

# PIX, TACACS+, and RADIUS Sample Configurations: 4.4.x

Document ID: 13819

## Contents

### Introduction

#### Prerequisites

- Requirements
- Components Used
- Conventions

#### Authentication vs. Authorization

#### What the User Sees with Authentication/Authorization On Security Server Configurations Used for All Scenarios

- CiscoSecure UNIX TACACS Server Configuration
- CiscoSecure UNIX RADIUS Server Configuration
- CiscoSecure NT 2.x RADIUS
- EasyACS TACACS+
- CiscoSecure 2.x TACACS+
- Livingston RADIUS Server Configuration
- Merit RADIUS Server Configuration
- TACACS+ Freeware Server Configuration

#### Debugging Steps

#### Network Diagram

#### Authentication Debug Examples from PIX

#### Adding Authorization

#### Authentication and Authorization Debug Examples from PIX

#### Adding Accounting

- TACACS+
- RADIUS

#### Use of Except Command

#### Max-sessions and Viewing Logged-in Users

#### Authentication and Enabling on the PIX Itself

#### Authentication on the Serial Console

#### Changing the Prompt Users See

#### Customizing the Message Users See on Success/Failure

#### Per-user Idle and Absolute Timeouts

#### Virtual HTTP

#### Virtual Telnet

#### Virtual Telnet Logout

#### Port Authorization

#### Related Information

## Introduction

RADIUS and TACACS+ authentication may be done for FTP, Telnet, and HTTP connections. Authentication for other less common TCP protocols can usually be made to work.

TACACS+ authorization is supported; RADIUS authorization is not. Changes in PIX 4.4.1 authentication, authorization, and accounting (AAA) over the previous version include: AAA server groups and failover, authentication for enable and serial console access, and accept and reject prompt messages.

# Prerequisites

## Requirements

There are no specific requirements for this document.

## Components Used

This document is not restricted to specific software and hardware versions.

## Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

## Authentication vs. Authorization

- Authentication is who the user is.
- Authorization is what the user can do.
- Authentication is valid without authorization.
- Authorization is not valid without authentication.

Suppose you have 100 users inside and you want only want 6 of these users to be able to do FTP, Telnet, or HTTP outside the network. You would tell the PIX to authenticate outbound traffic and give all 6 users IDs on the TACACS+/RADIUS security server. With simple authentication, these 6 users could be authenticated with username and password, then go out. The other 94 users could not go out. The PIX prompts users for username/password, then passes their username and password to the TACACS+/RADIUS security server, and depending on the response, opens or denies the connection. These 6 users could do FTP, Telnet, or HTTP.

But suppose one of these three users, "Terry," is not to be trusted. You would like to allow Terry to do FTP, but not HTTP or Telnet to the outside. This means having to add authorization, that is, authorizing what users can do in addition to authenticating who they are. When we add authorization to the PIX, the PIX would first send Terry's username and password to the security server, then send an authorization request telling the security server what "command" Terry is trying to do. With the server set up properly, Terry could be allowed to "FTP 1.2.3.4" but would be denied the ability to HTTP or Telnet anywhere.

## What the User Sees with Authentication/Authorization On

When trying to go from inside to outside (or vice versa) with authentication/authorization on:

- **Telnet** – The user sees a username prompt display, followed by a request for password. If authentication (and authorization) is successful at the PIX/server, the user is prompted for username and password by the destination host beyond.
- **FTP** – The user sees a username prompt come up. The user needs to enter "local\_username@remote\_username" for username and "local\_password@remote\_password" for password. The PIX sends the "local\_username" and "local\_password" to the local security server, and if authentication (and authorization) is successful at the PIX/server, the "remote\_username" and "remote\_password" are passed to the destination FTP server beyond.
- **HTTP** – A window is displayed in the browser requesting username and password. If authentication (and authorization) is successful, the user arrives at the destination web site beyond. Keep in mind that **browsers cache usernames and passwords**. If it appears that the PIX should be timing out an HTTP connection but is not doing so, it is likely that re-authentication actually is taking place with the browser "shooting" the cached username and password to the PIX, which then forwards this to the

authentication server. PIX syslog and/or server debug will show this phenomenon. If Telnet and FTP seem to work "normally", but HTTP connections do not, this is why.

## Security Server Configurations Used for All Scenarios

### CiscoSecure UNIX TACACS Server Configuration

Make sure that you have the PIX IP address or fully-qualified domain name and key in the CSU.cfg file.

```
user = ddunlap {
password = clear "rtp"
default service = permit
}

user = can_only_do_telnet {
password = clear "telnetonly"
service = shell {
cmd = telnet {
permit .*
}
}
}

user = can_only_do_ftp {
password = clear "ftponly"
service = shell {
cmd = ftp {
permit .*
}
}
}

user = httponly {
password = clear "httponly"
service = shell {
cmd = http {
permit .*
}
}
}
```

### CiscoSecure UNIX RADIUS Server Configuration

Use the advanced graphical user interface (GUI) to add the PIX IP and key to network access server (NAS) list.

```
user=adminuser {
radius=Cisco {
check_items= {
2="all"
}
reply_attributes= {
6=6
}
}
```

### CiscoSecure NT 2.x RADIUS

Complete these steps.

1. Obtain a password in User Setup GUI section.
2. From Group Setup GUI section, set attribute 6 (Service-Type) to Login or Administrative.
3. Add the PIX IP in the NAS Configuration GUI.

## EasyACS TACACS+

The EasyACS documentation describes setup.

1. In the group section, click on **Shell exec** (to give exec privileges).
2. To add authorization to the PIX, click **Deny unmatched IOS commands** at the bottom of the group setup.
3. Select the **Add/Edit** new command for each command you want to allow (for example, Telnet).
4. If you want to allow Telnet to specific sites, enter the IP(s) in the argument section in the form "permit #.#.#.#". To allow Telnet to all sites, click **Allow all unlisted arguments**.
5. Click **Finish editing command**.
6. Perform steps 1 through 5 for each of the allowed commands (for example, Telnet, HTTP and/or FTP).
7. Add the PIX IP in the NAS Configuration GUI section.

