Global Information Assurance Certification Paper

GIAC Level 2: Firewalls, Perimeter Protection and VPNs

Practical Assignment for SANS New Orleans 2001

Kevin Olree
May 10, 2001

1. Security Architecture for GIAC Enterprises

1.1 Scope of Work

GIAC Enterprises is a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings. The purpose of this assignment is to define a top-level network security architecture for GIAC that will provide for

- all networking services necessary to the conduct of the business
- a level of security adequate to prevent misuse of those services or compromise of company data

The first section specifies a logical network design that supports GIAC's business model. Basic services and user groups are defined, then tied together into a framework using security technologies.

The second section specifies a physical network design that will implement the logical design. Specific choices of hardware and software are justified here and estimates of time and equipment costs are given.

1.2 Logical Design

In general I have tried to follow the principle that "simpler is better" because in my experience uncluttered designs are cheaper and easier to manage. As a result the designs purposely do not include one of every new thing on the market. Nevertheless, "Defense in Depth" is provided for each service as will be explained in the relevant subsections.

1.2.1 Services

In order for GIAC to do business it must make certain network services available to its employees, suppliers, partners and customers. Some of these services are common to any company with an Internet presence while others are more specific to GIAC.

Standard services that GIAC will have to provide include a connection to the Internet, a company website, an email service and DNS name resolution. There are also some common internal services necessary to the employee and admin community such as print, backup, internal name resolution, user authentication, file sharing and security administration (including a logging service).

Finally, GIAC needs to provide fine-grained control of a database service that will manage access to all of the fortunes. This is really the heart of GIAC's business and receives more focus than any other service in the design. Although the specific security configuration of an enterprise database system is not within the scope of this paper, it is assumed that all access to the fortunes will be carefully controlled and monitored by the database administrator(s).

1.2.2 User Groups

The services described must be made available in different ways to five general user groups:

- local employees (at the office)
- remote employees (at home or on the road)
- suppliers
- partners
- customers

This design assumes that local and remote *employees* will require the same services. In other words, remote employees should be able to do all the same things that they can do with their workstations at the office. Since this will require network traffic that would normally be contained in the zone of highest security, an encrypted VPN solution is called for. The VPN connection for remote employees can be characterized by the following:

- no fixed source address
- full administrative control of the remote hosts
- connection pattern is Point-to-Network (access will be provided to the entire internal LAN)
- strongest possible security required (since a compromise here represents a worst case exposure scenario)

The services that should be limited to employees include print, backup, domain authentication (controlling access to shared files and applications), database administration and security administration (including IDS of some form and administrative access to network devices). Splitting the mail service into two pieces by way of a publicly accessible proxy will allow the back-end mail engine to be fully secured as well. These services will collectively enjoy the highest security level within the design.

Finally, employees will need general access to the Internet. In order to provide this service safely and efficiently a combination of packet filtering and stateful firewalling has been designed in.

The principle of "Defense in Depth" can be implemented for the employee services in several ways. The second line of defense (after the VPN security) includes all of the usual protective measures that should normally be in place on an office LAN. These would include a second phase domain authentication before shared files and applications could be accessed, as well as passwords further limiting access to network devices or database admin functions. All of these services will have some sort of logging capability as well that can be regularly checked for evidence of failed logins.

GIAC's *suppliers* are authors who connect to supply fortunes. They only need access to the database service and then only in a limited way. Their access to the database should require a two-phase authentication process. The first will be the establishment of a VPN connection – the second should be a password-based database authentication. Since GIAC will not have administrative control over these remote hosts, it is specified as a matter of policy that the source addresses be fixed. This will allow an additional level of security for the VPN design. The characteristics of the supplier VPN connections are:

- fixed source address
- no administrative control over the remote hosts
- connection pattern is Point-to-Point (access is only allowed to the database)
- Moderate security required (since a compromise would allow only limited access to data)

GIAC's *partners* are international affiliates that translate and resell fortunes. It is assumed that partners will need to connect to the database from multiple hosts on their corporate LANs. This can best be accomplished in a secure way with a VPN gateway device at each partner site that will provide an encrypted tunneling service between the networks. Connections from partners should also require a strong second phase authentication to the database service. The VPN connections from the partners have the following characteristics:

- fixed pool of source addresses
- fixed source address of remote VPN gateway
- administrative cooperation on the remote side
- connection pattern is Network-to-Point (access is only allowed to the database)
- strong security required (since a compromise allows access to all of the fortune data)

Finally, GIAC's *customers* must be able to make secure connections to an e-commerce website and purchase fortunes. Since the website will be available for public access from the Internet, it will be placed in a separate security zone with no default connection privileges to the internal LAN. It will share this zone with the mail proxy service, which must also be available for public connections. Any database access necessary to the web application will be tightly controlled to prevent abuse in the event of a successful intrusion. Likewise, the mail proxy's connection to the back-end mail engine will be strictly controlled.

The third service customers will need is DNS name resolution in order to "find" GIAC's web and mail services on the Internet. Since GIAC's external DNS needs are both static and simple, this design specifies that public name resolution will be outsourced to a reputable third-party provider. This will reduce cost and complexity of the GIAC network with no real penalty in terms of quality of service. It is understood that name resolution for the internal network will be handled internally (via DNS, WINS, host files, etc.).

1.2.3 Putting It All Together

Figure 1.1 presents a logical network design that organizes these services and user groups according to their various security needs. The broad outline of the physical requirements can be identified from this design, namely:

- a border device with the following capabilities
  - packet filtering
  - WAN interface for connection to the public network
  - LAN interface for connection to the firewall device
- a firewall device with the following capabilities
  - stateful packet inspection
  - NAT
  - VPN negotiation

    - LAN interface for connection to the border device
    - two LAN interfaces for connection to the two security zones
- several hosts that will provide the various internal and public services

