10

AAA Configuration

Terms you'll need to understand:

✓ AAA
✓ Cisco Secure Access Control Server (CSACS)
✓ TACACS+
✓ RADIUS
✓ Downloadable access control lists
✓ Cut-through proxy
✓ Virtual Telnet
✓ Virtual HTTP

Techniques you'll need to master:

✓ Configuring AAA services on the PIX
✓ Using the **group** tag
✓ Authenticating with RADIUS and TACACS+
✓ Using the **privilege** command
✓ Using named and unnamed downloadable ACLs

This chapter covers the powerful *triple A (AAA)* features the PIX firewall supports. Authentication, authorization, and accounting provide powerful control over who can access the network or the PIX and the capability to record when it happens. The capability to have downloadable access control lists (ACLs) gives PIX administrators limitless flexibility and granular control over users by restricting what they can and cannot do.

Introduction to AAA Services

In any Cisco environment, several features can provide administration overhead issues on your devices. The first administration feature your devices have is the enable or privilege exec password. This provides access to all the configuration commands on a device; however, managing this password could become an issue. For example, if Jack supports 300 Cisco routers and firewalls, trying to keep the enable password the same or changing it on all the devices could be an overwhelming task.

A second feature of Cisco devices is the capability to create several levels of access on the device. For example, if you have several different network engineers, each engineer can be restricted to a subset of commands and not the entire privilege exec list. However, if Jack had to create the same users on each of his 300 devices, it could take several days to configure and maintain the systems.

A third feature is providing access through devices. The PIX supports several features, such as cut-through proxy, that can require a user to authenticate before Internet access is granted. Again, if Jack had to manage 1,000 employee usernames and passwords on multiple PIX firewalls, he would have to spend all of his time managing and changing passwords.

AAA services help administrators manage Cisco devices by offloading authentication, authorization, and accounting tasks from local devices to centrally located servers. These servers can contain databases of users, passwords, dynamic ACLs, authorization settings, and account tracking features, just to name a few. Using central servers in the example with Jack, he could manage the central server username list in one location and just point all 300 devices to that location for AAA services.

Figure 10.1 displays two of these locations for usernames and passwords. One is called local and resides within the PIX configuration file; the other is called Cisco Secure ACS and is a user database located on a AAA server. The third place where users can exist, though not shown in Figure 10.1, is in external user databases such as Microsoft Windows.

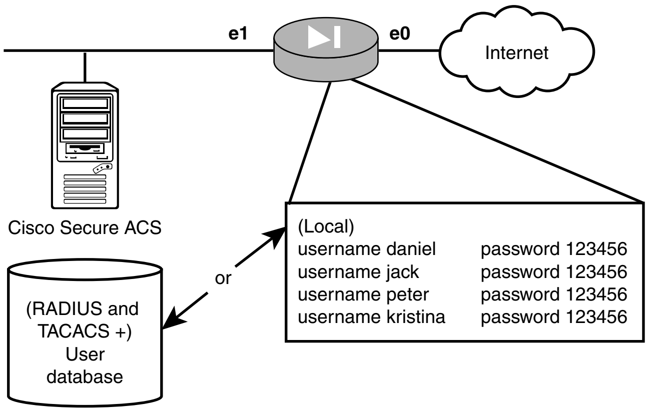

e1
e0
Internet
Cisco Secure ACS
(RADIUS and TACACS +) User database
or
(Local)
username daniel password 123456
username jack password 123456
username peter password 123456
username kristina password 123456

Figure 10.1 Username locations.

Authentication

Authentication is the process of validating a username and password. The PIX can check the local database or a remote AAA server database for valid usernames and passwords. After the user establishes positive authentication, authorization is the next step.

Authorization

Successful authentication is required before authorization can occur. *Authorization* defines what a user can and cannot do. For example, a dynamic ACL can be downloaded to the PIX, restricting a user to or from particular networks.