## CiscoSecure 2.x TACACS+

The user obtains a password in the User setup section of the GUI.

1. In the group section, click **Shell exec** (to give exec privileges).
2. To add authorization to the PIX, click **Deny unmatched IOS commands** at the bottom of the group setup.
3. Select **Add/Edit** for each command you want to allow (for example, Telnet).
4. If you want to allow Telnet to specific sites, enter the permit IP(s) in the argument rectangle (for example, "permit 1.2.3.4"). To allow Telnet to all sites, click **Allow all unlisted arguments**.
5. Click **Finish editing command**.
6. Perform steps 1 through 5 for each of the allowed commands (for example, Telnet, HTTP or FTP).
7. Add the PIX IP in the NAS Configuration GUI section.

## Livingston RADIUS Server Configuration

Add the PIX IP and key to the clients file.

```
adminuser Password="all"  
User-Service-Type = Shell-User
```

## Merit RADIUS Server Configuration

Add the PIX IP and key to the clients file.

```
adminuser Password="all"  
Service-Type = Shell-User
```

## TACACS+ Freeware Server Configuration

```
key = "cisco"  
  
user = adminuser {  
  login = cleartext "all"  
  default service = permit  
}
```

```
user = can_only_do_telnet {
login = cleartext "telnetonly"
cmd = telnet {
permit .*
}
}

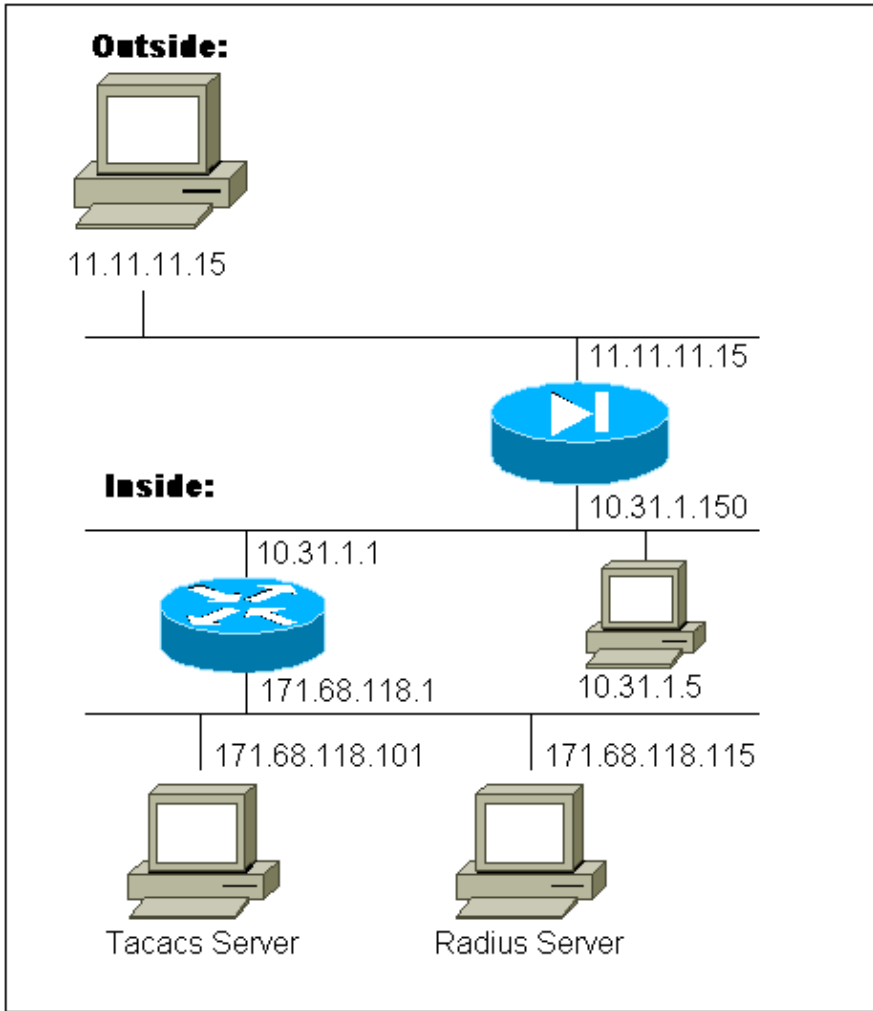
user = httponly {
login = cleartext "httponly"
cmd = http {
permit .*
}
}

user = can_only_do_ftp {
login = cleartext "ftponly"
cmd = ftp {
permit .*
}
}
```

## Debugging Steps

- Make sure that the PIX configurations are working before adding authentication, authorization, and accounting (AAA).
  - ◆ If you cannot pass traffic before instituting authentication and authorization, you will not be able to do so afterwards.
- Enable logging in the PIX:
  - ◆ The **logging console debugging** command should not be used on a heavily loaded system.
  - ◆ The **logging buffered debugging** command can be used. Output from the **show logging** or **logging** commands can be sent to a syslog server and examined.
- Make sure that debugging is on for the TACACS+ or RADIUS servers. All servers have this option.

## Network Diagram



### PIX Configuration

```

pix-5# write terminal
Building configuration...
: Saved
:
PIX Version 4.4(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 pix/intf2 security10
nameif ethernet3 pix/intf3 security15
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pix-5
fixup protocol ftp 21
fixup protocol http 80
fixup protocol smtp 25
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol sqlnet 1521
names
pager lines 24
no logging timestamp
logging console debugging
no logging monitor
no logging buffered
logging trap debugging
logging facility 20
interface ethernet0 auto
interface ethernet1 auto

