

PIX, TACACS+, and RADIUS Sample Configurations: 4.3.x

Document ID: 13818

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Authentication vs. Authorization

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

- Cisco Secure UNIX TACACS Server Configuration
- Cisco Secure UNIX RADIUS Server Configuration
- Cisco Secure NT 2.x RADIUS
- EasyACS TACACS+
- Cisco Secure 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

Add Accounting

- TACACS+
- RADIUS

Use of Except Command

Max-sessions and Viewing Logged-in Users

Authentication to the PIX Itself

Change the Prompt Users See

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 will usually work. TACACS+ *authorization* is supported; RADIUS *authorization* is not.

The syntax for authentication changed somewhat in PIX software 4.3.2. This document uses the syntax for software version 4.3.2.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

- PIX Software version 4.3.2

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

Conventions

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.

As an example, assume you have one hundred users inside and you only want six of these users to be able to do FTP, Telnet, or HTTP outside the network. You tell the PIX to authenticate outbound traffic and give all six users IDs on the TACACS+/RADIUS security server. With simple *authentication*, these six users can be authenticated with username and password, then go out. The other ninety-four users cannot go out. The PIX prompts users for username/password, then passes their username and password to the TACACS+/RADIUS security server. Depending on the response, the PIX then opens or denies the connection. These six users can 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 you have to add *authorization*, that is, authorizing *what* users can do in addition to authenticating *who* they are. When you add *authorization* to the PIX, the PIX first sends Terry's username and password to the security server, then sends an authorization request that tells the security server what "*command*" Terry is trying to do. With the server set up properly, Terry can be allowed to "FTP 1.2.3.4" but is 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 a 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 displays in the browser and requests a 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 time out an HTTP connection but does not do 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

In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only).

Cisco Secure 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 = all {
password = clear "all"
default service = permit
}

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

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

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

Cisco Secure UNIX RADIUS Server Configuration

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

```
user=all {
```

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

Cisco Secure NT 2.x RADIUS

Follow this procedure.

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 **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 section.
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 IP(s) in the argument section. To allow Telnet to all sites, click **Allow all unlisted arguments**.
5. Click **Finish editing command**.
6. Perform the 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.

Cisco Secure 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 the previous steps for each of the allowed commands (for example, Telnet, HTTP and/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.

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

Merit RADIUS Server Configuration

Add the PIX IP and key to the clients file.

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

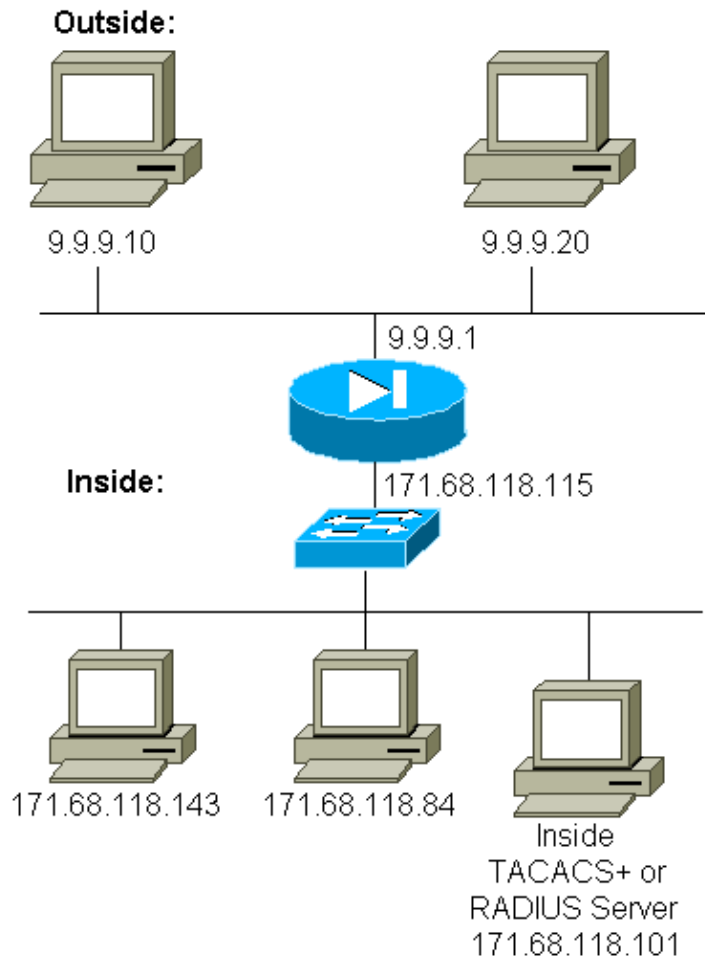
TACACS+ Freeware Server Configuration

```
# Handshake with router--PIX needs 'tacacs-server host #.#.#.# cisco':  
key = "cisco"  
  
user = all {  
login = cleartext "all"  
default service = permit  
}  
  
user = telnetonly {  
login = cleartext "telnetonly"  
cmd = telnet {  
permit .*  
}  
}  
  
user = httponly {  
login = cleartext "httponly"  
cmd = http {  
permit .*  
}  
}  
  
user = ftponly {  
login = cleartext "ftponly"  
cmd = ftp {  
permit .*  
}  
}
```

