# Configure a Cisco Router with TACACS+ Authentication

**Document ID: 13865**

## Contents

## Introduction

This document describes how to configure a Cisco router for authentication with the TACACS+ that runs on UNIX. TACACS+ does not offer as many features as the commercially available Cisco Secure ACS for Windows or Cisco Secure ACS UNIX.

TACACS+ software previously provided by Cisco Systems has been discontinued and is no longer supported by Cisco Systems.

Today, you can find many available TACACS+ freeware versions when you search for "TACACS+ freeware" on your favorite Internet search engine. Cisco does not specifically recommend any particular TACACS+ freeware implementation.

Cisco Secure Access Control Server (ACS) is available for purchase through regular Cisco sales and distribution channels worldwide. Cisco Secure ACS for Windows includes all the necessary components needed for an independent installation on a Microsoft Windows workstation. The Cisco Secure ACS Solution Engine is shipped with a pre−installed Cisco Secure ACS software license. Visit the Cisco Ordering Home Page (registered customers only) to place an order.

**Note:** You need a CCO account with an associated Service Contract to get the 90−day trial version for Cisco Secure ACS for Windows.

The router configuration in this document was developed on a router that runs Cisco IOS® Software Release 11.3.3. Cisco IOS Software Release 12.0.5.T and later uses **group tacacs+** instead of **tacacs+**, so statements such as **aaa authentication login default tacacs+ enable** appear as **aaa authentication login default group tacacs+ enable**.

Refer to the Cisco IOS Software documentation for more complete information on router commands.

## Prerequisites

## Requirements

There are no specific requirements for this document.

## Components Used

The information in this document is based on Cisco IOS Software Release 11.3.3 and Cisco IOS Software Release 12.0.5.T and later.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

# Authentication

Complete these steps:

1. Make sure you have compiled TACACS+ (TAC+) code on the UNIX server.

   The server configurations here assume you use the Cisco TAC+ server code. The router configurations should work whether or not the server code is Cisco server code. TAC+ must be run as root; su to root if necessary.

2. Copy the test_file at the end of this document, place it on the TAC+ server, and name it **test_file**.

   Check to be sure the **tac_plus_executable** daemon starts with **test_file**. In this command, the **−P** option checks for compile errors but does not start the daemon:

   ```
   tac_plus_executable −P −C test_file
   ```

   You might see the contents of test_file scroll down the window, but you should not see messages such as `cannot find file`, `cleartext expected--found cleartext`, or `unexpected }`. If there are errors, check paths to test_file, re−check your typing, and re−test before you continue.

3. Start to configure TAC+ on the router.

   Enter **enable** mode and type **configure terminal** before the command set. This command syntax ensures that you are not locked out of the router initially, providing the **tac_plus_executable** is not running:

   ```
   !--- Turn on TAC+.

       aaa new-model
       enable password whatever

   !--- These are lists of authentication methods.
       !--- "linmethod", "vtymethod", "conmethod", and
       !--- so on are names of lists, and the methods
       !--- listed on the same lines are the methods
       !--- in the order to be tried. As used here, if
       !--- authentication fails due to the
       !--- tac_plus_executable not being started, the
       !--- enable password is accepted because
   ```

```
                !--- it is in each list.

                !
                aaa authentication login linmethod tacacs+ enable
                aaa authentication login vtymethod tacacs+ enable
                aaa authentication login conmethod tacacs+ enable
                !

        !--- Point the router to the server, where #.#.#.#
            !--- is the server IP address.

                !
                tacacs-server host #.#.#.#
                line con 0
                    password whatever

        !--- No time-out to prevent being locked out
                !--- during debugging.

                exec-timeout 0 0
                login authentication conmethod
            line 1 8
                login authentication linmethod
                modem InOut
                transport input all
                rxspeed 38400
                txspeed 38400
                flowcontrol hardware
            line vty 0 4
                password whatever

        !--- No time-out to prevent being locked out
                !--- during debugging.

                exec-timeout 0 0
                login authentication vtymethod
```
4. Test to be sure you can still access the router with Telnet and through the console port before you continue. Because the **tac_plus_executable** is not running, the **enable** password should be accepted.

   **Note:** Keep the console port session active and remain in enable mode. This session should not time out. Access to the router is limited at this point, and you need to be able to make configuration changes without locking yourself out.

