

# PIX 6.2 : Authentication and Authorization Command Configuration Example

Document ID: 22923

## Contents

### Introduction

#### Before You Begin

- Conventions

- Prerequisites

- Components Used

#### Testing Prior to Adding Authentication/Authorization

#### Understanding Privilege Settings

#### Authentication/Authorization – Local Usernames

#### Authentication/Authorization with an AAA Server

- ACS – TACACS+

- CSUnix – TACACS+

- ACS – RADIUS

- CSUnix – RADIUS

#### Network Access Restrictions

#### Debug

#### Accounting

#### Information to Collect if You Open a TAC Case

#### Related Information

## Introduction

PIX command authorization and expansion of local authentication was introduced in version 6.2. This document provides an example of how to set this up on a PIX. Previously available authentication features are still available but not discussed in this document (for example, Secure Shell (SSH), IPsec client connection from a PC, and so on). The commands performed may be controlled locally on the PIX or remotely through TACACS+. RADIUS command authorization is not supported; this is a limitation of the RADIUS protocol.

Local command authorization is done by assigning commands and users to privilege levels.

Remote command authorization is done through a TACACS+ authentication, authorization, and accounting (AAA) server. Multiple AAA servers can be defined in the event that one is unreachable.

Authentication also works with previously configured IPsec and SSH connections. SSH authentication requires that you issue this command:

```
aaa authentication ssh console <LOCAL | server_tag>
```

**Note:** If you use a TACACS+ or RADIUS server group for authentication, you can configure the PIX to use the local database as a **FALLBACK** Method if the AAA server is unavailable.

For Example

```
pix(config)#aaa authentication ssh console TACACS+ LOCAL
```

You can alternatively use the local database as your main method of authentication (with no fallback) if you

enter LOCAL alone.

For example, issue this command in order to define a user account in the local database and to perform local authentication for an SSH connection:

```
pix(config)#aaa authentication ssh console LOCAL
```

Refer to [How To Perform Authentication and Enabling on the Cisco Secure PIX Firewall \(5.2 Through 6.2\)](#) for more information on how to create AAA–authenticated access to a PIX Firewall that runs PIX Software version 5.2 through 6.2 and for more information about enable authentication, syslogging, and gaining access when the AAA server is down.

Refer to [PIX/ASA : Cut–through Proxy for Network Access using TACACS+ and RADIUS Server Configuration Example](#) for more information on how to create AAA–authenticated (Cut–through Proxy) access to a PIX Firewall that runs PIX Software versions 6.3 and later.

If the configuration is done properly, you should not be locked out of the PIX. If the configuration is not saved, rebooting the PIX should return it to its pre–configuration state. If the PIX is not accessible due to misconfiguration, refer to [Password Recovery and AAA Configuration Recovery Procedure for PIX](#).

## Before You Begin

### Conventions

For more information on document conventions, see the [Cisco Technical Tips Conventions](#).

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

The information in this document is based on these software and hardware versions:

- PIX Software version 6.2
- Cisco Secure ACS for Windows version 3.0 (ACS)
- Cisco Secure ACS for UNIX (CSUnix) version 2.3.6

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Testing Prior to Adding Authentication/Authorization

Prior to implementing the new 6.2 authentication/authorization features, make sure that you are currently able to gain access to the PIX using these commands:

```
!--- IP range allowed to Telnet to the PIX (values depend on network).
```

```
telnet 172.18.124.0 255.255.255.0
```

```
!--- Telnet password.
```

```
passwd <password>
```

```
!--- Enable password.
```

```
enable password <password>
```

## Understanding Privilege Settings

Most commands in the PIX are at level 15, although a few are at level 0. To view current settings for all commands, use this command:

```
show privilege all
```

Most commands are at level 15 by default, as shown in this example:

```
privilege configure level 15 command route
```

A few commands are at level 0, as shown in this example:

```
privilege show level 0 command curpriv
```

