

PIX/ASA: Monitor and Troubleshoot Performance Issues

[TAC Notice: What's Changing on TAC Web](#)

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[Troubleshoot](#)

[Speed and Duplex Settings](#)

[CPU Utilization](#)

[High Memory Utilization](#)

[PortFast, Channeling, and Trunking](#)

[Network Address Translation \(NAT\)](#)

[Syslogs](#)

[Reverse DNS Lookups](#)

[show Commands](#)

[show cpu usage](#)

[show traffic](#)

[show perfmon](#)

[show blocks](#)

[show memory](#)

[show xlate](#)

[show conn count](#)

[show interface](#)

[show processes](#)

[Command Summary](#)

[NetPro Discussion Forums - Featured Conversations](#)

[Related Information](#)

Help us help you.

Please rate this document.

Excellent

Good

Average

Fair

Poor

This document solved my problem.

Yes

No

Just browsing

Suggestions for improvement:

(256 character limit)

Introduction

This document describes PIX/ASA commands that you can use to monitor and troubleshoot the performance of a Cisco PIX 500 Series/ASA 5500 Security Appliance.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on the Cisco PIX Firewall Software Version 6.2(1) and above.

Note: The information in this document can also be used with the Cisco ASA 5500 Series Security Appliance running 7.x and above version.

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

Conventions

Refer to [Cisco Technical Tips Conventions](#) for more information on document conventions.

Troubleshoot

In order to troubleshoot performance issues, check the basic areas described in this section.

Note: If you have the output of the **show** command from your Cisco device, you can use the [Output Interpreter Tool](#) ([registered](#) customers only) in order to display potential issues and fixes. The Output Interpreter Tool supports certain **show** commands. If you use the Output Interpreter Tool, you must be a [registered customer](#), you must be logged in to your Cisco account, and you must have JavaScript enabled within your browser.

Speed and Duplex Settings

The security appliance is preconfigured to autodetect the speed and duplex settings on an interface. However, several situations exist that can cause the autonegotiation process to fail, which results in either speed or duplex mismatches (and performance issues). For mission-critical network infrastructure, Cisco manually hardcodes the speed and duplex on each interface so there is no chance for error. These devices generally do not move around, so if you configure them properly, you should not need to change them.

On any network device, link speed can be sensed, but duplex must be negotiated. If two network devices are configured to autonegotiate speed and duplex, they exchange frames (called Fast Link Pulses, or FLPs) that advertise their speed and duplex capabilities. In order to a link partner that is not aware, these pulses are similar to regular 10 Mbps frames. In order to a link partner that can decode the pulses, the FLPs contain all the speed and duplex settings that the link partner can provide. The station that receives the FLPs acknowledges the frames, and the devices mutually agree on the highest speed and duplex settings that each can achieve. If one device does not support autonegotiation, the other device receives the FLPs and transitions to parallel detection mode. In order to sense the speed of the partner, the device listens to the length of pulses, and then sets the speed accordingly. The problem arises with the duplex setting. Since duplex must be negotiated, the device that is set to autonegotiate cannot determine the settings on the other device, so it defaults to half-duplex, as stated in the IEEE 802.3u standard.

For example, if you configure the PIX interface for autonegotiation and connect it to a switch that is hardcoded for 100 Mbps and full-duplex, the PIX sends out FLPs. However, the switch does not respond because it is hardcoded for speed and duplex and does not participate in autonegotiation. Because it receives no response from the switch, the PIX transitions into parallel detection mode and senses the length of the pulses in the frames that the switch sends out. That is, the PIX senses that the switch is set to 100 Mbps, so it sets the interface speed accordingly. However, because the switch does not exchange FLPs, the PIX cannot detect if the switch can run full-duplex, so the PIX sets the interface duplex to half-duplex, as stated in the IEEE 803.2u standard. Since the switch is hardcoded to 100 Mbps and full-duplex, and the PIX has just autonegotiated to 100 Mbps and half-duplex (as it should), the result is a duplex mismatch that can cause severe performance problems.

A speed or duplex mismatch is most frequently revealed when error counters on the interfaces in question increase. The most common errors are frame, cyclic redundancy checks (CRCs), and runts. If these values increment on your interface, either a speed/duplex mismatch or a cabling issue occurs. You must resolve this issue before you continue.