   Issue these commands to see server–to–router interaction at the router:

```
        terminal monitor
          debug aaa authentication
```
5. As root, start TAC+ on the server:

```
        tac_plus_executable -C test_file -d 16
```
6. Check to be sure TAC+ started:

```
        ps -aux | grep tac_plus_executable
```

   or

```
        ps -ef | grep tac_plus_executable
```

   If TAC+ does not start, it is usually a problem with syntax in the test_file. Return to step 1 to correct this.
7. Type **tail −f /var/tmp/tac_plus.log** to see router–to–server interaction at the server.

**Note:** The `-d 16` option in step 5 sends output of all transactions to the /var/tmp/tac_plus.log.

8. Telnet (VTY) users should now have to authenticate through TAC+.

   With debug going on the router and the server (steps 4 and 7), Telnet into the router from another part of the network.

   The router produces a username and password prompt, to which you reply:

   ```
   'authenuser'   (username from test_file)
   'admin'   (password from test_file)
   ```

   The user `authenuser` is in group `admin`, which has the password `admin`.

   Watch the server and the router where you can see the TAC+ interaction what is sent where, responses, requests, and so on. Correct any problems before you continue.

9. If you also want your users to authenticate through TAC+ in order to get into enable mode, make sure your console port session is still active and add this command to the router:

   ```
   !--- For enable mode, list 'default' looks to TAC+
     !--- then enable password if TAC+ does not run.

     aaa authentication enable default tacacs+ enable
   ```

   Users now have to enable through TAC+.

10. With debug going on the router and the server (steps 4 and 7), Telnet into the router from another part of the network. The router produces a username and password prompt, to which you reply:

    ```
    'authenuser'   (username from test_file)
    'admin'   (password from test_file)
    ```

    When you enter enable mode, the router requests a password, to which you reply:

    ```
    'cisco'   ($enable$ password from test_file)
    ```

    Watch the server and the router where you should see the TAC+ interaction   what is sent where, responses, requests, and so on. Correct any problems before you continue.

11. Bring down the TAC+ process on the server while still connected to the console port to be sure that your users can still access the router if TAC+ is down:

    ```
    ps -aux | grep tac_plus_executable
    ```

    or

    ```
    ps -ef | grep tac_plus_executable)
      kill -9 pid_of_tac_plus_executable
    ```

    Repeat the Telnet and enable of the previous step. The router then realizes that the TAC+ process is not responding and allows users to log in and enable with the default passwords.

12. Check for authentication of your console port users through TAC+. In order to do this, bring up the TAC+ server again (steps 5 and 6), and establish a Telnet session to the router (which should authenticate through TAC+).

    Remain connected through Telnet into the router in enable mode until you are sure you can log in to the router through the console port.

Log out of your original connection to the router through the console port, then reconnect to the console port. Console port authentication to log in and enable using user IDs and passwords (shown in step 10) should now be through TAC+.

13. While you remain connected through either a Telnet session or the console port and with debug going on the router and the server (steps 4 and 7), establish a modem connection to line 1.

Line users now have to log in and enable through TAC+.

The router produces a username and password prompt, to which you reply:

```
'authenuser'   (username from test_file)
'admin'   (password from test_file)
```

When you enter enable mode, the router requests a password.

Reply:

```
'cisco'   ($enable$ password from test_file)
```

Watch the server and the router where you see the TAC+ interaction   what is sent where, responses, requests, and so forth. Correct any problems before you continue.

Users now have to enable through TAC+.

# Add Authorization

Adding authorization is optional.

By default, there are three command–levels on the router:

- privilege level 0 which includes disable, enable, exit, help, and logout
- privilege level 1 – normal level on a Telnet – prompt says `router>`
- privilege level 15 – enable level – prompt says `router#`

Since commands available depend on the IOS feature set, version of Cisco IOS, model of router, and so on, there is not a comprehensive list of all commands at levels 1 and 15. For example, **show ipx route** is not present in an IP only feature set, **show ip nat trans** is not in Cisco IOS Software Release 10.2.x because NAT was not introduced at the time, and **show environment** is not present in router models without power supply and temperature monitoring. Commands available in a particular router at a particular level can be found when you enter a **?** at the prompt in the router when at that privilege level.

Console port authorization was not added as a feature until Cisco bug ID CSCdi82030 (registered customers only) was implemented. Console port authorization is off by default to lessen the likelihood that you become accidentally locked out of the router. If a user has physical access to the router through the console, console port authorization is not extremely effective. However, console port authorization can be turned on under line `con 0` in an image that Cisco bug ID CSCdi82030 (registered customers only) was implemented in with the command:

```
authorization exec default|WORD
```

1. The router can be configured to authorize commands through TAC+ at all or some levels.

This router configuration allows all users to have per–command authorization set up on the server. Here we authorize all commands through TAC+, but if the server is down, no authorization is

necessary.