The PIX can operate in enable and configure modes. Some commands, such as **show logging**, are available in both modes. To set privileges on these commands, you must specify the mode that the command exists in, as shown in the example. The other mode option is **enable**. You get the logging is a command available in multiple modes error message. If you do not configure the mode, use the **mode [enable|configure]** command:

```
privilege show level 5 mode configure command logging
```

These examples address the **clock** command. Use this command to determine the current settings for the **clock** command:

```
show privilege command clock
```

The output of the **show privilege command clock** command shows that the **clock** command exists in these three formats:

```
!--- Users at level 15 can use the show clock command.
```

```
privilege show level 15 command clock
```

```
!--- Users at level 15 can use the clear clock command.
```

```
Privilege clear level 15 command clock
```

```
!--- Users at level 15 can configure the clock
```

```
!--- (for example, clock set 12:00:00 Jan 01 2001).
```

```
privilege configure level 15 command clock
```

## Authentication/Authorization – Local Usernames

Before changing the privilege level of the **clock** command, you should go to the console port to configure an administrative user and turn on LOCAL login authentication, as shown in this example:

```
GOSS(config)# username poweruser password poweruser privilege 15
GOSS(config)# aaa-server LOCAL protocol local
GOSS(config)# aaa authentication telnet console LOCAL
```

The PIX confirms the addition of the user, as shown in this example:

```
GOSS(config)# 502101: New user added to local dbase:
      Uname: poweruser Priv: 15 Encpass: Nimj18wRa7VApm5
```

The user "poweruser" should be able to Telnet into the PIX and enable with the existing local PIX enable password (the one from the **enable password** <password> command).

You can add more security by adding authentication for enabling, as shown in this example:

```
GOSS(config)# aaa authentication enable console LOCAL
```

This requires the user to enter the password both for login and enable. In this example, the password "poweruser" is used for both login and enable. User "poweruser" should be able to Telnet into the PIX and also enable with the local PIX password.

If you want some users to be able to only use certain commands, you have to set up a user with lower privileges, as shown in this example:

```
GOSS(config)# username ordinary password ordinary privilege 9
```

Since practically all of your commands are at level 15 by default, you have to move some commands down to level 9 so that "ordinary" users can issue them. In this instance, you want your level 9 user to be able to use the **show clock** command, but not to reconfigure the clock, as shown in this example:

```
GOSS(config)# privilege show level 9 command clock
```

You also need your user to be able to log out of the PIX (the user might be in level 1 or 9 when wanting to do this), as shown in this example:

```
GOSS(config)# privilege configure level 1 command logout
```

You need the user to be able to use the **enable** command (the user is at in level 1 when attempting this), as shown in this example:

```
GOSS(config)# privilege configure level 1 mode enable command enable
```

By moving the **disable** command to level 1, any user between levels 2–15 can get out of enable mode, as shown in this example:

```
GOSS(config)# privilege configure level 1 command disable
```

If you Telnet in as the user "ordinary" and enable as the same user (the password is also "ordinary"), you should use the **privilege configure level 1 command disable**, as shown in this example:

```
GOSS# show curpriv
Username : ordinary
Current privilege level : 9
Current Mode/s : P_PRIV
```

If you still have the original session open (the one prior to adding any authentication), the PIX may not know who you are because you did not initially log in with a username. If that is the case, use the **debug** command to view messages about the user "enable\_15" or "enable\_1" if there is no associated username. Therefore,

Telnet into the PIX as the user "poweruser" (the "level 15" user) prior to configuring command authorization, because you need to be sure the PIX can associate a username with the commands being attempted. You are ready to test command authorization by using this command:

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

The user "poweruser" should be able to Telnet in, enable, and perform all commands. The user "ordinary" should be able to use the **show clock**, **enable**, **disable**, and **logout** commands but no others, as shown in this example:

```
GOSS# show xlate  
Command authorization failed
```

