Cisco PIX Firewall - Practical Guide

Author: Florin Prunoiu
Enterastream Communications Inc
Last update: March 25, 2004

About

This whitepaper is the result of hands-on working experience with various PIX platforms and versions and summarizes all the core concepts that a firewall administrator must know when working with Cisco PIX.

This whitepaper tries to complement other documentation sources by being geared toward the security administrator's direct understanding from a conceptual and presentational perspective.

So you may find this Cisco PIX Firewall - Practical Guide whitepaper tailored to your practical needs and easy to use when you are new to PIX or just looking for a memory refreshing reference.

All your suggestions and corrections are more than welcomed. Click here.

1. Essentials

Cisco PIX is a largely deployed firewall solution being specially preferred for security solutions that required high processing speed as are the semi-trusted connections between peering businesses. From a security features perspective it cannot be categorized as the most flexible platform but the latest versions (6.3 and later) cover most of the previously missing features.

PIX architecture is built around the ASA security engine that performs the inspection and maintains the session state information and handles the network translation.

The inspection sequence is performed as follows:

1. A packet is entering an interface and PIX evaluates the security level for the source and destination interfaces. A low-to-high is allowed only if there is an access-list/conduit that allows the connection and a high-to-low is allowed by default unless a specific access-list/outbound denies it.

2. The packet enters is checked against the statefull session table. If it is part of an already established flow is passed forward in order to be routed out and eventually translated if

specified.

If the packet is identified as part of a new session it is checked against the access-list applied to the inbound interface (or against the conduits for versions earlier than 6.3)

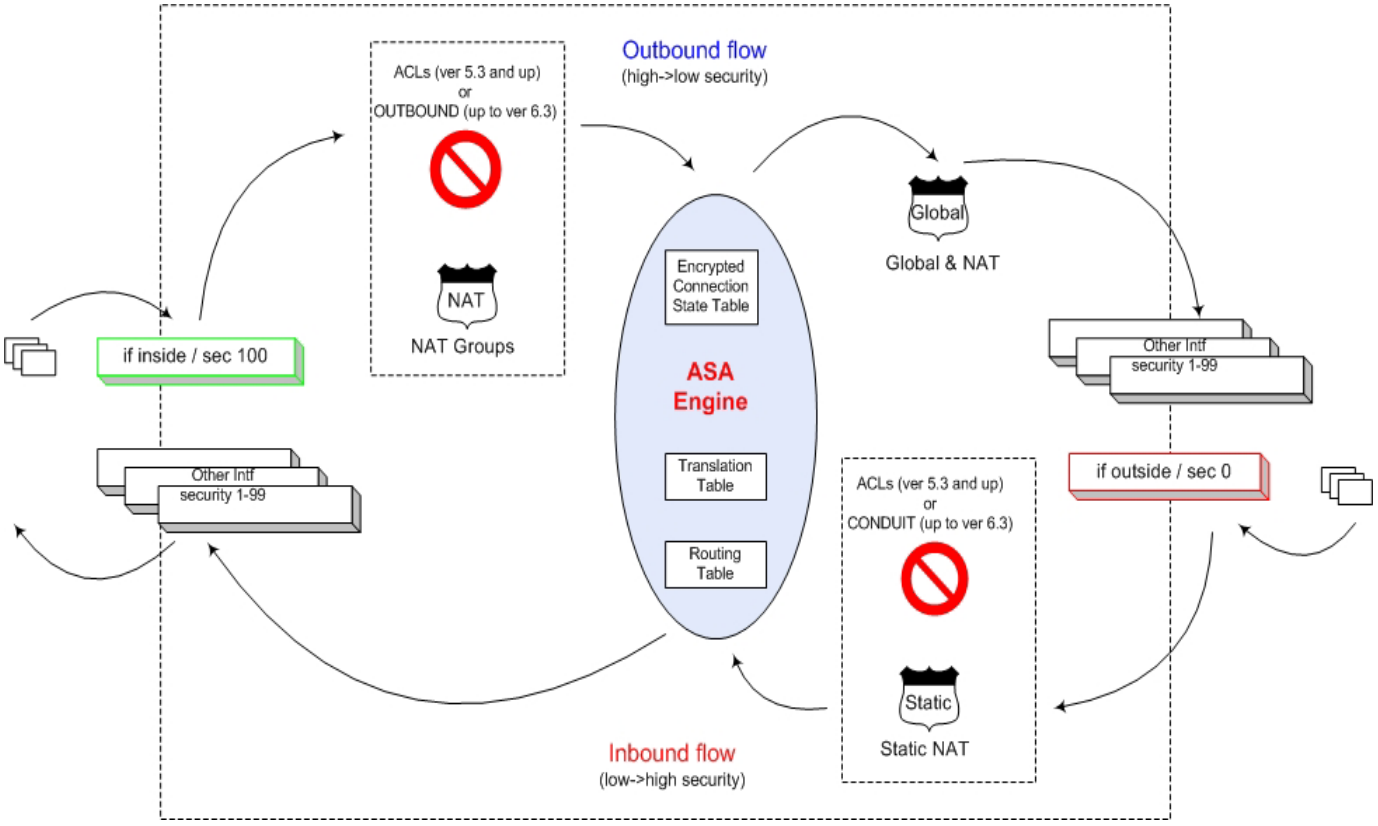
3. As the packet passed the inbound security check is passed to ASA that performs the inbound network translation (destination NAT).

