System Management

Terms you'll need to understand:

- ✓ Network Time Protocol (NTP)
- ✓ Secure Shell (SSH)
- ✓ Telnet
- ✓ Simple Network Management Protocol (SNMP)
- ✓ Syslog server

Techniques you'll need to master:

- ✓ Setting the clock
- ✓ Configuring NTP
- ✓ Using Telnet
- ✓ Configuring for SSH
- ✓ Logging to a syslog server
- ✓ Setting syslog timestamps

This chapter covers the ways of remotely accessing the PIX firewall, how to set up time servers, SNMP traps, and the capabilities of logging system messages to remote syslog servers.

The Importance of the Date and Time

The PIX firewall has the capability of logging system messages locally or remotely. This capability helps you to track what is happening across the PIX firewall. As with any tracking, you need the ability to display the correct date and time to know when the event took place. The date and time can be set using the local commands, or they can be acquired from a time server using the Network Time Protocol (NTP).

Setting the Date and Time

Manually setting the time requires three main steps:

- Setting the date and time
- Setting the time zone
- Setting whether to use daylight saving time (DST)

The `clock set` Command

The `clock set` command is used to manually set the date and time. Use of the `clock set` command is as follows:

```
pixfirewall(config)# clock set <hh:mm:ss> {<day> <month> ¦
              <month> <day>} <year>
```

Table 7.1 clock set Command Options

Option	Function
hh:mm:ss	The hour, minute, and second expressed in 24-hour format. For example, **13:10:11** is used for 1:10 p.m.
name_of_month	Using the first three characters of the month. For example, you'd use **aug** for August.

Table 7.1 clock set Command Options *(continued)*	
Option	**Function**
day	The number of the day of the month. For example, **31** would be used for the 31st day.
year	The year expressed with four characters. For example, **2003** would be used.

The following example sets the clock:

```
pixfirewall(config)# clock set 13:01:11 31 aug 2003
pixfirewall(config)#
pixfirewall(config)# show clock
13:01:16.920 UTC Sun Aug 31 2003
pixfirewall(config)#
```

The clock timezone Command

The `clock timezone` command sets the time zone of your PIX firewall. The command syntax is as follows:

```
pixfirewall(config)# [no] clock timezone <zone> <hours> [<minutes>]
```

Table 7.2 clock timezone Command Options	
Option	**Function**
zone	The name of the time zone, such as PST or EST. The default time zone, UTC, is also know as Greenwich Mean Time (GMT).
hours	This allows you to manually offset the hours from UTC.
minutes	This allows you to manually offset the minutes from UTC.

The following example sets the clock time zone:

```
pixfirewall(config)# clock timezone PST -8 0
pixfirewall(config)#
pixfirewall(config)# show clock
05:15:16.107 PST Sun Aug 31 2003
pixfirewall(config)#
```

The previous example sets the time zone name to `PST`, with a –8 hours and 0 minutes offset from UTC time. The `timezone` command is used for display purposes only; the actual time in the PIX firewall is UTC time.

The clock summer-time Command

The `clock timezone` command set supports the DST feature, which is turned off by default. The following is the command syntax:

```
pixfirewall(config)# [no] clock summer-time <zone> recurring
                [<week> <weekday> <month> <hh:mm>
                <week> <weekday> <month> <hh:mm>] [<offset>]

pixfirewall(config)# [no] clock summer-time <zone> date {<day>
                <month> ¦ <month> <day>} <year>
                <hh:mm> {<day> <month> ¦ <month> <day>}
                <year> <hh:mm> [<offset>]
```

The `summer-time` command enables you to offset the time once by using the `clock summer-time date` command or to perform the offset every year with the `clock summer-time recurring` command.

The show clock Command

The `show clock` command displays the current date and time on the firewall. By using the `detail` option, you can display the method used to set the clock, like so:

```
pixfirewall(config)# show clock [detail]
```

Table 7.3 show clock Command Option

Option	Function
detail	This option displays the source of the time, either user configured or NTP.