## Authentication/Authorization with an AAA Server

You can also authenticate and authorize users by using an AAA server. TACACS+ works best because command authorization is possible, but RADIUS can also be used. Check to see if there are previous AAA Telnet/console commands on the PIX (in the event that the **LOCAL AAA** command was previously used), as shown in this example:

```
GOSS(config)# show aaa  
AAA authentication telnet console LOCAL  
AAA authentication enable console LOCAL  
AAA authorization command LOCAL
```

If there are previous AAA Telnet/console commands, remove them by using these commands:

```
GOSS(config)# no aaa authorization command LOCAL  
GOSS(config)# no aaa authentication telnet console LOCAL  
GOSS(config)# no aaa authentication enable console LOCAL
```

As with configuring local authentication, test to make sure users can Telnet into the PIX by using these commands.

```
telnet 172.18.124.0 255.255.255.0  
  
!--- IP range allowed to telnet to the PIX (values would depend on network).  
  
passwd <password>  
  
!--- Telnet password.  
  
Enable password <password>  
  
!--- Enable password.
```

Depending on what server you are using, configure the PIX for authentication/authorization with an AAA server.

## ACS – TACACS+

Configure ACS to communicate with the PIX by defining the PIX in the Network Configuration with "Authenticate Using" TACACS+ (for Cisco IOS® Software). The configuration of the ACS user depends on the configuration of the PIX. At a minimum, the ACS user should be set up with a username and password.

On the PIX, use these commands:

```
GOSS(config)# enable password cisco123
GOSS(config)# aaa-server TACSERVER protocol tacacs+
GOSS(config)# aaa-server TACSERVER (inside) host timeout 10
GOSS(config)# aaa authentication telnet console TACSERVER
```

At this point, the ACS user should be able to Telnet into the PIX, enable it with the existing enable password on the PIX, and perform all commands. Complete these steps:

1. If there is a need to do PIX enable authentication with ACS, choose **Interface Configuration > Advanced TACACS+ Settings**.
2. Check the **Advanced TACACS+ Features in Advanced Configuration Options** box.
3. Click **Submit**. The Advanced TACACS+ Settings are now visible under the user configuration.
4. Set Max Privilege for any AAA Client to Level 15.
5. Choose the enable password scheme for the user (which could involve configuring a separate enable password).
6. Click **Submit**.

To turn on enable authentication through TACACS+ in the PIX, use this command:

```
GOSS(config)# aaa authentication enable console TACSERVER
```

At this point, the ACS user should be able to Telnet into the PIX and enable with the enable password configured in ACS.

Prior to adding PIX command authorization, ACS 3.0 must be patched. You can download the patch from the Software Center (registered customers only) . You can also view additional information about this patch by accessing Cisco bug ID CSCdw78255 (registered customers only) .