Debugging Steps

- Make sure that the PIX configurations work before you add authentication, authorization, and accounting (AAA).
 - ◆ If you cannot pass traffic before you institute 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 heavy 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

```

Building configuration...
: Saved
:
PIX Version 4.3(2)205
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd OnTrBUG1Tp0edmkr encrypted
hostname pixfirewall
fixup protocol ftp 21
fixup protocol http 80
fixup protocol smtp 25
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol sqlnet 1521
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 0.0.0.0
failover ip address 0.0.0.0
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
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
ip address 127.0.0.1 255.255.255.255
ip address 127.0.0.1 255.255.255.255
arp timeout 14400
global (outside) 1 9.9.9.5-9.9.9.9 netmask 255.0.0.0
static (inside,outside) 9.9.9.12 171.68.118.143 netmask 255.255.255.255 0 0
static (inside,outside) 9.9.9.13 171.68.118.84 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
timeout xlate 3:00:00 conn 1:00:00 udp 0:02:00
timeout rpc 0:10:00 h323 0:05:00
timeout uauth 0:00:00 absolute
!
!
!--- The next entry depends on whether TACACS+ or RADIUS is used.
!
tacacs-server (inside) host 171.68.118.101 cisco timeout 5
!
!
!--- We have decided to authenticate all inbound and outbound FTP,
!--- HTTP, and Telnet traffic.
!
aaa authentication ftp outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication http outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication telnet outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication ftp inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication http inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication telnet inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
telnet 171.68.118.143 255.255.255.255
telnet timeout 5
mtu outside 1500
mtu inside 1500
mtu 1500
mtu 1500
Cryptochecksum:cd1759f9f5f78af763e9ce565e6d2c75
: end

```

Authentication Debug Examples from PIX

The inside user at 171.68.118.143 initiates traffic to the outside 9.9.9.10 network or the 9.9.9.20 network.

PIX Debug – Good Authentication – TACACS+

This example shows a PIX debug with good authentication:

```

109001: Auth start for user '???' from 171.68.118.143/3494 to 9.9.9.20/23
109011: Authen Session Start: user 'all', sid 18
109005: Authentication succeeded for user 'all' from 171.68.118.143/3494 to 9.9.9.20/23

```

```
302001: Built TCP connection 31 for faddr 9.9.9.20/23 gaddr 9.9.9.12/3494
```

PIX Debug – Bad Authentication (Username or Password) – TACACS+

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

```
109001: Auth start for user '???' from 171.68.118.143/3502 to 9.9.9.20/23
109006: Authentication failed for user '' from 171.68.118.143/3502 to 9.9.9.20/23
```

PIX Debug – Can Ping Server, No Response – TACACS+

This example shows a PIX debug for a pingable server that is not speaking to PIX. The user sees and can enter a username four times, but PIX *never asks for a password* (this is on a Telnet). Immediately, the "Error: Max number of tries exceeded" message displays.

```
109001: Auth start for user '???' from 171.68.118.143/3515 to 9.9.9.10/23
109006: Authentication failed for user '' from 171.68.118.143/3515 to 9.9.9.10/23
```

PIX Debug – Unable to Ping Server – TACACS+

This example shows a 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 "Timeout to TACACS+ server" and "Error: Max number of tries exceeded" messages display.

```
109001: Auth start for user '???' from 171.68.118.143/3522 to 9.9.9.10/23
109002: Auth from 171.68.118.143/3522 to 9.9.9.10/23 failed (server 1.1.1.1 failed)
109002: Auth from 171.68.118.143/3522 to 9.9.9.10/23 failed (server 1.1.1.1 failed)
109002: Auth from 171.68.118.143/3522 to 9.9.9.10/23 failed (server 1.1.1.1 failed)
109006: Authentication failed for user '' from 171.68.118.143/3522 to 9.9.9.10/23
```