Listing 7.1 sets the clock and then displays the clock details:

Listing 7.1 Using the clock set Command

```
pixfirewall(config)# clock set 13:01:11 31 aug 2003
pixfirewall(config)#
pixfirewall(config)# clock timezone PST -8 0
pixfirewall(config)#
pixfirewall(config)# show clock
05:01:53.671 PST Sun Aug 31 2003
pixfirewall(config)#
pixfirewall(config)# show clock detail
05:01:59.941 PST Sun Aug 31 2003
Time source is user configuration
pixfirewall(config)#
```

The clear clock Command

The `clear clock` command removes user configuration from the clock to set the display time back to UTC time zone, and it also removes any DST settings.

Network Time Protocol

Network Time Protocol (NTP) servers enable computers and devices such as the PIX firewall to synchronize their internal clocks with a centralized timing server. NTP works off a hierarchy in which one master clock server dictates the time settings and sends them down to several NTP servers, which synchronize with the master server. These lower NTP servers help to balance the load for hundreds or thousands of possible NTP clients looking to synchronize their clocks. The PIX firewall can become an NTP client, allowing NTP to set the clock instead of manually configuring it with the `clock` command. Figure 7.1 displays a simple NTP hierarchy.

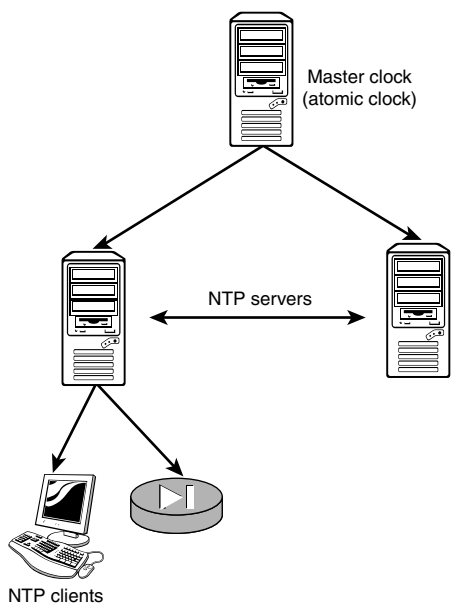

Figure 7.1 An NTP hierarchy.

Configuring NTP Clients on the PIX

To configure the PIX firewall as an NTP client, the use of several commands might be necessary. The basic NTP command set designates the NTP server

itself. If security is needed, a second set of commands is required to configure authentication keys.

The ntp server Command

The `ntp server` command enables you to designate the NTP server; its syntax is as follows:

```
pixfirewall(config)# [no] ntp server <ip_address> [key <number>]
                source <if_name> [prefer]
```

Table 7.4 ntp server Command Options

Option	Function
ip_address	The IP address of the NTP server
key number	A number, between 1 and 4,294,967,295, used to authenticate with the NTP server
if_name	The name of the interface on which the NTP server resides
prefer	Allows you to define a preference for a specific time server

Listing 7.2 configures the PIX to use three possible time servers and to give preference to the last time server for synchronizing time.

Listing 7.2 NTP Server Configuration Example

```
pixfirewall(config)# ntp server 192.168.1.100 source inside
pixfirewall(config)# ntp server 192.168.1.101 source inside
pixfirewall(config)# ntp server 192.168.1.102 source inside prefer
pixfirewall(config)#
pixfirewall(config)# show ntp
ntp server 192.168.1.100 source inside
ntp server 192.168.1.101 source inside
ntp server 192.168.1.102 source inside prefer
pixfirewall(config)#
pixfirewall(config)# show clock detail
14:17:31.014 UTC Sun Aug 31 2003
Time source is NTP
pixfirewall(config)#
```

In Listing 7.2, the `show ntp` command displays the configured NTP servers and the `show clock detail` displays the time source as being NTP rather than user configured.

NTP Authentication Commands

In secure environments, the NTP data can be sent using authentication between the NTP server and the PIX, allowing an MD5 hash against the time information passed. To do so, the following commands are required:

- `ntp authenticate`
- `ntp trusted-key <number>`
- `ntp authentication-key <number> md5 <value>`

The ntp authenticate Command

The `ntp authenticate` command enables authentication for NTP communications. When this command is used, the PIX and the NTP server must authenticate to allow the PIX firewall to accept the NTP information.