Accounting

Accounting is the feature that enables administrators to keep track of what their users do. For example, when a user logs in to the system, a log entry can be made.

AAA Server Protocols

When using AAA services, requests can be sent to remote AAA servers for authentication, authorization, and accounting. Cisco supports two main protocols for these requests: RADIUS and TACACS+. The request is sent to the servers, and the responses are used to allow the users into or out of the device. For example, in Figure 10.2 the PIX is configured to authenticate users before entering a privileged exec mode. The request is sent to the AAA server using RADIUS or TACACS+. Next, the AAA server authenticates the user either with its own database or, as shown in Figure 10.2, another database. After authentication is approved, authorization is checked and the responses are sent back to the PIX. Throughout all these transactions, accounting is working in the background logging and tracking user actions.

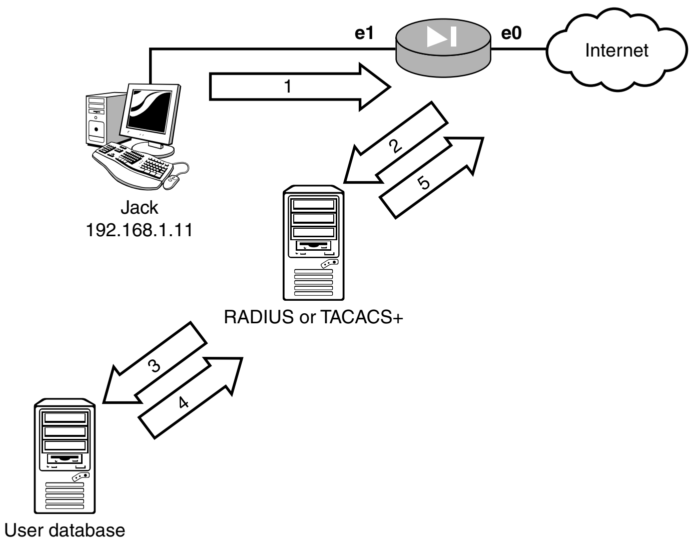


Figure 10.2 AAA services transaction.

AAA stands for authentication, authorization, and accounting. You cannot have authorization without successful authentication first.

Remote Access Dial-in User Service

The Remote Access Dial-in User Service (RADIUS) protocol was originally developed by Livingston Enterprises, Inc., as an access protocol. This protocol provides authentication and accounting services and can be used by just about any size network or vendor. The protocol is a client/server configuration, and the PIX devices are the clients and the AAA server would be the RADIUS server itself. The protocol uses a UDP connection and encrypts only the password and leaves the username in clear text.

Terminal Access Controller Access Control System Plus

Terminal Access Controller Access Control System (TACACS) was originally created by the U.S. government and is an open standard security protocol. Cisco uses a modified version of TACACS called Terminal Access Controller Access Control System Plus (TACACS+). In contrast to RADIUS, which uses UDP, the TACACS+ protocol provides a reliable TCP connection between the client and the server for AAA service requests. These requests are more secure than RADIUS because the body of the transaction is always encrypted.

TIP

For a detailed comparison of RADIUS and TACACS+, see Cisco's Web site at **www.cisco.com/warp/public/480/10.html**.

EXAM ALERT

TACACS+ uses TCP port 49 for connections between AAA servers and clients, whereas RADIUS uses UDP port 1812 for authentication and UDP port 1813 for accounting.