Example

```
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 00d0.b78f.d579
  IP address 192.168.1.1, subnet mask 255.255.255.0
```

```
MTU 1500 bytes, BW 100000 Kbit half duplex
  7594 packets input, 2683406 bytes, 0 no buffer
Received 83 broadcasts, 153 runts, 0 giants
378 input errors, 106 CRC, 272 frame, 0 overrun, 0
ignored, 0 abort
  2997 packets output, 817123 bytes, 0 underruns
  0 output errors, 251 collisions, 0 interface resets
  0 babbles, 150 late collisions, 110 deferred
```

CPU Utilization

If you noticed the CPU utilization is high, follow these steps in order to troubleshoot:

1. Verify that the connection count in **show xlate count** is low.
2. Verify that the memory block is normal.
3. Verify that the number of ACLs is higher.
4. Issue the **show memory detail** command, and verify that the memory used by the PIX is normal utilization.
5. Verify that the counts in **show processes cpu-hog** and **show processes memory** are normal.
6. Any host present inside or outside the security appliance can generate the malicious or mass traffic that can be a broadcast/multicast traffic and cause the high CPU utilization. In order to resolve this issue, configure an access list to deny the traffic between the hosts (end to end) and check the [usage](#).
7. Check the duplex and speed settings in PIX interfaces. The mismatch setting with the remote interfaces can increase the CPU utilization.

This example shows the higher number in *input error* and *overruns* due to the speed mismatch. Use the **show interface** command in order to verify the errors:

```
pix#show int e1
interface ethernet1 "inside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0050.54ff.d053
  IP address 172.16.1.5, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 100000 Kbit full duplex
    154755357 packets input, 3132291269 bytes, 0 no
```

```
buffer
    Received 5352738 broadcasts, 0 runts, 0 giants
    7182 input errors, 0 CRC, 0 frame, 7182
overrun, 0 ignored, 0 abort
    2595913856 packets output, 3842928626 bytes, 0
underruns
    0 output errors, 0 collisions, 0 interface
resets
    0 babbles, 0 late collisions, 0 deferred
```

In order to resolve this issue, set speed as *auto* to the corresponding interface.

Note: Cisco recommends that you enable the [ip verify reverse-path interface](#) command on all the interfaces as it will drop packets that do not have a valid source address, which results in less CPU usage.

8. Another reason for high CPU usage can be due to too many multicast routes. Issue the [show mroute](#) command in order to check if PIX/ASA receives too many multicast routes.
9. Use the [show local-host command](#) in order to see if the network experiences a denial-of-service attack, which can indicate a virus attack in the network.

High Memory Utilization

Here are some possible causes and resolutions for high memory utilization:

- **Event logging:** Event logging can consume large amounts of memory. In order to resolve this issue, install and log all events to an external server, such as a syslog server.
- **Memory Leakage:** A known issue in the security appliance software can lead to high memory consumption. In order to resolve this issue, upgrade the security appliance software.
- **Debugging Enabled:** Debugging can consume large amounts of memory. In order to resolve this issue, disable debugging with the `undebg all` command.
- **Blocking Ports:** Blocking ports on the outside interface of a security appliance cause the security appliance to consume high amounts of memory to block the packets through the specified ports. In order to resolve this issue, block the offending traffic at the ISP end.
- **:** The threat detection feature consists of different levels of statistics gathering for various threats, as well as scanning threat detection, which determines when a host is performing a scan. **Turn**

off this feature to consume less memory.

PortFast, Channeling, and Trunking

By default, many switches, such as Cisco switches that run the Catalyst operating system (OS), are designed to be plug-and-play devices. As such, many of the default port parameters are not desirable when a PIX is plugged into the switch. For example, on a switch that runs the Catalyst OS, default channeling is set to Auto, trunking is set to Auto, and PortFast is disabled. If you connect a PIX to a switch that runs the Catalyst OS, disable channeling, disable trunking, and enable PortFast.

Channeling, also known as Fast EtherChannel or Giga EtherChannel, is used to bind two or more physical ports in a logical group in order to increase the overall throughput across the link. When a port is configured for automatic channeling, it sends out Port Aggregation Protocol (PAgP) frames as the link becomes active in order to determine if it is part of a channel. These frames can cause problems if the other device tries to autonegotiate the speed and duplex of the link. If channeling on the port is set to Auto, it also results in an additional delay of about 3 seconds before the port starts to forward traffic after the link is up.

Note: On the Catalyst XL Series Switches, channeling is not set to Auto by default. For this reason, you should disable channeling on any switch port that connects to a PIX.

Trunking, also known by the common trunking protocols Inter-Switch Link (ISL) or Dot1q, combines multiple virtual LANs (VLANs) on a single port (or link). Trunking is typically used between two switches when both switches have more than one VLAN defined on them. When a port is configured for automatic trunking, it sends out Dynamic Trunking Protocol (DTP) frames as the link comes up in order to determine if the port that it connects to wants to trunk. These DTP frames can cause problems with autonegotiation of the link. If trunking is set to Auto on a switch port, it adds an additional delay of about 15 seconds before the port starts to forward traffic after the link is up.