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

2. While the TAC+ server runs, Telnet into the router with userid **authenuser**.

Because authenuser has default service = permit in test_file, this user should be able to perform all functions.

While in the router, enter **enable** mode, and turn on authorization debugging:

```
terminal monitor
  debug aaa authorization
```

3. Telnet into the router with userid **authoruser** and password **operator**.

This user is not able to do the two show commands **traceroute** and **logout** (see the test_file).

Watch the server and the router where you should see the TAC+ interaction (what is sent where, responses, requests, and so on). Correct any problems before you continue.

4. If you want to configure a user for an autocommand, eliminate the commented−out user transient in the test_file, and put a valid IP address destination in place of the #.#.#.#.

Stop and start the TAC+ server.

On the router:

```
aaa authorization exec default tacacs+
```

Telnet to the router with userid **transient** and password **transient**. The **telnet #.#.#.#** executes and user transient is sent to the other location.

# Add Accounting

Adding accounting is optional.

Reference to the accounting file is in test_file   accounting file = /var/log/tac.log. But accounting does not take place unless configured in the router (provided the router runs a version of Cisco IOS software later than 11.0).

1. Enable accounting in the router:

```
aaa accounting exec default start-stop tacacs+
  aaa accounting connection default start-stop tacacs+
  aaa accounting network default start-stop tacacs+
  aaa accounting system default start-stop tacacs+
```

**Note:** AAA accounting does not do per−command accounting in some versions. A workaround is to use per−command authorization and log the occurrence in the accounting file. (Refer to Cisco bug ID CSCdi44140.) If you use an image in which this fixed is used [Cisco IOS Software Releases 11.2(1.3)F, 11.2(1.2), 11.1(6.3), 11.1(6.3)AA01, 11.1(6.3)CA as of September 24, 1997] you can also enable command−accounting.

2. While TAC+ runs on the server, enter this command on the server to see the entries that go into the accounting file:

```
tail -f /var/log/tac.log
```

Then log into and out of the router, Telnet out of the router, and so forth. If necessary, on the router enter:

```
terminal monitor
 debug aaa accounting
```

## Test File

```
- - - - - - - (cut here) - - - - - - -

# Set up accounting file if enabling accounting on NAS
accounting file = /var/log/tac.log

# Enable password setup for everyone:
user = $enable$ {
        login = cleartext "cisco"
        }

# Group listings must be first:
group = admin {
# Users in group 'admin' have cleartext password
        login = cleartext "admin"
        expires = "Dec 31 1999"
}

group = operators {
# Users in group 'operators' have cleartext password
        login = cleartext "operator"
        expires = "Dec 31 1999"
}

group = transients {
# Users in group 'transient' have cleartext password
        login = cleartext "transient"
        expires = "Dec 31 1999"
}

# This user is a member of group 'admin' & uses that group's password to log in.
#  The $enable$ password is used to enter enable mode.  The user can perform all commands.
 user = authenuser {
        default service = permit
        member = admin
        }

# This user is limitted in allowed commands when aaa authorization is enabled:
 user = telnet {
        login = cleartext "telnet"
        cmd = telnet {
        permit .*
        }
        cmd = logout {
        permit .*
        }
        }

# user = transient {
#       member = transients
#       service = exec {
        # When transient logs on to the NAS, he's immediately
        # zipped to another site
#       autocmd = "telnet #.#.#.#"
#       }
#       }

# This user is a member of group 'operators'
```

```
# & uses that group's password to log in
 user = authenuser {
        member = operators
# Since this user does not have 'default service = permit' when command
# authorization through TACACS+ is on at the router, this user's commands
# are limited to:
        cmd = show {
        permit ver
        permit ip
        }
        cmd = traceroute {
        permit .*
        }
        cmd = logout {
        permit .*
        }
}
- - - - (end cut here) - - - -
```

**Note:** This error message is generated if your TACACS server is not reachable:
`%AAAA-3-DROPACCTSNDFAIL: accounting record dropped,send to server failed:`
`system-start`. Verify that the TACACS+ server is operational.

# Related Information

- **Single−User Network Access Security TACACS+**
- **Terminal Access Controller Access Control System (TACACS+)**
- **Cisco Secure Access Control Server for Windows**
- **Technical Support & Documentation − Cisco Systems**

Updated: Jul 21, 2006                                        Document ID: 13865