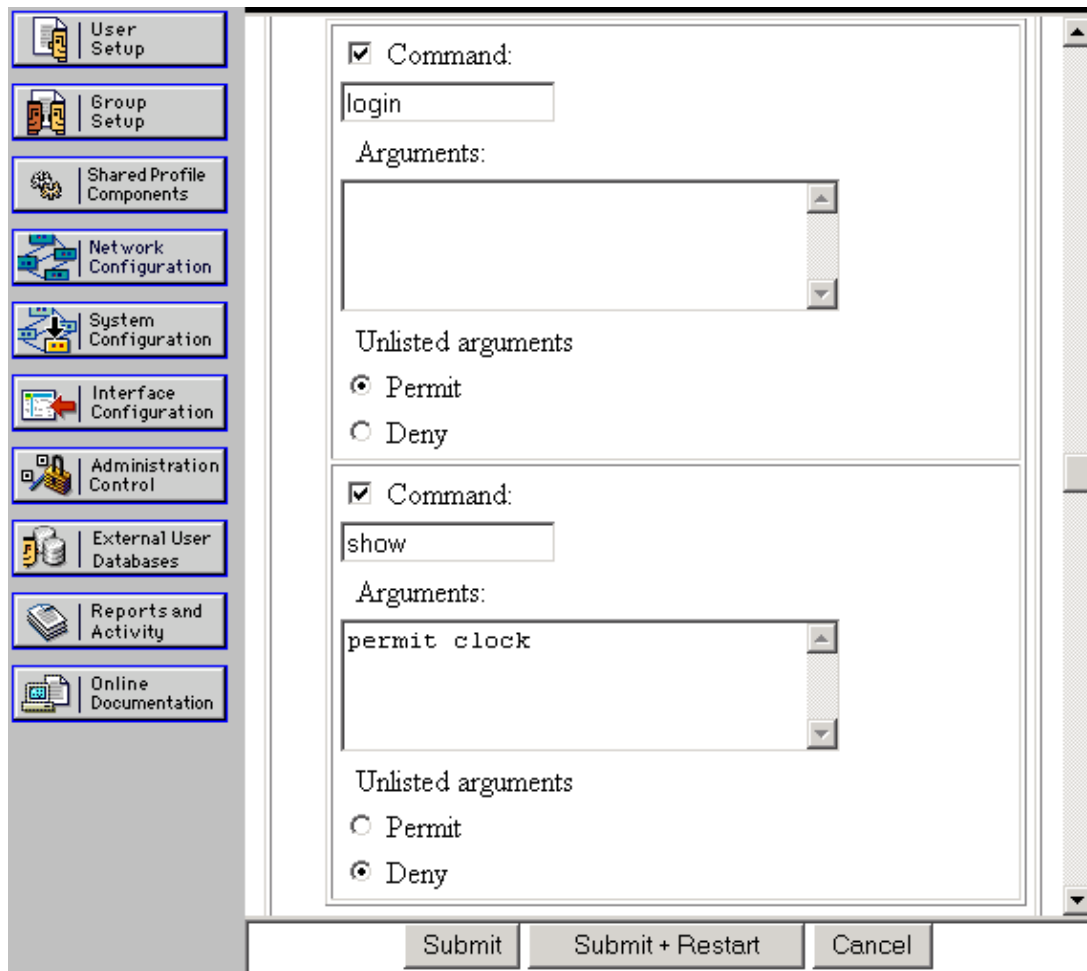
Authentication must be working prior to doing command authorization. If there is a need to perform command authorization with ACS, choose **Interface Configuration > TACACS+ (Cisco) > Shell (exec) for user and/or group** and click **Submit**. The shell command authorization settings are now visible under the user (or group) configuration.

It is a good idea to set up at least one powerful ACS user for command authorization and to permit unmatched Cisco IOS commands.

Other ACS users can be set up with command authorization by permitting a subset of commands. This example uses these steps:

1. Choose Group Settings to find the desired group from the drop-down box.
2. Click **Edit Settings**.
3. Choose **Shell Command Authorization Set**.
4. Click the **Command** button.
5. Enter **login**.
6. Choose Permit under Unlisted Arguments.
7. Repeat this process for the **logout**, **enable**, and **disable** commands.
8. Choose Shell Command Authorization Set.
9. Click the **Command** button.
10. Enter **show**.
11. Under Arguments , enter **permit clock**.
12. Choose deny for Unlisted Arguments.
13. Click **Submit**.

Here is an example of these steps:



If you still have your original session open (the one prior to adding any authentication), the PIX may not know who you are because you did not initially log in with a ACS username. If that is the case, use the **debug** command to view messages about user "enable\_15" or "enable\_1" if there is no username associated. You need to be sure the PIX can associate a username with the commands being attempted. You can do this by Telnetting into the PIX as the level 15 ACS user prior to configuring command authorization. You are ready to test command authorization by using this command:

```
aaa authorization command TACSERVER
```

At this point, you should have one user who should be able to Telnet in, enable, and use all of the commands, and a second user who can only do five commands.