PortFast, also known as Fast Start, is an option that informs the switch that a Layer 3 device is connected out of a switch port. The port does not wait the default 30 seconds (15 seconds to listen and 15 seconds to learn); instead, this action causes the switch to put the port into forwarding state immediately after the link comes up. It is important to understand that when you enable PortFast, spanning tree is not disabled. Spanning tree is still active on that port. When you enable PortFast, the switch is informed only that there is not another switch or hub (Layer 2-only device) connected at the other end of the link. The switch bypasses the normal 30-second delay while it attempts to determine if a Layer 2 loop results if it brings up that port. After the link is brought up, it still participates in spanning tree. The port sends out bridge packet data units (BPDUs), and the switch still listens for BPDUs on that port. For these reasons, it is recommended that you enable PortFast on any switch port that connects to a PIX.

Note: Catalyst OS releases 5.4 and later include the **set port host** *<mod>/<port>* command that allows you to use a single command to disable channeling, disable trunking, and enable PortFast.

Network Address Translation (NAT)

All sessions that connect through the security appliance must undergo some form of network address translation, or NAT. Each NAT or NAT Overload (PAT) session is assigned a translation slot known as an *xlate*. These *xlates* can persist even after you make changes to the NAT rules that affect them. This can lead to a depletion of translation slots or unexpected behavior or both by traffic that undergoes translation. This section explains how to view and clear *xlates* on the security appliance.

Note: Always clear *xlates* after you add, change, or remove the **aaa-server**, **access-list**, **alias**, **global**, **nat**, **route**, or **static** commands in your configuration.



Caution: A momentary interruption of the flow of all traffic through the device may occur when you globally clear *xlates* on the security appliance.

Sample PIX configuration for PAT that use the outside interface IP Address:

```
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0
```

Traffic that flows through the security appliance most likely undergoes NAT. In order to view the translations that are in use on the security appliance, issue the **show xlate** command:

```
pix#show xlate
1 in use, 1 most used
PAT Global 192.168.1.2(1) Local 10.2.2.2 ICMP id 21
```

```
pix#show xlate detail
1 in use, 1 most used
Flags: D - DNS, d - dump, I - identity, i - dynamic, n - no
random,
      r - portmap, s - static
ICMP PAT from inside:10.2.2.2/22 to outside:192.168.1.2/2
flags ri
```

Translation slots can persist after key changes are made. In order to clear current translation slots on the security appliance, issue the **clear xlate** command:

```
pix#clear xlate
```

```
pix#show xlate
0 in use, 1 most used
```

The **clear xlate** command clears all the current dynamic translation from the xlate table. In order to clear a particular IP translation, you can use the **clear xlate** command with the **global [ip address]** keyword.

Here is a sample PIX configuration for NAT:

```
global (outside) 1 10.10.10.10-10.10.10.100
nat (inside) 1 0.0.0.0 0.0.0.0
```

Observe the **show xlate** output for the translation for inside 10.2.2.2 to outside global 10.10.10.10:

```
pixfirewall#show xlate detail
2 in use, 2 most used
Flags: D - DNS, d - dump, I - identity, i - dynamic, n - no
random,
      r - portmap, s - static
NAT from inside:10.2.2.2 to outside:10.10.10.10 flags i
NAT from inside:10.5.5.5 to outside:10.10.10.11 flags i
```

Clear the translation for 10.10.10.10 global IP address:

```
pixfirewall# clear xlate global 10.10.10.10
```

In this example, the translation for inside 10.2.2.2 to outside global 10.10.10.10 is gone:

```
pixfirewall#show xlate detail
1 in use, 2 most used
Flags: D - DNS, d - dump, I - identity, i - dynamic, n - no
random,
      r - portmap, s - static
NAT from inside:10.5.5.5 to outside:10.10.10.11 flags i
```


When you clear translation slots, be sure to take into account the different types of translations:

- A *static xlate* is a persistent xlate that is created with the **static** command. In order to remove static xlates, you must remove the **static** command from the configuration. The **clear xlate** command does not remove the static translation rule. If you remove a **static** command from the configuration, preexisting connections that use the static rule can still forward traffic. Use the **clear local-host** command in order to deactivate these connections.

- A *dynamic xlate* is an xlate that is created on demand with traffic processing (through the **nat** or **global** command). The **clear xlate** command removes dynamic xlates and their associated connections. If you remove a **nat** or a **global** command from the configuration, the dynamic xlate and associated connections *might remain active*.

Syslogs

Syslogs allow you to troubleshoot issues on the PIX. Cisco offers a free syslog server for Windows NT called PIX Firewall Syslog Server (PFSS). You can download PFSS from the [Software Downloads](#) ([registered](#) customers only) page.

