may be free from claims of third parties for infringement of intellectual property rights, including but not limited to, direct and contributory infringement as well as for active inducement to infringe. In addition, the reader's attention is drawn to the following disclaimer with regard to the reader's use of the foregoing material in products and/or systems. That is:

ADTRAN SPECIFICALLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL ADTRAN BE LIABLE FOR ANY LOSS OR DAMAGE, AND FOR PERSONAL INJURY, INCLUDING BUT NOT LIMITED TO, COMPENSATORY, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.