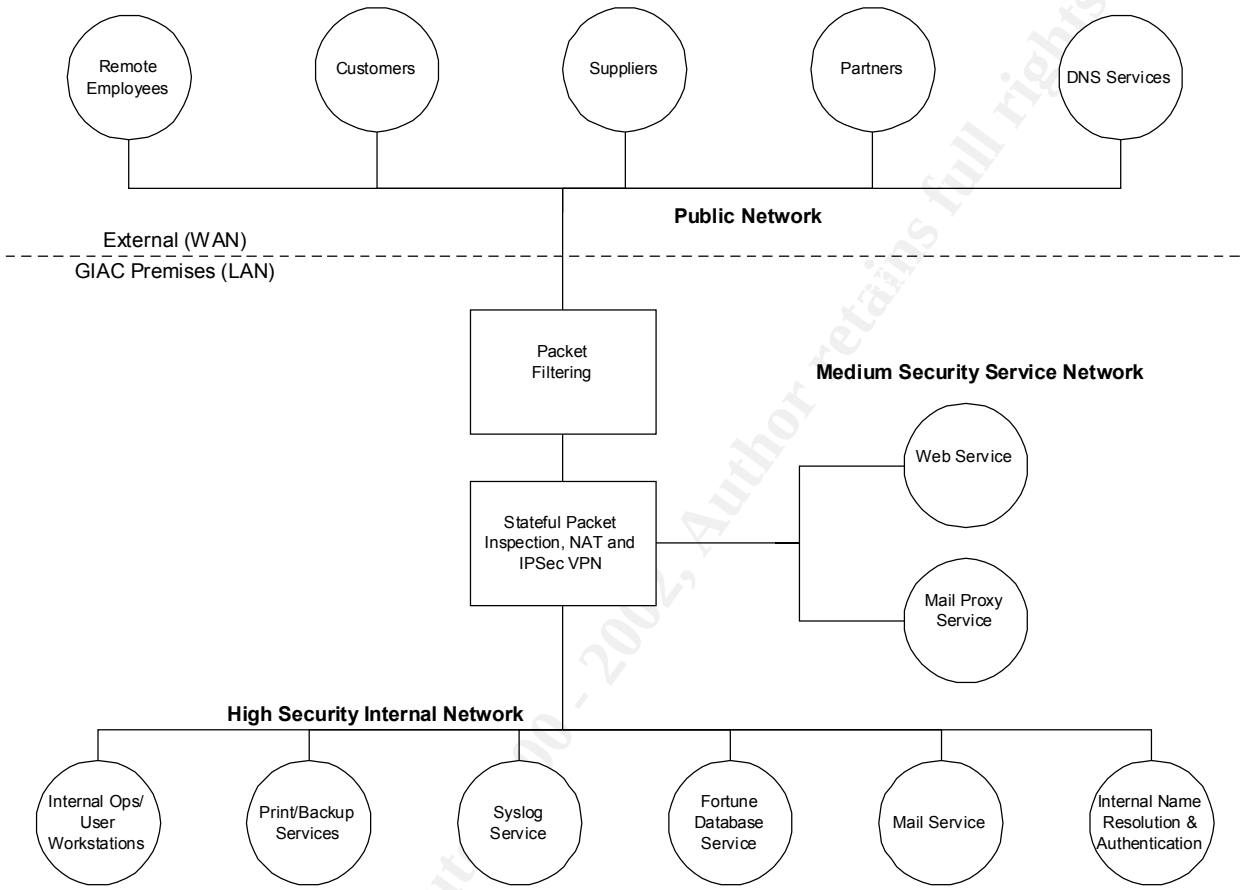


Figure 1.1: Logical Network Design

The logical design does not dictate how the services will be spread among hosts, other than the necessary separation of the two zones of trust (Service and Internal). It also does not dictate the level of hardware performance needed – this will have to be based on some educated guesses about GIAC's current and future traffic patterns.

1.3 Physical Design

Figure 1.2 presents a physical design that provides all of the features of the logical design, is as simple as reasonably possible and will scale well as GIAC grows. The following design decisions are explained in more detail below:

- Dedicated T1 connection to the Internet
- Cisco 3620 border router running IOS 12.1
- Cisco PIX 515 primary firewall running v. 5.3
- Centralized SYSLOG service
- SafeNet SoftPK as the VPN client software
- Cisco 827 DSL router as a lightweight Partner VPN gateway
- MailSweeper for SMTP v. 4.21 as mail proxy software
- Hardening of the Web and Mail Proxy Servers

1.3.1 Dedicated T1 Connection

It is assumed that at this point GIAC's traffic can be handled by a dedicated T1. This is the starting point from which GIAC will grow. Depending on the pattern of that growth, the VPN traffic could have a separate circuit dedicated to it in the future. The assumption here is that these two traffic types may scale differently over time. This approach would require two synchronous serial interfaces on the border device, and expansion capability has been designed in for that purpose (a second WIC slot). If GIAC chooses to scale the single circuit instead they can do so using the second network module slot as is described below.

1.3.2 Cisco 3620 Border Router

This router is a good choice for the border device. It is designed to process packets at LAN speeds so it will allow for traffic growth in the future. It is also flexible enough to provide the interfaces GIAC needs now while still allowing for future expansion. Specifically, GIAC will need the router chassis (CISCO3620) and a network module providing a fast ethernet interface and a synchronous serial interface (NM-1FE2W=,WIC-1T=). The unused WIC slot will allow the addition of another dedicated T1 later on if GIAC decided to split off the VPN traffic (WIC-1T= cost approx $300). Although a 10Mbps LAN interface would be sufficient to support the T1, it would not allow for future expansion. The 100Mbps LAN interface will support larger WAN upgrades later on (using the second network module slot) such as a 25Mbps or even 45Mbps ATM connection (NM-1ATM-25= or NM-1A-T3=). The cost of the initial system is approximately $5,000 including an extended support agreement, with WAN interface upgrades to 25Mbps and 45Mbps running about $1,600 and $4,500 respectively. Carrying costs of the dedicated circuits and of the ISP fees must be budgeted for as well. The circuit costs will vary considerably based on locality and distance from the serving facility.

The 3620 will allow for the packet filtering at the border called for by the logical design. Cisco ACLs are a very standard way to accomplish this, as will be seen in the configuration details in the Policy

section. The router will also be hardened to prevent attacks against itself since it will have no other protection from the Internet.