4. ASA creates an entry in the statefull session table and the timers are started for that session.
The packet gets routed out to the interface designated by the routing table.

5. At the exit interface eventual source translation is performed - if specified by using global statements and nat groups

6. The packet is delivered out to the next hop router or to the final destination if it is present in the local firewall's subnets.

The following diagram presents all PIX components:



2. Interfaces and security levels

Each physical or logical (VLANed from ver 6.3) interface has a security level assigned.
There are two interfaces whose names cannot be changed and are present by default in any system:

Outside interface is always defined as interface no. 0 (ie ethernet0) and has the security level 0 assigned (the least secure)
Inside interface is always defined as interface no. 1 (ie ethernet1) and has the security level 100 assigned (the most secure)
Other interfaces can be defined and named as desired and must have a security level between 1 and 99.

3. Naming convention

Outbound data flow: initiated from a higher security interface toward a lower security interface.
Inbound data flow: initiated from a lower security interface toward a higher security interface.
Inbound and outbound concepts are used in the logging messages generated by the firewall.

4. Default security mechanism

PIX firewall allows by default any sessions or data flows to pass from a higher security interface to a lower security interface without restrictions. This approach is no longer a valid feature in today's security developments when an already compromised host may initiate outbound sessions and infect other hosts. It is strongly recommended to disable this feature by using access-list on all interfaces and define the legitimate traffic while dropping anything else.

5. Defining and enforcing the security policy

The default security policy ensures that the packets originating from higher security interfaces are allowed to flow through lower security interfaces and any packets originating from lower security interfaces are not allowed to flow through higher security interfaces.

PIX-OS later than 5.3

Access lists are the newly recommended security enforcement mechanism.

An access list is applied to an interface and checks all traffic with no difference between the direction of traffic as outbound (high-to-low security) and inbound (low-to-high security).

Access lists are statefull and are part of the ASA engine.

The access list is applied only when a packet enters the firewall through an interface. No checking is performed when it exits the firewall using the destination interface. The flow is defined only once in the access-list that applies for the interface where the flow enters the firewall.

We can make a comparison with Checkpoint FW1/NG which has the option to check a flow when it enters and also when it exits the firewall. This increases the security but downgrades the processing speed.

Some of the features of ACLs in the new PIX-OS ver 6.3 and later:

1. Accept comments (remarks) so that each statement that is part of an ACL can be commented for a more readable security policy (essential feature for a firewall administrator that was missing in the previous PIX versions)
2. Statements are numbered permitting insertion of new statements at any desired position.
3. Accepts TCP/UDP port ranges.
4. Introduces the use of groups of objects for an easier management.

The ACL statements are checked in a sequential order exactly as they have been defined.

All hits that qualify for a specific statement are logged. In order to log all dropped traffic visibly specify the implied *#deny ip any any* statement at the end of the ACL. An ACL becomes active and assigned to an interface when it is associated with it using access groups.

