Common Application Guide (CAG)
Configuring TACACS+ Authentication for Device Administration

Configuring TACACS+ Authentication for Device Administration

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 login to the administrative interface of an AOS-based device using TACACS+ authentication.

Command Line Configuration

Enabling the Service & Defining a Server

AOS-based devices require that the AAA process be enabled and that at least one TACACS+ server is defined. The AOS device and the TACACS+ server being accessed must agree on the Pre-Shared Key for the process to operate successfully; the key is used to encrypt the entire communication between the client and the server. The configuration is as such:

```
aaa on
!
tacacs-server host <TACACS+ Server IP> key <Pre-Shared Key>
```

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

Define Authentication Methods

The next step is to define the desired authentication methods. This can be done to all interfaces using one command that defines the default list, or by specifying specific lists that can be applied to individual administrative services.

The lists define the order of operations for authentication. The primary method will only fail over to the next defined method if the previous method is unavailable. For instance, if the TACACS+ server is specified first and then the Local-User List, the Local-User List will only be consulted if the TACACS+ server does not respond to TACACS+ request messages. If the TACACS+ 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 TACACS+ server is defined within the list, and the TACACS+ 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 TACACS+ server is interrupted because the Local-User List will never be unavailable.

The following example will apply the "default" authentication method as TACACS+ authentication first and the Local-User List second in the event TACACS+ communication fails. This will affect all administrative services that do not have a defined login method:

```
aaa authentication login default group tacacs+ local
```

The following will show the same authentication method used as a named list:

```
aaa authentication login LoginUseTacacsLocal group tacacs+ local
```

Apply Authentication Methods to Services

The next step is to apply the defined named authentication method list to the specific interfaces, if the "default" list method is not used.

NOTE: *The HTTP authentication method will apply to both HTTP and HTTPS connections to the router.*

```
line con 0
  login authentication LoginUseTacacsLocal
!
line telnet 0 4
  login authentication LoginUseTacacsLocal
!
line ssh 0 4
```

```
  login authentication LoginUseTacacsLocal
!
ip http authentication LoginUseTacacsLocal
!
ftp authentication LoginUseTacacsLocal
```

Defining the Enable Password Method

The enable password is uniquely associated with the command-line interface, and is irrelevant to the GUI. If command-line access is not being made available for user access or if TACACS+ enable authentication is not required, then this section is optional.

Regardless of whether RADIUS or TACAS+ servers are used, they are inherently username & password systems. This poses a specific challenge for enable authentication, where the user is not asked for a username when entering enable mode. Both RADIUS and TACACS+ therefore define a username to be sent with the enable password authentication requests; the TACACS+ standard requires that this username be set to "$enab15$" and therefore it cannot be changed.

NOTE: *Because the TACACS+ enable username cannot have a custom value, it is recommended that enable authentication is not used with TACACS+. Instead, the "Auto-Exec" feature is recommended to place the users into enable mode, which is covered under the "Authorization & Accounting" document.*

The authentication methodology is the same as before, and it is recommended in this case that you define the final method as the locally-configured enable password, as such:

```
aaa authentication enable default group tacacs+ enable
```

Web Interface Configuration

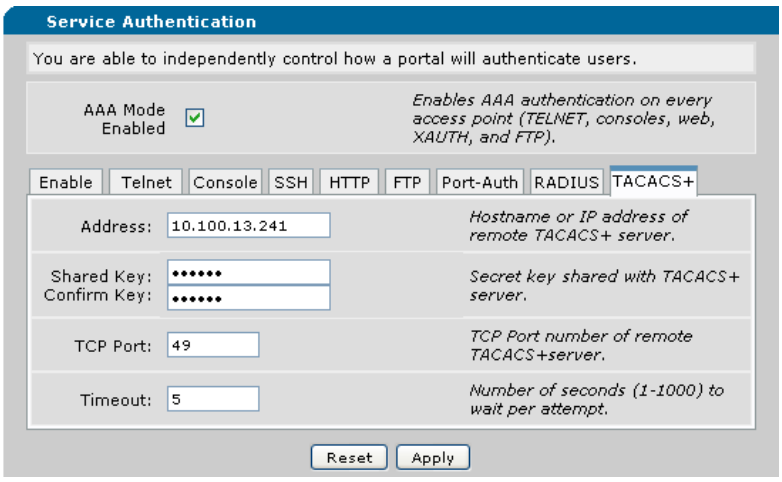
This section will define the methods for configuring this functionality through the GUI. The technology definitions and explanations will not be repeated; please refer to the relevant command line configuration section for more information.

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 TACACS+ server be unavailable. For this reason, CLI configuration is recommended.*

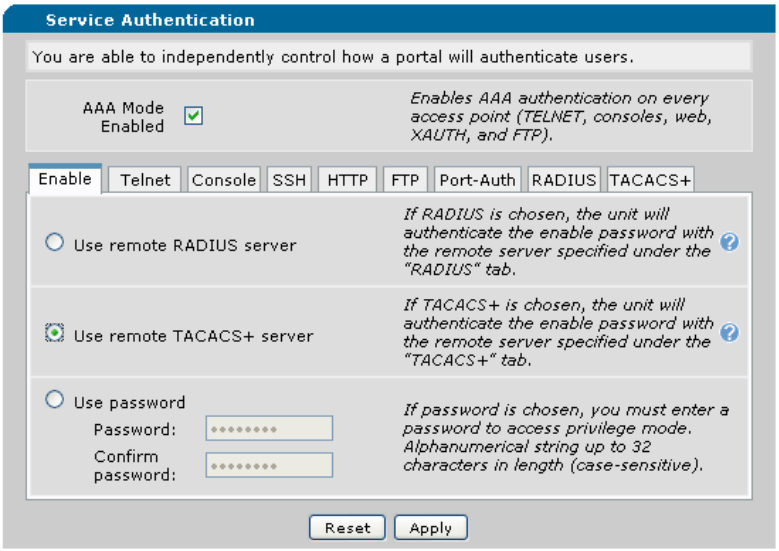
Enabling the Service, Defining a Server, & the Enable Username