Several other vendors, such as [Kiwi Enterprises](#) , offer syslog servers for various Windows platforms, such as Windows 2000 and Windows XP. Most UNIX and Linux machines have syslog servers installed by default.

When you set up the syslog server, configure the PIX in order to send logs to it.

For example:

```
logging on
logging host <ip_address_of_syslog_server>
logging trap debugging
```

Note: This example configures the PIX to send Debugging (level 7) and more critical syslogs to the syslog server. Since these PIX logs are the most verbose, use them only when you troubleshoot an issue. For normal operation, configure the logging level to Warning (level 4) or Error (level 3).

If you experience an issue with slow performance, open the syslog in a text file and search for the source IP address associated with the performance issue. (If you use UNIX, you can `grep` through the syslog for the source IP address.) Check for messages that indicate the external server tried to access the internal IP address on TCP port 113 (for Identification Protocol, or Ident), but the PIX denied the packet. The message should be similar to this example:

```
%PIX-2-106001: Inbound TCP connection denied from
10.64.10.2/35969 to 172.17.110.179/113 flags SYN
```

If you receive this message, issue the **service reset inbound** command to the PIX. The PIX does not silently drop packets; instead, this command causes the PIX to immediately reset any inbound connection that is denied by the security policy. The server does not wait for the Ident packet to time out its TCP connection; instead, it immediately receives a reset packet. Refer to [PIX Performance Issues](#)

[Caused by IDENT Protocol](#) for more information about the PIX and Ident.

Reverse DNS Lookups

If you experience slow performance with the PIX, verify that you have Domain Name System Pointer (DNS PTR) records, also known as Reverse DNS Lookup records, in the authoritative DNS server for the external addresses that the PIX uses. This includes any address in your global Network Address Translation (NAT) pool (or the PIX outside interface if you overload on the interface), any static address, and internal address (if you do not use NAT with them). Some applications, such as File Transfer Protocol (FTP) and Telnet servers, may use reverse DNS lookups in order to determine where the user comes from and if it is a valid host. If the reverse DNS lookup does not resolve, then performance is degraded as the request times out.

In order to ensure that a PTR record exists for these hosts, issue the **nslookup** command from your PC or UNIX machine; include the global IP address you use to connect to the Internet.

Example

```
% nslookup 198.133.219.25
25.219.133.198.in-addr.arpa      name = www.cisco.com.
```

You should receive a response back with the DNS name of the device assigned to that IP address. If you do not receive a response, contact the person that controls your DNS in order to request the addition of PTR records for each of your global IP addresses. Refer to [Poor or Intermittent FTP/HTTP Performance Through a PIX](#) for more information about performance issues on the PIX caused by PTR records that are lost.

show Commands

show cpu usage

The **show cpu usage** command was first introduced in PIX 6.0(1) and is used to determine the traffic load placed on the PIX CPU. During peak traffic times, network surges, or attacks, the CPU usage can spike.

The PIX has a single CPU to process a variety of tasks; for example, it processes packets and prints debug messages to the console. Each process has its own purpose, and some processes require more CPU time than other processes. Encryption is probably the most CPU-intensive process, so if your PIX passes a lot of traffic through encrypted tunnels, you should consider a faster PIX, a VPN Accelerator Card (VAC) [Part # PIX-VPN-ACCEL] for the PIX, or a dedicated VPN Concentrator, such as the VPN 3000. The VAC offloads the encryption and decryption from the PIX CPU and performs it in hardware

on the card. This allows the PIX to encrypt and decrypt 100 Mbps of traffic with 3DES (168-bit encryption).

Logging is another process that can consume large amounts of system resources. Because of this, it is recommended that you disable console, monitor, and buffer logging on the PIX. You can enable these processes when you troubleshoot a problem, but disable them for day-to-day operation, especially if you run out of CPU capacity. It is also suggested that syslogging or Simple Network Management Protocol (SNMP) logging (logging history) should be set to level 5 (Notification) or lower. In addition, you can disable specific syslog message IDs with the **no logging message** `<syslog_id>` command.

PIX Device Manager (PDM) also provides a graph on the Monitoring tab that allows you to view the CPU usage of the PIX over time. You can use this graph in order to determine the load on your PIX.

The **show cpu usage** command can be used to display CPU utilization statistics.

Example

```
pixfirewall#show cpu usage
```

```
CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5
minutes: 1%
```

Description of Output

This table describes the fields in the **show cpu usage** output.