Supported AAA Servers

The Cisco PIX firewall can support several AAA servers. Most third-party AAA servers support the RADIUS protocol, making installations in multivendor environments very flexible. The following is a list of some supported AAA servers:

- Cisco Secure ACS for Windows
- Cisco Secure ACS for Unix
- Livingston
- Merit

Cisco Secure Access Control Server

The Cisco Secure Access Control Server (CSACS) is Cisco's AAA server that supports both the RADIUS and TACACS+ protocols. The software provides centralized AAA services for AAA clients such as the PIX firewall. It is also very scalable, with the option to use its own user database or connect to an external user database, such as one of these:

➤ Axent token server

➤ Generic LDAP

➤ Novell NDS

➤ SafeWord token server

➤ Windows NT/2000 local or domain controller

Installing CSACS

The CSACS can be installed onto Unix or Microsoft Windows Server. Cisco uses a Web page front-end to configure the system. The following are some of the Windows requirements:

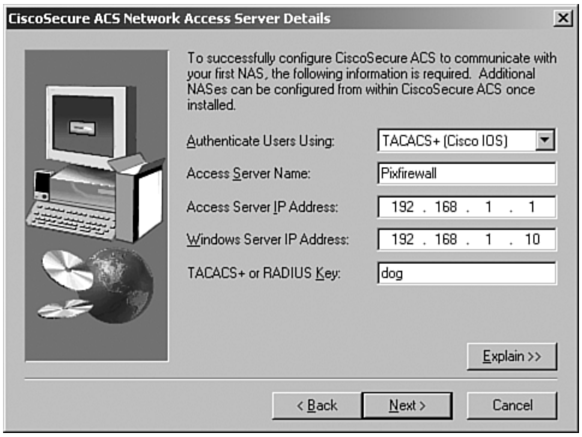
➤ Pentium III processor with 550MHz or better

➤ 256MB of RAM

➤ 250MB of available disk space

➤ Windows 2000 with SP1 or Windows NT with SP6a

During the installation, the software asks for at least one network access server (NAS) to be set up. A NAS is an AAA client, and in this case it's the PIX firewall (see Figure 10.3). CSACS can support up to 2,000 AAA clients.



To successfully configure CiscoSecure ACS to communicate with your first NAS, the following information is required. Additional NASes can be configured from within CiscoSecure ACS once installed.

Authenticate Users Using: TACACS+ (Cisco IOS)
Access Server Name: Pixfirewall
Access Server IP Address: 192 . 168 . 1 . 1
Windows Server IP Address: 192 . 168 . 1 . 10
TACACS+ or RADIUS Key: dog

Figure 10.3 The CSACS NAS dialog box.

During installation, the NAS's details dialog box has a prompt that states **Access Server IP Address**. This is the address of the PIX firewall or other network access server that will be using the CSACS server.

Cisco supports several solutions for CSACS. The latest version is CSACS for Windows 3.2 and v2.3 for Unix. CSACS may also be purchased in a 1 RU hardware solution called the CSCAS Solution Engine. More information can be found in the "Need to Know More?" at the end of the chapter.

Configuring AAA Services

Several steps are involved in configuring a system for AAA services. The next section walks you through some of the steps needed to configure a system for AAA services, but here is an overview of the basic steps:

1. Configure AAA usernames and passwords.
2. Configure AAA server locations.
3. Configure authentication.
4. Configure authorization (optional).
5. Configure accounting (optional).

Configuring AAA Usernames

The first step when configuring AAA services is to create usernames and passwords. As stated previously, three main locations exist for usernames and passwords—local, an AAA server, or an external database. However, in this chapter we discuss only local and CSACS.

Local Username and Password Commands

The PIX enables you to create a username and password list inside the configuration file. This list can be referenced by the AAA commands when users log in. The syntax used to accomplish this is shown here:

```
pixfirewall(config)# username <username> {nopassword¦password
              <password>}[privilege <level>]
```

Table 10.1 displays the possible options for the username command.

Table 10.1 username Command Options

Option	Function
username	The name of the user.
nopassword|password	The **nopassword** option specifies that no password is necessary. The **password** option specifies the password for the username.
privilege	This is the level of access you want to give the user. The default is **2**, and level 15 is for the privileged exec access level.

Listing 10.1 clears all users from the PIX, creates four users, and shows all the users that exist inside the PIX configuration file.

