

Using SNMP with the Security Appliances PIX/ASA

Document ID: 13822

Introduction

Prerequisites

- Requirements
- Components Used
- Related Products
- Conventions

SNMP Through the PIX/ASA

- Traps Outside to Inside
- Traps Inside to Outside
- Polling Outside to Inside
- Polling Inside to Outside

SNMP to the PIX/ASA

- MIB Support by Version
- Turning on SNMP in the PIX/ASA
- SNMP to the PIX/ASA – Polling
- SNMP to the PIX/ASA – Traps

SNMP Issues

- PIX Discovery
- Discover Devices Inside the PIX
- Discover Devices Outside the PIX

Version 6.2 snmpwalk of PIX

Information to Collect if You Open a TAC Case

Related Information

Introduction

You can monitor system events on the PIX using Simple Network Management Protocol (SNMP). This document describes how to use SNMP with the PIX, which includes:

- Commands to run SNMP *through* the PIX or *to* the PIX
- Sample PIX output
- Management Information Base (MIB) support in PIX Software Release 4.0 and later
- Trap levels
- syslog severity level examples
- PIX and SNMP device discovery issues

Note: The port for snmpget/snmpwalk is UDP/161. The port for SNMP traps is UDP/162.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco Secure PIX Firewall Software Releases 4.0 and later.

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

Related Products

- Cisco Adaptive Security Appliance (ASA) version 7.x

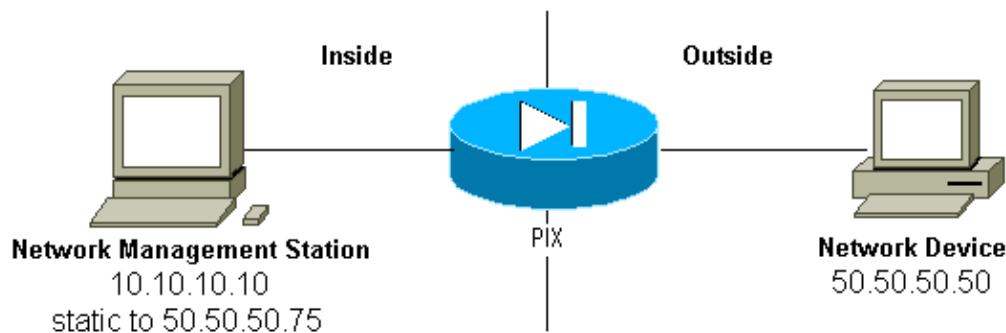
Conventions

Some lines of output and log data in this document have been wrapped for spacing considerations.

Refer to Cisco Technical Tips Conventions for more information on document conventions.

SNMP Through the PIX/ASA

Traps Outside to Inside



In order to allow traps in from 50.50.50.50 to 10.10.10.10:

```
conduit permit udp host 50.50.50.75 eq snmptrap host 50.50.50.50
static (inside,outside) 50.50.50.75 10.10.10.10 netmask 255.255.255.255 0 0
```

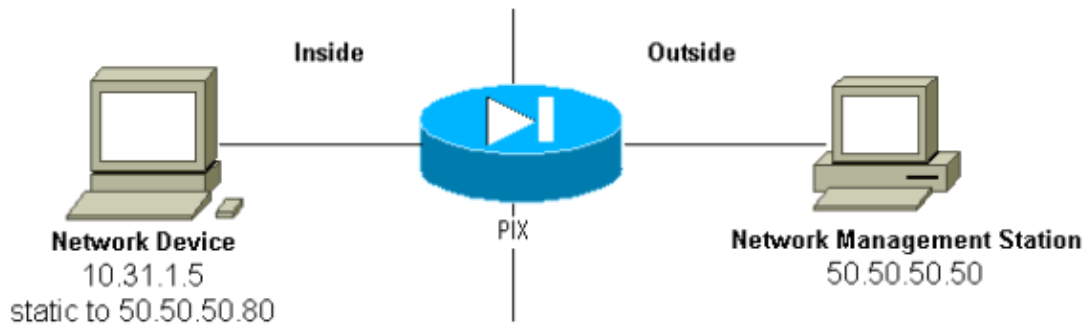
If you use access control lists (ACLs), available in PIX 5.0 and later, instead of conduits:

```
access-list Inbound permit udp host 50.50.50.50 host 50.50.50.75 eq snmptrap
access-group Inbound in interface outside
```