The ntp trusted-key Command

The `ntp trusted-key` command sets a number that must match in the `ntp server` command's `key` option. This same value must be sent by the NTP server in every packet for the PIX to accept the NTP information.

The ntp authentication-key Command

The `ntp authentication-key` command enables you to match an MD5 string with an NTP server. This match is made with the `number` option that corresponds to an `ntp trusted-key` command with the name `number`. In Listing 7.3, the NTP server is using `123` as its key and `timebandits` as the MD5 algorithm string. Listing 7.3 displays the commands used to create a secure connection.

Listing 7.3 Example of Configuring Secure NTP

```
pixfirewall(config)# ntp server 192.168.1.100 key 123 source inside
pixfirewall(config)# ntp authenticate
pixfirewall(config)# ntp trusted-key 123
pixfirewall(config)# ntp authentication-key 123 md5 timebandits
pixfirewall(config)#
```

MD5 is used to hash the NTP information and allow secure NTP traffic to be passed between the PIX and the NTP server.

Displaying NTP Information

Now that the PIX firewall is configured for NTP, the following three commands will enable you to verify its operational status:

- `show ntp`
- `show ntp status`
- `show ntp associations [detail]`

The show ntp Command

The `show ntp` command displays the current NTP configurations. The following example displays the NTP configuration created in Listing 7.3:

```
pixfirewall(config)# show ntp
ntp authentication-key 123 md5 ********
ntp authenticate
ntp trusted-key 123
ntp server 192.168.1.100 key 123 source inside
pixfirewall(config)#
```

The show ntp status Command

The `show ntp status` command displays the current clock status, like so:

```
pixfirewall(config)# show ntp status

Clock is synchronized, stratum 5, reference is 192.168.1.100
nominal freq is 99.9967 Hz, actual freq is 99.9967 Hz, precision is 2**6
reference time is a13124b9.46c2936b (06:28:16.000 UTC Thu Feb 7 2036)
clock offset is 0.3213 msec, root delay is 52.32 msec
root dispersion is 32.1 msec, peer dispersion is 4.4 msec
pixfirewall(config)#
```

The previous status shows the IP address of the NTP server as `192.168.1.100`.

The show ntp associations Command

The `show ntp associations` command displays information about the servers you have configured. Here is an example of the command:

```
pixfirewall(config)# show ntp associations

address         ref clock     st  when  poll reach  delay  offset    disp
*~192.168.1.100 0.0.0.0       5   30    64   377    5.0    -3.00     4.2.
* master (synced), # master (unsynced), + selected, - candidate,
~ configured
```

Notice the ledger that is displayed with the command. The asterisk symbol (*) designates that the master has synced.

Accessing the PIX

You can access the PIX firewall in several ways, such as using console ports, Telnet, Secure Shell (SSH), and HTTP. All these ways enable you to configure and manage the firewall, but by default only console port access is permitted. Figure 7.2 displays the preferred methods of access and shows that console access from the outside is allowed only when using SSH.

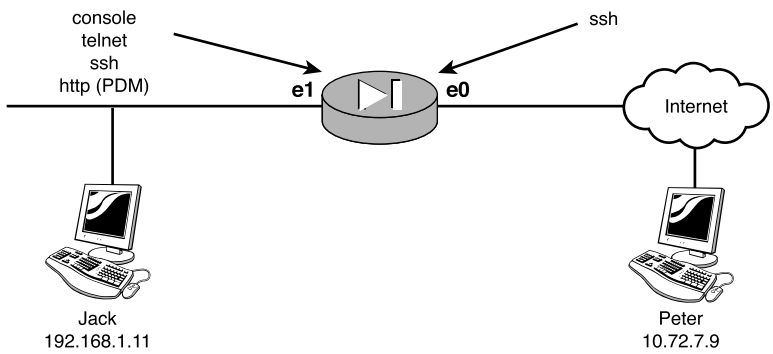

Figure 7.2 Accessing the PIX.

