Attack Guards and Intrusion Detection

Terms you'll need to understand:

- ✓ Attack guards
- ✓ Fragmentation guard
- ✓ Mail Guard
- ✓ Embryonic connections
- ✓ TCP intercept
- ✓ Signatures
- ✓ False positives
- ✓ Shunning

Techniques you'll need to master:

- ✓ Setting embryonic connections
- ✓ Setting IP audits to an interface
- ✓ Configuring the Mail Guard feature
- ✓ Disabling signatures

In addition to ACL filtering and application inspection, the PIX firewall has attack guards and intrusion detection built in to protect against access and denial-of-service (DoS) attacks. *Attack guards* help prevent penetration and DoS attacks from taking advantage of basic security threats such as weaknesses and security holes found in commonly used applications. Intrusion detection techniques are used by the PIX firewall to monitor and shun possible attacks by reviewing the IP signatures that pass through the device. This chapter reviews these features within the PIX firewall products.

Attack Guards

Attack guards enable the PIX firewall to monitor and reject requests or messages sent to commonly used applications or protocols. These requests and messages have been discovered and identified by hackers as a potential means to cause some form of harm to a computer or network. Over the life of the Internet, hackers have tended to focus on applications and protocols that have been readily accepted by the public; if a hacker can find a security hole or possible weakness in a widely used protocol or application, he could have the power to compromise several other systems across the Internet. For example, if a hacker found a security hole in a basic email request, he could exploit that hole on not one but thousands of servers.

Several attack guards are provided on the PIX firewall that help prevent hackers from taking advantage of known security holes. Although the guards use different commands to enable and disable them, they all help protect your environment from malicious attacks.

Table 9.1 lists the attack guards covered in this chapter and the commands that enable them.

Table 9.1 Attack Guards and Commands

Guard	Command
DNS Guard	None; it's enabled by default and cannot be turned off.
Mail Guard	Use the **fixup protocol smtp 25** command.
Fragmentation Guard	Use the **sysopt security fragguard** command.
Syn Guard	Use the **max connections** and **embryonic connections** parameters of the **static** and **nat** commands.
AAA Floodguard	Use the **floodguard enable** command.

DNS Guard

Clients send UDP requests to resolve names, such as `www.examcram.com`, to an IP address before actually traveling to the Web site. This is called *domain name resolution* and is performed on domain name service (DNS) servers. These DNS servers maintain zones of name spaces that contain the actual name-to-IP-address mappings for the computers the client is looking for. A client might send out several UDP requests to resolve one name. Recall that with UDP traffic, the PIX uses an idle timer to monitor whether there is traffic passing between two computers. If the timer expires before any traffic has passed, the connection is assumed to be ended and the connection slot entry is removed from the connection table. Because DNS requests use UDP, a dynamic opening is created in the PIX firewall for 2 minutes to allow the return UDP traffic. If a response from a DNS server is received in 1 second, the opening created normally doesn't close until the 2-minute idle timer has expired. This leaves an open hole through which hackers can send attacks using a method called *hijacking*.

DNS Guard prevents DoS and UDP session hijacking by closing the UDP port after the first received DNS response.

The DNS Guard feature in the PIX firewall helps prevent hijacking by closing the dynamically opened port immediately after the first DNS response.

The DNS Guard feature is enabled by default and cannot be disabled.

Mail Guard

The Mail Guard feature is used to protect Simple Mail Transfer Protocol (SMTP) servers from known potentially harmful security problems. The guard performs application inspection using fixup protocols as discussed in Chapter 8, "Advanced Protocol Handling and PIX Firewall Features."

The `fixup protocol smtp` command provides a function known as Mail Guard which inspects SMTP traffic and allows only the seven commands defined in RFC 821 section 4.5.1 to pass. These commands are `DATA`, `HELO`, `MAIL`, `NOOP`,

QUIT, RCPT, and RSET. All other commands result in a 500 command unrecognized response to the client and a discarding of the packet before the SMTP server ever receives it.

By default, fixup protocol smtp command is enabled for port 25. The commands shown here display how to enable and disable this guard using the fixup protocol commands:

```
pixfirewall(config)# fixup protocol smtp 25
```

or

```
pixfirewall(config)# no fixup protocol smtp 25
```

If the SMTP Mail Guard is turned off or disabled, hackers can send attachments to your email servers with unsecure email commands.

Fragmentation Guard

One form of attack committed by hackers uses packets that are broken down into hundreds and thousands of IP fragments. These fragments, when assembled, can amount to absolutely nothing or be harmless. More seriously, they can reassemble into a packet that causes another attack. The fragmented packets require resources to assemble back together and can cause a DoS if too many of them are allowed to reach the targeted devices.

The PIX firewall provides a guard against receiving too many fragmented packets by following the RFC 1858 recommendation. The guard allows only 100 fragments per internal destination host per second. Also, the guard expects to receive the first fragment before receiving other fragments. For example, if the middle fragment is received first, the packets are dropped.

