

Configuring Access Service Security

The access service security paradigm presented in this guide uses the authentication, authorization, and accounting (AAA) facility. Authentication requires dial in users to identify themselves and prove their identity. Requiring authentication before users can access the network prevents users from either accessing lines on the access server or connecting through the lines directly to network resources. You need to secure every access point.

Authorization prevents each user from gaining access to services and devices on the network that they do not need to or are not supposed to access. Accounting provides records for billing and other recording purposes of who is connected and how long they have been connected. This chapter does not describe how to configure accounting.

This chapter describes how to configure security using a local database resident on the access server or using a remote security database for TACACS+ and RADIUS. To understand the concept of local versus remote authentication, refer to the section “Local Versus Remote Server Authentication” later in this chapter.



Caution This chapter does not provide a comprehensive security overview. For example, it does not describe how to configure TACACS, Extended TACACS, Kerberos, or access lists. It presents the most commonly used security mechanisms to prevent unauthenticated and unauthorized access to network resources through Cisco access servers. For a comprehensive overview of Cisco security tools, refer to the *Security Configuration Guide*.

This chapter describes the following specific topics:

- Local Versus Remote Server Authentication
- Configuring Authentication

Assumptions

- Configuring Authorization
- Security Examples

Assumptions

This chapter assumes the following:

- You know which network protocols will be allowed access to your network. For example, you know if you will be allowing clients to dial in using modems to access IP, IPX, or AppleTalk networks.
- You are not an advanced user of the Cisco AAA security facility.

Local Versus Remote Server Authentication

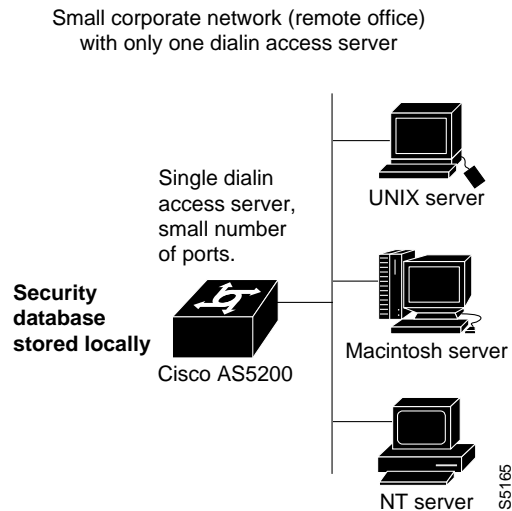
This section describes the differences between local and remote security databases and the basic authentication process for each. Remote security databases described in this chapter include Terminal Access Controller Access Control System with Cisco proprietary enhancements (TACACS+) and Remote Authentication Dial-In User Service (RADIUS).

Generally the size of the network and type of corporate security policies and control determines whether you use a local or remote security database.

Local Security Database

If you have one or two access servers providing access to your network, you should store username and password security information on the Cisco access server. This is referred to as local authentication. See Figure 4-1.

Figure 4-1 Local Security Database



A local security database is useful if you have very few access servers providing network access. A local security database does not require a separate (and costly) security server.

Remote Security Database

As your network expands, you need a centralized security database that provides username and password information to each of the access servers on the network. This centralized security database resides in a security server. See Figure 4-2.

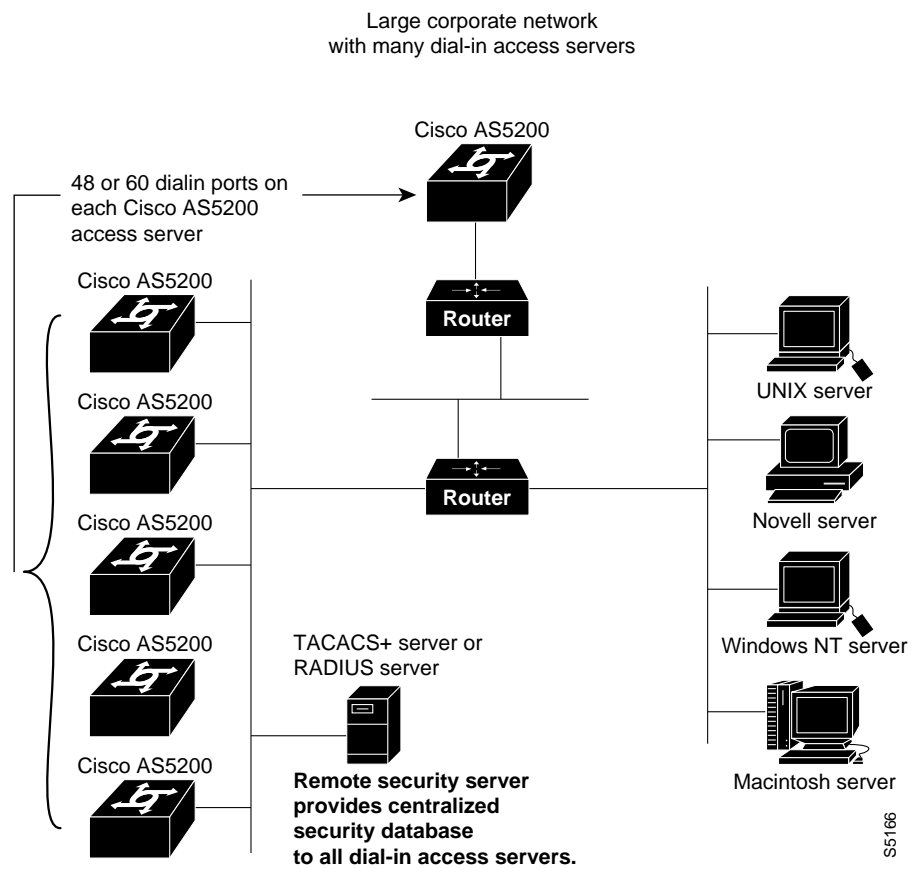
An example of a remote security database server is the CiscoSecure product from Cisco Systems, Inc. CiscoSecure is a UNIX security daemon, with which the administrator creates a database that defines the network users and their privileges. CiscoSecure uses a central database that stores user and group profiles with authentication and authorization information.