The Console Port

The console port allows access for a local serial connection connected directly into the PIX firewall. Procedures such as password breaking and loading new images are recommended via this connection point, but the physical distance a technician can be from the firewall is limited. Chapter 3, "Basics of the PIX Firewall," describes how to connect to the PIX via the console cable.

Telnet

Telnet enables you to remotely connect to the firewall using TCP/IP to create a remote console. With TCP/IP, physical distance is no longer a concern, making Telnet a convenient way to configure and manage your firewall without ever getting up from your chair.

Telnet communications are carried out in clear text. So, if hackers are sniffing the network, they could intercept passwords or configuration information about your PIX. Telnet access is therefore not recommended from the outside interface.

To enable Telnet access to the PIX, you must first use the `telnet` command to specify which IP addresses are allowed access. The following is the `telnet` command syntax:

```
pixfirewall(config)# [no] telnet <local_ip> [<mask>] [<if_name>]
```

Table 7.5 telnet Command Options	
Option	**Function**
local_ip	The subnet or IP address allowed to Telnet into the PIX.
mask	The optional mask allows you to specify an exact host with 255.255.255.255 or a subnet with a mask such as 255.255.255.0.
if_name	The name of the interface to accept the Telnet access.

The following three examples all allow 192.168.1.11 Telnet access on the inside interface to the PIX firewall:

```
pixfirewall(config)# telnet 192.168.1.11
```

or

```
pixfirewall(config)# telnet 192.168.1.11 255.255.255.255
```

or

```
pixfirewall(config)# telnet 192.168.1.11 255.255.255.255 inside
```

The following command allows all addresses on the inside interface Telnet access:

```
pixfirewall(config)# telnet 0.0.0.0 0.0.0.0 inside
```

Telnet also has the **who** command, which displays the current active Telnet sessions, and the **kill** command, which forces a Telnet session to disconnect.

Secure Shell

Secure Shell, like Telnet, allows remote console connections; however, with SSH, the connections are secure. SSH provides encryption of traffic from the PIX to the client, creating a secure environment in which to manage your PIX. To create this secure environment, you must create a public key and private RSA keys. Four main steps are required to configure SSH:

1. Configure a hostname.
2. Configure a domain name.
3. Create a public and private RSA key pair.
4. Specify which IP addresses are allowed SSH access.

When connecting to the PIX using SSH, you are prompted to enter a username and password. Cisco uses the username **pix**, which can't be changed, and the current Telnet password for these prompts. The default Telnet password is **cisco**, in all lowercase.

The hostname Command

SSH requires a hostname to be configured; the command shown here configures a hostname for the PIX:

```
Cisco(config)# hostname pixfirewall
pixfirewall(config)#
```

The domain-name Command

The PIX firewall needs a domain name that will be used inside the RSA key pairs. After you generate the keys, be sure you never change the domain name of the PIX; otherwise, you will have to regenerate the RSA keys. The following command sets the domain name to `newman.cla`:

```
pixfirewall(config)# domain-name newman.cla
pixfirewall(config)#
pixfirewall(config)# show domain-name
domain-name newman.cla
pixfirewall(config)#
```

The hostname and domain name are combined to form a fully qualified domain name (FQDN) that is used during key generation. For example, the FQDN in the previous example would be **pixfirewall.newman.cla**.

The ca generate rsa key Command

The `ca generate rsa key` command creates a pair of keys that are used to help create a secure connection between the client and the PIX. The values you used for the hostname and domain name are used inside the keys and should not be changed after the keys are generated. You can create modulus sizes for 512, 768, 1024, or 2048 bits. Also note that this command can take quite some time to execute.

Use the **ca zeroize rsa** command to remove any current RSA key pairs from the PIX.