Listing 10.1 Configuring Username and Passwords

```
pixfirewall(config)# clear user
pixfirewall(config)# username daniel password 1234 privilege 15
pixfirewall(config)# username kristina password 1234 privilege 15
pixfirewall(config)# username jack password 1234
pixfirewall(config)# username peter password 1234
pixfirewall(config)# show user
username daniel password fOpmsyD0svtnnlr/ encrypted privilege 15
username peter password GCbahPTC/hAylaFE encrypted privilege 2
username kristina password wTsW2QC6pRXaJrTT encrypted privilege 15
username jack password 5zVJhD5YUb6zO7VH encrypted privilege 2
```

Adding Users with CSACS

To add a user to the CSACS database, you must open the user Web interface. First, you turn on the TACACS+ advanced features so you can configure privileged exec access on your user accounts. The steps to enable these advanced features are as follows:

1. On the left menu, click the Interface Configuration button.
2. Click the TACACS+ (Cisco IOS) link.
3. Scroll down until you find the Advanced Configuration Options section and place a check in the box labeled Advanced TACACS+ Features.
4. Click the Submit button at the bottom of the screen to enable the features.

The following are the steps to create a user account in Cisco Secure ACS:

1. Click the User Setup button from the menu list on the left side.
2. Enter the username **`daniel`** into the User: box and click the Add/Edit button.

3. Scroll down to the User Setup section and enter a password of **123456789**.
4. Now scroll down to the Advanced TACACS+ Settings section and select the Max Privilege for Any AAA Client radio button option. Then use the pull-down menu to select level 15. This will provide privilege-level access to the user named `daniel`.
5. Just below the advanced TACACS+ settings is a section called TACACS+ Enable Password. Select Use CiscoSecure PAP Password. This will use the password you previously set; alternatively, you could set a separate enable password if you wanted.
6. Click the Submit button to save the user.

Now that the usernames have been created, you are ready to configure AAA services on the PIX firewall.

Configuring the aaa-server Command

The `aaa-server` command is used to configure group tags that define where the AAA server is located. If the AAA server is remote, either RADIUS or TACACS+ security protocols can be used to request the remotely located AAA services. However, if the AAA server is local, no security protocol is required to communicate with the internal PIX configuration file. Table 10.2 describes the three possible locations and protocol used to acquire AAA services.

Table 10.2 aaa-server Locations

Locations	Description
Local	This tells the PIX to look locally in the PIX configuration file for the usernames and passwords.
RADIUS	This configures the PIX to use a RADIUS security protocol and to request remotely located AAA services.
TACACS+	This configures the PIX to use a TACACS+ security protocol and to request remotely located AAA services.

The following is the syntax of the `aaa-server` command:

```
pixfirewall(config)# [no] aaa-server <group_tag> protocol tacacs+¦radius
pixfirewall(config)# [no] aaa-server <group_tag> [<(if_name)>]
               host <ip_address> [<key>]
```

Table 10.3 displays the possible options for the `aaa-server` command.

Table 10.3 aaa-server Command Options

Option	Function
group_tag	This option is a grouping of server settings that can be referenced by all the AAA commands. You can create up to 14 group tags.
if_name	This is the interface where the AAA server is located.
ip_address	This is the IP address of the AAA server.
key	This is the key value used on the TACACS+ server to encrypt the data between the client and server.
protocol	This option defines one of the three protocol locations: local, RADIUS, or TACACS+.

The following example displays the current `aaa-server` group tags and then creates a new group tag called `PIXAuth`. This new group tag configures TACACS+ as the security protocol and 192.168.1.10 as the AAA server:

```
pixfirewall(config)# show aaa-server
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local

pixfirewall(config)# aaa-server PIXAuth protocol tacacs+
pixfirewall(config)# aaa-server PIXAuth (inside) host 192.168.1.10 dog
```

Configuring Authentication

AAA authentication can be used to control access into the PIX console, privileged exec mode, and access through the PIX. This section covers console access and access technologies used to control traffic through the PIX.