The Cisco access server exchanges user authentication information with a TACACS+ or RADIUS database on the security server by transmitting encrypted TACACS+ or RADIUS packets across the network.

Local Versus Remote Server Authentication

For specific information about the interaction between the security server and the access server, refer to the *Security Configuration Guide*.

Figure 4-2 Remote Security Database



A remote, centralized security database is useful when you have a large number of access servers providing network access. It prevents having to update each access server with new or changed authentication and authorization information for potentially hundreds of thousands of dialin network users. A centralized security database also helps establish consistent remote access policies throughout a corporation.

Configuring Authentication

Using the AAA facility, you can authenticate users with either a local or a remote security database. For more information about what a local and remote security database are, refer to the previous section “Local Versus Remote Server Authentication.”

Whether you maintain a local or remote security database, or use TACACS+ or RADIUS authentication and authorization, the process of configuring the access server for these different databases and protocols is similar. The basic process of configuring the Cisco IOS software for authentication requires the following tasks:

- 1 Securing Access to Privileged EXEC and Configuration Mode
- 2 Communicating between the Access Server and the Security Server
- 3 Enabling AAA Globally on the Access Server
- 4 Defining Authentication Method Lists
 - Issue the `aaa authentication` Command
 - Specify Protocol or Login Authentication
 - Identify a List Name
 - Specify the Authentication Method
- 5 Applying Authentication Method Lists to Lines and Interfaces
 - Apply Login Lists to VTY Lines and the console port
 - Apply Authentication Lists to *Interfaces* Configured for PPP
 - Apply Authentication Lists Asynchronous (TTY) *Lines* Configured for ARA

Securing Access to Privileged EXEC and Configuration Mode

The first step is to secure access to privileged EXEC (enable) mode. Enable mode provides access to configuration mode, which enables any type of configuration change to the access server. To secure Privileged EXEC mode, use one of the commands listed in Table 4-1.

Table 4-1 **Commands Used to Secure Access to Privileged EXEC Mode**

Command	Purpose
enable password <i>password</i>	Requires that network administrators enter a password to access privileged EXEC mode. Do not provide access to users who are not administrators.
enable secret <i>password</i>	Specifies a secret password that is encrypted, so that the password cannot be read when crossing a network. After you issue this command, the encryption cannot be reversed. The encrypted version of the password appears in output of the show running-config and show startup-config commands. The enable secret password has precedence over the enable password. Do not enter the same password as the enable password. If the two passwords are the same, the enable secret password is not a secret, because the enable password is not encrypted and appears in output of show running-config and show startup-config commands.

For more information about the enable password and enable secret commands and their complete syntax, refer to the *Security Command Reference*.



Caution If you use the **enable secret** command and specify an encryption type, you *must* enter the *encrypted version* of a specific password. Do not enter the cleartext version of the password after specifying an encryption type. You must comply with the following procedure when you specify an encryption type or you will be locked irretrievably out of privileged EXEC (enable) mode. The only way to regain access to privileged EXEC mode will be to erase the contents of NVRAM, erase your entire configuration, and reconfigure the 5200 again.

To enter an encryption type with the **enable secret** command, follow these steps:

Step 1 From within global configuration mode, enter the **enable secret** command, followed by the cleartext password that you will use to gain access to privileged EXEC mode. Do not specify an encryption type.