```

```

interface ethernet2 auto
interface ethernet3 auto
mtu outside 1500
mtu inside 1500
mtu pix/intf2 1500
mtu pix/intf3 1500
ip address outside 11.11.11.1 255.255.255.0
ip address inside 10.31.1.150 255.255.255.0
ip address pix/intf2 127.0.0.1 255.255.255.255
ip address pix/intf3 127.0.0.1 255.255.255.255
no failover
failover timeout 0:00:00
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address pix/intf2 0.0.0.0
failover ip address pix/intf3 0.0.0.0
arp timeout 14400
global (outside) 1 11.11.11.10-11.11.11.14 netmask 255.255.255.0
static (inside,outside) 11.11.11.20 171.68.118.115 netmask 255.255.255.255 0 0
static (inside,outside) 11.11.11.21 171.68.118.101 netmask 255.255.255.255 0 0
static (inside,outside) 11.11.11.22 10.31.1.5 netmask 255.255.255.255 0 0
conduit permit icmp any any
conduit permit tcp any any
no rip outside passive
no rip outside default
no rip inside passive
no rip inside default
no rip pix/intf2 passive
no rip pix/intf2 default
no rip pix/intf3 passive
no rip pix/intf3 default
route inside 0.0.0.0 0.0.0.0 10.31.1.1 1
timeout xlate 3:00:00 conn 1:00:00 half-closed 0:10:00 udp 0:02:00
timeout rpc 0:10:00 h323 0:05:00
timeout uauth 0:00:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
!

!--- For any given list, multiple AAA servers can
!--- be configured. They will be
!--- tried sequentially if any one of them is down.

!
aaa-server Outgoing protocol tacacs+
aaa-server Outgoing (inside) host 171.68.118.101 cisco timeout 10
aaa-server Incoming protocol radius
aaa-server Incoming (inside) host 171.68.118.115 cisco timeout 10
aaa authentication ftp outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
aaa authentication http outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
aaa authentication telnet outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
aaa authentication ftp inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Incoming
aaa authentication http inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Incoming
aaa authentication telnet inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Incoming
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
telnet timeout 5
terminal width 80
Cryptochecksum:b287a37a676262275a4201cac52399ca
: end

```

# Authentication Debug Examples from PIX

In these debug examples:

## Outbound

Inside user at 10.31.1.5 initiates traffic to outside 11.11.11.15 and is authenticated through TACACS+ (outbound traffic uses server list "Outgoing" which includes TACACS server 171.68.118.101).

## Inbound

Outside user at 11.11.11.15 initiates traffic to inside 10.31.1.5 (11.11.11.22) and is authenticated through RADIUS (inbound traffic uses server list "Incoming" which includes RADIUS server 171.68.118.115).

## PIX Debug – Good Authentication – TACACS+

The example below shows PIX debug with good authentication:

```
109001: Auth start for user '???' from 10.31.1.5/11004 to 11.11.11.15/23
109011: Authen Session Start: user 'ddunlap', sid 3
109005: Authentication succeeded for user 'ddunlap'
from 10.31.1.5/11004 to 11.11.11.15/23
109012: Authen Session End: user 'ddunlap', sid 3, elapsed 1 seconds
302001: Built outbound TCP connection 4 for faddr 11.11.11.15/23 gaddr
11.11.11.22/11004 laddr 10.31.1.5/11004
```

## PIX debug – Bad Authentication (Username or Password) – TACACS+

The example below shows PIX debug with bad authentication (username or password). The user sees four username/password sets. The following message displays: "Error: max number of tries exceeded".

```
109001: Auth start for user '???' from 10.31.1.5/11005 to 11.11.11.15/23
109006: Authentication failed for user '' from 10.31.1.5/11005 to 11.11.11.15/23
```

## PIX debug – Can Ping, but no Response – TACACS+

The example below shows PIX debug for a pingable server that is not speaking to the PIX. The user sees the username once, and PIX never asks for a password (this is on Telnet).

```
'Error: Max number of tries exceeded'
109001: Auth start for user '???' from 10.31.1.5/11006 to 11.11.11.15/23
109002: Auth from 10.31.1.5/11006 to 11.11.11.15/23 failed
(server 171.68.118.101 failed)
109002: Auth from 10.31.1.5/11006 to 11.11.11.15/23 failed
(server 171.68.118.101 failed)
304006: URL Server 171.68.118.101 not responding, trying 171.68.118.101
109002: Auth from 10.31.1.5/11006 to 11.11.11.15/23 failed
(server 171.68.118.101 failed)
109006: Authentication failed for user '' from 10.31.1.5/11006 to 11.11.11.15/23
```

## PIX Debug – Can't Ping Server – TACACS+

The example below shows PIX debug for a server that is not pingable. The user sees the username once. PIX never asks for a password (this is on Telnet). The following message displays: "Timeout to TACACS+ server" and "Error: Max number of tries exceeded" (the configuration in this example reflects a bogus server).

```
109001: Auth start for user '???' from 10.31.1.5/11007 to 11.11.11.15/23
```

```
109002: Auth from 10.31.1.5/11007 to 11.11.11.15/23 failed
(server 171.68.118.199 failed)
109002: Auth from 10.31.1.5/11007 to 11.11.11.15/23 failed
(server 171.68.118.199 failed)
304006: URL Server 171.68.118.199 not responding, trying 171.68.118.199
109002: Auth from 10.31.1.5/11007 to 11.11.11.15/23 failed
(server 171.68.118.199 failed)
109006: Authentication failed for user '' from 10.31.1.5/11007 to 11.11.11.15/23
```

## PIX Debug – Good Authentication – RADIUS

The example below show PIX debug with good authentication:

```
109001: Auth start for user '???' from 11.11.11.15/11003 to 10.31.1.5/23
109011: Authen Session Start: user 'adminuser', sid 4
109005: Authentication succeeded for user 'adminuser'
from 10.31.1.5/23 to 11.11.11.15/11003
109012: Authen Session End: user 'adminuser', sid 4, elapsed 1 seconds
302001: Built inbound TCP connection 5 for faddr
11.11.11.15/11003 gaddr 11.11.11.22/23 laddr 10.31.1.5/23
```

## PIX Debug – Bad Authentication (Username or Password) – RADIUS

The example below shows PIX debug with bad authentication (username or password). The user sees a request for Username and Password. If either is wrong, the message "Incorrect password" displays four times. Then, the user is disconnected. This problem has been assigned bug ID #CSCdm46934.

```
'Error: Max number of tries exceeded'
109001: Auth start for user '???' from 11.11.11.15/11007 to 10.31.1.5/23
109006: Authentication failed for user '' from 10.31.1.5/23 to 11.11.11.15/11007
```