PIX Debug – Good Authentication – RADIUS

This example shows a PIX debug with good authentication. This example shows an HTTP session where you browse the router/server beyond. The IP HTTP server enabled.

```
109001: Auth start for user '???' from 171.68.118.143/3095 to 9.9.9.10/80
109011: Authen Session Start: user 'all', sid 3
109005: Authentication succeeded for user 'all' from 171.68.118.143/3095
to 9.9.9.10/80
302001: Built TCP connection 6 for faddr 9.9.9.10/80 gaddr 9.9.9.12/3095
laddr 171.68.118.143/3095 (all)
304001: all@171.68.118.143 Accessed URL 9.9.9.10:/exec/show/tech-support/cr
```

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

This example shows a PIX debug with bad authentication (username or password). The user sees four Username/Password sets. The PIX indicates: "Incorrect password". The "Error: Max number of tries exceeded" finally displays.

```
109001: Auth start for user '???' from 171.68.118.143/3139 to 9.9.9.10/23
109006: Authentication failed for user '' from 171.68.118.143/3139 to 9.9.9.10/23
```

PIX Debug – Server Does Not Communicate With PIX

This example shows a PIX debug for a server that does not communicate with the PIX. The username displays once, followed by the password four times. The "RADIUS server failed" displays and then finally the "Error:

Max number of tries exceeded" message displays.

```
109001: Auth start for user '???' from 171.68.118.143/2675 to 9.9.9.10/23
109002: Auth from 171.68.118.143/2675 to 9.9.9.10/23 failed
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.143/2675 to 9.9.9.10/23 failed
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.143/2675 to 9.9.9.10/23 failed
(server 171.68.118.101 failed)
109006: Authentication failed for user '' from 171.68.118.143/2675 to 9.9.9.10/23
```

Adding Authorization

As authorization is not valid without authentication, authorization is required 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
aaa authorization ftp inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
aaa authorization http inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
aaa authorization telnet inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

Authentication and Authorization Debug Examples from PIX

PIX Debug – Good Authentication and Authorization Succeeds – TACACS+

This example shows a PIX debug with good authentication and successful authorization.

```
109001: Auth start for user '???' from 171.68.118.143/3574 to 9.9.9.20/23
109011: Authen Session Start: user 'telnetonly', sid 21
109005: Authentication succeeded for user 'telnetonly' from
171.68.118.143/3574 to 9.9.9.20/23
109011: Authen Session Start: user 'telnetonly', sid 21
109007: Authorization permitted for user 'telnetonly' from
171.68.118.143/3574 to 9.9.9.20/23
109012: Authen Session End: user 'telnetonly', sid 21, elapsed 1 seconds
```

PIX Debug – Good Authentication, But Authorization Fails – TACACS+

This example shows a PIX debug with good authentication but failed authorization.

```
109001: Auth start for user '???' from 171.68.118.143/3551 to 9.9.9.10/23
109011: Authen Session Start: user 'httponly', sid 19
109005: Authentication succeeded for user 'httponly'
from 171.68.118.143/3551 to 9.9.9.10/23
109008: Authorization denied for user 'httponly'
from 171.68.118.143/3551 to 9.9.9.10/23
```

Add Accounting

TACACS+

```
aaa accounting any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa accounting any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
```

Debug looks 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.

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

```
Sat Mar 6 07:25:54 1999 171.68.118.115 timeout_143 PIX 171.68.118.143
start task_id=0x1f
foreign_ip=9.9.9.20 local_ip=171.68.118.143 cmd=telnet
Sat Mar 6 07:26:24 1999 171.68.118.115 timeout_143 PIX 171.68.118.143
stop task_id=0x1f
foreign_ip=9.9.9.20 local_ip=171.68.118.143 cmd=telnet elapsed_time=29
bytes_in=74 bytes_out=24
```

RADIUS

```
aaa accounting any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 radius
aaa accounting any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 radius
```

Debug looks 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 this output (these are from Cisco Secure UNIX; the ones in Cisco Secure NT may be comma-delimited instead):

```
Sat Mar 6 06:37:59 1999
Acct-Status-Type = Start
NAS-IP-Address = 171.68.118.115
Login-IP-Host = 171.68.118.143
Login-TCP-Port = 23
Acct-Session-Id = 0x00000009
User-Name = all

Sat Mar 6 06:38:10 1999
Acct-Status-Type = Stop
NAS-IP-Address = 171.68.118.115
Login-IP-Host = 171.68.118.143
Login-TCP-Port = 23
Acct-Session-Id = 0x00000009
User-Name = all
Acct-Session-Time = 11
Acct-Input-Octets = 72
Acct-Output-Octets = 23
```