Step 2 Exit from global configuration mode and enter the command **show running-config** to view the encrypted version of the password. The following example illustrates these first two steps:

```
5200(config)# enable secret mypassword
5200(config)# exit
5200# show running-config
Building configuration...

Current configuration:
!
version 11.1
! some of the configuration skipped
enable secret 5 $1$h7dd$VTNs4.BAfQMUU0Lrvw6570
! the rest of the configuration skipped
```

Step 3 At this point, select and copy the encrypted password after you enter **enable secret 5** in the configuration output (\$1\$h7dd\$VTNs4.BAfQMUU0Lrvw6570).

Step 4 Enter global configuration mode and enter the **enable secret** command, followed by the encryption type (5 is the only valid encryption type for **enable secret**), then paste in the encrypted version of the password, as shown in the following example:

```
5200(config)# enable secret 5 $1$h7dd$VTNs4.BAfQMUU0Lrvw6570
```

Step 5 Exit from global configuration mode and copy the running configuration to NVRAM.

```
5200(config)# exit
5200# copy running-config startup-config
```

Configuring Authentication

You can also specify additional protection for privileged EXEC mode, including the following:

- Privilege levels for Cisco IOS commands
- Privileged EXEC passwords for different privilege levels
- Privilege levels for specific lines on the access server
- Encrypt passwords using **service password-encryption**

For more information about these security tools, refer to the *Security Configuration Guide* in the Cisco IOS configuration guides and command references documentation.

Communicating between the Access Server and the Security Server

This section describes the Cisco IOS software commands that enable the access server to communicate with a security server. This process is similar for communicating with TACACS+ and RADIUS servers, and the following sections describe the process.

If you are using local authentication, refer to the section “Enabling AAA Globally on the Access Server.”

If you are using a remote security server for authentication and authorization, you must configure the security server before performing the tasks described in this chapter. The section “Security Examples” at the end of this chapter shows some typical TACACS+ and RADIUS server entries corresponding to the access server security configurations.

Communicating with a TACACS+ Server

To enable communication between the TACACS+ security (database) server and the access server, issue the commands listed in Table 4-2 in global configuration mode.

Table 4-2 Commands for Communicating with a TACACS+ Server

Command	Purpose
tacacs-server host { <i>hostname</i> <i>ip-address</i> }	Specifies the IP address or the host name of the remote TACACS+ server host. This host is typically a UNIX system running TACACS+ software.
tacacs-server key <i>shared-secret-text-string</i>	Specifies a shared secret text string used between the access server and the TACACS+ server. The access server and TACACS+ server use this text string to encrypt passwords and exchange responses.

For example, to enable the remote TACACS+ server to communicate with the access server, enter the commands as follows:

```
5200# configure terminal
5200(config)# tacacs-server host alcatraz
5200(config)# tacacs-server key abra2cad
```

The host name of the TACACS+ server in the previous example is *alcatraz*. The key (*abra2cad*) in the previous example is the encryption key shared between the TACACS+ server and the access server.

For more information about these commands, refer to the *Security Command Reference*, which is part of the Cisco IOS configuration guides and command references documentation.

Communicating with a RADIUS Server

To enable communication between the RADIUS security (database) server and the access server, issue the commands listed in Table 4-3 in global configuration mode.

Table 4-3 RADIUS Server Commands

Command	Purpose
radius-server host { <i>hostname</i> <i>ip-address</i> }	Specifies the IP address or the host name of the remote RADIUS server host. This host is normally a UNIX system running RADIUS software.
radius-server key <i>shared-secret-text-string</i>	Specifies a shared secret text string used between the access server and the RADIUS server. The access server and RADIUS server use this text string to encrypt passwords and exchange responses.

For example, to enable the remote RADIUS server to communicate with the access server, enter the commands as follows:

```
5200# configure terminal
5200(config)# radius-server host alcatraz
5200(config)# radius-server key abra2cad
```

The host name of the RADIUS server in the previous example is *alcatraz*. The key (*abra2cad*) in the previous example is the encryption key shared between the RADIUS server and the access server.

You can use any of the following optional commands to interact with the RADIUS server host:

- **radius-server retransmit** *number*

This command specifies the number of times that the access server transmits each RADIUS request to the server before the access server gives up.

- **radius-server timeout** *seconds*

This command specifies the number of seconds that an access server waits for a reply to a RADIUS request before the access server retransmits the request. The default is five seconds. If the RADIUS server's response is slow (because of support for a large number of users or large network latency), increase the timeout value.

For more information about these commands, refer to the *Security Command Reference*, which is part of the Cisco IOS configuration guides and command references.

Configuring Authentication on a TACACS+ Server

On most TACACS+ security servers, there are three ways to authenticate a user for login:

- Include a cleartext (DES) password for a user or for a group the user is a member of (each user can belong to only one group). Note that ARAP, CHAP, and global user authentication must be specified in cleartext.