To configure the Frag Guard, a system option command, sysopt, is necessary. This guard is disabled by default but can be enabled by using the following command:

```
pixfirewall(config)#sysopt security fragguard
```

The show sysopt command displays a list of all the system options configured. The following example displays an output of the default system options on a PIX with the Frag Guard enabled:

```
pixfirewall(config)# show sysopt
sysopt security fragguard
no sysopt connection timewait
sysopt connection tcpmss 1380
```

```
sysopt connection tcpmss minimum 0
no sysopt nodnsalias inbound
no sysopt nodnsalias outbound
no sysopt radius ignore-secret
no sysopt uauth allow-http-cache
no sysopt connection permit-ipsec
no sysopt connection permit-pptp
no sysopt connection permit-l2tp
no sysopt ipsec pl-compatible
no sysopt route dnat
```

SYN Floodguard

The SYN Floodguard protects hosts from TCP SYN attacks. TCP requires a three-way handshake to make a connection; therefore, hackers can exploit this technology by sending hundreds or thousands of SYN requests with no intention of ever responding to them. For example, when a host receives a SYN request, it responds with a SYN/ACK. Then, the host waits for a final acknowledgement from the initiating host. If that initiating host (the hacker) never responds with the final ACK, the internal host is left tied up waiting for the return ACK. In the end, the internal host could be left hanging with thousands of half-open connections—commonly called *embryonic* connections—which could cause a DoS attack on the host by consuming all available memory resources for each connection.

The PIX firewall implements protection against TCP SYN attacks with two main parameters at the end of the `static` and `nat` commands. These parameters are `max connections` and `embryonic limit`. The following are the `static` and `nat` commands with these parameters:

```
Pixfirewall(config)# [no] static [(internal_if_name, external_if_name)]
               {<global_ip>¦interface} <local_ip> [dns] [netmask <mask>]
               [<max_conns> [<emb_limit> [<norandomseq>]]]

Pixfirewall(config)# [no] nat [(<if_name>)] <nat_id> <local_ip> [<mask>
               [dns] [outside] [<max_conns> [emb_limit>
               [<norandomseq>]]]]
```

The maximum number of connections defines the number of connections allowed to a host. If the number is exceeded, all future connections above this number are dropped. A value of `0` states that an unlimited number of connections is allowed.

The `embryonic connections` parameter dictates not the number of connections, but the number of half-open connections allowed to an internal host. If a host reaches this embryonic limit, the PIX performs a function called TCP intercept.

TCP intercept doesn't actually send the three-way handshake to the internal host, nor does it absolutely block the request. Instead, it performs a special trick on behalf of the internal host. The PIX performs the three-way handshake with the external host in an attempt to determine whether the external host's intentions are genuine. If the three-way handshake turns out to be successful and not a dead embryonic connection, the PIX contacts the internal host to bind it with the external host, thus establishing a valid connection. If the connection turns out to be a dead embryonic connection, nothing is lost; the PIX drops it and the internal host was never actually bothered with the request.

Embryonic connections are half-open three-way handshake connections that could be left open intentionally by a hacker. If the embryonic limit is reached, TCP intercept on the PIX handles any new handshakes until they are proven to be valid requests.

The following is an example of setting the maximum number of connections to 500 and the maximum number of embryonic connections to 400. This would enable the host to receive only 400 embryonic connections before TCP intercept would start to be performed by the PIX:

```
Pixfirewall(config)# static (inside, outside) 169.254.8.1 192.168.1.11
               netmask 255.255.255.255 500 400
```

The embryonic limit should be set a little lower than what the internal server can actually handle so you never overload the internal servers.

AAA floodguard

The PIX can use triple-A services known as AAA to authenticate, authorize, and accounting. AAA is discussed in more detail in later chapters. AAA is a way to authenticate and authorize user access across the firewall. However, it provides an avenue for hackers to attack a system. If a hacker tries to overwhelm the system with too many authentication requests, a DoS attack on the PIX could occur.

The `floodguard` command is used to automatically reclaim PIX resources from other services to prevent DoS attacks on user authentications. The PIX monitors the uauth connections. If there are too many for it to handle, it drops other resources in an attempt to maintain all the uauth connections. This list displays the order in which the PIX drops or shuts down the four resources:

1. `Timewait`
2. `FinWait`
3. `Embryonic`
4. `Idle`

By default, the `floodguard` command is enabled. This example displays the `floodguard enable` and `show` commands:

```
pixfirewall(config)# floodguard enable
pixfirewall(config)#
pixfirewall(config)# show flood
floodguard enable
```

The AAA **floodguard** is sometimes called **flood defender** by Cisco.

Intrusion Detection System

The intrusion detection system (IDS) provides the functionality to monitor IP traffic passing across a network and listen for potentially malicious traffic. The system monitors this traffic similar to the way a network sniffer does, except that intrusion detection compares the flowing traffic to known signatures of attacks. If a match is found, one or a combination of several things can be done: An alarm can be set; the packet can be dropped; and the TCP reset flags can be set to cease the connection.