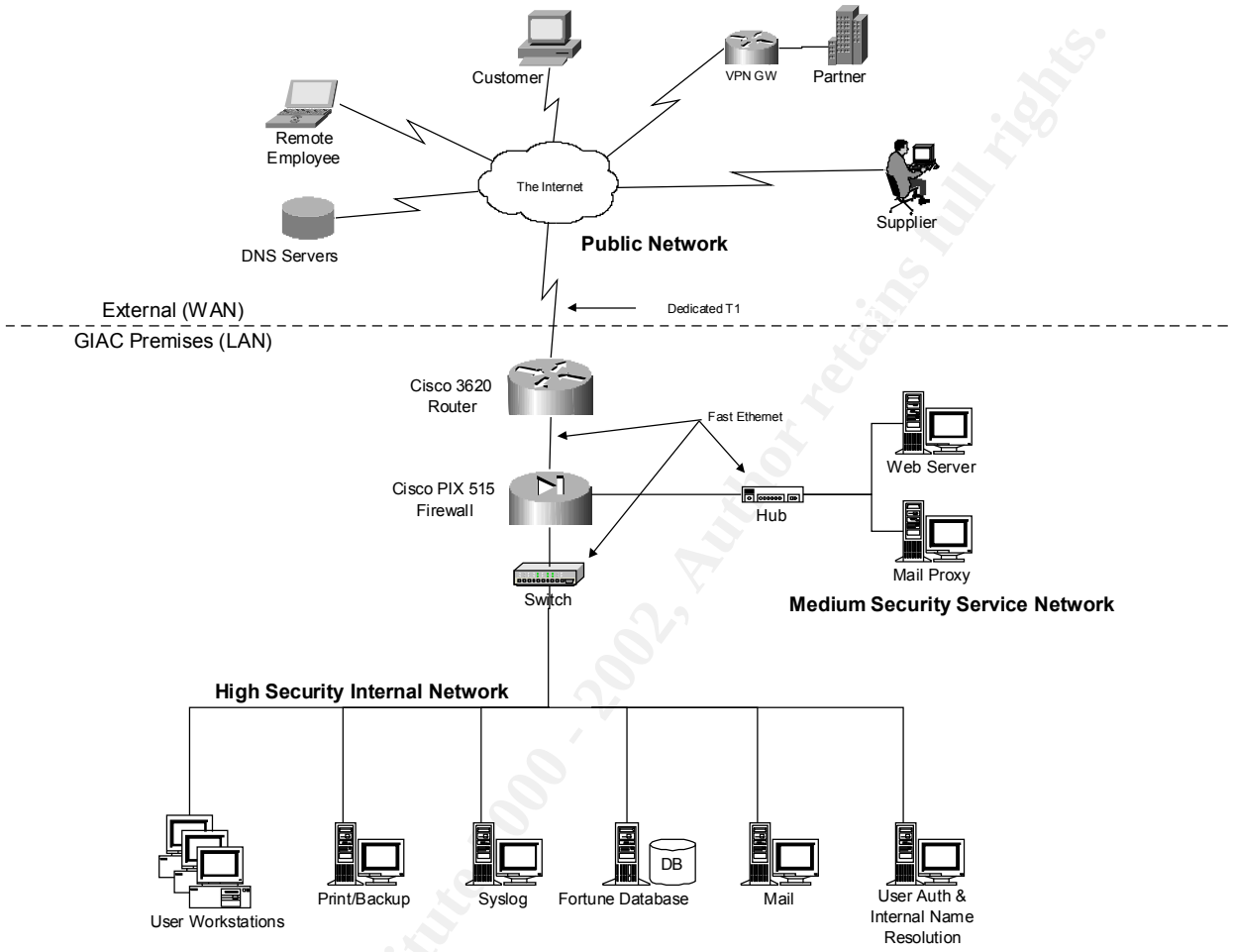


Figure 1.2: Physical Network Design

1.3.3 Cisco PIX Primary Firewall

The Cisco PIX 515 dedicated firewall device provides several key security services in the design.

It provides the stateful firewalling needed at the access point to the internal LAN to reduce the risks associated with use of the Internet by employees.

It also provides NAT (Network Address Translation) which will both conserve public address space and hide the details of the private addressing scheme.

The specific model used (PIX515-UR-BUN) has multiple backside interfaces allowing for the zone segmentation called for in the logical design. The three spare LAN interfaces will support any future needs for additional security zones.

It will provide fine-grained control of the exceptions necessary for

- the Mail Proxy's access to the internal Mail Server
- the Web Server's access to the internal Database Server
- public access to the Mail Proxy and Web Server
- access to the SYSLOG server for the border router and the service network hosts

And finally the PIX will act as a powerful VPN gateway device supporting the strongest IPSec protocols and algorithms while managing multiple connections simultaneously. The PIX has the computing power to handle all of these jobs at once – it is capable of thousands of simultaneous translations and connections. [1-1]

The specific device called for here will cost about $10,000 including an extended support agreement.

1.3.4 Centralized SYSLOG Service

This design calls for a centralized SYSLOG service as the main source of data for intrusion and fault detection. While it is certainly true that sifting through SYSLOG files is a more "crude" way of watching the perimeter than using more specialized IDS and device management tools, it is also a whole lot cheaper. Assuming that the clocks on all of the SYSLOG enabled devices are kept in synch and that the devices are in fact configured to log interesting events, the resulting log files should provide a wealth of information. As GIAC grows and money becomes available the systems administrators can migrate to more sophisticated tools.

Hardware requirements are simple. A host on the internal LAN (and it doesn't have to be especially powerful) can be dedicated to receiving SYSLOG messages. A host-based internal firewall and a thorough OS hardening would be very effective here since the types of traffic necessary to this server's operation are limited. For example, IP Chains could be used if the operating system is Linux.

1.3.5 SafeNet Soft-PK as VPN Client Software

SafeNet Inc.'s Soft-PK VPN client software is a good solution for the remote employees (and Suppliers if they choose). Version 5.1.3 was tested for use with this design and is available online from safenet-inc.com for approximately $100. This client works especially well with Cisco gateway devices – in fact the Cisco client is just a branded version of Soft-PK. The reason SafeNet's direct product is used here is that it will run under Windows 2000 while Cisco's still does not. At the time of this writing SafeNet's online knowledge base specifies minimum Cisco OS versions for interoperability as follows: IOS version 12.1.(4), PIX OS version 5.1(4).

While running on a remote computer, this software examines all traffic passing in or out of every network interface present (including dial-up). If traffic on an interface matches a profile specifying IPSec protection, then the traffic is either afforded protection (if it is headed out) or dropped if it lacks the proper protection (if it is headed in). The software handles IKE negotiations automatically whenever new IPSec SAs are needed.

It should be noted also that some form of personal firewall software should be running on these computers as well. Specific configurations for such a product are not included in this design, but I have had good luck with Network ICE's BlackICE Defender product. There is specific information on their website regarding interoperability with Soft-PK. [1-2]

1.3.6 Cisco 827 DSL Router as a Lightweight VPN Partner Gateway

The choice of the remote partner VPN gateway devices is not really GIAC's decision, but at least one example is needed in order to show how the configuration files are coordinated on either end. I have used the 827 for this purpose in order to show that such a device need not be terribly expensive (about $550 with IOS and DRAM upgrades). Of course this admittedly lightweight router will support only a few users, but for purposes of illustration it works well since the IOS configuration steps are the same for the more powerful Cisco routers.

1.3.7 MailSweeper for SMTP as the Mail Proxy Software

There are two significant benefits to using a mail proxy to handle all email going in or out of GIAC. The first is anti-virus protection.

The simplest way for systems administrators to prevent email viruses from getting into or out of the organization is to stop them at the border. Desktop anti-virus software is useful and important, but it is painful to coordinate on an enterprise-wide basis. A mail proxy system such as Baltimore Technologies' MailSweeper for SMTP fills this gap perfectly. It answers incoming and outgoing SMTP requests but does not act on them before checking messages against a set of pre-configured policies. For virus checking, MailSweeper calls on the services of one or more third-party anti-virus engines and either approves or quarantines based on the results. Baltimore Technologies is closely allied with Command Software and the interaction of MailSweeper v 4.21 and the Command engine was tested without problems. Another very nice feature of this software is that it will send notifications to the sender (and anyone else you want) in case of an infection or other policy violation. Attachment policies such as the blocking of .vbs or .exe files can be enforced here, as well as text based policies such as "dirty word" blocking and automatic attachment of disclaimers.

The second benefit to using a mail proxy is that the actual mail server can be fully isolated from any sort of direct connection from the Internet. This eliminates all possibility of attacks designed to exploit vulnerabilities in common mail systems such as MS Exchange Server or Sendmail. From the public's point of view the proxy is the mail server – the DNS MX record will point to the proxy's IP address. [1-3]

1.3.8 Hardening of the Web and Mail Proxy Servers

Many attacks against these sorts of "exposed" servers cannot be stopped by perimeter devices due to their in-band nature (i.e. they are accomplished by sending crafted content within otherwise normal traffic). In order for the servers to do their jobs, they must have certain standard ports open

to the Internet. As a result, an adequate security architecture must include provisions for hardening any servers that will be exposed to the Internet in this way.