The matching policy is the first pattern match is chosen to drop or allow the data flow. For improved performance define the most used statements first.

Configuration summary:

```
//define an ACL
#access-list acl_linside remark --- FP: Mar 16 2004: permit outbound  FTP and HTTP
#access-list acl_inside permit tcp 10.0.0.0 255.0.0.0 152.10.10.0 255.255.255.0 eq ftp
#access-list acl_inside permit tcp 10.0.0.0 255.0.0.0 152.10.10.0 255.255.255.0 eq http
```

```
#access-list acl_linside remark --- FP: Mar 16 2004: allow outbound DNS
#access-list acl_inside permit udp 10.0.0.0 255.0.0.0 any eq 53

#access-list acl_linside remark --- FP: Mar 16 2004: deny Mydoom virus spreading
#access-list acl_inside deny tcp 10.0.0.0 255.0.0.0 any range 3127 3198

//Apply the ACL to the interface
#access-group acl_inside in interface inside
```

For the complete syntax of access-list command click here for Cisco's documentation web site.

Note:
The "in" keyword in the access-group syntax does not have an opposite "out" option. The ACLs can be specified only as an inbound checking mechanism when a packet enters an interface and does not checks when exits the firewall.

PIX-OS up to version 5.3
The old security enforcement mechanism is based on using conduits and outbounds.

Conduits
Define a group of statements that enforces the security policy for all data flows moving from low to high security interfaces. The statements are not bound to a specific interface they are treated as a bulk which is checked for any packet entering any lower security interfaces and exiting through a higher security interface.
The conduits are statefull and accept tcp/udp/icmp/any-ip data flow definitions.

The conduits offer a limited manageability and readability comparing with the extended features presented by the ACLs. They are not numbered and you cannot insert and new statement wherever you want without removing all conduit group and reentering it in the desired order.

The conduits are checked in a sequential priority, exactly as they have been defined and the first match is chosen to allow or drop the data flow. For improved performance define the most used statements first.

Outbounds

Outbounds are used to control the outbound communication originating from a higher security level to a lower security level. The outbound connections are allowed by default but you must use the outbounds to define the legitimate connections and prevent the spread of any viruses or security threats coming from the inside.

The statements are not bound to a specific interface they are treated as a bulk which is checked for any packet entering any higher security interfaces and exiting through a lower security interface.

The outbounds are statefull and accept tcp/udp/icmp/any-ip data flows definitions.

The outbounds offer a limited manageability and readability comparing with the extended features presented by the ACLs. They are not numbered and you cannot insert and new statement wherever you want without removing all conduit group and reentering it in the desired order.

The outbounds are checked in a sequential priority, exactly as they have been defined and the first match is chosen to allow or drop the data flow. For improved performance define the most used statements first.

Configuration summary

Note: the conduits syntax could be confusing due to the order the source and destination is specified which is opposite to the ACL statements.

```
#conduit permit <proto> <destination>  <dest-mask> <port> <source> <src-mask> <src-port>

//define conduits to control the inbound traffic
#conduit permit tcp host 200.31.21.2 host 205.189.2.1 eq  http
#conduit deny tcp any host 205.189.2.1
```

For complete syntax information go to Cisco Documentation website

6. Static network translation

Static network translation is the feature that allows source or destination translation on a one-to-one basis. The translation entries created using statics are permanent mappings and they do not have an expiration timer associated.

Static translation is performed using the static command. Cisco is using the global and local significance attributes for an IP address. For a better understanding think of global and local terms as being related to the physical location in regards to an interface. A global address is an address that can be access externally through other interfaces. The global address hides the local IP address which is behind the interface we refer.

Static destination translation

The destination translation occurs when you define a so called global IP for a resource which is located behind one of the internal interfaces which is accessed through any of the other interfaces.

The firewall automatically handles the ARP requests for the global IP address and assigns its own interface MAC address to it. By default PIX performs gratuitous ARP which permits the ARP resolution for the NATed IP.