The PIX shows:

```
302005: Built UDP connection for faddr 50.50.50.50/2388
gaddr 50.50.50.75/162 laddr 10.10.10.10/162
```

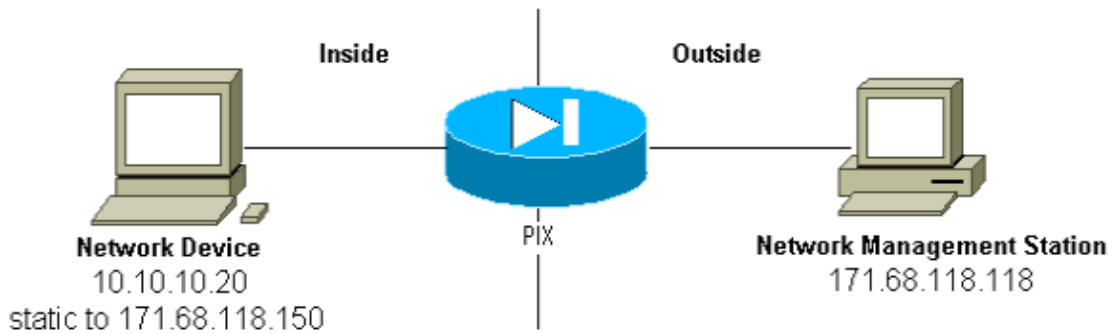
Traps Inside to Outside



Outbound traffic is allowed by default (in the absence of outbound lists) and the PIX shows:

```
305002: Translation built for gaddr 50.50.50.80 to laddr 10.31.1.5
302005: Built UDP connection for faddr 50.50.50.50/162
gaddr 50.50.50.80/2982 laddr 10.31.1.5/2982
```

Polling Outside to Inside



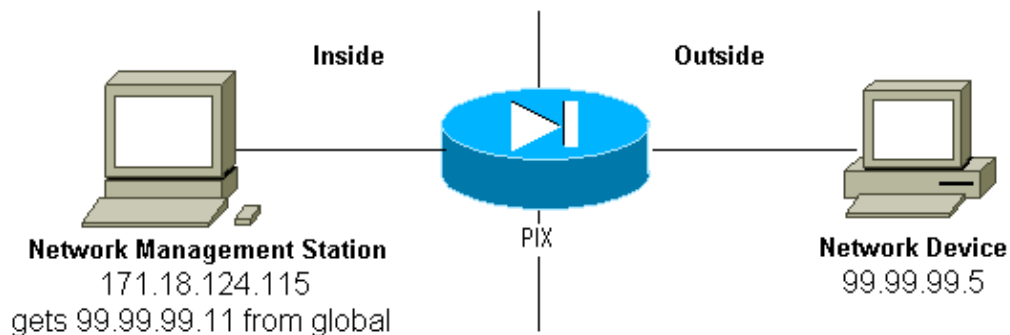
In order to allow polling from 171.68.118.118 to 10.10.10.20:

```
static (inside,outside) 171.68.118.150 10.10.10.20 netmask 255.255.255.255 0 0
conduit permit udp host 171.68.118.150 eq snmp host 171.68.118.118
```

If you use ACLs, available in PIX 5.0 and later, instead of conduits:

```
access-list Inbound permit udp host 171.68.118.118 host 171.68.118.150 eq snmp
access-group Inbound in interface outside
```

Polling Inside to Outside



Outbound traffic is allowed by default (in the absence of outbound lists) and the PIX shows:

```
305002: Translation built for gaddr 99.99.99.11 to laddr 172.18.124.115
302005: Built UDP connection for faddr 99.99.99.5/161
```

SNMP to the PIX/ASA

MIB Support by Version

These are the versions of MIB support in the PIX:

- PIX Firewall Software Versions 4.0 until 5.1 System and Interface groups of MIB-II (refer to RFC 1213) but not the AT, ICMP, TCP, UDP, EGP, transmission, IP, or SNMP groups
CISCO-SYSLOG-MIB-V1SML.my.
- PIX Firewall Software Versions 5.1.x and later Previous MIBs and
CISCO-MEMORY-POOL-MIB.my and the cfwSystem branch of the
CISCO-FIREWALL-MIB.my.
- PIX Firewall Software Versions 5.2.x and later Previous MIBs and the ipAddrTable of the IP group.
- PIX Firewall Software Versions 6.0.x and later Previous MIBs and modification of the MIB-II OID to identify PIX by model (and enable CiscoView 5.2 support). The new object identifiers (OIDs) are found in the CISCO-PRODUCTS-MIB; for example, the PIX 515 has the OID 1.3.6.1.4.1.9.1.390.
- PIX Firewall Software Versions 6.2.x and later Previous MIBs and
CISCO-PROCESS-MIB-V1SML.my.
- PIX/ASA Software Version 7.x Previous MIBs and IF-MIB, SNMPv2-MIB, ENTITY-MIB,
CISCO-REMOTE-ACCESS-MONITOR-MIB, CISCO-CRYPTO-ACCELERATOR-MIB,
ALTIGA-GLOBAL-REG.