Authentication and Console Access

Console access requires a username and password. Access methods for AAA authentication to the PIX can be via serial, Telnet, SSH, HTTP, and privileged exec mode. The `aaa authentication` command syntax is as follows:

```
pixfirewall(config)# [no] aaa authentication serial¦telnet¦ssh¦http¦enable
               console <group_tag>
```

Table 10.4 displays the possible options for the `aaa-authentication` command.

Table 10.4 aaa authentication Options

Option	Description
serial	This causes the user to be prompted to enter a username and password when connecting using the serial port.
telnet	This requires Telnet connections to enter a username and password before entering the PIX console.
ssh	When users SSH into the PIX, AAA authentication is required.
http	This option is used when connecting to the PIX using the PDM Web interface, requiring AAA logon.
enable	Before a user can enter the privileged exec mode, a username and password must be supplied.
group_tag	The tag option defines the group name to associate the command with.

The following example displays the commands needed to configure the PIX firewall to use a TACACS+ server (192.168.1.10) for console authentication. As users connect via Telnet or enter privileged exec mode, they will be prompted to enter a username and password before allowing them to proceed:

```
pixfirewall(config)# aaa-server PIXAuth protocol tacacs+
pixfirewall(config)# aaa-server PIXAuth (inside) host 192.168.1.10 dog
pixfirewall(config)# aaa authentication enable console PIXAuth
pixfirewall(config)# aaa authentication telnet console PIXAuth
```

Authorization and Console Access Commands

Console access can be controlled by the AAA authentication, whereas console commands can be controlled by using AAA authorization. Controlling command access enables the PIX to contain several levels of users, some with full command access (level 15) and others with specifically allocated commands.

The `privilege` command is used to associate a specific command with a level of access. The command syntax is shown here:

```
pixfirewall(config)# [no] privilege [{show ¦ clear ¦ configure}]
               level <level> [mode {enable¦configure}] command <command>
```

Table 10.5 displays the possible options for the `privilege command`.

Table 10.5 privilege command Options

Option	Function
show|clear|configure	These options enable you to specify the type of command you want to put in the list.
level	This option is used for the level of the list.
mode	This defines the area where the command is being restricted: enable mode or configure mode. This defines the command related to the first **show|clear|configure** option.

The `privilege` command shown in Listing 10.2 creates two users with different levels of access. Commands designated to access levels 11 and 12 are then assigned.

Listing 10.2 Creating Users and Setting Privilege Mode Commands

```
pixfirewall(config)# username jimmy password 123456 privilege 11
pixfirewall(config)# username richard password 123456 privilege 12

pixfirewall(config)# privilege show level 11 command access-list
pixfirewall(config)# privilege show level 11 command running-config
pixfirewall(config)# privilege show level 12 command interface

pixfirewall(config)# aaa authorization command LOCAL
```

In Listing 10.2, because Jimmy has privilege level 11, he will be able to execute any command assigned to level 11 and below. Richard, on the other hand, has a level that is higher and will therefore be able to execute any command associated with level 12 and below. The last `aaa authorization command LOCAL` designates the local username and password list for command authorization.

Authorization for commands can also be set up using the CSACS Web interface by clicking the Group Setup button, selecting the group options, and setting the commands.

The **privilege** command enables you to associate a command with an access level.

Authentication for Cut-through Proxy

Using cut-through proxy enables you to control HTTP, FTP, and Telnet services through the PIX firewall. Access lists provide general packet filtering, whereas cut-through proxy requires a username and password before allowing access. For example, if Jack wants to control which users can use HTTP through the PIX, he could implement cut-through proxy and prompt HTTP traffic with a username and password dialog box, as shown in Figure 10.4. This username and password would be forwarded to the AAA server for authentication and authorization using TACACS+ or RADIUS security protocols.