The hardening must address two general areas – the operating system and the service software itself. Fortunately there are some excellent online resources that explain these processes in great detail. Since I do not intend to include these procedures in the Policy section of this paper, I am including some links here for further research.

Operating Systems

A good overview of whole process:
http://www.sans.org/infosecFAQ/securitybasics/host_sec.htm

Windows NT/2000:
http://www.sans.org/newlook/publications/ntstep.htm
http://www.enteract.com/~lspitz/nt.html
http://www.cert.org/tech_tips/win_configuration_guidelines.html

Linux:
http://www.enteract.com/~lspitz/linux.html

Solaris:
http://www.enteract.com/~lspitz/armoring.html

Also take a look at the Step-by-Step Series Publications from SANS:
http://www.sansstore.org/

General UNIX hardening suggestions compiled by CERT:
http://www.cert.org/tech_tips/unix_configuration_guidelines.html

Services

General FTP hardening suggestions compiled by CERT:
http://www.cert.org/tech_tips/anonymous_ftp_config.html

General Web security information resources provided by CERT:
http://www.cert.org/other_sources/websec.html

Microsoft's Security Guides for Exchange Server:
http://www.microsoft.com/technet/security/email.asp

Microsoft's Security Guides for IIS:
http://www.microsoft.com/technet/security/web.asp

CERT summary including Sendmail issues:
http://www.cert.org/summaries/CS-96.04.html

Once the initial hardening process has been accomplished, administrators will need to stay current as new vulnerabilities are discovered and patches become available. I have included some links here for that purpose.

Searchable Securityfocus (Bugtraq) Archive:

http://www.securityfocus.com/ *** Vulnerabilities section

Searchable CERT Announcements:

http://www.cert.org/

Searchable Microsoft Security Bulletins:

http://www.microsoft.com/technet/security/current.asp

2. Security Policy for GIAC Enterprises

2.1 Introduction to the Cisco Command Line

The following sections assume a basic familiarity with command line navigation in Cisco IOS and PIX OS. For those readers who have not worked with Cisco hardware before, I have included my version of the basics and some references for further study here. If you already know this stuff just skip to the next section.

Basic Authentication and Navigation

Normally you will get to a command prompt on a Cisco router or PIX firewall in one of two ways – terminal session via a cable to the 'Console' port or telnet session via a network connection. Initial setup out of the box always requires a console session, if only to get the network interface up and ready for telnet. If you connect via telnet, you will normally be challenged for a 'login' password. After entering this password, you will gain access to a command prompt in what is known as the "User Exec" mode. If you connect via the console port you may have immediate access to the User Exec mode without a password (e.g. right out of the box). If the location of the device is not secure, then a password requirement for console sessions may be configured also. Assuming that we have set the hostname of the device to 'Cisco' the first command prompt should look like this:

Cisco>

To get to the next higher level of control you must enter the word 'enable' at this prompt, and answer the next password challenge. Once authenticated you will be given access to "Privileged Exec" mode (often referred to as "Enable" mode), and the prompt will now look like this:

Cisco#

From here we have access to many commands that will show the current status of the device's interfaces, file systems, memory, routing tables, etc. We can also access the next higher level of control by entering 'configure terminal' at this prompt. Doing this puts us in "Global Config" mode and the prompt looks like this:

Cisco(config)#

Finally, we can change the focus of what we want to work on by moving into one of the many interface or sub-interface configuration modes. For example, if we want to set the IP address of interface 'eth 0', we would enter the command 'interface eth 0' at the Global Config prompt. The prompt will now change to indicate that we are at one of the "Interface Config" modes (although it does not indicate which one – you need to stay aware of where you are as you navigate).

Cisco(config-if)#

If you want to backtrack, just enter the word 'exit'. That's basically it for IOS and PIX navigation. You have different commands available in each mode and you will constantly move back and forth between modes as you make changes and check your work. Knowing exactly which mode you have to be in and which command to use to accomplish a given task is the real challenge. Cisco's website has volumes of documentation on this stuff – I usually have several browser windows going at once while I am trying to figure out something new.

Basic Commands

Some basic commands you should be familiar with include the following:

Cisco#write term	-- lists the current Running Configuration of the router. You will use this a lot to see the effect of your configuration changes.
Cisco#write mem	-- copies the current Running Configuration to the Startup Configuration in NVRAM. You have to do this to make your changes permanent. (This is a good thing. The device always starts up with the Startup Configuration, so if you make an error you can't recover from just reboot.)
Cisco#sho int <int>	-- this command shows you the status of a particular interface.
Cisco#sho access-list	-- this shows the number of matches for each line of each access-list. Use this to see how your ACLs are working.
Cisco#copy run tftp:	-- IOS: this is the standard way to make a backup of your config file. Get a tftp server from the Cisco site or from a shareware site like tucows or davecentral.
Cisco#write net <ip>:file	-- PIX: this is the tftp backup command with PIX syntax. Most of the time PIX syntax is the same as IOS, but PIX OS is a different operating system, so it is always worth checking.
Cisco#full-help	-- by default, the help function (accessed with the question mark) only shows you a partial list of your options. This command tells the OS to reveal lots more options (but still not all!)
Cisco#?	-- this is probably the single most important command you can learn! At any prompt or at any point during command entry you can type a question mark to see what your options are. If you append it directly to something you are typing you get a list of word completion options. If you type it by itself (with a space in front of it) during command entry you will get a list of the possible options you can type next. This tool is absolutely invaluable.

Cisco and SYSLOG

IOS access lists can generate SYSLOG messages when rules are matched. You enable this feature on a line-by-line basis by appending the 'log' keyword. (Note: PIX access lists do not have this option.) IOS generates a message the first time a rule is matched and then once every five minutes after that indicating the total number of matches within that period. This is a good thing. If IOS generated a message for every single match, logfiles would quickly become full and routers would have to have a lot more horsepower to do the same job. If you need more detail you best bet is to study the raw

(unfiltered) packet flow with a packet sniffer or IDS tool. The commands used to turn on and configure the SYSLOG service are as follows:

Cisco(config)#logging on	-- both IOS and PIX: turns it on
Cisco(config)#logging trap debugging	-- both IOS and PIX: sets the max level of detail. You can specify less detail with other keywords.
Cisco(config)#logging <ip>	-- IOS only: specifies the IP of the SYSLOG server
Cisco(config)#logging host inside <ip>	-- PIX only: specifies the IP of the SYSLOG server as well as the interface through which it can be reached.

Even though the PIX does not provide a line-by-line logging option for ACLs, it gives you just as much and more. You get messages whenever a packet is dropped for any reason, for example. You also get messages every time the PIX creates or tears down connections, negotiates IPSec stuff, etc..

References for Further Study

Mark Tripod's book Cisco Router Configuration & Troubleshooting is excellent. I will also give a qualified recommendation for Gil Held and Kent Hundley's Cisco Access Lists Field Guide. The cover advertises it as a reference book but I would call it more of an idea book.

It is also worth spending some time playing with the debugging commands. Debugging is an art in itself and there are more ways to watch things happen on a Cisco box than you can believe. One of the most useful tricks I have found is using a filtering ACL with certain debugging commands that support it (particularly packet and NAT . On a production box there is really no other way to debug at the packet level, so it is worth learning how to do.