Syntax:

#static(destination_intf, source_intf) <global_IP > <local_IP> netmask <mask>

Where:

- Destination_intf is the interface where the destination/translated host resides.
- Source_intf is the interface where the hosts that requires access come from.
- Netmask could be a host type (255.255.255.255) or a subnet of a random mask length meaning that the whole subnet is published for being accessed. The relationship is one-to-one. You must have enough global_IP addresses available to match each local host address.

Example:

```
//NAT for a server located behind the inside interface which is accessed by a client which comes from outside. The global IP address 191.90.30.3 is accessible through the outside interface:

#static(inside,outside) 191.90.30.3 10.0.0.100 netmask 255.255.255.255

//NAT for the same server located behind the inside interface which is accessed by a client which comes from the dmz01 interface. The global IP address 172.16.20.100 is accessible through the dmz01 interface:

#static(inside,dmz01) 172.16.20.100 10.0.0.100 netmask 255.255.255.255
```

Very important:

```
1. As observed in the above example, PIX architecture requires individual static statements for each pair of interfaces in order allow access to that translated IP. The static statement is essential in “publishing” the local host to that specific interface and making it accessible through that interface.
```

2. It is mandatory to define a “transparent” static translation when you access any host from a lower security interface to higher security interface. The static translation could be called transparent because there is no real address translation; it is only an IP address publishing in order to allow the access.

Example:

In order to access the server 10.0.0.100 situated behind the inside interface from the dmz02 interface you have to define a “transparent” static translation as follows:

#static(inside,dmz02) 10.0.0.100 10.0.0.100 netmask 255.255.255.255

You can observe that the global and local IPs is one and the same and no real translation is done. This is mandatory and is part of PIX’s specific architecture. It brings an additional level of security by the fact that even you might have the access lists/conduits to allow the access, it will not work unless you specifically designate which host(s) are published for access.

This transparent static is required only for inbound transactions from a lower security to a higher security interface. No specific statics are required when you access from a higher security to a lower security interface, unless you want to do explicit source address translation.

Static source translation

Source static translation is used when the source IP address of the host (local IP) is changed to another IP (global IP) once the packet gets routed to the destination. This translation hides the real identity of the initiator and also allows private IP addresses to be translated to public IPs in order to get routed through public networks.

Syntax:
#static(source_intf, destination_intf) <global_IP> <local-IP> netmask <mask>

Example:

//Host 10.0.0.100 is source translated when connects to another host situated behind dmz03 interface.

#static(inside,dmz03) 90.30.2.10 10.0.0.100 netmask 255.255.255.255

7. Dynamic network translation

The dynamic network translation is exclusively used to translate the source IP for either inbound or outbound sessions. The translation is done on a many-to-one or many-to-many basis.

Example:

Outbound dynamic source translation: Your inside users (10.0.0.0/16) are source translated when they go out to Internet using a single IP address - 201.187.12.100 (many to one)

Inbound dynamic source translation: Your company has a semi-private connection with a customer company and the IP address schema overlaps for the two companies - they both use 10.0.0.0/16 address space. In order to avoid the routing issues that appear in this situation you will define source address translation for the customer company when it enters your network as they will be translated to 172.16.0.0 /16 address space on a many-to-many basis or you can translate all customer IPs into a single IP address 172.16.1.1 /32 (many-to-one)

Also the latest PIX versions allow the so called policy NAT which permits you to specify a layer 3/4 access to identify the transactions that you want to translate the source IP address.

Two steps are required in defining the dynamic translation:

1. On the source interface define the NAT Groups that include the definition of the hosts that will be subject to dynamic translation. You might need the same group of hosts to be source translated to different IPs depending on the destination they want to access. This kind of granularity can be achieved by defining NAT Groups for each type of access and for each group of hosts.

Syntax:

To specify only layer3 flows based on the source IP address only:
#nat <interface-name> <group-number> <IP> <mask> dns | outbound

To specify complex layer3/4 flows that you want to tgranslate the source IP use:
#nat <interface-name> <group-number> access-list <acl-name> dns|outbound

The options are:

1. dns option allows application layer NAT. PIX looks inside the DNS resolution replies and translates the IP address that is returned that is returned to client.
2. outbound option is required when the source IP is behind a lower security interface and accesses a higher security interface.