Intrusion detection on the PIX firewall is a small engine that monitors more than 50 types of attacks, whereas a full IDS system can monitor more than 600 types of attacks. This makes the PIX IDS suitable only for basic IDS monitoring.

Signatures

Signatures are patterns found inside packets that have been known to result in some form of attack. The two general classes of signatures on the PIX are informational and attack. The detection of *informational* signatures does not necessarily indicate an attack on the network, but it can indicate the passing of traffic that is typically turned off, such as ICMP requests. *Attack* signatures are matches to traffic that produces some type of harmful danger, such as fragmented ICMPs, ping-of-death attacks, and other DoS attacks. The PIX

firewall contains a subset of the possible instruction detection signatures that exists. The syslog error messages can be found in the range from 400000 to 407002. See "Cisco PIX Firewall System Log Messages" on the Cisco Web site for the current list.

The PIX firewall only contains a subset of signatures compared to a full IDS system.

Configuring Audit Policies

The PIX firewall enables you to configure specific and general global audit policies. These audit policies define what action the PIX should perform if an attack or informational signature match is found. Table 9.2 displays the three types of actions the PIX can take.

Table 9.2 IDS Actions

Action	Description
Alarm	Creates a syslog message and sends it to the syslog server configured
Drop	Drops the packet(s)
Reset	Drops the packet and closes the connection

Global Audit Policies

As stated previously, the PIX can have a global audit policy that defines what the PIX will do globally to any signature matches when a specific audit policy is not assigned to the offending interface. For example, if attack signatures are detected on the outside interface and no specific policy is set on that interface, the global policy defines what to do with those packets. The following is the command syntax for the global informational and attack audit policies:

```
pixfirewall(config)# [no] ip audit info [action [alarm] [drop] [reset]]
pixfirewall(config)# [no] ip audit attack [action [alarm] [drop] [reset]]
```

The `show ip audit {info ¦ attack}` command can be used to display the global settings. The example shown here sets attack and informational global policies to alarm and drop matching signatures:

```
pixfirewall(config)# ip audit info action alarm drop
pixfirewall(config)# ip audit attack action alarm drop
pixfirewall(config)# show ip audit info
```

```
ip audit info action alarm drop
pixfirewall(config)# show ip audit attack
ip audit attack action alarm drop
```

Specific Audit Policies

The PIX can create specific audit policies to define what action to take when signature matches are found on an interface. Typically, only one policy is created and assigned to all the external interfaces. However, the PIX is capable of supporting a different policy for each interface if so desired.

Two steps are involved when working with specific audit policies. Step one creates the named policies, and step two links the policies to the designated interfaces. The following example displays creating and linking two separate policies named `audit-info` and `audit-attack` on the outside interface:

```
pixfirewall(config)# ip audit name Audit-Info info action alarm
pixfirewall(config)# ip audit name Audit-Attack attack action
                alarm drop reset
pixfirewall(config)# ip audit interface outside Audit-Info
pixfirewall(config)# ip audit interface outside Audit-Attack
pixfirewall(config)# show ip audit interface
ip audit interface outside Audit-Info
ip audit interface outside Audit-Attack
```

In IDS language, a false positive is an alarm or a signature match against legitimate traffic.

Disabling Signatures from Policies

This section covers how to exclude individual signatures from being audited. By default, all signatures are enabled, which can cause several false alarms (called *false positives*). To prevent false positives, the `ip audit signature` command can be used to disable individual signatures. One thing to note is that, when disabling a signature, the signature becomes disabled for the entire PIX, not just an interface or a specific policy. This example demonstrates how to globally disable several audit signatures:

```
pixfirewall(config)# ip audit signature 2001 disable
pixfirewall(config)# ip audit signature 2002 disable
pixfirewall(config)# ip audit signature 2150 disable
pixfirewall(config)# show ip audit signature
ip audit signature 2001 disable
ip audit signature 2002 disable
ip audit signature 2150 disable
```

By default, all audit signatures are enabled. If you want to disable one, then the **ip audit signature <number> disable** command can be used.

The shun Command

When IDSs block traffic, they use a technique called *shunning*. The PIX can perform dynamic shunning, meaning it can block traffic if the traffic violates a signature. The `shun` command is used to manually block or unblock traffic. In the following, the `shun` command blocks traffic from source 169.254.70.1:

```
pixfirewall(config)# shun 169.254.70.1
pixfirewall(config)# show shun
Shun 169.254.70.1 0.0.0.0 0 0
```

The `shun` command is a powerful command that takes precedence over the conduits and ACLs. However, it's only a temporary command and is not actually shown or saved in the configuration: it is deleted when the device is restarted. To view active shunned addresses, use the `show shun` command. To clear all active shuns, use the `clear shun` command, and use the `no shun` for a specific entry.

The **shun** command is used to block traffic for instruction detection system (IDS).