## PIX Debug – Daemon Down, Will Not Communicate with PIX – RADIUS

The example below shows PIX debug with a pingable server, but daemon is down. The server will not communicate with PIX. The user sees Username, followed by password. The following messages display: "RADIUS server failed" and "Error: Max number of tries exceeded".

```
109001: Auth start for user '???' from 11.11.11.15/11008 to 10.31.1.5/23
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11008 failed
(server 171.68.118.115 failed)
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11008 failed
(server 171.68.118.115 failed)
304006: URL Server 171.68.118.115 not responding, trying 171.68.118.115
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11008 failed
(server 171.68.118.115 failed)
109006: Authentication failed for user '' from 10.31.1.5/23 to 11.11.11.15/11008
```

## PIX Debug – Can't Ping Server or Key/Client Mismatch – RADIUS

The example below shows PIX debug for a server that is not pingable or where there is a key/client mismatch. The user sees Username and Password. The following messages display: "Timeout to RADIUS server" and "Error: Max number of tries exceeded" (the server in the configuration is for example purposes only).

```
109001: Auth start for user '???' from 11.11.11.15/11009 to 10.31.1.5/23
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11009 failed
(server 171.68.118.199 failed)
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11009 failed
(server 171.68.118.199 failed)
304006: URL Server 171.68.118.199 not responding, trying 171.68.118.199
109002: Auth from 10.31.1.5/23 to 11.11.11.15/11009 failed
(server 171.68.118.199 failed)
```

```
109006: Authentication failed for user '' from 10.31.1.5/23 to 11.11.11.15/11009
```

## Adding Authorization

As authorization is not valid without authentication, we will require authorization for the same source and destination range:

```
aaa authorization ftp outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
aaa authorization http outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
aaa authorization telnet outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

### Outgoing

Note that we do not add authorization for "incoming" because incoming traffic is authenticated with RADIUS, and RADIUS authorization is not valid

## Authentication and Authorization Debug Examples from PIX

### PIX Debug With Good Authentication and Successful Authorization – TACACS+

The example below show PIX debug with good authentication and successful authorization:

```
109001: Auth start for user '???' from 10.31.1.5/11002 to 11.11.11.15/23
109011: Authen Session Start: user 'can_only_do_telnet', sid 7
109005: Authentication succeeded for user 'can_only_do_telnet'
from 10.31.1.5/11002 to 11.11.11.15/23
109011: Authen Session Start: user 'can_only_do_telnet', sid 7
109007: Authorization permitted for user 'can_only_do_telnet'
from 10.31.1.5/11002 to 11.11.11.15/23
109012: Authen Session End: user 'can_only_do_telnet', sid 7,
elapsed 1 seconds
302001: Built outbound TCP connection 6 for faddr 11.11.11.15/23
gaddr 11.11.11.22/11002 laddr 10.31.1.5/11002 (can_only_do_telnet)
```

### PIX Debug – Good Authentication, Failed Authorization – TACACS+

The example below shows PIX debug with good authentication, but failed authorization:

Here the user also sees the message "Error: Authorization Denied"

```
109001: Auth start for user '???' from 10.31.1.5/11000 to 11.11.11.15/23
109011: Authen Session Start: user 'can_only_do_ftp', sid 5
109005: Authentication succeeded for user 'can_only_do_ftp'
from 10.31.1.5/11000 to 11.11.11.15/23
109008: Authorization denied for user 'can_only_do_ftp' from
10.31.1.5/11000 to 11.11.11.15/23
109012: Authen Session End: user 'can_only_do_ftp', sid 5, elapsed 33 seconds
```

## Adding Accounting

### TACACS+

```
aaa accounting any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
```

Debug will look the same whether accounting is on or off. However, at the time of the "Built", there will be a "start" accounting record sent. At the time of the "Teardown", there will be a "stop" accounting record sent.

TACACS+ accounting records look like the following (these are from CiscoSecure UNIX; the ones in CiscoSecure NT may be comma-delimited instead):

```
Thu Jun  3 10:41:50 1999 10.31.1.150 can_only_do_telnet
PIX 10.31.1.5 start task_id=0x7 foreign_ip=11.11.11.15
local_ip=10.31.1.5 cmd=telnet
Thu Jun  3 10:41:55 1999 10.31.1.150 can_only_do_telnet PIX 10.31.1.5
stop task_id=0x7 foreign_ip=11.11.11.15
local_ip=10.31.1.5 cmd=telnet elapsed_time=4 bytes_in=74 bytes_out=27
```

## RADIUS

```
aaa accounting any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Incoming
```

Debug will look the same whether accounting is on or off. However, at the time of the "Built", a "start" accounting record is sent. At the time of the "Teardown", a "stop" accounting record is sent:

RADIUS accounting records look like the following: (these are from CiscoSecure UNIX; the ones in CiscoSecure NT may be comma-delimited instead):

```
10.31.1.150 adminuser -- start server=rtp-evergreen.rtp.cisco.com
time=14:53:11 date=06/3/1999 task_id=0x00000008
Thu Jun  3 15:53:11 1999
Acct-Status-Type = Start
Client-Id = 10.31.1.150
Login-Host = 10.31.1.5
Login-TCP-Port = 23
Acct-Session-Id = "0x00000008"
User-Name = "adminuser"
10.31.1.150 adminuser -- stop server=rtp-evergreen.rtp.cisco.com
time=14:54:24 date=06/ 3/1999 task_id=0x00000008
Thu Jun  3 15:54:24 1999
Acct-Status-Type = Stop
Client-Id = 10.31.1.150
Login-Host = 10.31.1.5
Login-TCP-Port = 23
Acct-Session-Id = "0x00000008"
User-Name = "adminuser"
Acct-Session-Time = 73
Acct-Input-Octets = 27
Acct-Output-Octets = 73
```

## Use of Except Command

In our network, if we decide that a particular source and/or destination does not need authentication, authorization, or accounting, we can do something like the following:

```
aaa authentication except outbound 10.31.1.60 255.255.255.255
11.11.11.15 255.255.255.255 Outgoing
aaa authorization except outbound 10.31.1.60 255.255.255.255
11.11.11.15 255.255.255.255 Outgoing
```

If you are "excepting" ip addresses from authentication and have authorization on, you must also except them from authorization!

## Max-sessions and Viewing Logged-in Users

Some TACACS+ and RADIUS servers have "max-session" or "view logged-in users" features. The ability to do max-sessions or check logged-in users is dependent on accounting records. When there is an accounting

"start" record generated but no "stop" record, the TACACS+ or RADIUS server assumes the person is still logged in (that is, has a session through the PIX).

This works well for Telnet and FTP connections because of the nature of the connections. This does not work well for HTTP because of the nature of the connection. In the following example, a different network configuration is used but the concepts are the same.

The user telnets through the PIX, authenticating on the way:

```
(pix) 109001: Auth start for user '???' from 171.68.118.100/1200
to 9.9.9.25 /23
(pix) 109011: Authen Session Start: user 'cse', sid 3
(pix) 109005: Authentication succeeded for user 'cse' from
171.68.118.100/12 00 to 9.9.9.25/23
(pix) 302001: Built TCP connection 5 for faddr 9.9.9.25/23
gaddr 9.9.9.10/12 00 laddr 171.68.118.100/1200 (cse)
(server start account) Sun Nov 8 16:31:10 1998 rtp-pinecone.rtp.cisco.com cse
PIX 171.68.118.100 start task_id=0x3 foreign_ip=9.9.9.25
local_ip=171.68.118.100 cmd=telnet
```

Because the server has seen a "start" record but no "stop" record (at this point in time), the server will show that the "Telnet" user is logged in. If the user attempts another connection that requires authentication (perhaps from another PC) and if max-sessions is set to "1" on the server for this user (assuming the server supports max-sessions), the connection will be refused by the server.

The user goes on with her Telnet or FTP business on the target host, then exits (spends 10 minutes there):

```
(pix) 302002: Teardown TCP connection 5 faddr 9.9.9.25/80 gaddr
9.9.9.10/128 1 laddr 171.68.118.100/1281 duration 0:00:00 bytes 1907 (cse)

(server stop account) Sun Nov 8 16:41:17 1998 rtp-pinecone.rtp.cisco.com cse

PIX 171.68.118.100 stop task_id=0x3 foreign_ip=9.9.9.25 local_ip=171.68.118.100
cmd=telnet elapsed_time=5 bytes_in=98 bytes_out=36
```

Whether uauth is 0 (authenticate every time) or more (authenticate once and not again during uauth period), an accounting record is cut for every site accessed.

However, HTTP works differently due to the nature of the protocol. Below is an example of HTTP.

The user browses from 171.68.118.100 to 9.9.9.25 through the PIX:

```
(pix) 109001: Auth start for user '???' from 171.68.118.100/1281
to 9.9.9.25 /80
(pix) 109011: Authen Session Start: user 'cse', sid 5
(pix) 109005: Authentication succeeded for user 'cse' from
171.68.118.100/12 81 to 9.9.9.25/80
(pix) 302001: Built TCP connection 5 for faddr 9.9.9.25/80 gaddr
9.9.9.10/12 81 laddr 171.68.118.100/1281 (cse)
(server start account) Sun Nov 8 16:35:34 1998 rtp-pinecone.rtp.cisco.com cse
PIX 171.68.118.100 start task_id=0x9 foreign_ip=9.9.9.25
local_ip=171.68.118.100 cmd=http
(pix) 302002: Teardown TCP connection 5 faddr 9.9.9.25/80 gaddr
9.9.9.10/128 1 laddr 171.68.118.100/1281 duration 0:00:00 bytes 1907 (cse)
(server stop account) Sun Nov 8 16:35:35 1998 rtp-pinecone.rtp.cisco .com
cse PIX 171.68.118.100 stop task_id=0x9 foreign_ip =9.9.9.25
local_ip=171.68.118.100 cmd=http elapsed_time=0 bytes_in=1907 bytes_out=223
```

The user reads the downloaded web page.

The start record posted at 16:35:34, and the stop record posted at 16:35:35. This download took one second (that is; there was less than one second between the start and the stop record). Is the user still logged in to the web site and the connection still open when they are reading the web page? No. Will max-sessions or view logged-in users work here? No, because the connection time (the time between the "Built" and "Teardown") in HTTP is too short. The "start" and "stop" record is sub-second. There will not be a "start" record without a "stop" record, since the records occur at virtually the same instant. There will still be "start" and "stop" record sent to the server for every transaction, whether uauth is set for 0 or something larger. However, max-sessions and view logged-in users will not work due to the nature of HTTP connections.

## Authentication and Enabling on the PIX Itself

The previous discussion was of authenticating Telnet (and HTTP, FTP) traffic through the PIX. In the example below, we make sure that Telnet to the pix works without authentication on:

```
telnet 10.31.1.5 255.255.255.255
passwd ww
```

Then, we add the command to authenticate users Telnetting to the PIX:

```
aaa authentication telnet console Outgoing
```

When users Telnet to the PIX, they are prompted for the Telnet password ("ww"). The PIX also requests the TACACS+ in this case (since the "Outgoing" server list is used) or RADIUS username and password.