Note: The supported section of the PROCESS MIB is the cpmCPUTotalTable branch of the cpmCPU branch of the ciscoProcessMIBObjects branch. There is no support for the ciscoProcessMIBNotifications branch, ciscoProcessMIBconformance branch, or the two tables, cpmProcessTable and cpmProcessExtTable, in the cpmProcess branch of the ciscoProcessMIBObjects branch of the MIB.

Turning on SNMP in the PIX/ASA

Issue these commands to permit polling/queries and traps in the PIX:

```
snmp-server host #.#.#.#  
  
!--- IP address of the host allowed to poll  
!--- and where to send traps.  
  
snmp-server community <whatever>  
snmp-server enable traps
```

PIX Software Versions 6.0.x and later allow more granularity with regard to traps and queries.

```
snmp-server host #.#.#.#  
  
!--- The host is to be sent traps and can query.  
  
snmp-server host #.#.#.# trap  
  
!--- The host is to be sent traps and cannot query.  
  
snmp-server host #.#.#.# poll  
  
!--- The host can query but is not to be sent traps.
```

PIX/ASA Software Versions 7.x allow more granularity with regard to traps and queries.

```

hostname(config)#snmp-server host <interface_name> <ip_address> trap community <community>

!--- The host is to be sent traps and cannot query
!--- with community string specified.

hostname(config)#snmp-server host <interface_name> <ip_address> poll community <community>

!--- The host can query but is not to be sent traps
!--- with community string specified.

```

Note: Specify **trap** or **poll** if you want to limit the NMS to receiving traps only or browsing (polling) only. By default, the NMS can use both functions.

SNMP traps are sent on UDP port 162 by default. You can change the port number with the **udp-port** keyword.

SNMP to the PIX/ASA – Polling

The variables that the PIX returns depends on mib support in the version. An example output of an snmpwalk of a PIX that runs 6.2.1 is at the end of this document. Earlier versions of software return only the previously noted mib values.

SNMP to the PIX/ASA – Traps

Note: An SNMP OID for PIX Firewall displays in SNMP event traps sent from the PIX Firewall. OID 1.3.6.1.4.1.9.1.227 was used as the PIX Firewall system OID until PIX Software version 6.0. The new model-specific OIDs are found in the CISCO-PRODUCTS-MIB.

Issue these commands to turn on traps in the PIX:

```

snmp-server host #.#.#.#

!--- IP address of the host allowed to do queries
!--- and where to send traps.

snmp-server community <whatever>
snmp-server enable traps

```

Traps Version 4.0 Until 5.1

When you use PIX Software 4.0 and later, you can generate these traps:

```

cold start = 1.3.6.1.6.3.1.1.5.1
link_up = 1.3.6.1.6.3.1.1.5.4
link_down = 1.3.6.1.6.3.1.1.5.3
syslog trap (clogMessageGenerated) = 1.3.6.1.4.1.9.9.41.2.0.1

```

Trap Changes (PIX 5.1)

In PIX Software version 5.1.1 and later, the trap levels are separated from the syslog levels for the syslog traps. The PIX still sends syslog traps, but more granularity can be configured. This example raw trapd.log file (and this is the same for HP OpenView [HPOV] or Netview) included 3 link_up traps and 9 syslog traps, with 7 different syslog ids: 101003, 104001, 111005, 111007, 199002, 302005, 305002.