Field	Description
CPU utilization for 5 seconds	CPU utilization for the last five seconds
1 minute	Average of 5 second samples of CPU utilization over the last minute
5 minutes	Average of 5 second samples of CPU utilization over the last five minutes

show traffic

The **show traffic** command shows how much traffic that passes through the PIX over a given period of time. The results are based on the time interval since the command was last issued. For accurate results, issue the **clear traffic** command first and then wait 1-10 minutes before you issue the **show traffic** command. You could also issue the **show traffic** command and wait 1-10 minutes before you issue the command again, but only the output from the second instance is valid.

You can use the **show traffic** command in order to determine how much traffic passes through your PIX. If you have multiple interfaces, the command can help you determine which interfaces send and receive the most data. For PIX appliances with two interfaces, the sum of the inbound and outbound traffic on the outside interface should equal the sum of the inbound and outbound traffic on the inside interface.

Example

```
pixfirewall#show traffic
outside:
    received (in 124.650 secs):
        295468 packets  167218253 bytes
        2370 pkts/sec   1341502 bytes/sec
    transmitted (in 124.650 secs):
        260901 packets  120467981 bytes
        2093 pkts/sec   966449 bytes/sec
inside:
    received (in 124.650 secs):
        261478 packets  120145678 bytes
        2097 pkts/sec   963864 bytes/sec
    transmitted (in 124.650 secs):
        294649 packets  167380042 bytes
        2363 pkts/sec   1342800 bytes/sec
```

If you come close to or reach the rated throughput on one of your interfaces, you need to upgrade to a faster interface or limit the amount of traffic that goes into or out of that interface. Failure to do so can result in dropped packets. As explained in the [show interface](#) section, you can examine the interface counters in order to find out about throughput.

show perfmon

The **show perfmon** command is used to monitor the amount and types of traffic that the PIX inspects. This command is the only way to determine the number of translations (xlates) and connections (conn) per second. Connections are further broken down into TCP and User Datagram Protocol (UDP) connections. See [Description of Output](#) for descriptions of the output that this command generates.

Example

PERFMON STATS	Current	Average
Xlates	18/s	19/s
Connections	75/s	79/s
TCP Conns	44/s	49/s
UDP Conns	31/s	30/s
URL Access	27/s	30/s
URL Server Req	0/s	0/s
TCP Fixup	1323/s	1413/s
TCPIntercept	0/s	0/s
HTTP Fixup	923/s	935/s
FTP Fixup	4/s	2/s
AAA Authen	0/s	0/s
AAA Author	0/s	0/s
AAA Account	0/s	0/s

Description of Output

This table describes the fields in the **show perfmon** output.

Field	Description
Xlates	Translations built up per second
Connections	Connections established per second
TCP Conns	TCP connections per second
UDP Conns	UDP connections per second
URL Access	URLs (websites) accessed per second
URL Server Req	Requests sent to Websense and N2H2 per second (requires filter command)
TCP Fixup	Number of TCP packets that the PIX forwards per second
TCPIntercept	Number of SYN packets per second that have exceeded the embryonic limit set on a static

HTTP Fixup	Number of packets destined to port 80 per second (requires fixup protocol http command)
FTP Fixup	FTP commands inspected per second
AAA Authen	Authentication requests per second
AAA Author	Authorization requests per second
AAA Account	Accounting requests per second

show blocks

Along with the **show cpu usage** command, you can use the **show blocks** command in order to determine whether the PIX is overloaded.

Packet-Processing Blocks (1550 and 16384 Bytes)

When it comes into the PIX interface, a packet is placed on the input interface queue, passed up to the OS, and placed in a block. For Ethernet packets, the 1550-byte blocks are used; if the packet comes in on a 66 MHz Gigabit Ethernet card, the 16384-byte blocks are used. The PIX determines whether the packet is permitted or denied based on the Adaptive Security Algorithm (ASA) and processes the packet through to the output queue on the outbound interface. If the PIX cannot support the traffic load, the number of available 1550-byte blocks (or 16384-byte blocks for 66 MHz GE) hovers close to 0 (as shown in the CNT column of the command output). When the CNT column hits zero, the PIX attempts to allocate more blocks, up to a maximum of 8192. If no more blocks are available, the PIX drops the packet.

Failover and Syslog Blocks (256 Bytes)

The 256-byte blocks are mainly used for stateful failover messages. The active PIX generates and sends packets to the standby PIX in order to update the translation and connection table. During periods of bursty traffic where high rates of connections are created or torn down, the number of available 256-byte blocks may drop to 0. This drop indicates that one or more connections are not updated to the standby PIX. This is generally acceptable because the next time around the stateful failover protocol catches the xlate or connection that is lost. However, if the CNT column for 256-byte blocks stays at or near 0 for extended periods of time, the PIX cannot keep up with the translation and connection tables that are synchronized because of the number of connections per second that the PIX processes. If this happens consistently, upgrade the PIX to a faster model.