```
aaa authentication enable console Outgoing
```

With this command, the user is prompted for a username and password which is sent to the TACACS or RADIUS server. In this case, since the "Outgoing" server list is used, the request goes to the TACACS server. Since the authentication packet for enable is the same as the authentication packet for login, the user can enable through TACACS or RADIUS with the same username/password, assuming the user can log in to the PIX with TACACS or RADIUS. This problem has been assigned bug ID #CSCdm47044.

In the event that the server is down, the user can gain access to the PIX enable mode by entering "PIX" for the username and the normal enable password from the PIX ("enable password whatever"). If "enable password whatever" is not in the PIX configuration, the user should enter "PIX" for the username and press the Enter key. If the enable password is set but not known, a password recovery disk will be required in order to reset.

## Authentication on the Serial Console

The **aaa authentication serial console** command requires authentication verification in order to access the serial console of the PIX. When the user performs configuration commands from the console, syslog messages will be cut (if the PIX is configured to send syslog at the debug level to a syslog host). Below is an example from the syslog server:

```
Jun  5 07:24:09 [10.31.1.150.2.2] %PIX-5-111008: User 'cse' executed
the 'hostname' command.
```

## Changing the Prompt Users See

If we have the command:

```
auth-prompt THIS_IS_PIX_5
```

the users going through the PIX see the sequence:

```
THIS_IS_PIX_5 [at which point one would enter the username]
Password:[at which point one would enter the password]
```

and then, on arrival at the ultimate destination box, the "Username:" and "Password:" prompt the destination box is presented.

This prompt only affects users going through the PIX, not to the PIX.

**Note:** There are no accounting records cut for access to the PIX.

## Customizing the Message Users See on Success/Failure

If we have the commands:

```
auth-prompt accept "You're allowed through the pix"
auth-prompt reject "You blew it"
```

Users will see the following on a failed/successful login through the PIX:

```
THIS_IS_PIX_5
Username: asjdkl
Password:
"You blew it"
"THIS_IS_PIX_5"
Username: cse
Password:
"You're allowed through the pix"
```

## Per-user Idle and Absolute Timeouts

Idle and absolute uauth timeouts can be sent down from the TACACS+ server on a per-user basis. If all the users in your network are to have the same "timeout uauth," then do not implement this! But, if you need different uauths per-user, read on.

In our example on the PIX, we use the **timeout uauth 3:00:00** command. This means that once a person authenticates, they will not have to reauthenticate for 3 hours. But if we set up a user with the following profile and have TACACS AAA authorization on in the PIX, the idle and absolute timeouts in the user profile override the timeout uauth in the PIX for that user. This does not mean that the Telnet session through the PIX gets disconnected after the idle/absolute timeout. It just controls whether or not re-authentication takes place.

```
user = timeout {
default service = permit
login = cleartext "timeout"
service = exec {
timeout = 2
idletime = 1
}
}
```

After authentication, issue a **show uauth** command on the PIX:

```
pix-5# show uauth
Current      Most Seen
Authenticated Users      1          1
Authen In Progress      0          1
user 'timeout' at 10.31.1.5, authorized to:
  port 11.11.11.15/telnet
  absolute timeout: 0:02:00
```

```
inactivity timeout: 0:01:00
```

After the user sits idle for one minute, the debug on the PIX shows:

```
109012: Authen Session End: user 'timeout', sid 19, elapsed 91 seconds
```

The user will have to re-authenticate when returning to the same target host or a different host.

## Virtual HTTP

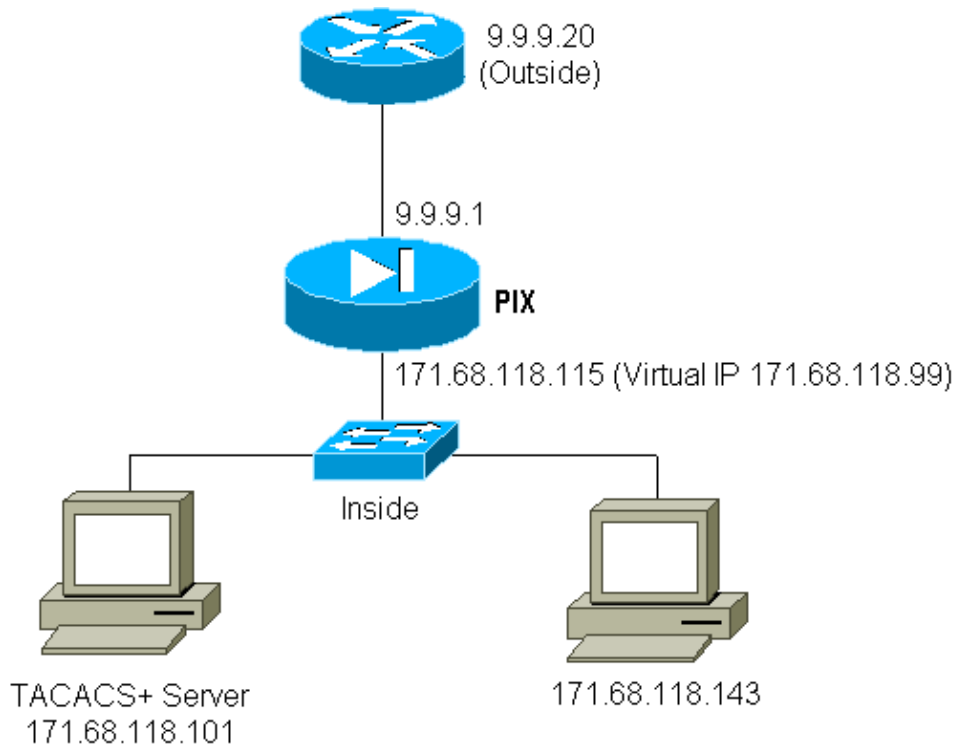
If authentication is required on sites outside the PIX, as well as on the PIX itself, unusual browser behavior can sometimes be observed since browsers cache the username and password.

To avoid this, you can implement virtual HTTP by adding an RFC 1918 address (that is, an address that is unroutable on the Internet, but valid and unique for the PIX inside network) to the PIX configuration using the following command:

```
virtual http #.#.#.# [warn]
```

When the user tries to go outside the PIX, authentication is required. If the warn parameter is present, the user receives a redirect message. The authentication is good for the length of time in the uauth. As indicated in the documentation, do not set the **timeout uauth** command duration to 0 seconds with virtual HTTP; this prevents HTTP connections to the real web server.

### Virtual HTTP Outbound Example:



### PIX Configuration Virtual HTTP Outbound:

```
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
global (outside) 1 9.9.9.5-9.9.9.9 netmask 255.0.0.0
timeout uauth 01:00:00
aaa-server TACACS+ protocol tacacs+
```