Use of Except Command

In our network, if we decide that one outgoing user (171.68.118.84) does not need to be authenticated, authorized, or accounted for when going to a specific host or network, we can do this:

```
aaa authentication except outbound 171.68.118.84 255.255.255.255 9.9.9.0
255.255.255.0 tacacs+
aaa authentication any outbound 171.68.118.0 255.255.255.0 9.9.9.0
255.255.255.0 tacacs+
aaa authorization except outbound 171.68.118.84 255.255.255.255 9.9.9.0
255.255.255.0
aaa authorization any outbound 171.68.118.0 255.255.255.0 9.9.9.0 255.255.255.0
aaa accounting except outbound 171.68.118.84 255.255.255.255 9.9.9.0
255.255.255.0 tacacs+
aaa accounting any outbound 171.68.118.0 255.255.255.0 9.9.9.0 255.255.255.0 tacacs+
```

If you except a box from authentication and have authorization on, you must also except the box 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, the user 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.

As an example of Telnet or FTP, the user Telnets from 171.68.118.100 to 9.9.9.25 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 shows 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, the connection is refused by the server.

The user goes on with regular 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 (that is, 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. This 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 a 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 (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 to the PIX Itself

We discussed authenticating Telnet (and HTTP, FTP) traffic through the PIX. With software version 4.3.2, Telnet connections to the PIX may also be authenticated. Here, we define the IPs of boxes that can Telnet to the PIX:

```
telnet 171.68.118.143 255.255.255.255
```

We then supply the Telnet password:

```
passwd ww
```

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

```
aaa authentication telnet console tacacs+|radius
```

When users Telnet to the PIX, they are prompted for the Telnet password ("ww"), then the PIX also requests the TACACS+ or RADIUS username and password.

Change the Prompt Users See

If we have the command:

```
auth-prompt THIS_IS_PIX_5
```

the users going through the PIX will see the sequence:

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

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

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.

Per-user Idle and Absolute Timeouts

Starting in software version 4.3.2.205, 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", you should not implement this! But if you need different uauths per-user, read on.

In our example, we set up two users – users "timeout_84" and "timeout_143". We want user "timeout_84" to

have an idle timeout of two and an absolute timeout of five. However, user "timeout_143" is a dawdler, so we give "timeout_143" an idle of one and absolute of two to discourage this. The user profiles (from the TACACS+ freeware) are as follows:

```
user = timeout_143 {
    default service = permit
    login = cleartext "timeout_143"
    service = exec {
        timeout = 2
        idletime = 1
    }
}

user = timeout_84 {
    default service = permit
    login = cleartext "timeout_84"
    service = exec {
        timeout = 5
        idletime = 2
    }
}
```

As the following debug shows, we clear uauths, allow both users to Telnet through the PIX (they are authenticated), and do a **show uauth** command to see the per-user uauths.

Both users sit idle. After approximately 1 minute, user timeout_143's uauth expires. After about 2 minutes, user timeout_84's uauth expires. This means that both users will have to re-authenticate whether they return to the same target host or to a different host.

```
pixfirewall# clear uauth
pixfirewall# 109001: Auth start for user '???' from
171.68.118.143/3382 to 9.9.9.20/23
109011: Authen Session Start: user 'timeout_143', sid 15
109005: Authentication succeeded for user 'timeout_143'
from 171.68.118.143/3382 to 9.9.9.20/23
302001: Built TCP connection 28 for faddr 9.9.9.20/23 gaddr
9.9.9.12/3382 laddr 171.68.118.143/3382 (timeout_143)
109001: Auth start for user '???' from 171.68.118.84/3330 to 9.9.9.10/23
109011: Authen Session Start: user 'timeout_84', sid 16
109005: Authentication succeeded for user 'timeout_84'
from 171.68.118.84/3330 to 9.9.9.10/23
302001: Built TCP connection 29 for faddr 9.9.9.10/23
gaddr 9.9.9.13/3330 laddr 171.68.118.84/3330 (timeout_84)

pixfirewall#
pixfirewall# show uauth
                Current      Most Seen
Authenticated Users      2          2
Authen In Progress      0          1
user 'timeout_143' at 171.68.118.143, authenticated
    absolute timeout: 0:02:00
    inactivity timeout: 0:01:00
user 'timeout_84' at 171.68.118.84, authenticated
    absolute timeout: 0:05:00
    inactivity timeout: 0:02:00
pixfirewall#
pixfirewall#
pixfirewall# 109012: Authen Session End: user 'timeout_143',
sid 15, elapsed 93 seconds
302002: Teardown TCP connection 27 faddr 10.31.1.99/1467 gaddr 9.9.9.13/3329
laddr 171.68.118.84/3329 duration 0:02:32 bytes 0 (timeout_84)
pixfirewall# 109012: Authen Session End: user 'timeout_84', sid 16,
elapsed 134 seconds
```