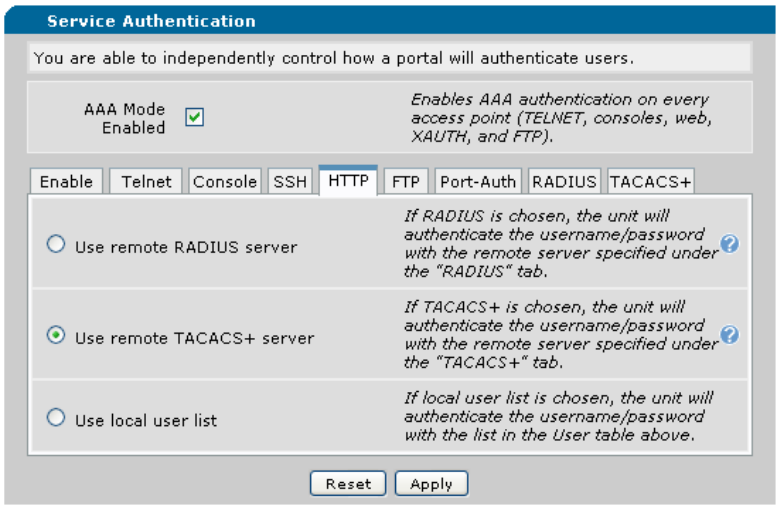
The “AAA Mode Enabled” checkbox must be checked and the TACACS+ server defined with a Pre-Shared Key, as shown:



Define & Apply Authentication Methods

The GUI does not support the use of a “default” list. Each service will have one of the predefined lists that the GUI creates applied to it, based upon the bullet-point chosen under the Service’s tab. Several examples are shown below:





Configuring the TACACS+ Server

The TACACS+ standard does not leave any room for vendor-specific options; AOS clients will formulate the message in the same manner as every other TACACS+ client. Therefore, please use the normal TACACS+ server setup specified by the TACACS+ server vendor.

Troubleshooting

This section will describe the relevant debug procedures involved when determining any issues with the AAA or TACACS+ configurations. The commands that will be used are:

- debug aaa
- debug tacacs+

A successful authentication attempt would be similar to the following:

```
AAA: New Session on portal 'SSH 0 (<Client IP>:<Port>)'.
AAA: Session using AUTHENTICATION list 'LoginUseTacacsLocal'.
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Authentication START pkt
TAC+ TX: Authentication START
  TAC+ TX: version=0xc0, type=Authentication, seq_no=1, flags=00
  TAC+ TX: action=Login
  TAC+ TX: level=1
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
```

```
  TAC+ TX: remote address=0.0.0.0
TAC+ EVENT: Received Authentication REPLY pkt
TAC+ RX: Authentication REPLY
  TAC+ RX: version=0xc0, type=Authentication, seq_no=2, flags=00
  TAC+ RX: status=GETPASS
  TAC+ RX: flags=0x01
  TAC+ RX: server msg=Password:
TAC+ EVENT: Sending Authentication CONTINUE pkt
TAC+ TX: Authentication CONTINUE
  TAC+ TX: version=0xc0, type=Authentication, seq_no=3, flags=00
  TAC+ TX: user message=********
  TAC+ TX: flags=0x00
TAC+ EVENT: Received Authentication REPLY pkt
TAC+ RX: Authentication REPLY
  TAC+ RX: version=0xc0, type=Authentication, seq_no=4, flags=00
  TAC+ RX: status=PASS
  TAC+ RX: flags=00
  TAC+ RX: server msg=
TAC+ EVENT: Authentication PASSED
AAA: TACACS+ authentication passed.
```

An unsuccessful authentication attempt would be similar to the following:

```
AAA: New Session on portal 'SSH 0 (<Client IP>:<Port>)'.
AAA: Session using AUTHENTICATION list 'LoginUseTacacsLocal'.
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Authentication START pkt
TAC+ TX: Authentication START
  TAC+ TX: version=0xc0, type=Authentication, seq_no=1, flags=00
  TAC+ TX: action=Login
  TAC+ TX: level=1
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
  TAC+ TX: remote address=0.0.0.0
TAC+ EVENT: Received Authentication REPLY pkt
TAC+ RX: Authentication REPLY
  TAC+ RX: version=0xc0, type=Authentication, seq_no=2, flags=00
  TAC+ RX: status=GETPASS
  TAC+ RX: flags=0x01
  TAC+ RX: server msg=Password:
TAC+ EVENT: Sending Authentication CONTINUE pkt
TAC+ TX: Authentication CONTINUE
  TAC+ TX: version=0xc0, type=Authentication, seq_no=3, flags=00
  TAC+ TX: user message=********
  TAC+ TX: flags=0x00
TAC+ EVENT: Received Authentication REPLY pkt
```

```
TAC+ RX: Authentication REPLY
  TAC+ RX: version=0xc0, type=Authentication, seq_no=4, flags=00
  TAC+ RX: status=FAIL
  TAC+ RX: flags=00
  TAC+ RX: server msg=
TAC+ EVENT: Authentication FAILED
AAA: TACACS+ authentication failed.
AAA: Closing Session on portal 'SSH 0 (<Client IP>:<Port>)'.
```


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.