```

aaa-server Outgoing protocol tacacs+
aaa-server Outgoing (inside) host 171.68.118.101 cisco timeout 10
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
virtual http 171.68.118.99
auth-prompt THIS_IS_PIX_5

```

## Virtual Telnet

Configuring the PIX to authenticate all inbound and outbound traffic is not a good idea because some protocols, such as "mail," are not easily authenticated. When a mail server and client try to communicate through the PIX when all traffic through the PIX is being authenticated, the PIX syslog for unauthenticatable protocols will show messages such as:

```

109001: Auth start for user '???' from 9.9.9.10/11094 to 171.68.118.106/25
109009: Authorization denied from 171.68.118.106/49 to 9.9.9.10/11094

(not authenticated)

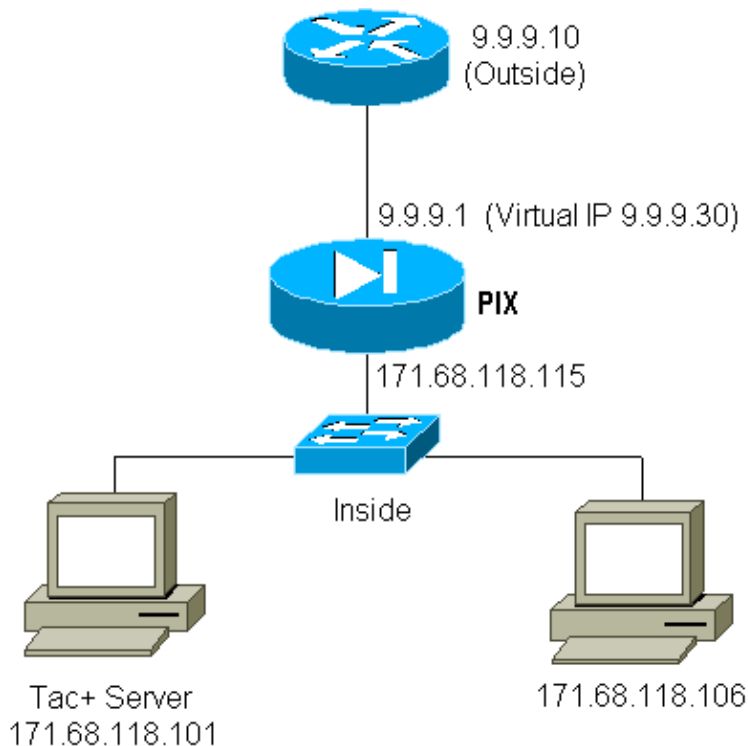
```

Since mail and some other services are not interactive enough to authenticate, one solution is to use the **except** command for authentication/authorization (authenticate all except for source/destination of the mail server/client).

But if there is really a need to authenticate some kind of unusual service, this can be done by use of the **virtual telnet** command. This command allows authentication to occur to the virtual Telnet IP. After this authentication, the traffic for the unusual service can go to the real server which is tied to the virtual IP.

In our example, we want to allow TCP port 49 traffic to flow from outside host 9.9.9.10 to inside host 171.68.118.106. As this traffic is not really authenticatable, we set up virtual Telnet.

### Virtual Telnet Inbound:



### PIX Configuration Virtual Telnet Inbound:

```

ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
static (inside,outside) 9.9.9.30 171.68.118.106 netmask 255.255.255.255 0 0
conduit permit tcp host 9.9.9.30 host 9.9.9.10
aaa-server TACACS+ protocol tacacs+
aaa-server Incoming protocol tacacs+
aaa-server Incoming (inside) host 171.68.118.101 cisco timeout 5
aaa authentication any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Incoming
virtual telnet 9.9.9.30

```

### TACACS+ Server User Configuration Virtual Telnet Inbound:

```

user = pinecone {
default service = permit
    login = cleartext "pinecone"
service = exec {
    timeout = 10
    idletime = 10
}
}

```

### PIX Debug Virtual Telnet Inbound:

The user at 9.9.9.10 must first authenticate by telnetting to the 9.9.9.30 address on the PIX:

```

pixfirewall# 109001: Auth start for user '???' from 9.9.9.10/11099
to 171.68.118.106/23
109011: Authen Session Start: user 'pinecone', sid 13
109005: Authentication succeeded for user 'pinecone' from
171.68.118.106/23 to 9.9.9.10/11099

```

After the successful authentication, the **show uauth** command shows the user has "time on the meter":

```

pixfirewall# show uauth

```

	Current	Most Seen
Authenticated Users	1	1
Authen In Progress	0	1
user 'pinecone' at 9.9.9.10, authenticated		
absolute timeout:	0:10:00	
inactivity timeout:	0:10:00	

And when the device at 9.9.9.10 wants to send TCP/49 traffic to the device at 171.68.118.106:

```

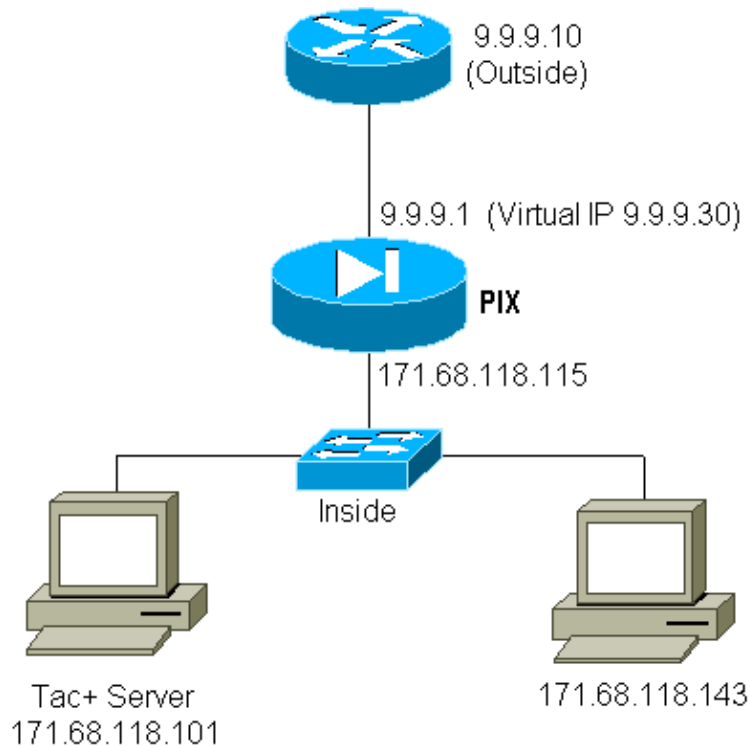
pixfirewall# 109001: Auth start for user 'pinecone'
from 9.9.9.10/11104 to 171.68.118.106/49
109011: Authen Session Start: user 'pinecone', sid 14
109007: Authorization permitted for user 'pinecone' from 9.9.9.10/11104
to 171.68.118.106/49
302001: Built TCP connection 23 for faddr 9.9.9.10/11104 gaddr
9.9.9.30/49 laddr 171.68.118.106/49 (pinecone)
302002: Teardown TCP connection 23 faddr 9.9.9.10/11104 gaddr 9.9.9.30/49
laddr 171.68.118.106/49 duration 0:00:10 bytes 179 (pinecone)

```

### Virtual Telnet Outbound:

Since outbound traffic is allowed by default, no static is required for use of virtual Telnet outbound. In the following example, the inside user at 171.68.118.143 will Telnet to virtual 9.9.9.30 and authenticate. The Telnet connection is immediately dropped.

Once authenticated, TCP traffic is allowed from 171.68.118.143 to the server at 9.9.9.10:



### PIX Configuration Virtual Telnet Outbound:

```
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
global (outside) 1 9.9.9.5-9.9.9.9 netmask 255.0.0.0
timeout uauth 00:05:00
aaa-server TACACS+ protocol tacacs+
aaa-server Outgoing protocol tacacs+
aaa-server Outgoing (inside) host 171.68.118.101 cisco timeout 10
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
virtual telnet 9.9.9.30
```

### PIX Debug Virtual Telnet Outbound:

```
109001: Auth start for user '???' from 171.68.118.143/1536 to 9.9.9.30/23
109011: Authen Session Start: user 'timeout_143', sid 25
109005: Authentication succeeded for user 'timeout_143' from
171.68.118.143/1536 to 9.9.9.30/23
302001: Built TCP connection 46 for faddr 9.9.9.10/80 gaddr 9.9.9.30/1537
laddr 171.68 .118.143/1537 (timeout_143)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302001: Built TCP connection 47 for faddr 9.9.9.10/80 gaddr 9.9.9.30/1538
laddr 171.68 .118.143/1538 (timeout_143)
302002: Teardown TCP connection 46 faddr 9.9.9.10/80 gaddr 9.9.9.30/1537
laddr 171.68. 118.143/1537 duration 0:00:03 bytes 625 (timeout_143)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302002: Teardown TCP connection 47 faddr 9.9.9.10/80 gaddr 9.9.9.30/1538
laddr 171.68. 118.143/1538 duration 0:00:01 bytes 2281 (timeout_143)
302009: 0 in use, 1 most used
```

## Virtual Telnet Logout

When the user Telnets to the virtual Telnet IP, the **show uauth** command shows his uauth. If the user wants to prevent traffic from going through after his session is finished (when there is time left in the uauth), he needs to Telnet to the virtual Telnet IP again. This toggles the session off.

# Port Authorization

You can require authorization on a range of ports. In the following example, authentication was still required for all outbound, but authorization is only required for TCP ports 23–49.

## PIX Configuration:

```
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
aaa authorization tcp/23-49 outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 Outgoing
```

So, when we Telnet from 171.68.118.143 to 9.9.9.10, authentication and authorization occurred because Telnet port 23 is in the 23–49 range. When we do an HTTP session from 171.68.118.143 to 9.9.9.10, we still have to authenticate, but the PIX does not ask the TACACS+ server to authorize HTTP because 80 is not in the 23–49 range.

## TACACS+ Freeware Server Configuration

```
user = telnetrange {
    login = cleartext "telnetrange"
    cmd = tcp/23-49 {
        permit 9.9.9.10
    }
}
```

Note that the PIX is sending "cmd=tcp/23–49" and "cmd–arg=9.9.9.10" to the TACACS+ server.

## Debug on the PIX:

```
109001: Auth start for user '???' from 171.68.118.143/1051 to 9.9.9.10/23
109011: Authen Session Start: user 'telnetrange', sid 0
109005: Authentication succeeded for user 'telnetrange' from
171.68.118.143/1051 to 9.9.9.10/23
109011: Authen Session Start: user 'telnetrange', sid 0
109007: Authorization permitted for user 'telnetrange' from
171.68.118.143/1051 to 9.9.9.10/23
302001: Built TCP connection 0 for faddr 9.9.9.10/23 gaddr 9.9.9.5/1051
laddr 171.68.1.18.143/1051 (telnetrange)
109001: Auth start for user '???' from 171.68.118.143/1105 to 9.9.9.10/80
109001: Auth start for user '???' from 171.68.118.143/1110 to 9.9.9.10/80
109011: Authen Session Start: user 'telnetrange', sid 1
109005: Authentication succeeded for user 'telnetrange' from
171.68.118.143/1110 to 9.9.9.10/80
302001: Built TCP connection 1 for faddr 9.9.9.10/80 gaddr 9.9.9.5/1110
laddr 171.68.1.18.143/1110 (telnetrange)
302001: Built TCP connection 2 for faddr 9.9.9.10/80 gaddr 9.9.9.5/1111
laddr 171.68.1.18.143/1111 (telnetrange)
302002: Teardown TCP connection 1 faddr 9.9.9.10/80 gaddr 9.9.9.5/1110
laddr 171.68.1.18.143/1110 duration 0:00:08 bytes 338 (telnetrange)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302002: Teardown TCP connection 2 faddr 9.9.9.10/80 gaddr 9.9.9.5/1111 laddr
171.68.1.18.143/1111 duration 0:00:01 bytes 2329 (telnetrange)
```

## Related Information

- [PIX Support Page](#)
- [Documentation for PIX Firewall](#)
- [PIX Command References](#)
- [Requests for Comments \(RFCs\)](#)
- [Technical Support – Cisco Systems](#)

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2010 – 2011 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Jan 17, 2006

Document ID: 13819

---