Example of a trapd.log

```
952376318 1 Mon Mar 06 15:58:38 2000 10.31.1.150 - 1=20 2=7
  3=Syslog Trap 4=199002:
PIX startup completed. Beginning operation. 5=0;1 .1.3.6.1.4.1.9.9.4 1.2.0.1 0

952376318 1 Mon Mar 06 15:58:38 [10.31.1.150.2.2] %PIX-1-104001: (Secondary)
Switching to ACTIVE - no failover cable.

952376332 1 Mon Mar 06 15:58:52 2000 10.31.1.150 - 1=20 2=2
  3=Syslog Trap 4=101003: (Secondary) Failover cable not connected (this unit)
  5=1400;1 .1.3.6.1.4.1.9.9.41.2.0.1 0

952376332 1 Mon Mar 06 15:58:52 [10.31.1.150.2.2] %PIX-1-101003: (Secondary)
Failover cable not connected (this unit)

952376345 1 Mon Mar 06 15:59:05 2000 10.31.1.150 - 1=20 2=7
  3=Syslog Trap 4=305002:
Translation built for gaddr 50.50.50.75 to laddr 171.68.118.118 5=2800;1
.1.3.6.1.4.1.9.9.41.2.0.1 0

952376345 1 Mon Mar 06 15:59:05 2000 10.31.1.150 - 1=20 2=7
  3=Syslog Trap 4=302005: Built UDP connection for faddr 50.50.50.50/2388
  gaddr 50.50.50.75/162 laddr 171.68.118.118/162
  5=2800;1 .1.3.6.1.4.1.9.9.41.2.0.1 0

952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 1;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0

952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 2;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0

952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 3;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0

952376360 1 Mon Mar 06 15:59:20 2000 10.31.1.150 - 1=20 2=6
  3=Syslog Trap 4=111007: Begin configuration: console reading from terminal
  5=4200;1 .1.3.6.1.4.1.9.9.41.2.0.1 0

952376365 1 Mon Mar 06 15:59:25 2000 10.31.1.150 - 1=20 2=6
  3=Syslog Trap 4=111005: console end configuration: OK
  5=4700;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

Description of Each Trap – trapd.log

```
199002 (syslog)
4=199002: PIX startup completed. Beginning operation.
5=0;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
104001 (syslog)
Mar 6 15:58:38 [10.31.1.150.2.2] %PIX-1-104001: (Secondary)
Switching to ACTIVE - no failover cable.
```

```
101003 (syslog)
952376332 1 Mon Mar 06 15:58:52 2000 10.31.1.150 - 1=20 2=2
  3=Syslog Trap 4=101003: (Secondary) Failover cable not connected (this unit)
  5=1400;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
101003 (syslog)
Mar 6 15:58:52 [10.31.1.150.2.2] %PIX-1-101003: (Secondary) Failover cable not
connected (this unit)
```

```
305002 (syslog)
952376345 1 Mon Mar 06 15:59:05 2000 10.31.1.150 - 1=20 2=7
3=Syslog Trap 4=305002: Translation built for gaddr 50.50.50.75
to laddr 171.68.118.118 5=2800;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
302005 (syslog)
952376345 1 Mon Mar 06 15:59:05 2000 10.31.1.150 - 1=20 2=7
3=Syslog Trap 4=302005: Built UDP connection for faddr 50.50.50.50/2388
gaddr 50.50.50.75/162 laddr 171.68.118.118/162
5=2800;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
Linkup (linkup)
952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 1;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0
```

```
Linkup (linkup)
952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 2;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0
```

```
Linkup (linkup)
952376347 1 Mon Mar 06 15:59:07 2000 10.31.1.150 - Agent Interface Up (linkUp
Trap) enterprise:ENTERPRISES.9.1.227 (.1.3.6.1.4.1.9.1.227) on interface 3;1
.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9.1.227 0
```

```
Linkup (syslog)
952376360 1 Mon Mar 06 15:59:20 2000 10.31.1.150 - 1=20 2=6
3=Syslog Trap 4=111007: Begin configuration: console reading from terminal
5=4200;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
111007 (syslog)
952376360 1 Mon Mar 06 15:59:20 2000 10.31.1.150 - 1=20 2=6
3=Syslog Trap 4=111007: Begin configuration: console reading from terminal
5=4200;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

```
111005 (syslog)
952376365 1 Mon Mar 06 15:59:25 2000 10.31.1.150 - 1=20 2=6
3=Syslog Trap 4=111005: console end configuration: OK
5=4700;1 .1.3.6.1.4.1.9.9.41.2.0.1 0
```

syslog Severity Level Examples

These are reproduced from the documentation to illustrate the seven messages.