Syslog messages sent out from the PIX also use the 256-byte blocks, but they are not generally released in such a quantity that causes a depletion of the 256-byte block pool. If the CNT column shows that the

number of 256-byte blocks is near 0, ensure that you do not log at Debugging (level 7) to the syslog server. This is indicated by the logging trap line in the PIX configuration. It is recommended that you set logging to Notification (level 5) or lower, unless you require additional information for debugging purposes.

Example

```
pixfirewall#show blocks
  SIZE      MAX      LOW      CNT
    4       1600    1597    1600
   80       400     399     400
  256       500     495     499
 1550      1444    1170    1188
16384      2048    1532    1538
```

Description of Output

This table describes the columns in the **show blocks** output.

Column	Description
SIZE	The size, in bytes, of the block pool.
MAX	The maximum number of blocks available for the specified byte block pool. Note that the maximum number of blocks are carved out of memory at bootup. Typically, the maximum number of blocks does not change. The exception is for the 256-byte and 1550-byte blocks, where the PIX can dynamically create more when needed, up to a maximum of 8192.
LOW	The low watermark. This value is the lowest number of this size blocks available since the PIX was powered up, or since the last time the blocks were cleared with the clear blocks command.
CNT	The current number of blocks available for that specific size block pool.

This table describes the SIZE row values in the **show blocks** output.

SIZE Value	Description
4	Use in order to duplicate blocks that exist in DNS, Internet Security Association and Key Management Protocol (ISAKMP), URL filtering, uauth, h323, tftp, and TCP modules.
80	Use in TCP intercept in order to generate acknowledgment (ACK) packets and for failover hello messages.
256	Use for stateful failover updates, syslogging, and other TCP functions.
1550	Use in order to store Ethernet packets that are processed through the PIX.
16384	Use for only the 64-bit, 66 MHz Gigabit Ethernet cards (i82543) in the PIX 535.

show memory

The **show memory** command displays the total physical memory (or RAM) for the PIX, along with the number of bytes currently available. In order to use this information, you must first understand how the PIX uses memory. When the PIX boots, it copies the OS from Flash into RAM and runs the OS from RAM (just like routers). Next, the PIX copies the startup configuration from Flash and places it into RAM. Finally, the PIX allocates RAM in order to create the block pools discussed in the [show blocks](#) section. Once this allocation is complete, the PIX needs additional RAM only if the configuration increases in size. In addition, the PIX stores the translation and connection entries in RAM.

During normal operation, the free memory on the PIX should change very little, if at all. Typically, the only time you should run low on memory is if you are under attack and hundreds of thousands of connections go through the PIX. In order to check the connections, issue the [show conn count](#) command, which displays the current and maximum number of connections through the PIX. If the PIX runs out of memory, it eventually crashes. Prior to the crash, you might notice memory allocation failure messages in the syslog (PIX-3-211001). If you run out of memory because you are under attack, contact the [Cisco Technical Assistance Center \(TAC\)](#).

Example

```
pixfirewall#show memory
1073741824 bytes total, 1022992384 bytes free
```

show xlate

The **show xlate count** command displays the current and maximum number of translations through the PIX. A translation is a mapping of an internal address to an external address and can be a one-to-one mapping, such as Network Address Translation (NAT), or a many-to-one mapping, such as Port Address Translation (PAT). This command is a subset of the **show xlate** command, which outputs each translation through the PIX. Command output shows translations "in use," which refers to the number of active translations in the PIX when the command is issued; "most used" refers to the maximum translations that have ever been seen on the PIX since it was powered on.

Note: A single host can have multiple connections to various destinations, but only one translation. If the xlate count is much larger than the number of hosts on your internal network, it is possible that one of your internal hosts has been compromised. If your internal host has been compromised, it spoofs the source address and sends packets out the PIX.

Note: When the vpnclient configuration is enabled and the inside host sends out DNS requests, the **show xlate** command might list multiple xlates for a static translation.

Example

```
pixfirewall#show xlate count
84 in use, 218 most used
```

This example shows the output from the **show xlate detail** command with three active Port Address Translations (PATs):

```
pixfirewall(config)#show xlate detail

3 in use, 3 most used

Flags: D - DNS, d - dump, I - identity, i - inside, n - no
random,

        o - outside, r - portmap, s - static

TCP PAT from inside:10.1.1.15/1026 to
outside:192.150.49.1/1024 flags ri

UDP PAT from inside:10.1.1.15/1028 to
outside:192.150.49.1/1024 flags ri

ICMP PAT from inside:10.1.1.15/21505 to
```

```
outside:192.150.49.1/0 flags ri
```

The first entry is a TCP Port Address Translation for host-port (10.1.1.15, 1025) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The "r" flag denotes the translation is a Port Address Translation. The "i" flag denotes that the translation applies to the inside address-port.