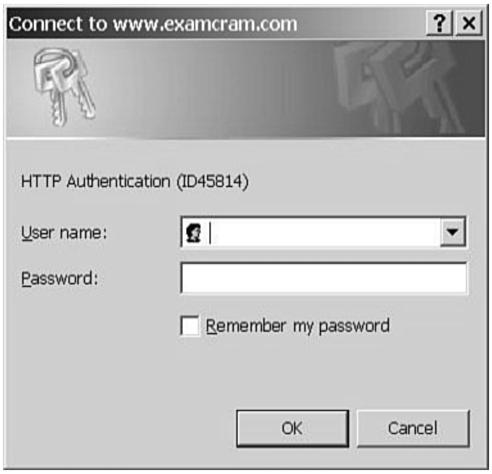


Figure 10.4 The cut-through proxy HTTP dialog box.

NOTE

Cut-through proxy works with FTP, HTTP, and Telnet.

The following shows two `aaa authentication` commands that can be used to implement cut-through proxy:

```
pixfirewall(config)# [no] authentication include¦exclude <service>
                          inbound¦outbound <if_name>
                          <internal_ip> <internal_mask>
                          [<external_ip> <external_mask>] <group_tag>

pixfirewall(config)# [no] aaa authentication match <access_list_name>
                          inbound¦outbound <if_name> <if_name> <group_tag>
```

Table 10.6 displays the possible options for the `aaa authentication` command.

Table 10.6 Cut-through Proxy aaa authentication Command Options

Option	Function
include|exclude	The **include** option is used to create a new rule, whereas the **exclude** option is used to create an exception to an **include** statement.
service	This option states the type of service to include or exclude. The following options are valid: **any**, **ftp**, **http**, and **telnet**.
if_name	This is the interface where the AAA server is located.
internal_ip internal_mask	This defines what internal traffic IP addresses are included. Using **0 0** defines all IP addresses.
external_ip external_mask	This defines what external traffic IP addresses are included. Using **0 0** defines all IP addresses.
group_tag	The tag option defines the group name to associate the command with.
match	This command works in conjunction with an access list. This allows you to use an access-list to define traffic.

The example shown here enables cut-through proxy authentication on the inside interface:

```
pixfirewall(config)# aaa-server PIXAuth protocol tacacs+
pixfirewall(config)# aaa-server PIXAuth (inside) host 192.168.1.10 dog
pixfirewall(config)# aaa authentication include http
               outbound 0 0 0 0 PIXAuth
```

NOTE

Cut-through proxy can be used to authenticate both users in the inside interface going out and users coming in on the outside interface.

Authorization for Cut-through Proxy

After cut-through proxy authentication is configured, users are automatically allowed to pass through the firewall. However, authorization can be added to further control where your users are allowed to go. The commands used are similar to the `authentication` commands; the command syntax for authorization is as follows:

```
pixfirewall(config)# [no] authorization include¦exclude <service>
                          inbound¦outbound <if_name>
                          <internal_ip> <internal_mask>
                          [<external_ip> <external_mask>] <group_tag>

pixfirewall(config)# [no] aaa authorization match <access_list_name>
                          inbound¦outbound <if_name> <if_name> <group_tag>
```

Configuring Accounting

After authentication and authorization have been configured, configuration of accounting is often the next step. Accounting information enables you to track users who have logged on and accessed the device and the amount of time they're logged on. These two commands enable the accounting process:

```
pixfirewall(config)# [no] accounting include¦exclude <service>
                          inbound¦outbound <if_name>
                          <internal_ip> <internal_mask>
                          [<external_ip> <external_mask>] <group_tag>

pixfirewall(config)# [no] aaa accounting match <access_list_name>
                          inbound¦outbound <if_name> <if_name> <group_tag>
```

The format of the commands is similar to the authorization and authentication commands. The `include¦exclude` parameters define the connections that need accounting, whereas the `match` parameter uses an ACL to define who needs accounting. The `service` parameter specifies the connection, such as `any`, `ftp`, `http`, `telnet`, or a protocol/port number.