Virtual HTTP

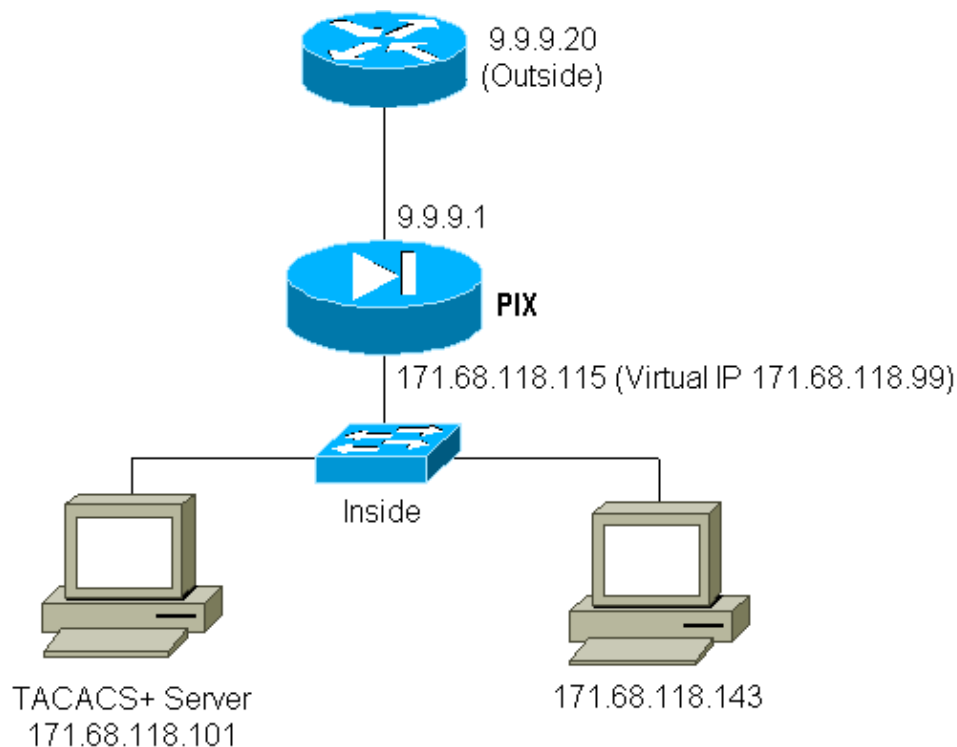
If authentication is required on sites outside the PIX, as well as on the PIX itself, unusual browser behavior can sometimes be observed since browsers cache the username and password.

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
tacacs-server (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 tacacs+
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
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 during a time when all traffic through the PIX is being authenticated, PIX syslog for unauthenticatable protocols will show messages such as:

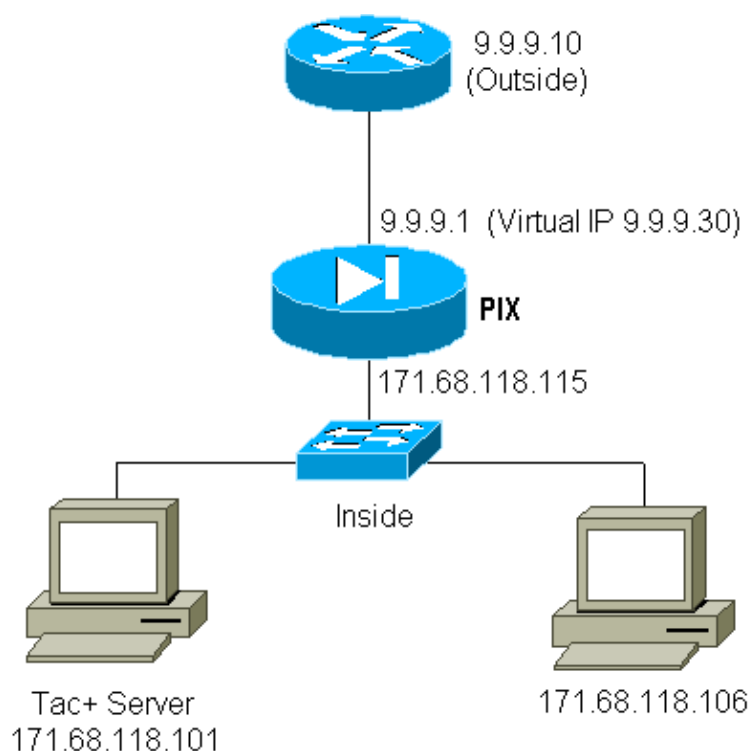
```
109001: Auth start for user '???' from 9.9.9.10/11094 to 171.68.118.106/25
109009: Authorization denied from 171.68.118.106/49 to 9.9.9.10/11094
(not authenticated)
```