The following commands are needed to create a pair of keys:

```
pixfirewall(config)# ca zeroize rsa
pixfirewall(config)#
pixfirewall(config)# ca generate rsa key 1024
For <key_modulus_size> >= 1024, key generation could
  take up to several minutes. Please wait.
..
pixfirewall(config)#
```

The ssh Command

The `ssh` command is used to define which IP addresses are allowed access to the Secure Shell console on the PIX firewall. The `ssh` command also defines the idle timeout of an SSH connection, like so:

```
pixfirewall(config)# [no] ssh <local_ip> [<mask>] [<if_name>]
```

Table 7.6 ssh Command Options

Option	Function
local_ip	The subnet or IP address allowed to SSH into the PIX.
mask	The optional **mask** allows you to specify an exact host with 255.255.255.255 or a subnet with a mask such as 255.255.255.0.
if_name	The name of the interface to accept the Telnet access.

The following example allows SSH secure access to 10.72.7.9 on the outside interface:

```
pixfirewall(config)# ssh 10.72.7.9 255.255.255.255 outside
pixfirewall(config)#
pixfirewall(config)# show ssh
10.72.7.9 255.255.255.255 outside
pixfirewall(config)#
```

The `ssh timeout` command can be used to limit the idle timeout for SSH sessions. The command example shown here sets the timeout to 10 minutes:

```
pixfirewall(config)# ssh timeout 10
pixfirewall(config)#
pixfirewall(config)# show ssh timeout
ssh timeout 10 minutes
pixfirewall(config)#
```

When you first connect to the PIX firewall, you might see a prompt with periods (.). This means the firewall is busy generating server keys and it could take several seconds before you are prompted for username and password.

Displaying and Saving SSH Information

After you have configured SSH, four more commands are available that you can use to verify its operation and disconnect users. They are

- `show ca mypubkey rsa`
- `ca save all`
- `show ssh sessions`
- `disconnect ssh session`

The `show ca mypubkey rsa` Command

The `show ca mypubkey rsa` command enables you to view the public key that was generated with your hostname and domain name. The command syntax is shown here:

```
pixfirewall(config)# show ca mypubkey rsa
% Key pair was generated at: 09:05:34 UTC Aug 31 2003
Key name: pixfirewall.newman.cla
 Usage: General Purpose Key
 Key Data:
  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00c5af11
  97e073ae ece530d1 cfea4649 84521282 768557e3 c1bb1315 8f6627cc 50224607
  14b1b9cd bf7a9c61 3e28d997 ea92b816 c04c63fd 0751748e 588cbcd2 0659675b
  ece86f2b 6592bc39 f707de5e b040e889 cc350b03 ab1a8582 ca329402 31ce17a3
  26a4c8be 3c72cd25 a80612d6 19e7419f afa68301 6c2c7682 d26a39c7 6b020301
  0001
pixfirewall(config)#
```

The `ca save all` Command

After you have generated your key, you must execute the `ca save all` command to save the key to flash memory. The following displays the command:

```
pixfirewall(config)# ca save all
```

The `ca save all` command might take several seconds to save, so be patient.

The `show ssh sessions` Command

The `show ssh sessions` command can be used to show who is currently connected to the PIX. Following is an example of this command:

```
pixfirewall(config)# show ssh sessions

Session ID      Client IP       Version Encryption      State   Username
    0           192.168.1.11      1.5     DES             6       pix
pixfirewall(config)#
```

The ssh disconnect session Command

After you have viewed who has an active SSH session using the `show ssh sessions` command, you can use the `ssh disconnect session` command to drop a specific session. The following is an example of this command:

```
pixfirewall(config)# show ssh sessions

Session ID      Client IP       Version Encryption      State   Username
    0           192.168.1.11      1.5     DES             6       pix
pixfirewall(config)#
pixfirewall(config)# ssh disconnect session 0
pixfirewall(config)# show ssh sessions
pixfirewall(config)#
```

HTTP PDM Access

The PIX firewall allows several methods of console access, but it also has a Web browser interface that can be used to monitor and configure the firewall. This interface is called PIX Device Manager (PDM). The PDM interface Web pages are hosted from PIX firewalls and downloaded to client browsers that support HTTPS (secure socket layer). The PIX firewall must have the HTTP server feature enabled to host the PDM Web pages. The following two steps are needed to configure HTTP access:

1. Turn on the HTTP server capability.
2. Specify which hosts can connect using HTTP.

The http server Command

To allow the clients to access the system using HTTP browsers you first must use the `http server enable` command to turn on the service. Here's an example of the command:

```
pixfirewall(config)# http server enable
```

The http Command

Now that the PIX is enabled to host the PDM interface, the next step, as in Telnet, is to specify which hosts can connect to the PIX using HTTP. The `http` command's syntax is as follows:

```
pixfirewall(config)# [no] http <local_ip> [<mask>] [<if_name>]
```

Table 7.7 http Command Options	
Option	**Function**
local_ip	The subnet or IP address allowed to use HTTP to access the PIX.
mask	The optional mask allows you to specify an exact host with 255.255.255.255 or a subnet with a mask such as 255.255.255.0.
if_name	The name of the interface on which to accept the HTTP access.

The first example shown here allows 192.168.1.11 HTTP access on the inside interface, whereas the second example allows HTTP access to the PIX for all addresses on the 192.168.1.0 subnet:

```
pixfirewall(config)# http 192.168.1.11 255.255.255.255 inside
```

and

```
pixfirewall(config)# http 192.168.1.0 0.0.0.0 inside
```

You can use the `show http` command to display what has been configured, like so:

```
pixfirewall(config)# show http
http server enabled
192.168.1.11 255.255.255.255 inside
0.0.0.0 0.0.0.0 inside
pixfirewall(config)#
```

The PIX Device Manager is covered in more detail in later chapters.

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) was designed to help centrally manage devices using network management stations (NMSs). These stations can poll information, accept events from devices known as *traps*, and even configure the devices remotely. Devices such as hubs, routers, printers, firewalls, and even Microsoft computers can allow an NMS to collect information about them.

SNMP on the PIX

The three main versions of SNMP are 1, 2, and 3. The PIX firewall supports only versions 1 and 2. The PIX also supports only the reading of information, meaning you cannot remotely configure the PIX firewall using NMS as you can with other devices.

By default, SNMP is enabled on the PIX with a community name of `public`. Any NMS can read information about the PIX. Therefore, to provide some basic security, you should change the default community name to something other than `public`.

Configuring SNMP

Listing 7.4 is an example of configuring SNMP on the PIX firewall. The community setting should be the same as on the NMS, so that information can be polled from the PIX firewall. The location and contact settings provide basic information about where and who to contact about this device. The enable traps allow messages to be sent from the PIXs to the NMS. Finally, the host setting defines where to send the SNMP traps (which is the IP address of the NMS server).

Listing 7.4 Configuring SNMP

```
pixfirewall(config)# snmp-server community myarea
pixfirewall(config)# snmp-server location oregon
pixfirewall(config)# snmp-server contact Mr. Newman
pixfirewall(config)# snmp-server enable traps
pixfirewall(config)# snmp-server host inside 192.168.1.11
pixfirewall(config)# show snmp
snmp-server host inside 192.168.1.11
snmp-server location oregon
snmp-server contact Mr. Newman
snmp-server community myarea
snmp-server enable traps
pixfirewall(config)#
```

Logging PIX Firewall Information

The PIX firewall enables you to log just about every type of event that takes place on the device. Events such as changing passwords, ACL hits, debug events, or even when someone just views the log itself can all be recorded.

Most of the logging commands in the following sections contain a severity level setting. The severity level setting enables you to specify how much detail you want to log.

Severity Levels

The PIX contains several logging security levels that help determine how much information should be logged. The higher the security level number, the more detail that is logged. Table 7.8 displays the eight severity level settings.

Table 7.8 PIX Logging Severity Levels		
Number	**Name**	**Description**
0	Emergencies	The system is becoming unstable.
1	Alerts	Take immediate action.
2	Critical	Critical conditions.
3	Errors	Error messages.
4	Warnings	Warning messages.
5	Notifications	Normal but significant conditions.
6	Informational	Information messages.
7	Debugging	Log debug messages, FTP commands, and WWW URLs.