The following is the configuration for global authentication:

```
user = mswartz {global = cleartext "mswartz global password"}
```

To assign different passwords for ARAP, CHAP, and a normal login, you must enter a string for each user. Each string must specify the security protocols, state whether the password is cleartext, and specify if the authentication is performed via a DES card. The following example shows a user aaaa, who has authentication configured for ARAP, CHAP, and login. Her ARAP and CHAP passwords, “arap password” and “chap password”, are shown in cleartext. Her login password has been encrypted.

```
user = aaaa {arap = cleartext "arap password"
             chap = cleartext "chap password"
             login = des XQj4892fjk}
```

- Use password (5) files instead of entering the password into the configuration file directly.

The default authentication is to deny authentication. You can change this at the top level of the configuration file to have the default use passwd(5) file, by issuing the following command:

```
default authentication = /etc/passwd
```

- Authenticate using an s/key. If you have built and linked in an s/key library and compiled TACACS+ to use the s/key, you can specify that a user be authenticated via the s/key, as shown in the following example:

```
user= bbbb {login = skey}
```

Configuring Authentication

On the access server, configure authentication on all lines including the VTY and Console lines by entering the following commands, beginning in privileged EXEC mode:

```
5200# configure terminal
5200(config)# aaa new-model
5200(config)# aaa authentication login default tacacs+ enable
```



Caution When you issue the **aaa authentication login default tacacs+ enable** command, you are specifying that if your TACACS+ server fails to respond (because it is set up incorrectly), you can log in to the access server by using your enable password. If you do not have an enable password set on the access server, you will not be able to log in to it until you have a functioning TACACS+ daemon configured with usernames and passwords. The enable password in this case is a last-resort authentication method. You can also specify **none** as the last-resort method, which means that no authentication is required if all other methods failed.

Enabling AAA Globally on the Access Server

To use the AAA security facility in the Cisco IOS software, you must issue the **aaa new-model** command from global configuration mode.

When you issue the **aaa new-model** command, all lines on the access server receive the implicit **login authentication default** method list, and all interfaces with PPP enabled have an implicit **ppp authentication pap default** method list applied.



Caution If you intend to authenticate users via a security server, make sure you do not inadvertently lock yourself out of the access server ports after you issue the **aaa new-model** command. Enter line configuration mode and issue the **aaa authentication login default tacacs+ enable** global configuration command. This command specifies that if your TACACS+ (or RADIUS) server is not functioning properly, you can enter your enable password to log in to the access server. In general, make sure you have a last-resort access method before you are certain that your security server is set up and functioning properly. For more information about the **aaa authentication** command, refer to the “Defining Authentication Method Lists” section.

Note Cisco recommends that you use CHAP authentication with PPP, rather than PAP. CHAP passwords are encrypted when they cross the network, whereas PAP passwords are cleartext when they cross the network. The Cisco IOS software selects PAP as the default, so you must manually select CHAP. The process for specifying CHAP is described in the “Applying Authentication Method Lists” section.

For example, enter the following commands to enable AAA in the Cisco IOS software:

```
5200# configure terminal
5200(config)# aaa new-model
```

Defining Authentication Method Lists

After you enable AAA globally on the access server, you need to define authentication method lists, which you then apply to lines and interfaces. These authentication method lists are security profiles that indicate the protocol (ARAP or PPP) or login and authentication method (TACACS+, RADIUS, or local authentication).

To define an authentication method list, follow these steps, which are described in this section:

- 1 Issue the **aaa authentication** command.
- 2 Specify protocol (ARAP or PPP) or login authentication.
- 3 Identify a list name or **default**. A list name is any alphanumeric string you choose. You assign different authentication methods to different named lists.
- 4 Specify the authentication method. You can specify multiple methods, such as **tacacs+**, followed by **local** in case a TACACS+ server is not available on the network.
- 5 Populate the local username database if you specified **local** as the authentication method (or one of the authentication methods). To use a local username database, you must issue the **username** global configuration command. Refer to the task “Populate the Local Username Database if Necessary.”

Configuring Authentication

After defining these authentication method lists, apply them to one of the following:

- Lines—VTY lines or the console port for login and asynchronous lines (in most cases) for ARA
- Interfaces—Interfaces (synchronous or asynchronous) configured for PPP

The section “Applying Authentication Method Lists” describes how to apply these lists.

Issue the `aaa authentication` Command

To define an authentication method list, start by issuing the **`aaa authentication`** global configuration command, as shown in the following example:

```
5200# configure terminal
5200(config)# aaa authentication
```

Specify Protocol or Login Authentication

After you issue **`aaa authentication`**, you must specify one of the following dialin protocols as applicable for your network:

- If you are enabling dialin PPP access, specify **`ppp`**
- If you are enabling dialin ARA access, specify **`arap`**
- If you are enabling users to connect to the EXEC facility, specify **`login`**

You can specify only one dialin protocol per authentication method list. However, you can create multiple authentication method lists with each of these options. You must give each list a different name, as described in the next section “Identify a List Name.”