## CSUnix – TACACS+

Configure CSUnix to communicate with the PIX as you would with any other network device. The configuration of the CSUnix user depends on the configuration of the PIX. At a minimum, the CSUnix user should be set up with a username and password. In this example, three users have been set up:

```
!--- This is our "poweruser" who can enable, use all commands, and log in.
!--- The login password is in the 'clear "*****"' statement.
!--- The enable password is in the 'clear "*****" 15' statement.

user = pixtest{
password = clear "*****"
privilege = clear "*****" 15
```

```

service=shell {
default cmd=permit
default attribute=permit
}
}

```

*!--- This user can Telnet in, enable, and use four commands  
!--- (such as **show clock**, **logout**, **exit**, and **enable**).  
!--- The login password is in the 'clear "\*\*\*\*\*"' statement.  
!--- The enable password is in the 'clear "\*\*\*\*\*" 15' statement.*

```

user = limitpix{
password = clear "*****"
privilege = clear "*****" 15
service=shell {
cmd=show {
permit "clock"
}
cmd=logout {
permit ".*"
}
cmd=enable {
permit ".*"
}
cmd=exit {
permit ".*"
}
}
}
}

```

*!--- This user can Telnet in, but not enable. This user can use any  
!--- **show** commands in non-enable mode as well as **logout**, **exit**, and **?**.*

```

user = oneuser{
password = clear "*****"
service=shell {
cmd=show {
permit ".*"
}
cmd=logout {
permit ".*"
}
cmd="?" {
permit ".*"
}
cmd=exit {
permit ".*"
}
}
}
}

```

On the PIX, use these commands:

```

GOSS(config)# enable password cisco123
GOSS(config)# aaa-server TACSERVER protocol tacacs+
GOSS(config)# aaa-server TACSERVER (inside) host <ip> <key> timeout 10
GOSS(config)# aaa authentication telnet console TACSERVER

```

At this point, any of the CSUnix users should be able to Telnet into the PIX, enable with the existing enable password on the PIX, and use all of the commands.

Enable authentication through TACACS+ in the PIX:

```
GOSS(config)# aaa authentication enable console TACSERVER
```

At this point, the CSUnix users who have "privilege 15" passwords should be able to Telnet into the PIX and enable with those "enable" passwords.

If you still have your original session open (the one prior to adding any authentication), the PIX may not know who you are because you did not initially log in with a username. If that is the case, issuing the **debug** command may show messages about user "enable\_15" or "enable\_1" if there is no username associated. Telnet into the PIX as the user "pixtest" (our "level 15" user) prior to configuring command authorization, because we need to be sure the PIX can associate a username with the commands being attempted. Enable authentication must be on prior to doing command authorization. If there is a need to perform command authorization with CSUnix, add this command:

```
GOSS(config)# aaa authorization command TACSERVER
```

Of the three users, "pixtest" can do everything, and the other two users can do a subset of commands.

## ACS – RADIUS

RADIUS command authorization is not supported. Telnet and enable authentication is possible with ACS. ACS can be configured to communicate with the PIX by defining the PIX in Network Configuration with "Authenticate Using" RADIUS (any variety). The configuration of the ACS user depends on the configuration of the PIX. At a minimum, the ACS user should be set up with a username and password.

On the PIX, use these commands:

```
GOSS(config)# enable password cisco123
GOSS(config)# aaa-server RADSERVER protocol radius GOSS(config)
                # aaa-server RADSERVER (inside)
                host <ip> <key> timeout 10
GOSS(config)# aaa authentication telnet console RADSERVER
```

At this point, the ACS user should be able to Telnet into the PIX, enable with the existing enable password on the PIX, and use all commands (the PIX does not send commands to the RADIUS server; RADIUS command authorization is not supported).

If you want to enable with ACS and RADIUS on the PIX, add this command:

```
aaa authentication enable console RADSERVER
```

Unlike with TACACS+, the same password is used for RADIUS enable as for RADIUS login.