The second entry is a UDP Port Address Translation for host-port (10.1.1.15, 1028) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The "r" flag denotes the translation is a Port Address Translation. The "i" flag denotes that the translation applies to the inside address-port.

The third entry is an ICMP Port Address Translation for host-ICMP-id (10.1.1.15, 21505) on the inside network to host-ICMP-id (192.150.49.1, 0) on the outside network. The "r" flag denotes the translation is a Port Address Translation. The "i" flag denotes that the translation applies to the inside address-ICMP-id.

The inside address fields appear as source addresses on packets that traverse from the more secure interface to the less secure interface. Conversely, they appear as destination addresses on packets that traverse from the less secure interface to the more secure interface.

show conn count

The **show conn count** command shows the current and maximum number of connections through the PIX. A connection is a mapping of Layer 4 information from an internal address to an external address. Connections are built up when the PIX receives a SYN packet for TCP sessions or when the first packet in a UDP session arrives. Connections are torn down when the PIX receives the final ACK packet, which occurs when the TCP session handshake closes or when the timeout expires in the UDP session.

Extremely high connection counts (50-100 times normal) may indicate that you are under attack. Issue the **show memory** command in order to ensure that the high connection count does not cause the PIX to run out of memory. If you are under attack, you can limit the maximum number of connections per static entry and also limit the maximum number of embryonic connections. This action protects your internal servers, so they do not become overwhelmed. Refer to [Cisco Secure PIX Firewall Command References](#) for more information.

Example

```
pixfirewall#show conn count
2289 in use, 44729 most used
```

show interface

The **show interface** command can help determine duplex mismatch problems and cable issues; it can also provide further insight as to whether or not the interface is overrun. If the PIX runs out of CPU capacity, the number of 1550-byte blocks hovers close to 0. (Look at the 16384-byte blocks on the 66 MHz Gig cards.) Another indicator is the increase of "no buffers" on the interface. The no buffers message indicates that the interface is unable to send the packet to the PIX OS because there is no available block for the packet, and the packet is dropped. If an increase in no buffer levels occurs regularly, issue the **show proc cpu** command in order to check the CPU usage on the PIX. If the CPU usage is high because of a heavy traffic load, upgrade to a more powerful PIX that can handle the load.

When a packet first enters an interface, it is placed in the input hardware queue. If the input hardware queue is full, the packet is placed in the input software queue. The packet is passed from its input queue up to the PIX OS and placed in a 1550-byte block (or in a 16384-byte block on 66 MHz Gigabit Ethernet interfaces). The PIX then determines the output interface for the packet and places the packet in the appropriate hardware queue. If the hardware queue is full, the packet is placed in the output software queue. If the maximum blocks in either of the software queues are large, then the interface is overrun. For example, if 200 Mbps come into the PIX and all go out a single 100 Mbps interface, the output software queue indicates high numbers on the outbound interface, which indicates that the interface cannot handle the traffic volume. If you experience this situation, upgrade to a faster interface.

Example

```
pixfirewall#show interface
interface ethernet0 "inside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0002.b31b.99ff
  IP address 9.9.9.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 100000 Kbit full duplex
    4630 packets input, 803174 bytes, 0 no buffer
    Received 2 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0
  ignored, 0 abort
    4535 packets output, 445424 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collisions, 0 deferred
    0 lost carrier, 0 no carrier
    input queue (curr/max blocks): hardware (128/128)
software (0/1)
    output queue (curr/max blocks): hardware (0/2)
software (0/1)
```

You should also check the interface for errors. If you receive runts, input errors, CRCs, or frame errors,

it is likely that you have a duplex mismatch. (The cable could be faulty as well.) See [Speed and Duplex Settings](#) for more information on duplex issues. Remember that each error counter represents the number of packets that are dropped because of that particular error. If you see a specific counter that increments regularly, the performance on your PIX most likely suffers, and you must find the root cause of the problem.

While you examine the interface counters, note that if the interface is set to full-duplex, you should not experience any collisions, late collisions, or deferred packets. Conversely, if the interface is set to half-duplex, you should receive collisions, some late collisions, and possibly some deferred packets. The total number of collisions, late collisions, and deferred packets should not exceed 10% of the sum of the input and output packet counters. If your collisions exceed 10% of your total traffic, then the link is overutilized, and you must upgrade to full-duplex or to a faster speed (10 Mbps to 100 Mbps). Remember that collisions of 10% mean that the PIX drops 10% of the packets that go through that interface; each of these packets must be retransmitted.