If you specify the **`ppp`** option, the default authentication method for PPP is PAP. For greater security, specify CHAP. The full command is **`aaa authentication ppp chap`**. If you specify the **`arap`** option, the authentication method built into ARA is used. The full command is **`aaa authentication arap`**.

For example, if you specify PPP authentication, the configuration looks like this:

```
5200# configure terminal
5200(config)# aaa authentication ppp
```

Identify a List Name

A list name identifies each authentication list. You can choose either to use the keyword **default**, or choose any other name that describes the authentication list. For example, you might give it the name `ppp-radius` if you intend to apply it to interfaces configured for PPP and RADIUS authentication. The list name can be any alphanumeric string. Use **default** as the list name for most lines and interfaces, and use different names on an exception basis.

You can create different authentication method lists and apply them to lines and interfaces selectively. You can even create a named authentication method list that you do not apply to a line or interface, but which you intend to apply at some later point, such as when you deploy a new login method for users.

After you define a list name, you must identify additional security attributes (such as local authentication versus TACACS+ or RADIUS).

In the following example, the default authentication method list for PPP dialin clients uses the local security database:

```
5200# configure terminal
5200(config)# aaa authentication ppp default
```

In the following example, the PPP authentication method list name is `insecure`:

```
5200# configure terminal
5200(config)# aaa authentication ppp insecure
```

In the following example, the ARA authentication method list name is `callback` (because asynchronous callback is used on the access server):

```
5200# configure terminal
5200(config)# aaa authentication arap callback
```

In the following example, the login authentication method list name is `deveng`:

```
5200# configure terminal
5200(config)# aaa authentication login deveng
```

Specify the Authentication Method

After you identify a list name, you must specify an authentication method. An authentication method identifies how users are authenticated. For example, will users be authenticated by a local security database resident on the access server (local method)? Will they be authenticated by a remote security database, such as by a TACACS+ or RADIUS daemon? Will guest access to an AppleTalk network be permitted?

Authentication methods are defined with optional keywords in the **aaa authentication** command. The available authentication methods for PPP are described in Table 4-4. The available authentication methods for ARA are described in Table 4-5.

Table 4-4 PPP Authentication Methods

Authentication Methods for PPP	Purpose
if-needed	Authenticates only if not already authenticated. No duplicate authentication.
krb5	Specifies Kerberos 5 authentication.
local	Uses the local username database in the access server. This is defined with the username global configuration command.
none	No authentication is required. Do not prompt for a username or password.
radius	Use RADIUS authentication as defined on a RADIUS security server.
tacacs+	Use TACACS+ authentication as defined on a TACACS+ security server.



Timesaver If you are not sure whether you should use TACACS+ or RADIUS, here are some comparisons: TACACS+ encrypts the entire payload of packets passed across the network, whereas RADIUS only encrypts the password when it crosses the network. TACACS+ can query the security server multiple times, whereas a RADIUS server gives one response only and is therefore not as flexible regarding per-user authentication and authorization attempts. Moreover, RADIUS does not support authentication of ARA.

Table 4-5 ARA Authentication Methods

Authentication Methods for ARA	Purpose
auth-guest	Allows guests to log in only if they have already been authenticated at the EXEC.
guest	Allows guests to log in.
line	Uses the line (login) password for authentication.
local	Uses the local username database in the access server for authentication. This database is defined with the username global configuration command.
tacacs+	Use TACACS+ authentication as defined on a TACACS+ security server.

Note RADIUS does not support ARA. If you want to authenticate Macintosh users with RADIUS, you must configure AppleTalk to run over PPP, which is referred to as ATCP.

You can specify multiple authentication methods for each authentication list. The following example authentication method list for PPP first queries a TACACS+ server, then a RADIUS server, then the local security database. Multiple authentication methods can be useful if you have multiple types of security servers on the network and one or more types of security server do not respond:

```
5200(config)# aaa authentication ppp testbed tacacs+ radius local
```

If you specify more than one authentication method and the first method (TACACS+ in the previous example) is not available, the Cisco IOS software attempts to authenticate using the next method (such as RADIUS). If in the previous example the RADIUS server has no information about the user, or if no RADIUS server can be found, the user is authenticated using the local username database that was populated with the **username** command.

However, if authentication *fails* using the first method listed, the Cisco IOS software does *not* permit access. It does not attempt to authenticate using the subsequent security methods if the user entered the incorrect password.

Configuring Authentication

Populate the Local Username Database if Necessary

If you specify **local** as the security method, you must specify username profiles for each user who might log in. An example of specifying local authentication is as follows:

```
5200(config)# aaa authentication login deveng local
```

This command specifies that any time a user attempts to log in to a line on an access server, the Cisco IOS software checks the username database. To create a local username database, define username profiles using the **username** global configuration command.