## CSUnix – RADIUS

Configure CSUnix to talk to the PIX as you would with any other network device. The configuration of the CSUnix user depends on the configuration of the PIX. This profile works for authentication and enabling:

```
user = pixradius{
profile_id = 26
profile_cycle = 1
```

```
!--- The login password is in the 'clear "*****"' statement;
!--- this is used for the login, enable, and non-enable commands.
```

```
password = clear "*****" < pixradius
}
```

On the PIX, use these commands:

```
GOSS(config)# enable password cisco123
GOSS(config)# aaa-server RADSERVER protocol radius
GOSS(config)# aaa-server RADSERVER (inside) host <ip> <key> timeout 10
```

If you want to enable with ACS and RADIUS on the PIX, use this command:

```
GOSS(config)# aaa authentication enable console RADSERVER
```

Unlike with TACACS+, the same password is used for RADIUS enable as for RADIUS login.

## Network Access Restrictions

Network access restrictions can be used in both ACS and CSUnix to limit who may connect to the PIX for administrative purposes.

- **ACS** The PIX would be configured in the Network Access Restrictions area of the Group Settings. The PIX configuration is either "Denied Calling/Point of Access Locations" or "Permitted Calling/Point of Access Locations" (depending on the security plan).
- **CSUnix** This is an example of a user who is permitted access to the PIX, but not other devices:

```
user = naruser{
  profile_id = 119
  profile_cycle = 1
  password = clear "*****"
  privilege = clear "*****" 15
  service=shell {
    allow "10.98.21.50" ".*" ".*"
    refuse ".*" ".*" ".*"
    default cmd=permit
    default attribute=permit
  }
}
```

## Debug

To turn on debug, use this command:

```
logging on
logging <console|monitor> debug
```

These are examples of good and bad debugs:

- **Good debug** The user is able to use the **log in**, **enable**, and **perform** commands.

```
307002: Permitted Telnet login session from 172.18.124.111
111006: Console Login from pixpartial at console
502103: User priv level changed: Uname: pixpartial From: 1 To: 15
111009: User 'pixpartial' executed cmd: show clock
```

- **Bad debug** Authorization fails for user, as shown in this example:

```
610101: Authorization failed: Cmd: uauth Cmdtype: show
```

- **The remote AAA server is unreachable:**

## Accounting

There is no actual command accounting available, but by having syslog activated on the PIX, you can see what actions were performed, as shown in this example:

```
307002: Permitted Telnet login session from 172.18.124.111
111006: Console Login from pixtest at console
611103: User logged out: Uname: pixtest
307002: Permitted Telnet login session from 172.18.124.111
111006: Console Login from pixtest at console
502103: User priv level changed: Uname: pixtest From: 1 To: 15
111008: User 'pixtest' executed the 'enable' command.
111007: Begin configuration: 172.18.124.111 reading from terminal
111008: User 'pixtest' executed the 'configure t' command.
111008: User 'pixtest' executed the 'write t' command.
```

## Information to Collect if You Open a TAC Case

If you still need assistance after following the troubleshooting steps above and want to open a case with the Cisco TAC, be sure to include the following information for troubleshooting your PIX Firewall.

- Problem description and relevant topology details
- Troubleshooting performed before opening the case
- Output from the **show tech-support** command
- Output from the **show log** command after running with the **logging buffered debugging** command, or console captures that demonstrate the problem (if available)

Please attach the collected data to your case in non-zipped, plain text format (.txt). You can attach information to your case by uploading it using the Case Query Tool (registered customers only) . If you cannot access the Case Query Tool, you can send the information in an email attachment to [attach@cisco.com](mailto:attach@cisco.com) with your case number in the subject line of your message.

---

## Related Information

- [PIX Command Reference](#)
- [Cisco PIX Firewall Software – Technical Support & Documentation](#)
- [Cisco Secure Access Control Server for Windows – Technical Support & Documentation](#)
- [Cisco Secure Access Control Server for Unix – Technical Support & Documentation](#)
- [Technical Support & Documentation – Cisco Systems](#)

