Configuring TACACS+ Authorization & Accounting

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 command usage through TACACS+ authorization and logging command history through TACACS+ accounting.

Authorization & Accounting Considerations

Both TACACS+ authorization and accounting are relevant to the Command Line Interface (CLI) only. The Graphical User Interface (GUI) accesses the configuration via a different method and bypasses both of these functions entirely. For that reason, it is recommended that access to the GUI be disabled if these features are utilized.

NOTE: *It is recommended that TACACS+ authentication also be used with these features, although it is not a requirement. This feature is covered in another document entitled "Configuring TACACS+ Authentication for Device Administration".*

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>
```

Enabling Authorization & Accounting on the Console

The console port connections are by default not subject to the Authorization & Accounting configurations. This is typically the case because it is assumed that the device is already physically secured, and a user with physical access does not need to be subject to the restraints that Authorization & Accounting place on the user.

The typical scenario that requires console access is a site down situation. In this situation, the TACACS+ server would typically be unavailable anyway and the user would have to wait for the TACACS+ server authorization attempts to timeout to each server, for each command entered. It can exponentially extend the time a relatively simple operation might take.

If console port connections should be subject to the Authorization & Accounting configurations, the following command must be entered:

```
aaa authorization console
```

Define Command Authorization & Accounting Method

AOS currently only supports Authorization & Accounting features to a TACACS+ server. For the purposes of this document, the only relevant option for methodology is the case of when the TACACS+ server cannot be contacted. The correct answer is matter of network security policy as to whether a secondary TACACS+ server is required to be contacted, if the Authorization & Accounting features should be disabled, or a combination of both.

NOTE: *The Accounting feature is not required to be successful for the commands to be issued, so that function has a built-in "none" option that is not shown at the end of "aaa accounting" commands.*

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

The next question is what commands will be sent to the TACACS+ server for authorization. The commands within AOS are grouped into Level 1 (unprivileged) and Level 15 (privileged). Level 1 commands can be accessed from User mode (Router>); Level 15 commands require Enable mode access (Router#). Typically, only Level 15 commands are sent for authorization, as no configuration-affecting or processor-intensive commands can be issued from Level 1.

An example configuration for authorizing Level 15 commands only, and allow all commands should the TACACS+ server be unavailable, is:

```
aaa authorization commands 1 default none
aaa authorization commands 15 default group tacacs+ none
```

An example configuration for accounting all commands is:

```
aaa accounting commands 1 default stop-only group tacacs+
aaa accounting commands 15 default stop-only group tacacs+
```

Define Exec Authorization & Accounting Method

The "Auto-Exec" feature is used automatically during login. After the user has logged into the unit via any method, the TACACS+ server is queried to determine the administrative level of the user in question. If the response indicates the user has Level 15 privileges, the user is placed directly into enable mode (Router#).

As described in the "Configuring TACACS+ Authentication for Device Administration" document, using the "Auto-Exec" feature is preferable to the standard TACACS+ enable authentication method.

An example configuration for using the "Auto-Exec" feature, and automatically place the user into enable mode should the TACACS+ server be unavailable, is:

```
aaa authorization exec default group tacacs+ none
```

An example configuration for accounting all "Auto-Exec" attempts is:

```
aaa accounting exec default start-stop group tacacs+
```

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 & "Auto-Exec" attempt would be similar to the following:

```
AAA: New Session on portal 'SSH 0 (<Client IP>:<Port>)'.
AAA: No list mapped to 'SSH 0'. Using 'default'.
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=<Client IP>
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.
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Authorization REQUEST pkt
TAC+ TX:Authorization REQUEST
  TAC+ TX: version=0xc0, type=Authorization, seq_no=1, flags=00
  TAC+ TX: method=TACACSPLUS
  TAC+ TX: level=1
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: arg_cnt=2
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
  TAC+ TX: remote address=<Client IP>
  TAC+ TX: arg:service=shell
  TAC+ TX: arg:cmd*
TAC+ EVENT: Received Authorization RESPONSE pkt
TAC+ RX: Authorization RESPONSE
  TAC+ RX: version=0xc0, type=Authorization, seq_no=2, flags=00
  TAC+ RX: status=PASS_ADD
  TAC+ RX: arg_cnt=1
  TAC+ RX: server msg=
  TAC+ RX: arg=priv-lvl=15
TAC+ EVENT: Authorization PASSED
AAA: TACACS+ command authorization passed.
AAA: Authorization passed. Entering command level 15.
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Accounting REQUEST pkt
TAC+ TX: Accounting REQUEST
  TAC+ TX: version=0xc0, type=Accounting, seq_no=1, flags=00
  TAC+ TX: flags=0x02
  TAC+ TX: method=TACACSPLUS
  TAC+ TX: level=15
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: arg_cnt=5
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
  TAC+ TX: remote address=console
  TAC+ TX: arg:timezone=GMT+05:30
  TAC+ TX: arg:service=shell
  TAC+ TX: arg:start_time=1267169511
  TAC+ TX: arg:priv-lvl=15
  TAC+ TX: arg:task-id=3
TAC+ EVENT: Received Accounting REPLY pkt
TAC+ RX: Accounting REPLY
  TAC+ RX: version=0xc0, type=Accounting, seq_no=2, flags=00
  TAC+ RX: status=SUCCESS
  TAC+ RX: arg_cnt=5
  TAC+ RX: server msg=
TAC+ EVENT: Accounting SUCCESSFUL
AAA: TACACS+ command accounting 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=<Client IP>
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>)'.
```

A successful command authorization would be similar to the following:

```
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Authorization REQUEST pkt
TAC+ TX:Authorization REQUEST
  TAC+ TX: version=0xc0, type=Authorization, seq_no=1, flags=00
  TAC+ TX: method=TACACSPLUS
  TAC+ TX: level=15
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: arg_cnt=4
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
  TAC+ TX: remote address=<Client IP>
  TAC+ TX: arg:service=shell
  TAC+ TX: arg:cmd=show
  TAC+ TX: arg:cmd-arg=running-config
  TAC+ TX: arg:cmd-arg=<cr>
TAC+ EVENT: Received Authorization RESPONSE pkt
TAC+ RX: Authorization RESPONSE
  TAC+ RX: version=0xc0, type=Authorization, seq_no=2, flags=00
  TAC+ RX: status=PASS_ADD
  TAC+ RX: arg_cnt=0
  TAC+ RX: server msg=
TAC+ EVENT: Authorization PASSED
AAA: TACACS+ command authorization passed.
```

An unsuccessful command authorization would be similar to the following:

```
TAC+ EVENT: Trying group 'tacacs+'
TAC+ EVENT: Attempting connection to host '<Server IP>'.
TAC+ EVENT: Sending Authorization REQUEST pkt
TAC+ TX:Authorization REQUEST
  TAC+ TX: version=0xc0, type=Authorization, seq_no=1, flags=00
  TAC+ TX: method=TACACSPLUS
  TAC+ TX: level=15
  TAC+ TX: authen type=ASCII
  TAC+ TX: requested service=Login
  TAC+ TX: arg_cnt=4
  TAC+ TX: username=<username>
  TAC+ TX: port=SSH 0 (<Client IP>:<Port>)
  TAC+ TX: remote address=<Client IP>
  TAC+ TX: arg:service=shell
  TAC+ TX: arg:cmd=configure
  TAC+ TX: arg:cmd-arg=terminal
  TAC+ TX: arg:cmd-arg=<cr>
```

```
TAC+ EVENT: Received Authorization RESPONSE pkt
TAC+ RX: Authorization RESPONSE
  TAC+ RX: version=0xc0, type=Authorization, seq_no=2, flags=00
  TAC+ RX: status=FAIL
  TAC+ RX: arg_cnt=0
  TAC+ RX: server msg=
TAC+ EVENT: Authorization FAILED
AAA: TACACS+ command authorization failed.
```