Refer to the **interface** command in the [Cisco Secure PIX Firewall Command References](#) for detailed information on the interface counters.

show processes

The **show processes** command on the PIX displays all the active processes that run on the PIX at the time the command is executed. This information is useful in order to determine which processes receive too much CPU time and which processes do not receive any CPU time. In order to get this information, issue the **show processes** command twice; wait about 1 minute between each instance. For the process in question, subtract the Runtime value displayed in the second output from the Runtime value displayed in the first output. This result tells you how much CPU time (in milliseconds) the process received in that interval of time. Note that some processes are scheduled to run at particular intervals, and some processes only run when they have information to process. The 577poll process most likely has the largest Runtime value of all your processes. This is normal because the 577poll process polls the Ethernet interfaces in order to see if they have any data that needs to be processed.

Note: An examination of each PIX process is out of the scope of this document, but is mentioned briefly for completeness. Refer to [The PIX show processes Command](#) for more information about the PIX processes.

Command Summary

In summary, use the **show cpu usage** command in order to identify the load that the PIX is under. Remember that the output is a running average; the PIX can have higher spikes of CPU usage that are masked by the running average. Once the PIX reaches 80% CPU usage, the latency through the PIX slowly increases to about 90% CPU. When CPU usage is more than 90%, the PIX starts to drop packets.

If the CPU usage is high, use the **show processes** command in order to identify the processes that use the most CPU time. Use this information in order to reduce some of the time that is consumed by the intensive processes (like logging).

If the CPU does not run hot, but you believe packets are still dropped, use the **show interface** command in order to check the PIX interface for no buffers and collisions, possibly caused by a duplex mismatch. If the no buffer count increments, but the CPU usage is not low, the interface cannot support the traffic that flows through it.

If the buffers are fine, check the blocks. If the current CNT column in the **show blocks** output is close to 0 on the 1550-byte blocks (16384-byte blocks for 66 MHz Gig cards), the PIX most likely drops Ethernet packets because it is too busy. In this instance, the CPU spikes high.

If you experience trouble when you make new connections through the PIX, use the **show conn count** command in order to check the current count of connections through the PIX.

If the current count is high, check the **show memory** output in order to ensure that the PIX does not run out of memory. If memory is low, investigate the source of the connections with the **show conn** or **show local-host** command in order to verify that your network has not experienced a denial-of-service attack.

You can use other commands in order to measure the amount of traffic that passes through the PIX. The **show traffic** command displays the aggregate packets and bytes per interface, and the **show perfmon** breaks the traffic down into different types that the PIX inspects.

NetPro Discussion Forums - Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums - Featured Conversations for Security

Security: Intrusion Detection [Systems]

[IPS 4260 failover](#) - Oct 26, 2009

[100% Sensor load during some period](#) - Oct 26, 2009

[IPS 4260 sensor - NIC ports](#) - Oct 26, 2009

[Unable to took action for some sig. in IPS-4260](#) - Oct 26, 2009

[IPS Manager Express \(IME\) issue](#) - Oct 26, 2009

Security: AAA

[802.1x with EAP-TLS Fails on Wired](#) - Oct 26, 2009

[ACS Group mapping and restrictions](#) - Oct 26, 2009

[ACS using RSA keyfobs issue](#) - Oct 25, 2009

[Router VPN authenticaiton fail via windows 2008 NPS radius](#) - Oct 25, 2009

[IOS VPN authenticaiton via w2k3 IAS radius](#) - Oct 25, 2009

Security: General

[Cannot access subnet](#) - Oct 26, 2009

[How many days does CS-MARS retain device logs?](#) - Oct 26, 2009

[Smartcare and switch discovery](#) - Oct 26, 2009

[ACS Group mapping and restrictions](#) - Oct 26, 2009

[DHCP snooping](#) - Oct 26, 2009

Security: Firewalling

[FWSM : One Way Communication Issue](#) - Oct 26, 2009


[Monitoring WebVPN](#) - Oct 26, 2009

[ASA WCCP multipkle service groups same interface](#) - Oct 26, 2009

[Blocking users using mac address](#) - Oct 26, 2009

[IPS-SSM password recovery](#) - Oct 26, 2009

Related Information

- [Cisco PIX Firewall Documentation](#)
- [Cisco Secure PIX Firewall Command References](#)
- [Most Common L2L and Remote Access IPsec VPN Troubleshooting Solutions](#)
- [Cisco PIX 500 Series Security Appliances Product Support Page](#)
- [IETF Requests for Comments \(RFCs\)](#) 
- [Technical Support - Cisco Systems](#)

[Home](#)

[How to Buy](#)

[Login](#)

[Profile](#)

[Feedback](#)

[Site Map](#)

[Help](#)

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

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