Alert:

```
%PIX-1-101003:(Primary) failover cable not connected (this unit)
%PIX-1-104001:(Primary) Switching to ACTIVE (cause:reason)
```

Notification:

```
%PIX-5-111005:IP_addr end configuration: OK
%PIX-5-111007:Begin configuration: IP_addr reading from device.
```

Informational:

```
%PIX-6-305002:Translation built for gaddr IP_addr to laddr IP_addr
```

```
%PIX-6-302005:Built UDP connection for faddr faddr/fport gaddr gaddr/gport
laddr laddr/lport
%PIX-6-199002:Auth from laddr/lport to faddr/fport failed
(server IP addr failed) in interface int name.
```

Interpret syslog Severity Levels

| Level | Meaning |
|-------|---|
| 0 | System unusable – emergency |
| 1 | Take immediate action – alert |
| 2 | Critical condition – critical |
| 3 | Error message – error |
| 4 | Warning message – warning |
| 5 | Normal but significant condition – notification |
| 6 | Informational – informational |
| 7 | Debug message – debug |

Configure PIX 5.1 and Later for a Subset of Traps

If the PIX configuration has:

```
snmp-server host inside #.#.#.#
```

the only traps that are generated are the standard traps: cold start, link up and link down (not syslog).

If the PIX configuration has:

```
snmp-server enable traps
logging history debug
```

then all standard and all syslog traps are generated. In our example, these are syslog entries 101003, 104001, 111005, 111007, 199002, 302005, and 305002, and whatever other syslog output the PIX generated. Because the logging history set for debug and these trap numbers are in the notification, alert, and informational levels, level debug includes these:

If the PIX configuration has:

```
snmp-server enable traps
logging history (a_level_below_debugging)
```

then all standard and all traps at the level below debug are generated. If the **logging history notification** command is used, this would include all syslog traps at emergency, alert, critical, error, warning, and notification levels (but not informational or debug levels). In our case, 111005, 111007, 101003, and 104001 (and whatever others the PIX would generate in a live network) would be included.

If the PIX configuration has:

```
snmp-server enable traps
logging history whatever_level
no logging message 305002
no logging message 302005
no logging message 111005
```

then messages 305002, 302005, 111005 are not produced. With PIX set for **logging history debug**, you see messages 104001, 101003, 111007, 199002, and all other PIX messages, but not the 3 listed (305002, 302005, 111005).

Configure PIX/ASA 7.x for a Subset of Traps

If the PIX configuration has:

```
snmp-server host <interface name> <ip address> community <community string>
```

the only traps that are generated are the standard traps: authentication, cold start, link up and link down (not syslog).

The remaining configuration is similar as PIX Software version 5.1 and later, except in PIX/ASA version 7.x, the **snmp-server enable traps** command has additional options such as **ipsec**, **remote-access** and **entity**

Note: Refer to the Enabling SNMP section of Monitoring the Security Appliance in order to learn more about the SNMP traps in PIX/ASA

SNMP Issues

PIX Discovery

If the PIX responds to an SNMP query and reports its OID as 1.3.6.1.4.1.9.1.227, or in PIX Firewall Software versions 6.0 or later, as an ID listed in the CISCO-PRODUCTS-MIB for that model, then the PIX is working as designed.

In versions of PIX code prior to 5.2.x when there was support added for the ipAddrTable of the IP group, network management stations might not be able to draw the PIX on the map as a PIX. A network management station should always be able to detect the fact that the PIX exists if it is able to ping the PIX, but it might not draw it as a PIX – a black box with 2 lights. In addition to needing support of the ipAddrTable of the IP group, HPOV, Netview, and most other network management stations need to understand that the OID being returned by the PIX is that of a PIX for the proper icon to appear.

CiscoView support for PIX management was added in CiscoView 5.2; PIX version 6.0.x is also required. In earlier PIX versions, a third-party management application allows the HPOV Network Node Manager to identify PIX Firewalls and systems that run PIX Firewall Manager.

Discover Devices Inside the PIX

If the PIX is properly configured, it passes SNMP queries and traps from outside to inside. Because Network Address Translation (NAT) is usually configured on the PIX, statics would be required to do this. The problem is when the network management station does an snmpwalk of the public address, which statics to a private address inside the network, the outside header of the packet does not agree with the information in the ipAddrTable. Here 171.68.118.150 is walked, which is static to 10.10.10.20 inside the PIX and you can see where device 171.68.118.150 reports that it has two interfaces: 10.10.10.20 and 10.31.1.50:

```
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.10.10.10.20 : IpAddress: 10.10.10.20
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.10.31.1.50 : IpAddress: 10.31.1.50
```

Will this make sense to a network management station? Probably not. The same issue will be present for traps: if the 10.31.1.50 interface were to go down, device 171.68.118.150 would report interface 10.31.1.50 was down.