Since mail and some other services are not interactive enough to authenticate, one solution is to use the **except** command for authentication/authorization. That means we would 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
tacacs-server (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 tacacs+
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
```

```
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 that the user has "time on the meter":

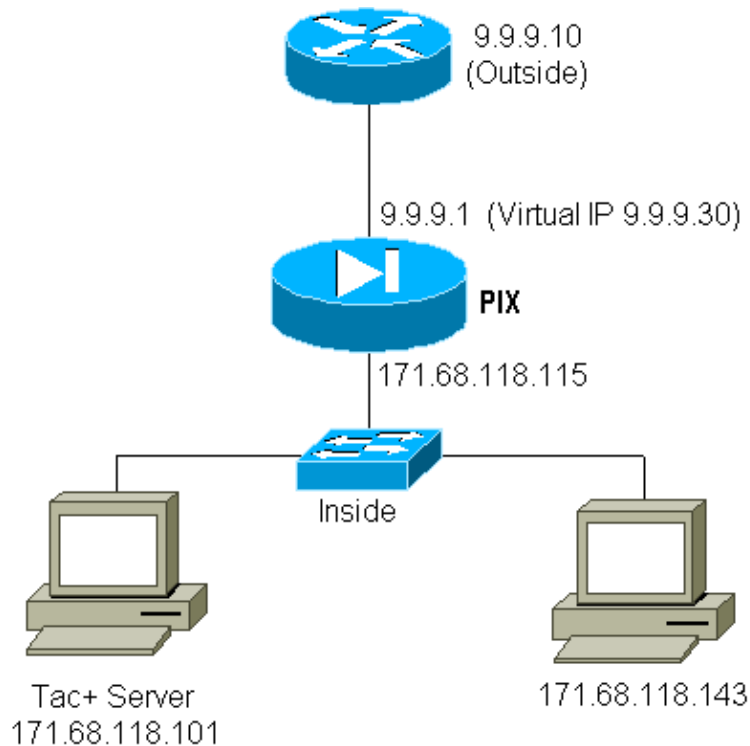
```
pixfirewall# show uauth
Current      Most Seen
Authenticated Users      1          1
Authen In Progress      0          1
user 'pinecone' at 9.9.9.10, authenticated
  absolute timeout: 0:10:00
  inactivity timeout: 0:10:00
```

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
tacacs-server (inside) host 171.68.118.101 cisco timeout 5
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+
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 you Telnet to the virtual Telnet IP, the **show uauth** command displays who has been authenticated and the time left. If you want to prevent traffic from going through after your session is finished (when time is left in the uauth), Telnet to the virtual Telnet IP again. This toggles the session off.

Port Authorization

With PIX 4.3.2, 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 tacacs+
aaa authorization tcp/23-49 outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

When a Telnet is performed from 171.68.118.143 to 9.9.9.10, authentication and authorization occur because Telnet port 23 is in the 23–49 range.

When an HTTP session is initiated from 171.68.118.143 to 9.9.9.10, authentication still happens, 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 171.68.118.143/1051 (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 171.68.118.143/1110 (telnetrange)
302001: Built TCP connection 2 for faddr 9.9.9.10/80 gaddr 9.9.9.5/1111
laddr 171.68.118.143/1111 (telnetrange)
302002: Teardown TCP connection 1 faddr 9.9.9.10/80 gaddr 9.9.9.5/1110
laddr 171.68.118.143/1110 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 171.68.118.143/1111 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](#)

© 2007 – 2008 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Sep 26, 2008

Document ID: 13818