The following is an example of enabling accounting for all internal traffic:

```
pixfirewall(config)# aaa accounting include any inbound 0 0 0 0 PIXAuth
pixfirewall(config)# aaa accounting include any outbound 0 0 0 0 PIXAuth
```

You do not need to configure any settings on the CSACS server itself; accounting requests should automatically be accepted. However, you should verify that traffic is being accounted by clicking the Reports and Activity button and the TACACS+ Accounting link.

Downloadable Access Control Lists

The PIX and CSACS support the capability to use downloadable ACLs, allowing you to create ACLs that are downloaded for a specific user or groups of users. The ACL can be downloaded during the authentication phase of a RADIUS connection, but TACACS+ does not support this feature. There are two types of downloadable access lists; named and unnamed.

EXAM ALERT

Downloadable ACLs are supported only on RADIUS and not TACACS+.

Named ACL

Named ACL gives you the ability to name an ACL that is downloaded once to the PIX and shared between many users. If a newer ACL is on the server, the newer version is downloaded and shared among the users assigned to the named ACL. Named ACLs work best for several users who all need the same ACL control and when several PIX access servers need that same ACL list.

The following is an example of a downloaded named ACL:

```
pixfirewall(config)# show access-list
access-list #ACSACL#-PIX-MySharedACL-3ef2957b; 3 elements
access-list #ACSACL#-PIX-MySharedACL-3ef2957b deny tcp any
               host 10.0.0.2 eq ftp (hitcnt=0)
access-list #ACSACL#-PIX-MySharedACL-3ef2957b deny tcp any
               host 10.0.0.3 eq www (hitcnt=0)
access-list #ACSACL#-PIX-MySharedACL-3ef2957b deny tcp any
               host 10.0.0.4 eq telnet (hitcnt=0)
```

The name of the list in the previous example is `MySharedACL`, and this list will be shared for all users who have been assigned the `MySharedACL` on the CSACS. The two tasks to configure named downloadable ACL within CSACS should be included here:

1. From Shared Profile Components in CSACS, define the named downloadable ACL.
2. From User Setup, apply the downloadable ACL to the corresponding users.

Unnamed ACLs

Unnamed ACLs are used to specify ACLs for individual users. The list created is used only by a single user, as opposed to a named ACL, which is shared. These lists are recommended only if each user requires an individual list.

An example of a downloaded unnamed ACL is shown here:

```
pixfirewall(config)# show access-list
access-list  AAA-user-daniel; 1 elements
access-list  AAA-user-daniel deny tcp any any eq www (hitcnt=0)
```

In the previous unnamed ACL example, an ACL is downloaded for a user named `daniel` and is used only by `daniel`.

Named ACLs are shared among several users and are downloaded only once during authentication. Unnamed ACLs are not shared and are downloaded during authentication.

Authentication of Other Services and Authentication Issues

Normally, services such as HTTP, FTP, and Telnet can be authenticated using cut-through proxy. However, other services might need access through the PIX firewall. For example, if Jack's users need to access TFTP servers or Microsoft servers using NetBIOS, ports 69 and 139 would be used. Cut-through proxy does not work in these cases. You do have can use virtual telnet to allow users through. Also cut-through proxy may have some authentication issues with HTTP and Web browsers. If this is a problem the another service called Virtual HTTP can be used. This and Virtual Telnet are covered in the next section.

Virtual Telnet

Virtual Telnet enables users to preauthenticate using a virtual Telnet session before executing the application that needs to pass through the PIX. For example, when Jack needs TFTP access, he must first open a Telnet session with PIX to a virtual Telnet IP address and then enter his username and password. The PIX caches the successful user logon and allows TFTP traffic through.