Another problem in trying to manage an inside network from outside is "drawing" the network. If the management station is Netview or HPOV, these products use a "netmon" daemon to read the route tables from devices. The route table is used in discovery. The PIX does not support enough of RFC 1213 to return a routing table to a network management station, and for security reasons, this is not a good idea anyhow. While devices inside the PIX report their route-tables when the static is queried, all the public IP devices (statics) report all private interfaces. If the other private addresses inside the PIX do not have statics, they cannot be queried. If they do have statics, the network management station has no way of knowing what the statics are.

Discover Devices Outside the PIX

Since a network management station inside the PIX queries a public address which reports "public" interfaces, the discovery outside to inside issues do not apply.

Here, 171.68.118.118 was inside and 10.10.10.25 was outside. When 171.68.118.118 walked 10.10.10.25, the box correctly reported its interfaces, that is, the header is the same as inside the packet:

```
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.10.10.10.25 : IpAddress: 10.10.10.25
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.10.31.1.50 : IpAddress: 10.31.1.50
```

Version 6.2 snmpwalk of PIX

The `snmpwalk -c public <pix_ip_address>` command was used on a HPOV management station to perform snmpwalk. All MIBs available for PIX 6.2 were loaded prior to performing the snmpwalk.

```
system.sysDescr.0 : DISPLAY STRING- (ascii):
Cisco PIX Firewall Version 6.2(1)
system.sysObjectID.0 : OBJECT IDENTIFIER:
.iso.org.dod.internet.private.enterprises.cisco.ciscoProducts.390
system.sysUpTime.0 : Timeticks: (6630200) 18:25:02.00
system.sysContact.0 : DISPLAY STRING- (ascii):
system.sysName.0 : DISPLAY STRING- (ascii): satan
system.sysLocation.0 : DISPLAY STRING- (ascii):
system.sysServices.0 : INTEGER: 4
interfaces.ifNumber.0 : INTEGER: 3
interfaces.ifTable.ifEntry.ifIndex.1 : INTEGER: 1
interfaces.ifTable.ifEntry.ifIndex.2 : INTEGER: 2
interfaces.ifTable.ifEntry.ifIndex.3 : INTEGER: 3
interfaces.ifTable.ifEntry.ifDescr.1 : DISPLAY STRING- (ascii):
PIX Firewall 'outside' interface
interfaces.ifTable.ifEntry.ifDescr.2 : DISPLAY STRING- (ascii):
PIX Firewall 'inside' interface
interfaces.ifTable.ifEntry.ifDescr.3 : DISPLAY STRING- (ascii):
PIX Firewall 'intf2' interface
interfaces.ifTable.ifEntry.ifType.1 : INTEGER: ethernet-csmacd
interfaces.ifTable.ifEntry.ifType.2 : INTEGER: ethernet-csmacd
interfaces.ifTable.ifEntry.ifType.3 : INTEGER: ethernet-csmacd
interfaces.ifTable.ifEntry.ifMtu.1 : INTEGER: 1500
interfaces.ifTable.ifEntry.ifMtu.2 : INTEGER: 1500
interfaces.ifTable.ifEntry.ifMtu.3 : INTEGER: 1500
interfaces.ifTable.ifEntry.ifSpeed.1 : Gauge32: 10000000
interfaces.ifTable.ifEntry.ifSpeed.2 : Gauge32: 10000000
interfaces.ifTable.ifEntry.ifSpeed.3 : Gauge32: 10000000
interfaces.ifTable.ifEntry.ifPhysAddress.1 : OCTET STRING-
(hex): length = 6
      0:  00 50 54 fe ea 30 -- -- -- -- -- -- -- --
.PT..0.....

interfaces.ifTable.ifEntry.ifPhysAddress.2 : OCTET STRING- (hex): length = 6
      0:  00 50 54 fe ea 31 -- -- -- -- -- -- -- --
.PT..1.....
```

```

interfaces.ifTable.ifEntry.ifPhysAddress.3 : OCTET STRING- (hex): length = 6
      0: 00 90 27 42 fb be -- -- -- -- -- -- -- -- -- --