The following example shows how to use the **username** command for a user jnieters with password nlvriti:

```
5200(config)# username jnieters password nlvriti
```

The **show running-config** command shows the encrypted version of the password, as follows:

```
5200# show running-config  
Building configuration...  
  
Current configuration:  
!  
version 11.1  
! most of config omitted  
username jnieters password 7 0215055500070C294D
```

Note The Cisco IOS software adds the encryption type of 7 automatically for passwords. If you were to manually enter the number 7 to represent an encryption type, you must follow the 7 with the *encrypted* version of the password. If you specify the number 7, then enter a cleartext password, the user will not have access to the line, interface, or the network the user is trying to access, and you must reconfigure the user's authentication profile.

Authentication Method List Examples

This section shows some examples of authentication lists.

Authentication Method List Examples for Users Logging in to the Access Server

The following example creates a local authentication list for users logging in to any line on the access server.

```
5200(config)# aaa authentication login default local
```

The following example specifies login authentication using RADIUS (the RADIUS daemon is polled for authentication profiles):

```
5200(config)# aaa authentication login default radius
```

The following example specifies login authentication using TACACS+ (the TACACS+ daemon is polled for authentication profiles):

```
5200(config)# aaa authentication login default tacacs+
```

Authentication List Examples for Dialin Users Using ARA to Access Network Resources

The following example creates a local authentication list for Macintosh users dialing in to an AppleTalk network through the access server.

```
5200(config)# aaa authentication arap default local
```

The following example specifies that Macintosh users dialing in to an AppleTalk network through the access server be authenticated by a TACACS+ daemon:

```
5200(config)# aaa authentication arap default tacacs+
```

The following example creates an authentication method list that does the following:

- Enables guest access if the guest has been authenticated at the EXEC facility
- Queries a TACACS+ daemon for authentication

Configuring Authentication

- Polls the line (login) authentication password if the TACACS+ server has no information about the user or if no TACACS+ server on the network responds
- Uses the local security database if there is no line password

```
5200(config)# aaa authentication arap default auth-guest tacacs+ line  
local
```

Authentication Method List Examples for Users Dialing In Using PPP

The following example creates a TACACS+ authentication list for users connecting to interfaces configured for dialin using PPP. The name of the list is **marketing**. This example specifies that a remote TACACS+ daemon be used as the security database. If this security database is not available, the Cisco IOS software then polls the RADIUS daemon. Users are not authenticated if they are already authenticated on a TTY line.

```
5200(config)# aaa authentication ppp marketing if-needed tacacs+ radius
```

In this example, **default** can be substituted for **marketing** if the administrator wants this list to be the default list.

Applying Authentication Method Lists

As described in the “Defining Authentication Method Lists” section, the **aaa authentication** global configuration command creates authentication method lists or profiles. You apply these authentication method lists to lines or interfaces by issuing the **login authentication**, **arap authentication**, or **ppp authentication** command, as described in Table 4-6.

Table 4-6 Line and Interface Authentication Method Lists

Interface and Line Command	Action	Port to which List is Applied	Corresponding Global Configuration Command
login authentication	Logs directly in to the access server	Console Port or VTY lines	aaa authentication login
arap authentication	Uses ARA to access AppleTalk network resources	TTY line	aaa authentication arap
ppp authentication ¹	Uses PPP to access IP or IPX network resources	Interface	aaa authentication ppp

1. If you issued the **ppp authentication** command, you must specify either CHAP or PAP authentication. PAP is enabled by default, but Cisco recommends that you use CHAP because CHAP is more secure. For more information, refer to the *Security Configuration Guide*.

You can create more than one authentication list or profile for login and protocol authentication and apply them to different lines or interfaces. The following examples show the line or interface authentication commands that correspond to the **aaa authentication** global configuration command.

Login Authentication Examples

The following example shows the default login authentication list applied to the console port and the default virtual terminal (VTY) lines on the access server:

```
5200(config)# aaa authentication login default local
5200(config)# line console 0
5200(config-line)# login authentication default
5200(config-line)# line vty 0 4
5200(config-line)# login authentication default
```

Configuring Authentication

In the following example, the login authentication list named `rtp2-office`, which uses RADIUS authentication, is created. It is applied to all 54 lines on a Cisco AS5200 access server configured with a Dual T1 PRI card, including the console (CTY) port, the 48 physical asynchronous (TTY) lines, the auxiliary (AUX) port, and 5 virtual terminal (VTY) lines:

```
5200(config)# aaa authentication login rtp2-office radius
5200(config)# line 0 54
5200(config-line)# login authentication rtp2-office
```