And finally, I will say that the absolute best way to learn this stuff is to play, play, play!

2.2 IP Addressing Scheme

Figure 2.1 presents the IP addressing scheme I will be using for all of the examples.

The external interface of the 3620 router (serial 0) will have an IP address assigned by the ISP out of the same block as its default gateway (5.5.5.0/24).

The ISP has also provided GIAC with a block of 16 public IP addresses (9.9.9.0/28). The web and mail proxy servers will have addresses assigned to them out of this block for use in the external DNS zone files, but the IP addresses configured on the hosts themselves will be assigned out of a private address space dedicated to the service network (10.2.2.0/24). The outside interface of the PIX, the inside interface of the router and the NAT address internal network users will use to connect to the Internet will also be assigned out of the public block. The SYSLOG server will have one public address assigned to it also, but only to provide a conduit for the log messages from the router. This address will not appear in the zone files and no traffic will be allowed to it from the Internet.

Another private block of addresses are dedicated to VPN users (10.3.3.0/24). Part of this block will be used as a pool of dynamically assigned addresses and part will be used for fixed addresses. This block is not shown on the diagram.

Finally, all of the hosts on the internal network (as well as the inside interface of the PIX) will be assigned addresses from another private block dedicated for this use (10.1.1.0/24).

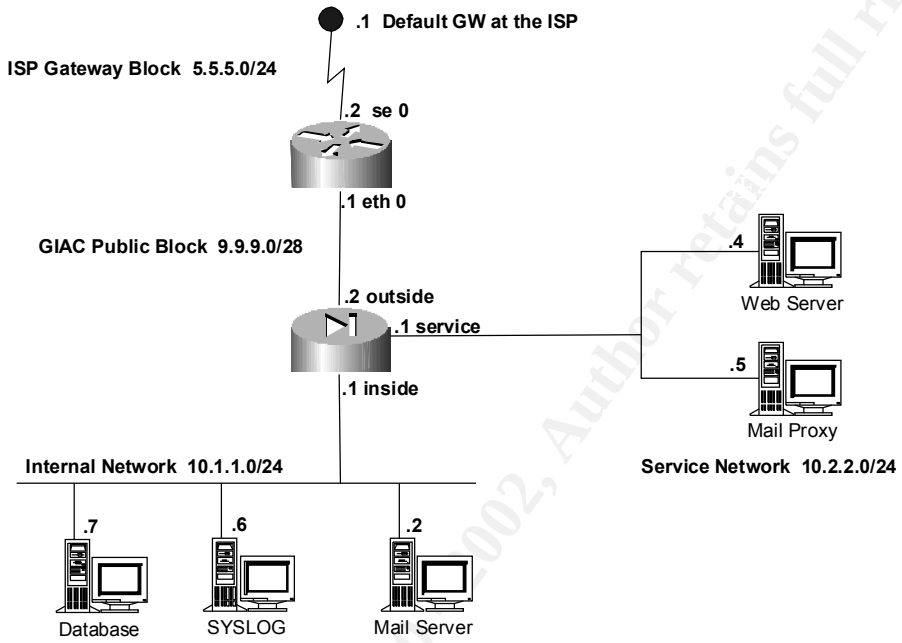


Figure 2.1: IP Addressing Scheme

Here is the complete list of the significant IP addresses for this design:

IP Address	Host/Interface
5.5.5.1	Default Gateway at the ISP
5.5.5.2	Outside interface of the 3620 (se 0)
9.9.9.1	Inside interface of the 3620 (eth 0)
9.9.9.2	'Outside' interface of the PIX
9.9.9.3	NAT address that internal users will use to reach the Internet (managed by the PIX)
9.9.9.4	Public address of the Web Server (External DNS will 'know' this address)
9.9.9.5	Public address of the Mail Proxy Server (External DNS will 'know' this address)
9.9.9.6	'Public' address of the SYSLOG server (no internet access will be allowed to this IP)
10.2.2.1	'Service' interface of the PIX
10.2.2.4	Private address of the Web Server
10.2.2.5	Private address of the Mail Proxy Server
10.2.2.6	Service network address of the internal Mail server

10.2.2.7 Service network address of the SYSLOG server
10.2.2.8 Service network address of the Database server

10.1.1.1 'Inside' interface of the PIX
10.1.1.2 Private address of the Mail Server
10.1.1.6 Private address of the SYSLOG Server
10.1.1.7 Private address of the Database Server

10.3.3.0/25 VPN: Dynamic pool
10.3.3.128/25 VPN: Static pool

2.3 The Border Router

The border router configuration will address three requirements:

- Hardening of the router itself
- Anti-spoofing (both ingress and egress)
- Blocking of vulnerable service ports

All IOS commands assume that the router's hostname is '3620'.

2.3.1 Hardening of the Router Itself

Being the first line of defense, the router itself must be hardened against attacks from the Internet.[2-1]

1. Limit Administrative (Telnet) Access to internal hosts

We will specify that only hosts on the internal network can connect to the router via telnet. All internal hosts have their source addresses translated by NAT in the PIX and will appear to the router as originating from the same address.

```
3620(config)#access-list 10 permit host 9.9.9.3
3620(config)#line vty 0 4
3620(config-line)#access-class 10
3620(config-line)#login
```

2. Disable SNMP

SNMP is a network management protocol. If GIAC grows as much as it expects to it may make sense later on to look at an SNMP management platform like HP OpenView, but for now we are going to turn it off. This will prevent malicious use of it from outside.

```
3620(config)#no snmp
```

3. Disable Loose Source Routing and Proxy-ARP

Unless blocked, loose source routing can be used by an attacker to bypass ACLs. Proxy ARP can reveal MAC address information about your internal network.

3620(config)#no ip source-route
3620(config)#no ip proxy-arp

4. Disable the HTTP service

This method of connecting to the router does not restrict the number of bad logon attempts, so it must be shut down.

3620(config)#no ip http

5. Encrypt Passwords

By default, passwords in IOS configs are printed in cleartext. They can be encrypted during display by using this command. This will make accidental password exposure much less likely (but don't forget to write them down somewhere safe).

3620(config)#service password-encryption

6. Enable SYSLOG

All log messages are directed to the internal SYSLOG server via its 'public' address. The PIX is going to translate this traffic and send it on to the private inside address (10.1.1.6).

3620(config)#logging on
3620(config)#logging 9.9.9.6
3620(config)#logging trap debugging

7. Configure a Warning Banner

A warning banner will help preserve GIACs legal rights in the event of an intrusion.