...B.....

interfaces.ifTable.ifEntry.ifAdminStatus.1 : INTEGER: up
interfaces.ifTable.ifEntry.ifAdminStatus.2 : INTEGER: up
interfaces.ifTable.ifEntry.ifAdminStatus.3 : INTEGER: down
interfaces.ifTable.ifEntry.ifOperStatus.1 : INTEGER: up
interfaces.ifTable.ifEntry.ifOperStatus.2 : INTEGER: up
interfaces.ifTable.ifEntry.ifOperStatus.3 : INTEGER: down
interfaces.ifTable.ifEntry.ifLastChange.1 : Timeticks: (6630200) 18:25:02.00
interfaces.ifTable.ifEntry.ifLastChange.2 : Timeticks: (6630200) 18:25:02.00
interfaces.ifTable.ifEntry.ifLastChange.3 : Timeticks: (6630200) 18:25:02.00
interfaces.ifTable.ifEntry.ifInOctets.1 : Counter: 0
interfaces.ifTable.ifEntry.ifInOctets.2 : Counter: 19120151
interfaces.ifTable.ifEntry.ifInOctets.3 : Counter: 0
interfaces.ifTable.ifEntry.ifInUcastPkts.1 : Counter: 0
interfaces.ifTable.ifEntry.ifInUcastPkts.2 : Counter: 1180
interfaces.ifTable.ifEntry.ifInUcastPkts.3 : Counter: 0
interfaces.ifTable.ifEntry.ifInNUcastPkts.1 : Counter: 0
interfaces.ifTable.ifEntry.ifInNUcastPkts.2 : Counter: 246915
interfaces.ifTable.ifEntry.ifInNUcastPkts.3 : Counter: 0
interfaces.ifTable.ifEntry.ifInDiscards.1 : Counter: 0
interfaces.ifTable.ifEntry.ifInDiscards.2 : Counter: 0
interfaces.ifTable.ifEntry.ifInDiscards.3 : Counter: 0
interfaces.ifTable.ifEntry.ifInErrors.1 : Counter: 0
interfaces.ifTable.ifEntry.ifInErrors.2 : Counter: 0
interfaces.ifTable.ifEntry.ifInErrors.3 : Counter: 0
interfaces.ifTable.ifEntry.ifOutOctets.1 : Counter: 60
interfaces.ifTable.ifEntry.ifOutOctets.2 : Counter: 187929
interfaces.ifTable.ifEntry.ifOutOctets.3 : Counter: 0
interfaces.ifTable.ifEntry.ifOutUcastPkts.1 : Counter: 1
interfaces.ifTable.ifEntry.ifOutUcastPkts.2 : Counter: 2382
interfaces.ifTable.ifEntry.ifOutUcastPkts.3 : Counter: 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.1 : Counter: 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.2 : Counter: 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.3 : Counter: 0
interfaces.ifTable.ifEntry.ifOutDiscards.1 : Counter: 0
interfaces.ifTable.ifEntry.ifOutDiscards.2 : Counter: 0
interfaces.ifTable.ifEntry.ifOutDiscards.3 : Counter: 0
interfaces.ifTable.ifEntry.ifOutErrors.1 : Counter: 0
interfaces.ifTable.ifEntry.ifOutErrors.2 : Counter: 0
interfaces.ifTable.ifEntry.ifOutErrors.3 : Counter: 0
interfaces.ifTable.ifEntry.ifSpecific.1 : OBJECT IDENTIFIER:
.ccitt.zeroDotZero
interfaces.ifTable.ifEntry.ifSpecific.2 : OBJECT IDENTIFIER:
.ccitt.zeroDotZero
interfaces.ifTable.ifEntry.ifSpecific.3 : OBJECT IDENTIFIER:
.ccitt.zeroDotZero
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.212.3.3.1 : IpAddress:
212.3.3.1
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.10.48.66.47 : IpAddress:
10.48.66.47
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1 : IpAddress:
127.0.0.1
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.212.3.3.1 : INTEGER: 1
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.10.48.66.47 : INTEGER: 2
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.127.0.0.1 : INTEGER: 3
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.212.3.3.1 : IpAddress:
255.255.255.0
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.10.48.66.47 : IpAddress:
255.255.254.0
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.127.0.0.1 : IpAddress:
255.255.255.255
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.212.3.3.1 : INTEGER: 0