To use virtual Telnet, the `virtual telnet` command is needed. Its syntax is as follows:

```
pixfirewall(config)# [no] virtual telnet <ip>
```

The `ip` option is the IP address of the virtual Telnet server running on the PIX firewall. This address is the IP address clients use to enter their usernames and passwords. After authentication takes place, the user is allowed to pass traffic through the PIX. To log out, the user only has to connect using Telnet again and reenter her username and password.

The example shown here requires TFTP traffic to be authenticated before TFTP traffic is allowed through the PIX firewall. The user will create a Telnet session with 192.168.1.252:

```
pixfirewall(config)# aaa-server PIXAuth protocol tacacs+
pixfirewall(config)# aaa-server PIXAuth host 192.168.1.10 dog
pixfirewall(config)# aaa authentication include tcp/69
```

```
               outbound 0 0 0 0 PIXAuth
pixfirewall(config)# virtual telnet 192.168.1.252
```

Virtual Telnet allows support for applications that don't use the typical HTTP, Telnet, and FTP ports.

Virtual HTTP

Virtual HTTP enables browser and Web server authentication to work correctly with the PIX when authentication with cut-through proxy is problematic. Web browsers can cache authentication requests, potentially causing future authentication problems. The Virtual HTTP works by redirecting the user's initial internal Web server request to a virtual HTTP server on the PIX. The user then authenticates his username and password and is redirected back to the original URL.

This example creates a virtual HTTP server that is used to catch HTTP traffic:

```
pixfirewall(config)# aaa-server PIXAuth protocol tacacs+
pixfirewall(config)# aaa-server PIXAuth host 192.168.1.10 dog
pixfirewall(config)# aaa authentication include any inbound 0 0 0 0 PIXAuth
pixfirewall(config)# virtual http 192.168.1.251
```

Virtual HTTP helps to correct Web browser problems with HTTP authentication.

General AAA Commands

Several useful commands can help you view and confirm that your AAA services are configured correctly. Table 10.7 displays some of these.

Table 10.7 General AAA Commands

Command	Description
show aaa	This command displays all the currently configured AAA authentication, authorization, and accounting commands.
clear aaa	This deletes all the configured AAA authentication, authorization, and accounting commands.
show aaa-server	This command displays all the configured AAA servers.
show uauth	The output of this command displays the usernames and IP addresses of users who are currently logged in.

Table 10.7 General AAA Commands *(continued)*

Command	Description
clear uauth	This can remove a single user or all authentication users currently logged in.

Authentication Prompts

The authentication `prompt` command enables you to modify login prompts during AAA authentication. This command configures text for accepted, rejected, and basic prompts, and its syntax is as follows:

```
pixfirewall(config)# [no ¦ clear] auth-prompt [prompt ¦ accept ¦ reject]
               "<prompt text>"
```

Table 10.8 displays the possible options for the `auth-prompt` command.

Table 10.8 auth-prompt Command Options

Option	Function
prompt	After this option, use quotes around the text you want to display to the user during general AAA logon attempts.
accept	This defines the accepted text after authentication is successful.
reject	This defines the text displayed after failed authentication attempts.

Here are some basic examples of setting the prompts:

```
pixfirewall(config)# auth-prompt prompt "AUTHORIZED PERSONNEL ONLY"
pixfirewall(config)# auth-prompt reject "WRONG"
pixfirewall(config)# auth-prompt accept "Welcome to the PIX firewall"
```

Authentication Timeouts

AAA authentication connections support two timeouts: inactivity and absolute. The *inactivity* timeout is used to disconnect the connection when the user is idle or inactive. The *absolute* timeout sets the total duration that the user is allowed to be logged in. Here is the command syntax:

```
pixfirewall(config)# timeout uauth hh:mm:ss [absolute¦inactivity]
```

After timeouts are set, the `show timeout` command can be used to display all the values for the `timeout` command. The output of the command is shown here:

```
pixfirewall(config)# show timeout
timeout uauth 0:05:00 absolute
```