The following sample output shows lines and their status on the access server:

```
5200#sho line
Tty Typ      Tx/Rx      A Modem  Roty AccO AccI  Uses    Noise  Overruns
*  0 CTY                -      -      -      -      -      0        0        0/0
I  1 TTY  57600/57600 - inout      -      -      -      0        0        0/0
I  2 TTY  57600/57600 - inout      -      -      -      0        0        0/0
...
I 48 TTY  57600/57600 - inout      -      -      -      0        0        0/0
 49 AUX   9600/9600  -      -      -      -      -      0        0        0/0
 50 VTY                -      -      -      -      -      0        0        0/0
 51 VTY                -      -      -      -      -      0        0        0/0
 52 VTY                -      -      -      -      -      0        0        0/0
 53 VTY                -      -      -      -      -      0        0        0/0
 54 VTY                -      -      -      -      -      0        0        0/0
```

ARA Authentication Examples

In the following example, the ARA authentication list `bldg-d-list` is created, then applied to lines 1 through 48 (the physical asynchronous lines) on a Cisco AS5200 access server:

```
5200(config)# aaa authentication arap bldg-d-list auth-guest tacacs+
5200(config)# line 1 48
5200(config-line)# arap authentication bldg-d-list
```

PPP Authentication Examples

The following example creates the PPP authentication list marketing, which uses TACACS+, then RADIUS authentication. The list marketing requires authentication only if the user has not already been authenticated on another line. It is then applied to asynchronous lines 1 through 48 on a Cisco AS5200 access server and uses CHAP authentication, instead of the default of PAP:

```
5200(config)# aaa authentication ppp marketing if-needed tacacs+ radius
5200(config)# line 1 48
5200(config-line)# ppp authentication chap marketing
```

Configuring Authorization

You can configure the access server to restrict user access to the network so that users can only perform certain functions after successful authentication. As with authentication, authorization can be used with either a local or remote security database. This guide describes only remote security server authorization.

A typical configuration most likely uses the EXEC facility and network authorization. EXEC authorization restricts access to the EXEC, and network authorization restricts access to network services, including PPP and ARA.

Authorization must be configured on both the access server and the security daemon. The default authorization is different on the access server and the security server:

- By default, the access server *permits* access for every user until you configure the access server to make authorization requests to the daemon.
- By default, the daemon *denies* authorization of anything that is not explicitly permitted. Therefore, you have to explicitly allow all per-user attributes on the security server.



Timesaver If authentication has not been set up for a user, per-user authorization attributes are not enabled for that user. That is, if you want a user to obtain authorization before gaining access to network resources, you must first require that the user provide authentication. For example, if you want to specify the **aaa authorization network tacacs+** (or **radius**) command, you must first specify the **aaa authentication {ppp | arap} default if-needed tacacs+** (or **radius**) command.

Configuring Authorization on the Security Server

You typically have three methods for configuring default authorization on the security server:

- To override the default denial or authorization from a non-existent user, specify authorization at the top level of the configuration file:

```
default authorization = permit
```

- At the user level, inside the braces of the user declaration, the default for a user who does not have a service or command explicitly authorized is to deny that service or command. To permit it:

```
default service = permit
```

- At the service authorization level, arguments are processed according to the following algorithm: For each AV pair sent from the access server, the following process occurs:
 - (a) If the AV pair from the access server is mandatory, look for an exact match in the daemon's mandatory list. If found, add the AV pair to the output.
 - (b) If an exact match doesn't exist, look in the daemon's optional list for the first attribute match. If found, add the access server AV pair to the output.
 - (c) If no attribute match exists, deny the command if the default is to deny, or if the default is permit, add the access server AV pair to the output.
 - (d) If the AV pair from the access server is optional, look for an exact attribute, value match in the mandatory list. If found, add the daemon's AV pair to output.
 - (e) If not found, look for the first attribute match in the mandatory list. If found, add daemon's AV pair to output.
 - (f) If no mandatory match exists, look for an exact attribute, value pair match among the daemon's optional AV pairs. If found add the daemon's matching AV pair to the output.
 - (g) If no exact match exists, locate the first attribute match among the daemon's optional AV pairs. If found add the daemon's matching AV pair to the output.

- (h) If no match is found, delete the AV pair if default is deny, or if the default is permit, add the access server AV pair to the output.
- (i) If there is no attribute match already in the output list after all AV pairs have been processed for each mandatory daemon AV pair, add the AV pair (add only one AV pair for each mandatory attribute).

Configuring Authorization (Network or EXEC) on the Access Server

To specify network authorization, which means that you are preventing unauthorized users from accessing network resources, issue the **aaa authorization network** command. To restrict users from logging into the EXEC facility, issue the **aaa authorization exec** command. See the following example:

```
5200(config)# aaa authorization network
5200(config)# aaa authorization exec
```

Note You can also require authorization before a user can issue specific commands by using the **aaa authorization** command. For more information, refer to the *Security Configuration Guide*, which is part of the Cisco IOS configuration guides and command references.

Specifying the Authorization Method

Authorization methods are defined as optional keywords in the **aaa authorization** command. You can specify any of the authorization methods listed in Table 4-7 for both network and EXEC authorization.