```

```
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.10.48.66.47 : INTEGER: 0
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.127.0.0.1 : INTEGER: 0
ip.ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.212.3.3.1 : INTEGER:
65535
ip.ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.10.48.66.47 : INTEGER:
65535
ip.ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.127.0.0.1 : INTEGER:
65535
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolName.1 :
DISPLAY STRING- (ascii): PIX system memory
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolAlternate.1 :
INTEGER: 0
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolValid.1 :
INTEGER: true
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolUsed.1 :
Gauge32: 21430272
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolFree.1 :
Gauge32: 12124160
cisco.ciscoMgmt.ciscoMemoryPoolMIB.ciscoMemoryPoolObjects.
ciscoMemoryPoolTable.ciscoMemoryPoolEntry.ciscoMemoryPoolLargestFree.1 :
Gauge32: 0
cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.
cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUTotalPhysicalIndex.1 : INTEGER: 0
cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.
cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUTotal5sec.1 : Gauge32: 0
cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.
cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUTotal1min.1 : Gauge32: 0
cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.
cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUTotal5min.1 : Gauge32: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareInformation.
6 : OCTET STRING- (ascii):
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareInformation.
7 : OCTET STRING- (ascii):
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareStatusValue.
6 : INTEGER: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareStatusValue.
7 : INTEGER: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareStatusDetail.
6 : OCTET STRING- (ascii): Failover Off
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatus.cfwHardwareStatusTable.cfwHardwareStatusEntry.cfwHardwareStatusDetail.
7 : OCTET STRING- (ascii): Failover Off
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
4.3 : OCTET STRING- (ascii): maximum number of allocated 4 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
4.5 : OCTET STRING- (ascii): fewest 4 byte blocks available
since system startup
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
4.8 : OCTET STRING- (ascii): current number of available 4 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
80.3 : OCTET STRING- (ascii): maximum number of allocated 80 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
```

cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
80.5 : OCTET STRING- (ascii): fewest 80 byte blocks available
since system startup
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
80.8 : OCTET STRING- (ascii): current number of available 80 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
256.3 : OCTET STRING- (ascii): maximum number of allocated 256 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
256.5 : OCTET STRING- (ascii): fewest 256 byte blocks available
since system startup
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
256.8 : OCTET STRING- (ascii): current number of available 256 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
1550.3 : OCTET STRING- (ascii): maximum number of allocated 1550 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
1550.5 : OCTET STRING- (ascii): fewest 1550 byte blocks available
since system startup
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatInformation.
1550.8 : OCTET STRING- (ascii): current number of available 1550 byte blocks
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
4.3 : Gauge32: 1600
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
4.5 : Gauge32: 1599
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
4.8 : Gauge32: 1600
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
80.3 : Gauge32: 400
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
80.5 : Gauge32: 374
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
80.8 : Gauge32: 400
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
256.3 : Gauge32: 500
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
256.5 : Gauge32: 498
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
256.8 : Gauge32: 500
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
1550.3 : Gauge32: 1252
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
1550.5 : Gauge32: 865
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwBufferStatsTable.cfwBufferStatsEntry.cfwBufferStatValue.
1550.8 : Gauge32: 867
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
cfwConnectionStatDescription.40.6 :
OCTET STRING- (ascii): number of connections currently in use

```
by the entire firewall
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
  cfwConnectionStatDescription.40.7 :
OCTET STRING- (ascii):      highest number of connections in use
  at any one time since system startup
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
  cfwConnectionStatCount.40.6 :
Counter: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
  cfwConnectionStatCount.40.7 :
Counter: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
  cfwConnectionStatValue.40.6 :
Gauge32: 0
cisco.ciscoMgmt.ciscoFirewallMIB.ciscoFirewallMIBObjects.cfwSystem.
cfwStatistics.cfwConnectionStatTable.cfwConnectionStatEntry.
  cfwConnectionStatValue.40.7 :
Gauge32: 0
End of MIB View.
```

Information to Collect if You Open a TAC Case

If you still need assistance after you complete the troubleshooting steps in this document and want to open a case with the Cisco TAC, make sure to include this information to troubleshoot your PIX Firewall.

- Problem description and relevant topology details
- Troubleshooting performed before you opened the case
- Output from the **show tech-support** command
- Output from the **show log** command after running with the **logging buffered debugging** command, or console captures that demonstrate the problem (if available)

Attach the collected data to your case in non-zipped, plain text format (.txt). You can attach information to your case by uploading it using the Case Query Tool (registered customers only). If you cannot access the Case Query Tool, you can send the information in an email attachment to attach@cisco.com with your case number in the subject line of your message.

Related Information

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