2. On the destination interface associate the NAT group with the global IP address(es) that will translate the source addresses. The global IPs could be:

- A single IP address assigned to a whole NAT group
- A pool of IPs associated to a NAT group
- PIX's own interface IP address is used for translation

Syntax:

Define a single/pool of global IPs :
#global <destination-intf> <nat-group-id> <IP1> [- <IPn>] netmask <mask>

Use PIX's own interface IP address as global IP.
#global <destination_intf> <nat-group> interface

Example:

Define two NAT groups on the inside interface that will be translated differently on the outside interface. First group is a policy NAT group and the second one is a standard NAT group.

```
 //define the access-list that identifies the policy NAT flows:
#access-list policy-nat-01 permit tcp 10.0.0.0 255.0.0.0 12.10.1.10 eq http
#access-list policy-nat-01 permit tcp 10.0.0.0 255.0.0.0 12.10.1.10 eq https

//define the policy NAT group 1
#nat (inside) 1 access-list policy-nat-01

//define the NAT group 2
#nat (inside) 2 10.0.10.0 255.255.255.0

//define the policy NAT global that translates all source IPs to 201.100.1.10
#global (outside) 1 201.100.1.10 netmask 255.255.255.255

//define the global for NAT group 2 that translates all source IP using PIX's own interface IP address.
#global (outside) 2 interface
```

Using NAT 0

NAT group 0 is a specific PIX feature that allows to define the group of source IP addresses that will never be translated when initiate outbound connections to any destinations and any interfaces.

Example:

```
//Host 10.0.0.5 is never translated
#nat (inside) 0 10.0.0.5 255.255.255.255
```

NOTE:
If you do not specify any NAT/GLOBAL statements, all communications will be performed without source IP address translation. It is not mandatory for communication between private subnets that belong to the same corporation but it is absolutely needed when internal hosts initiate Internet or other public networks connections.

8. Routing

PIX performs the routing process based on the directly connected routes, static routes and OSPF - available in the latest PIX 6.3 version.

Static routes are directly linked to the outbound interface that connects to the next hope router. This adds more extra security.

Syntax:
#route <exit_intf> <destination-IP> <destination-mask> <next-hop-IP> <metric>

```
//Example: default route is allowed only through the outside interface.
#route outside 0.0.0.0 0.0.0.0 201.19.20.1
```

OSPF routing is a new feature available in ver 6.3 but because PIX it is usually deployed to delimitate external/perimeter connections and the path the data may flow could pose a security risk it is recommended to rely only on static routes for security and accuracy reasons. OSPF or other routing protocols could be implemented at the edge routers that connect back and forth to the PIX firewall.

9. Implementing VLANs

VLANs are also a new feature available in 6.3 or later and bring functionality and deployment flexibility.

The limitations are:

1. The number of VLAN interfaces that may be deployed per PIX varies between 3-12 based on the model and license you have.
2. Performance and throughput might be an issue when defining multiple VLANs on a single physical interface. Do proper bandwidth evaluation prior to migrating to VLANs.

The VLANs are treated as logical interfaces and can be configured and handled as any other physical interface. PIX does not allow traffic to pass between two VLAN interfaces that are defined on the same physical interface unless you specify access-lists that allow that.

Important: for security reasons PIX does not use the native VLAN. When define the VLAN interfaces do not use the native VLAN. PIX treats the trunk link a little bit differently than a regular router that performs trunking. It just does the tagging/de-tagging operation.

Configuration summary:

On the physical interface define the VLAN as logical interfaces.
At least one VLAN has to be defined as physical in order to instruct PIX to perform tagging on the physical interface. The other VLANs are defines as logical interfaces.

```
//Define the VLANs on the physical interface
#interface ethernet1 vlan10 physical
#interface ethernet1 vlan20 logical
#interface ethernet1 vlan30 logical
#interface ethernet1 vlan40 logical
//Assign a name and a security level the VLAN interfaces.
#nameif vlan10 dmz01 security10
#nameif vlan20 dmz01 security20
#nameif vlan30 dmz01 security30
#nameif vlan40 dmz01 security40
```

From this point on you handle the new dmz0x interfaces as any other interface.

For complete command syntax, access the Cisco Documentation website.