If you select severity level 3 in the logging command, level 3 and all the levels below it, including levels 2, 1, and 0, will be logged.

Similar to most Cisco products, the PIX can log information to several locations simultaneously. Figure 7.3 shows some of the locations where information can be logged:

- Internal buffer
- Console port
- SNMP management stations
- Syslog servers

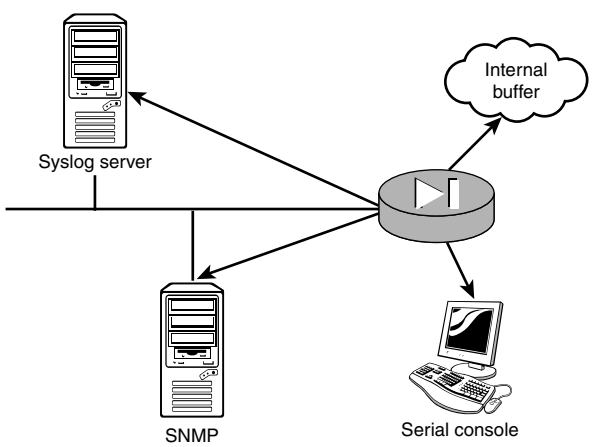

Figure 7.3 Places to log information.

Internal Buffers

You can log information to internal buffers maintained in RAM. The following commands enable this location for logging:

```
pixfirewall(config)# logging on
pixfirewall(config)# logging buffered 4
pixfirewall(config)#
```

The `logging on` command enables logging, and the `logging buffered 4` command enables logging severity level 4 messages to the internal buffer. Severity levels were described previously in Table 7.8.

The **show logging** command displays the internal buffer messages, whereas the **clear logging** command flushes the local logging buffer.

Console Port Logging

Logging to the console port enables your serial connection to display the messages being generated. Although this is fun to watch, it usually shouldn't be left on for too long. The following commands enable console logging:

```
pixfirewall(config)# logging on
pixfirewall(config)# logging console 4
pixfirewall(config)#
```

SNMP Management Station

By using the `logging history` command, you can send syslog traps to an SNMP management station, like so:

```
pixfirewall(config)# logging on
pixfirewall(config)# logging history 4
pixfirewall(config)#
```

Syslog Servers

Syslog servers are typically the primary location to log data. These are remote servers that can store your log messages to disk or other methods of storage. Syslog server software is freely available from several vendors, including Cisco. After the software is installed on a remote computer, you'll need to configure your PIX.

To enable messages to be sent to a syslog server, the `logging host` command needs to be executed. The following is the command syntax:

```
pixfirewall(config)# [no] logging host [<in_if>] <l_ip> [tcp¦udp/port#]
```

Table 7.9 logging host Command Options

Option	Function
in_if	This is the interface name the messages will exit.
l_ip	This is the IP address of the host.
tcpludp	You can specify TCP or UDP. TCP helps to guarantee your messages are delivered. This option also requires a port number.

The following example enables logging to a remote syslog server with an IP address of 192.168.1.15 and specifies that each message sent should have a timestamp value appended to it:

```
pixfirewall(config)# logging host inside 192.168.1.15
pixfirewall(config)# logging on
pixfirewall(config)# logging timestamp
pixfirewall(config)#
```

Use the **logging host** command to direct log messages to a remote syslog server.

General Logging Commands

Several other logging commands are available. Table 7.9 displays a few of the most common commands.

Table 7.10 Logging Command Options

Command	Description
logging on	Enables logging
logging timestamp	Works with syslog servers and adds a timestamp to each message to make them unique
logging monitor	Used to set which messages are sent to Telnet sessions
logging trap	Sets log levels for syslog traps
logging standby	Allows the standby PIX to send messages to the syslog server
clear logging	Clears all the log messages in the internal buffers
show logging	Displays the current logging settings and the messages located in the internal buffers

The **logging timestamp** command places a timestamp on messages before they are sent to a syslog server.