3620(config)#banner login / Warning: Authorized access only /

2.3.2 Anti-Spoofing

It is important to block any inbound traffic that claims to be sourced from inside GIAC and any outbound traffic that claims to be sourced from outside GIAC because this is a common method of attack. The ingress list blocks any packets with source addresses 1) from an RFC 1918 private range or 2) from GIACs public address block and is applied as packets enter the serial interface. The egress list allows packets sourced from GIACs public block and denies all others. The ingress list will be extended for other purposes in section 2.3.3.

```
3620(config)#access-list 101 remark Ingress Filter
3620(config)#access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
3620(config)#access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
3620(config)#access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
3620(config)#access-list 101 deny ip 9.9.9.0 0.0.0.15 any log
3620(config)#access-list 101 permit ip any any
3620(config)#int serial 0
3620(config-if)#access-group 101 in

3620(config)#access-list 201 remark Egress Filter
3620(config)#access-list 201 permit ip 9.9.9.0 0.0.0.15 any
3620(config)#access-list 201 deny ip any any log
3620(config)#int eth 0
3620(config-if)#access-group 201 in
```

2.3.3 Blocking of Vulnerable Service Ports

For this section I have combined a requirement specific to GIAC with the CERT recommendations for the blocking of vulnerable service ports.[2-2] GIAC specifically needs to block Internet access to the public SYSLOG address. The following are additions to ingress list 101. They should be inserted just before the last line (permit ip any any). IOS does not allow you to insert lines after the fact, however, so the whole access-list will have to be dropped and reentered and the access-group command reapplied.

```
3620(config)#access-list 101 remark Block access to SYSLOG server
3620(config)#access-list 101 deny ip any host 9.9.9.6
3620(config)#access-list 101 remark Block TFTP
3620(config)#access-list 101 deny udp any any eq 69
3620(config)#access-list 101 remark Block LINK
3620(config)#access-list 101 deny tcp any any eq 87
3620(config)#access-list 101 remark Block SunRPC and NFS
3620(config)#access-list 101 deny udp any any eq 111
3620(config)#access-list 101 deny tcp any any eq 111
3620(config)#access-list 101 deny udp any any eq 2049
3620(config)#access-list 101 deny tcp any any eq 2049
3620(config)#access-list 101 remark Block BSD 'r' commands and LPD
3620(config)#access-list 101 deny tcp any any range 512 515
3620(config)#access-list 101 remark Block UUCPD
3620(config)#access-list 101 deny tcp any any eq 540
3620(config)#access-list 101 remark Block Openwindows
3620(config)#access-list 101 deny udp any any eq 2000
3620(config)#access-list 101 deny tcp any any eq 2000
3620(config)#access-list 101 remark Block X-Windows
3620(config)#access-list 101 deny udp any any range 6000 6100
3620(config)#access-list 101 deny tcp any any range 6000 6100
```

The other service port closures and ICMP controls are accomplished with other IOS commands as follows:

```
3620(config)#no service tcp-small-servers
3620(config)#no service udp-small-servers
3620(config)#no ip bootp
3620(config)#no service finger
3620(config)#no ip direct-broadcast
3620(config)#no ip unreachables
```

2.4 The Primary Firewall

The primary firewall configuration will address three requirements:

1. Separation of the network into three different zones of trust
2. NAT translation and stateful firewalling for hosts connecting to the Internet
3. Negotiation of IPSec VPN connections for partners, suppliers and remote employees

All PIX OS commands assume that the PIX's hostname is 'Pix'.

2.4.1 Separation of the Network into Three Different Zones of Trust

Each interface on the PIX is assigned a number indicating its relative level of security. By default, the 'Outside' interface is at the lowest level 0 and the 'Inside' inteface is at the highest level, 100. The other four interfaces can be anywhere in between. We will be using one of them for the service network. The following commands will configure IP addressing for the PIX according to the scheme presented earlier. The IP addresses of the inside and service interfaces will function as the default gateway addresses for all hosts on the related networks. Note the 'route' command pointing to the ethernet interface of the 3620. This defines the default gateway for the PIX itself.

```
Pix(config)#nameif ethernet0 outside security0
Pix(config)#interface ethernet0 auto
Pix(config)#ip address outside 9.9.9.2 255.255.255.240
Pix(config)#nameif ethernet1 inside security100
Pix(config)#interface ethernet1 auto
Pix(config)#ip address inside 10.1.1.1 255.255.255.0
Pix(config)#nameif ethernet2 service security50
Pix(config)#interface ethernet2 auto
Pix(config)#ip address service 10.2.2.1 255.255.255.0
Pix(config)#route outside 0.0.0.0 0.0.0.0 9.9.9.1
```

These commands enable logging on the PIX.

```
Pix(config)#logging on
```

```
Pix(config)#logging trap debugging
Pix(config)#logging host inside 10.1.1.3
```

2.4.2 NAT translation and Stateful Firewalling for hosts connecting to the Internet

Unless specifically configured otherwise, the PIX allows connections to be initiated only from a zone of higher security to one of lower security. But for hosts on the inside and service networks to connect to the Internet they will also need their source addresses translated to something public. There are also some special exceptions to the connection rules that need to be considered.

1. The two hosts on the service network will need their own static NAT assignments so that the public addresses can be published in the external DNS zone files. They also need special exceptions to allow Internet hosts to connect to their service ports. The following commands will accomplish this. We are allowing access to ports 80 and 443 on the web server and port 25 on the mail proxy. The last parameter in the static commands specifies an 'embryonic' connection limit of 1000 for the web server and 50 for the mail proxy server. This tells the PIX to disallow new connection requests if the number of incomplete connections has reacheed this limit. Ideally we have set these values high enough to accommodate the peak traffic load while still protecting the hosts from SYN attacks.

```
Pix(config)#static (service,outside) 9.9.9.4 10.2.2.4 netmask 255.255.255.255 1000
Pix(config)#static (service,outside) 9.9.9.5 10.2.2.5 netmask 255.255.255.255 50
Pix(config)#conduit permit tcp host 9.9.9.4 eq www any
Pix(config)#conduit permit tcp host 9.9.9.4 eq 443 any
Pix(config)#conduit permit tcp host 9.9.9.5 eq smtp any
```

2. The SYSLOG server needs a public address for use by the router, but this address will not be accessible from the Internet and it will not be published in the DNS zone files. (Internet access to this address was blocked at the border router in section 2.3.3.) It also needs an address on the service network so the web and mail proxy servers can send their SYSLOG messages there also. In both cases we will have to configure exceptions to allow the messages through. Note that the conduit for the public side will pass SYSLOG traffic only from the 3620 and the conduit for the service network will pass it only for source addresses on the service network.

```
Pix(config)#static (inside,outside) 9.9.9.6 10.1.1.6 netmask 255.255.255.255 0
Pix(config)#conduit permit udp host 9.9.9.6 eq syslog host 9.9.9.1
Pix(config)#static (inside,service) 10.2.2.7 10.1.1.6 netmask 255.255.255.255 0
Pix(config)#conduit permit udp host 10.2.2.7 eq syslog 10.2.2.0 255.255.255.0
```

3. The inside Mail Server needs an address on the service network and a matching exception to allow SMTP connections from the Mail Proxy.

```
Pix(config)#static (inside,service) 10.2.2.6 10.1.1.2 netmask 255.255.255.255 0
Pix(config)#conduit permit tcp host 10.2.2.6 eq smtp host 10.2.2.5
```

4. The Database Server needs an address on the service network and a matching exception to allow SQL-Net connections from the Web Server (assuming that the DB is Oracle).