Configuring Authorization

Table 4-7 Authorization Methods

Authorization Methods	Purpose
if-authenticated	User is authorized if already authenticated.
local	Uses the local database for authorization. The local database is created using the username privilege command to assign users to a privilege level from 0 to 15 and the privilege level command to assign commands to these different levels.
none	Authorization always succeeds.
radius	Uses RADIUS authorization as defined on a RADIUS server.
tacacs+	Uses TACACS+ authorization as defined on a TACACS+ server.

Specifying Authorization Parameters on a TACACS+ Server

When you configure authorization, you must ensure that the parameters established on the access server correspond with those set on the TACACS+ server.

Authorization Examples

The following example uses a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or has no information about a user, no authorization is performed, and the user can use all network services:

```
5200(config)# aaa authorization network tacacs+ none
```

The following example permits the user to run the EXEC process if the user is already authenticated. If the user is not already authenticated, the Cisco IOS software defers to a RADIUS server for authorization information:

```
5200(config)# aaa authorization exec if-authenticated radius
```

The following example configures network authorization. If the TACACS+ server does not respond or has no information about the username being authorized, the RADIUS server is polled for authorization information for the user. If the RADIUS server does not respond, the user still can access all network resources without authorization requirements.

```
5200(config)# aaa authorization network tacacs+ radius none
```

Security Examples

This series of examples shows complete security configuration components of a configuration file on an access server. Each of these examples shows authentication and authorization.

Simple Local Security Example

This sample configuration uses AAA to configure default authentication using a local security database on the access server. All lines and interfaces have the default authentication lists applied. Users aaaa, bbbb, and cccc have been assigned privilege level 7, which prevents them from issuing the **ppp arap**, and **slip** commands, because these commands have been assigned to privilege level 8.

```
aaa new-model
aaa authentication login default local
aaa authentication arap default local
aaa authentication ppp default local
aaa authorization exec local
aaa authorization network local
aaa authorization
!
username aaaa privilege exec level 7 privilege network level 8 password 7
095E470B1110
username bbbb privilege network level 7 password 7 0215055500070C294D
username cccc privilege network level 7 password 7 095E4F10140A1916
!
privilege exec level 8 ppp
privilege exec level 8 arap
privilege exec level 8 slip

line console 0
 login authentication default
!
```

Security Examples

```
line 1 48
  arap authentication default
!
interface Group-Async1
  ppp authentication chap default
  group-range 1 48
```

With this configuration, the sign-on dialog from a remote PC appears as follows:

```
atdt5551234
CONNECT 14400/ARQ/V32/LAPM/V42BIS
User Access Verification
Username: aaaa
Password:
5200> enable
Password:
5200#
```

TACACS+ Security Example for Login, PPP, and ARA

The following example shows how to create and apply the following authentication lists:

- A TACACS+ server named dog-house is polled for authentication information (so you do not need to define a local username database). The shared key between the access server and the TACACS+ security server is shepard4.
- A login authentication list named rtp2-office is created, then applied to the console port.
- A PPP authentication list named marketing is created, then applied to group async interface 0, which includes asynchronous interfaces 1 to 48.
- An ARA list named los-banos-office is created and applied to lines 1 to 48.

Note The authentication method lists used in this example use names other than default. However, you generally specify **default** as the list name for most lines and interfaces, and apply different named lists on an exception basis. These names are used only for illustrative purposes.

```
hostname 5200
!
tacacs-server host dog-house
tacacs-server key shepard4
!
aaa authentication login rtp2-office tacacs+
aaa authentication ppp marketing if-needed tacacs+
aaa authentication arap los-banos-office tacacs+
!
line console0
  login authentication rtp2-office
!
interface group-async0
  ppp authentication chap marketing
  group-range 1 48
!
line 1 48
  arap authentication los-banos-office RADIUS Example for Login and PPP
```

The following example shows how to create the following authentication lists:

- A RADIUS server named pig-pen is polled for authentication information (so you do not need to define a local username database). The shared key between the access server and the RADIUS security server is BaBe218.
- A login authentication list named fly is created, then applied to all lines that users can log in to, except the console port. In this example, the console port is physically secure and does not need password protection. The access server is locked in a closet and secured behind a deadbolt lock.
- A PPP authentication list maaaa is created, then applied to group async interface 658, which includes asynchronous interfaces 1 to 48. CHAP authentication is used, because it is more secure than PAP.

```
radius-server host pig-pen
radius-server key BaBe218
!
privilege exec level 14 configure
privilege exec level 14 reload
privilege exec level 8 arap
privilege exec level 8 ppp
!
```

Security Examples

```
aaa authentication login fly radius
aaa authentication ppp maaaa if-needed radius
aaa authorization network radius
aaa authorization exec radius
!
line 1 54
  login authentication fly
!
interface group-async658
  ppp authentication chap maaaa
  group-range 1 48
```