

PIX, TACACS+, and RADIUS Sample Configurations: 4.2.x

Document ID: 13817

Introduction

Prerequisites

- Requirements
- Components Used
- Network Diagram
- Conventions

Authentication vs. Authorization

What the User Sees with Authentication/Authorization On 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 NT 2.x TACACS+
- Livingston RADIUS Server Configuration
- Merit RADIUS Server Configuration
- TACACS+ Freeware Server Configuration

Debugging Steps

Authentication Debug Examples from PIX

Adding Authorization

Authentication and Authorization Debug Examples From PIX

Add Accounting

- TACACS+
- RADIUS

Max Sessions and Viewing Logged-in Users

Use of the Except Command

Authentication to the PIX Itself

Changing the Prompt the Users See

Related Information

Introduction

RADIUS and TACACS+ authentication may be done for FTP, Telnet, and HTTP connections. TACACS+ authorization is supported; RADIUS authorization is not.

The syntax for authentication changed slightly in PIX software 4.2.2. This document uses the syntax for software versions 4.2.2.

Prerequisites

Requirements

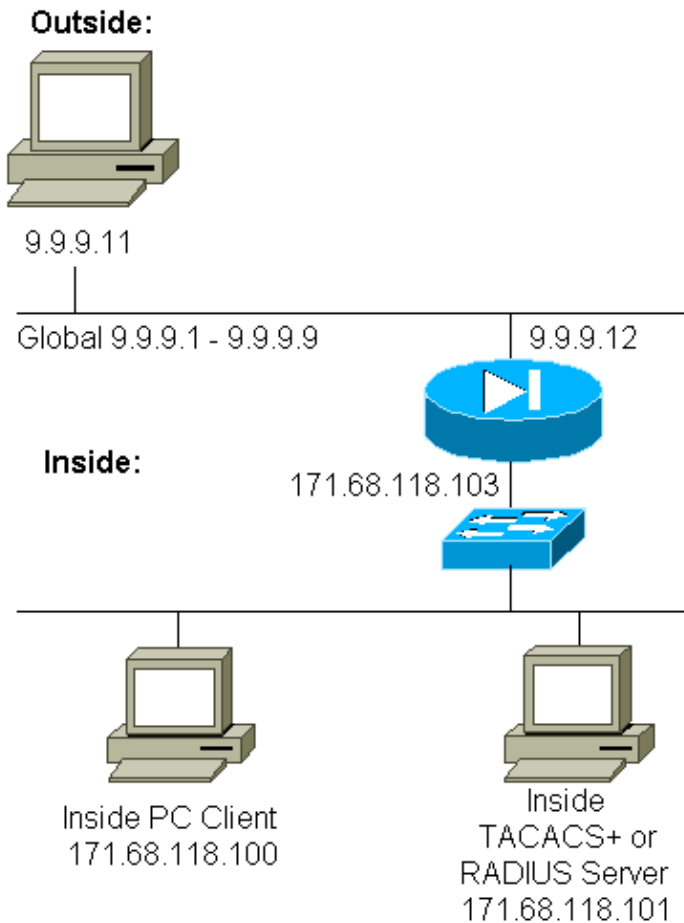
There are no specific requirements for this document.

Components Used

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

Network Diagram

This document uses this network setup:



PIX Configuration

```
pix2# write terminal
Building configuration
: Saved
:
PIX Version 4.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd OnTrBUG1Tp0edmkr encrypted
hostname pix2
fixup protocol http 80
fixup protocol smtp 25
no fixup protocol ftp 21
no fixup protocol h323 1720
no fixup protocol rsh 514
no 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
names
pager lines 24
logging console debugging
no logging monitor
logging buffered debugging
logging trap debugging
logging facility 20
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto
ip address outside 9.9.9.12 255.255.255.0
ip address inside 171.68.118.103 255.255.255.0
ip address 0.0.0.0 0.0.0.0
arp timeout 14400
global (outside) 1 9.9.9.1-9.9.9.9 netmask 255.0.0.0
static (inside,outside) 9.9.9.10 171.68.118.100 netmask 255.255.255.255 0 0
conduit permit icmp any any
conduit permit tcp host 9.9.9.10 eq telnet 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
radius-server (inside) host 171.68.118.101 cisco timeout 10
!

!--- The focus of concern is with hosts on the inside network
!--- accessing a particular outside host.

!
aaa authentication any outbound 171.68.118.0 255.255.255.0 9.9.9.11
255.255.255.255 tacacs+|radius
!

!--- It is possible to be less granular and authenticate
!--- all outbound FTP, HTTP, Telnet traffic with:

aaa authentication ftp outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
tacacs+|radius
aaa authentication http outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
tacacs+|radius
aaa authentication telnet outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
tacacs+|radius
!

!--- Accounting records are sent for
!--- successful authentications to the TACACS+ or RADIUS server.

!
aaa accounting any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs+|radius
!
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
telnet 171.68.118.100 255.255.255.255
mtu outside 1500

```

```
mtu inside 1500
mtu 1500
Smallest mtu: 1500
floodguard 0
tcpchecksum silent
Cryptochecksum:be28c9827e13baf89a937c617cfe6da0
: end
[OK]
```

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. 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. Also, depending on the response, it opens or denies the connection. These six users could do FTP, Telnet, or HTTP.

However, assume 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 need 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 you try 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 that 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 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. It then forwards this to the authentication server. PIX syslog and/or server debugs show this phenomenon. If Telnet and FTP seem to work normally, but HTTP connections do not, this is the reason.

Server Configurations Used for All Scenarios

In the TACACS+ server configuration examples, if only authentication is on, users "all", "telnetonly", "httponly", and "ftponly" all work. In the RADIUS server configuration examples, user "all" works.

When authorization is added to the PIX, in addition to sending the username and password to the TACACS+ authentication server, the PIX sends commands (Telnet, HTTP, or FTP) to the TACACS+ server. The TACACS+ server then checks to see if that user is authorized for that command.

In a later example, the user at 171.68.118.100 issues the command **telnet 9.9.9.11**. When this is received at the PIX, the PIX passes the username, password, and command to the TACACS+ server for processing.

So with authorization on in addition to authentication, user "telnetonly" can perform Telnet operations through the PIX. However, users "httponly" and "ftponly" cannot perform Telnet operations through the PIX.

(Again, authorization is not supported with RADIUS due to the nature of the protocol specification).

Cisco Secure UNIX TACACS+ Server Configuration

Cisco Secure 2.x

- User stanzas are displayed here.
- Add the PIX IP address or fully-qualified domain name and key to CSU.cfg.

```
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. The user stanza appears as seen here:

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

Cisco Secure NT 2.x RADIUS

The Sample Configurations section of the CiscoSecure 2.1 online and web documentation describes setup; attribute 6 (Service-Type) would be Login or Administrative.

Add the IP of the PIX in the NAS Configuration section using the GUI.

EasyACS TACACS+

The EasyACS documentation provides setup information.

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 (Telnet, for example).
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 steps 1 through 5 for each of the allowed commands (Telnet, HTTP and/or FTP, for example).
7. Add the IP of the PIX in the NAS Configuration section using the GUI.

Cisco Secure NT 2.x TACACS+

The Cisco Secure 2.x documentation provides setup information.

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 the **command** checkbox at the bottom and enter the command that you want to allow (Telnet, for example).
4. If you want to allow Telnet to specific sites, enter the IP in the argument section (for example, "permit 1.2.3.4"). To allow Telnet to all sites, click **Permit unlisted arguments**.
5. Click **Submit**.
6. Perform steps 1 through 5 for each of the allowed commands (Telnet, FTP, and/or HTTP, for example).
7. Add the IP of the PIX in the NAS Configuration section using the GUI.

Livingston RADIUS Server Configuration

Add the PIX IP and key to 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 {  
default service = permit  
login = cleartext "all"  
}  
  
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 are working before adding authentication, authorization, and accounting (AAA).
 - ◆ If you cannot pass traffic before instituting AAA, 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 then 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.

Authentication Debug Examples from PIX

PIX Debug – Good Authentication – RADIUS

This is an example of a PIX debug with good authentication:

```
109001: Auth start for user '???' from 171.68.118.100/1116 to 9.9.9.11/23
```

```
109011: Authen Session Start: user 'bill', sid 1
109005: Authentication succeeded for user 'bill'
      from 171.68.118.100/1116 to 9.9.9.11/23
109012: Authen Session End: user 'bill', sid 1, elapsed 1 seconds
302001: Built TCP connection 1 for faddr 9.9.9.11/23 gaddr 9.9.9.10/1116
      laddr 171.68.118.100/1116 (bill)
```

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

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

Note: If this is an FTP attempt, one try is allowed. For HTTP, infinite retries are allowed.

```
109001: Auth start for user '???' from 171.68.118.100/1132 to 9.9.9.11/23
109006: Authentication failed for user '' from
      171.68.118.100/1132 to 9.9.9.11/23
```

PIX Debug – Server Down – RADIUS

This is an example of a PIX debug with the server down. The user sees the username once. The server then "hangs" and asks for a password (three times).

```
109001: Auth start for user '???' from 171.68.118.100/1151 to 9.9.9.11/23
109002: Auth from 171.68.118.100/1151 to 9.9.9.11/23 failed
      (server 171.68.118.101 failed)
109002: Auth from 171.68.118.100/1151 to 9.9.9.11/23 failed
      (server 171.68.118.101 failed)
```

PIX Debug – Good Authentication – TACACS+

This is an example of a PIX debug with good authentication:

```
109001: Auth start for user '???' from 171.68.118.100/1200 to 9.9.9.11/23
109011: Authen Session Start: user 'cse', sid 3
109005: Authentication succeeded for user 'cse'
      from 171.68.118.100/1200 to 9.9.9.11/23
109012: Authen Session End: user 'cse', sid 3, elapsed 1 seconds
302001: Built TCP connection 3 for faddr 9.9.9.11/23 gaddr 9.9.9.10/1200
      laddr 171.68.118.100/1200 (cse)
```

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

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

Note: If this is an FTP attempt, one try is allowed. For HTTP, infinite retries are allowed.

```
109001: Auth start for user '???' from 171.68.118.100/1203 to 9.9.9.11/23
109006: Authentication failed for user ''
      from 171.68.118.100/1203 to 9.9.9.11/23
```

PIX Debug – Server Down – TACACS+

This is an example of a PIX debug with the server down. The user sees the username once. Immediately, the "Error: Max number of tries exceeded" message is displayed.

```
109001: Auth start for user '???' from 171.68.118.100/1212 to 9.9.9.11/23
109002: Auth from 171.68.118.100/1212 to 9.9.9.11/23 failed
```

```
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.100/1212 to 9.9.9.11/23 failed
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.100/1212 to 9.9.9.11/23 failed
(server 171.68.118.101 failed)
109006: Authentication failed for user ' ' from 171.68.118.100/1212 to 9.9.9.11/23
```

Adding Authorization

Because authorization is not valid without authentication, authorization is required for the same source and destination:

```
aaa authorization any outbound 171.68.118.0 255.255.255.0 9.9.9.11
255.255.255.255 tacacs+|radius
```

Or, if all three outbound services were originally authenticated:

```
aaa authorization http outbound 0.0.0.0 0.0.0.0 0.0.0.0
0.0.0.0 tacacs+|radius
aaa authorization ftp outbound 0.0.0.0 0.0.0.0 0.0.0.0
0.0.0.0 tacacs+|radius
aaa authorization telnet outbound 0.0.0.0 0.0.0.0 0.0.0.0
0.0.0.0 tacacs+|radius
```

Authentication and Authorization Debug Examples From PIX

PIX Debug – Good Authentication and Authorization – TACACS+

This is an example of a PIX debug with good authentication and authorization:

```
109001: Auth start for user '???' from 171.68.118.100/1218 to 9.9.9.11/23
109011: Authen Session Start: user 'telnetonly', sid 5
109005: Authentication succeeded for user 'telnetonly' from
171.68.118.100/1218 to 9.9.9.11/23
109011: Authen Session Start: user 'telnetonly', sid 5
109007: Authorization permitted for user 'telnetonly' from
171.68.118.100/1218 to 9.9.9.11/23
109012: Authen Session End: user 'telnetonly', sid 5, elapsed 1 seconds
302001: Built TCP connection 4 for faddr 9.9.9.11/23 gaddr 9.9.9.10/1218
laddr 171.68.118.100/1218 (telnetonly)
```

PIX Debug – Good Authentication, but Failure in Authorization – TACACS+

This is an example of a PIX debug with good authentication but failure in authorization:

```
109001: Auth start for user '???' from 171.68.118.100/1223 to 9.9.9.11/23
109011: Authen Session Start: user 'httponly', sid 6
109005: Authentication succeeded for user 'httponly'
from 171.68.118.100/1223 to 9.9.9.11/23
109008: Authorization denied for user 'httponly'
from 171.68.118.100/1223 to 9.9.9.11/23
```

PIX Debug – Bad Authentication, Authorization Not Attempted – TACACS+

This is an example of a PIX debug with authentication and authorization, but authorization not attempted due to bad authentication (username or password). The user sees four username/password sets. The "Error: max number of retries exceeded." message is displayed

Note: If this is an FTP attempt, one try is allowed. For HTTP, infinite retries are allowed.

```
109001: Auth start for user '???' from 171.68.118.100/1228 to 9.9.9.11/23
109006: Authentication failed for user '' from 171.68.118.100/1228
to 9.9.9.11/23
```

PIX Debug – Authentication/Authorization, Server Down – TACACS+

This is an example of a PIX debug with authentication and authorization. The server is down. The user sees username once. Immediately, the "Error: Max number of tries exceeded." is displayed.

```
109001: Auth start for user '???' from 171.68.118.100/1237 to 9.9.9.11/23
109002: Auth from 171.68.118.118.100/1237 to 9.9.9.11/23 failed
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.100/1237 to 9.9.9.11/23 failed
(server 171.68.118.101 failed)
109002: Auth from 171.68.118.100/1237 to 9.9.9.11/23 failed
(server 171.68.118.101 failed)
109006: Authentication failed for user '' from 171.68.118.100/1237
to 9.9.9.11/23
```

Add Accounting

TACACS+

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

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

```
109011: Authen Session Start: user 'telnetonly', sid 13
109005: Authentication succeeded for user 'telnetonly'
from 171.68.118.100/1299 to 9.9.9.11/23
109011: Authen Session Start: user 'telnetonly', sid 13
109007: Authorization permitted for user 'telnetonly'
from 171.68.118.100/1299 to 9.9.9.11/23
109012: Authen Session End: user 'telnetonly', sid 13, elapsed 1 seconds
302001: Built TCP connection 11 for faddr 9.9.9.11/23 gaddr 9.9.9.10/1299
laddr 171.68.118.100/1299 (telnetonly)
302002: Teardown TCP connection 11 faddr 9.9.9.11/23 gaddr 9.9.9.10/1299
laddr 171.68.118.100/1299 duration 0:00:02 bytes 112
```

The TACACS+ accounting records look like this output (these are from CiscoSecure UNIX; the records in Cisco Secure Windows may be comma-delimited instead):

```
Tue Sep 29 11:00:18 1998 redclay cse PIX 171.68.118.103
start task_id=0x8 foreign_ip=9.9.9.11
local_ip=171.68.118.100 cmd=telnet
Tue Sep 29 11:00:36 1998 redclay cse PIX 171.68.118.103
stop task_id=0x8 foreign_ip=9.9.9.11
local_ip=171.68.118.100 cmd=telnet elapsed_time=17
bytes_in=1198 bytes_out=62
Tue Sep 29 11:02:08 1998 redclay telnetonly PIX 171.68.118.103
start task_id=0x9 foreign_ip=9.9.9.11
local_ip=171.68.118.100 cmd=telnet
Tue Sep 29 11:02:27 1998 redclay telnetonly PIX 171.68.118.103
stop task_id=0x9 foreign_ip=9.9.9.11
local_ip=171.68.118.100 cmd=telnet elapsed_time=19
bytes_in=2223 bytes_out=64
```

The fields break down as seen here:

```
DAY MO DATE TIME YEAR NAME_OF_PIX USER SENDER PIX_IP START/STOP
UNIQUE_TASK_ID DESTINATION SOURCE
SERVICE <TIME> <BYTES_IN> <BYTES_OUT>
```

RADIUS

```
aaa accounting any outbound 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. Also, at the time of the "Teardown," a "stop" accounting record is sent:

```
109001: Auth start for user '???' from 171.68.118.100/1316 to 9.9.9.11/23
109011: Authen Session Start: user 'bill', sid 16
109005: Authentication succeeded for user 'bill'
      from 171.68.118.100/1316 to 9.9.9.11/23
109012: Authen Session End: user 'bill', sid 16, elapsed 1 seconds
302001: Built TCP connection 14 for faddr 9.9.9.11/23 gaddr 9.9.9.10/1316
      laddr 171.68.118.100/1316 (bill)
302002: Teardown TCP connection 14 faddr 9.9.9.11/23 gaddr 9.9.9.10/1316
      laddr 171.68.118.100/1316 duration 0:00:03 bytes 112
```

RADIUS accounting records look like this output (these are from Cisco Secure UNIX; the ones in Cisco Secure Windows are comma-delimited):

```
Mon Sep 28 10:47:01 1998
Acct-Status-Type = Start
Client-Id = 171.68.118.103
Login-Host = 171.68.118.100
Login-TCP-Port = 23
Acct-Session-Id = "0x00000004"
User-Name = "bill"
```

```
Mon Sep 28 10:47:07 1998
Acct-Status-Type = Stop
Client-Id = 171.68.118.103
Login-Host = 171.68.118.100
Login-TCP-Port = 23
Acct-Session-Id = "0x00000004"
User-Name = "bill"
Acct-Session-Time = 5
```

The fields break down as seen here:

```
Acct-Status-Type = START or STOP
Client-ID = IP_OF_PIX
Login_Host = SOURCE_OF_TRAFFIC
Login-TCP-Port = #
Acct-Session-ID = UNIQUE_ID_PER_RADIUS_RFC
User-name = <whatever>
<Acct-Session-Time = #>
```

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. As an example:

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 about 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), there will be an accounting record cut for every site accessed.

But the HTTP works differently due to the nature of the protocol. This is an example:

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.

Note the time. 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? No.

Will max-sessions or view logged-in users work here? No, because the connection time in HTTP is too short. The time between the "Built" and "Teardown" (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, the max-sessions and view logged-in users will not work due to the nature of HTTP connections.

Use of the Except Command

In our network, if we decide that one outgoing user (171.68.118.100) does not need to be authenticated, we can do this:

```
aaa authentication any outbound 171.68.118.0 255.255.255.0 9.9.9.11
255.255.255.255 tacacs+
aaa authentication except outbound 171.68.118.100 255.255.255.255 9.9.9.11
255.255.255.255 tacacs+
```

Authentication to the PIX Itself

The previous discussion is of authenticating Telnet (and HTTP, FTP) traffic through the PIX. With 4.2.2, Telnet connections to the PIX may also be authenticated. Here, we define the IPs of boxes that can Telnet to the PIX:

```
telnet 171.68.118.100 255.255.255.255
```

Then supply the Telnet password: **passwd ww**.

Add the new 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"). The PIX also requests the TACACS+ or RADIUS username and password.

Changing the Prompt the Users See

If you add the command: **auth-prompt YOU_ARE_AT_THE_PIX**, users going through the PIX will see the sequence:

```
YOU_ARE_AT_THE_PIX [at which point you enter the username]
Password:[at which point you enter the password]
```

Upon arrival at the ultimate destination, the "Username:" and "Password:" prompts will be displayed. 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.

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](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 17, 2006

Document ID: 13817