Pix(config)#static (inside,service) 10.2.2.8 10.1.1.7 netmask 255.255.255.255 0
Pix(config)#conduit permit tcp host 10.2.2.8 eq 1521 host 10.2.2.4

5. All of the other hosts on the internal network need a NAT translation to connect to the Internet, but we don't have enough public addresses in our block to assign one to everybody. The answer to this is a feature called PAT (port address translation, also referred to as 'overloaded NAT'). In this case the PIX is going to use a single public address as the translated source address for all of the internal hosts. It will do so by assigning distinct source ports to each connection and keeping up with all the mappings. The 'global' command assigns the public address and the 'nat' command defines the group of hosts that will have translation applied. This situation will become a little more complicated in section 2.4.3 when VPN connections are defined.

Pix(config)#global (outside) 9.9.9.3 netmask 255.255.255.240
Pix(config)#nat (inside) 1 10.1.1.0 255.255.255.0 0

2.4.3 Negotiation of IPSec VPN connections

The PIX commands that will enable IPSec VPN connections will be very similar for partners, suppliers and remote employees. Looking back at the Security Architecture we see that we can count on fixed IP addresses for partners and suppliers but not for remote employees. It will also make sense to use unique keys for each partner and supplier, so that if one becomes compromised we won't have to ask all of them to change their configurations. The remote employee pool can share a key since we have full administrative control over that group of host configurations (but we can always go to a unique key scheme for employees if it becomes a problem).

Regarding all of the other IKE and IPSec parameters, we are going to simplify as much as possible. All of the VPN connections we want to support will require encryption of the packet payloads to prevent snoopers from getting database passwords or other sensitive company information. Therefore ESP will be used instead of AH. This design will be simplified even further by requiring a consistent set of parameters for all connections, regardless of their origin. To accomplish this a parameter set has been chosen that will meet the needs of the most sensitive connection.

IKE SA negotiation (Phase 1)	Authentication:pre-shared key	
	Encryption Alg:	3DES
	Hash Alg:	SHA-HMAC
	Diffie-Hellman Grp:	2
	SA lifetime:	86400 sec (24 hours)
IPSec SA negotiation (Phase 2)	Mode:	ESP in tunnel mode
	Encryption Alg:	3DES
	Hash Alg:	SHA-HMAC

The phase 1 parameters are configured as part of an 'isakmp policy' as follows:

```
Pix(config)#isakmp policy 1 authentication pre-share
Pix(config)#isakmp policy 1 encryption 3des
Pix(config)#isakmp policy 1 hash sha
Pix(config)#isakmp policy 1 group 2
Pix(config)#isakmp policy 1 lifetime 86400
```

The phase 2 parameters are configured together as a 'transform set' as follows:

```
Pix(config)#crypto ipsec transform-set GIAC esp-3des esp-sha-hmac
```

Next we associate pre-shared key values with all of the fixed source addresses using 'isakmp key' commands. At this point GIAC has two partners and three suppliers with fixed addresses as follows:

IP Address	Host
4.4.4.1	Supplier 1
4.4.4.2	Supplier 2
4.4.4.3	Supplier 3
7.7.7.1	Partner 1 – VPN gateway router address
7.7.7.2	Partner 2 - VPN gateway router address

```
Pix(config)#isakmp key s1key address 4.4.4.1 netmask 255.255.255.255 no-xauth no-config-mode
Pix(config)#isakmp key s2key address 4.4.4.2 netmask 255.255.255.255 no-xauth no-config-mode
Pix(config)#isakmp key s3key address 4.4.4.3 netmask 255.255.255.255 no-xauth no-config-mode
Pix(config)#isakmp key p1key address 7.7.7.1 netmask 255.255.255.255 no-xauth no-config-mode
Pix(config)#isakmp key p2key address 7.7.7.2 netmask 255.255.255.255 no-xauth no-config-mode
```

The no-xauth parameter turns off a secondary authentication feature that is not compatible with all VPN clients and the no-config-mode turns off IP address autoassignment.

We have to create one more "global" key for remote employees as follows:

```
Pix(config)#isakmp key remoteemployee address 0.0.0.0 netmask 0.0.0.0
```

Now we set up the address pool that remote employees will be autoassigned out of when they connect and make it available to the IKE service.

```
Pix(config)#ip local pool remotepool 10.3.3.1-10.3.3.127
Pix(config)#isakmp client configuration address-pool local remotepool outside
```

We now need to create 'crypto maps' that will associate all of the relevant parameters together.

```
Pix(config)#crypto dynamic-map vpnclients 4 set transform-set GIAC
Pix(config)#crypto map partner-map 20 ipsec-isakmp dynamic vpnclients
```

```
Pix(config)# crypto map partner-map client configuration address initiate
Pix(config)# crypto map partner-map client configuration address respond
Pix(config)# crypto map partner-map inteface outside
```

Finally, we enable the IKE and IPSec services on the outside inteface, and change the NAT settings to exclude the VPN traffic.

```
Pix(config)#sysopt connection permit-ipsec
Pix(config)#isakmp enable outside
Pix(config)#access-list nonat permit ip 10.1.1.0 255.255.255.0 10.3.3.0 255.255.255.0
Pix(config)#nat (inside) 0 access-list nonat
Pix(config)#nat (inside) 1 10.1.1.0 255.255.255.0 0
```

At this point we have a working VPN solution. When partners or suppliers connect they will use back-end IP addresses assigned to them in advance by GIAC from the 10.3.3.128/25 pool. When remote employees connect they will be assigned an IP address automatically by the PIX from the 10.3.3.0/25 pool. The only thing left to do is restrict the partner and supplier visibility to the database server. We can do this with an ACL applied to the inside interface.

```
Pix(config)#access-list 200 permit ip host 10.1.1.7 10.3.3.128 255.255.255.128
Pix(config)#access-list 200 deny ip any 10.3.3.128 255.255.255.128
Pix(config)#access-list 200 permit ip any any
Pix(config)#access-group 200 in interface inside
```

3. Primary Firewall Audit

I have been assigned to provide technical support for a comprehensive information system audit for GIAC Enterprises. My part of the assessment is to audit the primary firewall described in sections 1 and 2.

3.1 Plan

An audit of the primary firewall should provide answers to two questions:

1. Is the firewall functioning as called for in the design?
2. Are there vulnerabilities present that were not addressed in the design?

The first question can be answered with a good measure of certainty by testing with scanning tools and packet capture utilities. The second is more open-ended, partly because it presents a moving target and partly because it requires knowledge of what all of the other pieces of the design are supposed to do. Since my role in the overall assessment is limited, this section will focus mainly on the first of these questions. Some opinions as to the second are presented as part of my summary at the end.

Technical Approach

The assessment will occur in two phases. The first will involve a review of the goals for the firewall as presented in the architecture and policy sections. The goal of this phase is to state as clearly as possible what it is that needs to be tested. This list will also include any items found during a search for known vulnerabilities for either the PIX 515 or version 5.3 software. During the second phase appropriate tests will be run against the design to verify each item identified in phase 1.

For the second phase testing, I will craft specific traffic patterns and attempt to send them either into or through the firewall. Results will be gathered using packet capture software and the logging capabilities of the firewall itself. The crafted traffic patterns will be generated with nmap for NT, and the packets captured with MS network monitor. [3-1] The SafeNet Soft-PK VPN client software will be used on the source laptop for VPN testing.

Time and Materials

Phase 1 will happen offsite and is estimated at 10 hours. The result will be a specific list of testable policy items.

Phase 2 will happen onsite and will require a hub, two laptops and cables (all of which will be provided by me). While sending crafted traffic to the outside interface the hub will be inserted between the border router and the firewall device. I will connect a laptop (using one of the spare public IP addresses) to this hub in order to send in the traffic. The other laptop will run the packet capture software and be placed on either the internal or service network as appropriate. Time estimate for this phase is 30 hours, 4 for testing and the rest for analysis and report preparation.

The timing of phase 2 is an important consideration and will be coordinated carefully with onsite administrative staff. Parts of the testing may cause very brief interruptions of service of either the

firewall device or the service network hosts. Therefore the testing should happen 1) at the time of day when GIAC's business traffic is at its lowest and 2) with GIAC administrators familiar with the service network hosts standing by to address any interruptions. Phase 2 should also be scheduled so as not to conflict with any other part of the overall information systems audit.

The total cost to GIAC for my time is estimated at 40 hours (at $125/hour).

3.2 Implementation

Results of Phase 1

The architecture defines the security services provided by the PIX as follows:

- provide stateful firewalling at the access point to the internal LAN
- use NAT to hide private addressing scheme
- segment the network into three zones of trust
- manage exceptions for cases requiring special access
- manage VPN connections

The list of testable policy items gathered from my analysis of the above requirements and of the more detailed explanations given in the policy section are as follows:

1. it should be impossible to initiate a connection from the external network to any port on any host on the internal network except where specifically configured
    i. SYSLOG Server, 9.9.9.6, udp/514, sourced only from 9.9.9.1
2. it should be impossible to initiate a connection from the external network to any port on the service network hosts not specifically configured to be open
    i. Web Server, 9.9.9.4, tcp/80, tcp/443
    ii. Mail Proxy Server, 9.9.9.5, tcp/25
3. it should be impossible to initiate a connection from the service network to any port on any host on the internal network except where specifically configured
    i. Internal Mail Server, 10.2.2.6, tcp/25, sourced only from 10.2.2.5
    ii. Database Server, 10.2.2.8, tcp/1521, sourced only from 10.2.2.4
    iii. SYSLOG Server, 10.2.2.7, udp/514
4. it should be impossible to discover the private addressing scheme of either the internal or service networks from the external network
5. it should be impossible to discover the private addressing scheme of the internal network from the service network
6. it should be impossible to negotiate a partner or supplier VPN connection from the wrong source address
7. it should be impossible to negotiate a VPN connection of any type without using the proper parameter set

Items 2,4,6 and 7 will be checked first using Test Setup 1 shown in figure 3-1.

Item1 will be checked next using Test Setup 2 shown in figure 3-2.

Items 3 and 5 will be checked next using Test Setup 3 shown in figure 3-3.

Results of Phase 2

Test Setup 1: Policy Items 2,4,6 and 7

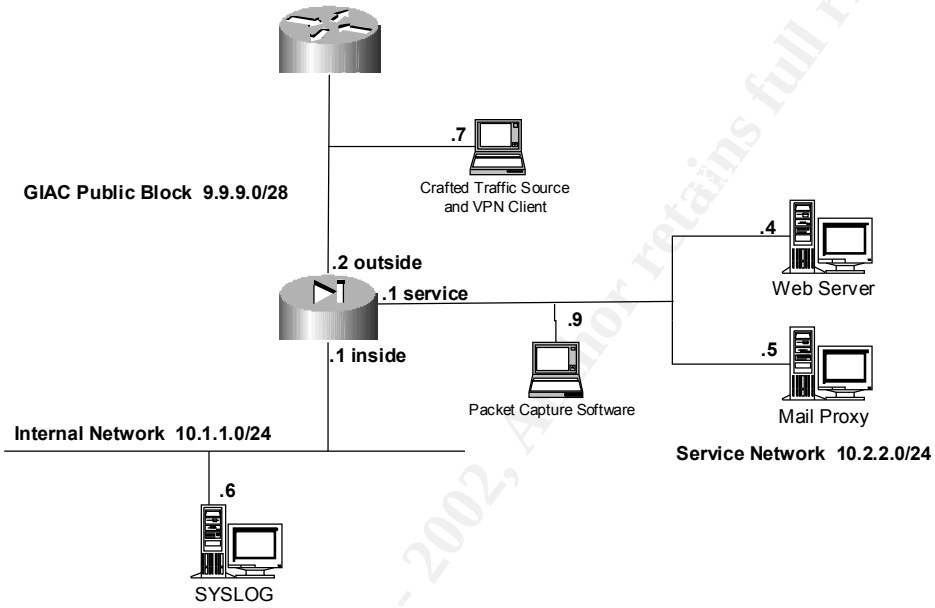


Figure 3.1: Test Setup 1

Policy item 2 was confirmed by the following tcp and udp scan tests. Note that the web server shows http and https open and that the mail proxy has smtp open. As a double-check, the SYSLOG server was checked and showed the packets being dropped. The packet capture laptop was also checked and showed those connections that were successful.

```
------ begin output ------
c:>nmapnt -sT 9.9.9.4/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Interesting ports on  (9.9.9.4):
(The 65533 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
443/tcp    open        https

Nmap run completed -- 1 IP address (1 host up) scanned in 2251 seconds
```

```
c:>nmapnt -sU 9.9.9.4/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
All 65535 scanned ports on  (9.9.9.4) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2081 seconds

c:>nmapnt -sT 9.9.9.5/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Interesting ports on  (9.9.9.5):
(The 65534 ports scanned but not shown below are in state: closed)
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 1832 seconds

c:>nmapnt -sU 9.9.9.5/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
All 65535 scanned ports on  (9.9.9.5) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2177 seconds

c:>
------ end output ------
```

Policy item 4 was checked by running an nmap ping scan against the private address ranges we know to be in use behind the firewall. (This is a lot more information than an attacker would have, but there is no need for us to grind through possibilities that we already know do not exist.) The results of the ping scan confirm that the private addressing scheme is not being revealed by the PIX even when pings are directed at the correct private addresses. A double-check of the PIX logs on the SYSLOG server shows the pings being dropped.

```
------ begin output ------
c:>nmapnt -sP 10.1.1.0/24

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Nmap run completed -- 256 IP addresses (0 hosts up) scanned in 48 seconds
```

```
c:>nmapnt -sP 10.2.2.0/24

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Nmap run completed -- 256 IP addresses (0 hosts up) scanned in 45 seconds
c:>
------ end output ------
```

Item 6 was tested using VPN client software on the source laptop. Since the IP address of the source laptop on the external network (9.9.9.7) does not equal the fixed address of any of GIAC's suppliers or partners, an attempt to establish a VPN connection using each of the partner and supplier keys was made. All such attempts failed even though all of the other parameters were correct, confirming that this restriction is functioning as expected.

Item 7 was tested using the VPN client software on the source laptop. This time the remote employee key was used (which will work with any source address). First, a positive result was confirmed by setting all other parameters to their correct values and establishing a successful connection. Next, other mixes of parameters were tried with the remote employee key, but all such attempts were unsuccessful.

Test Setup 2: Policy Item 1

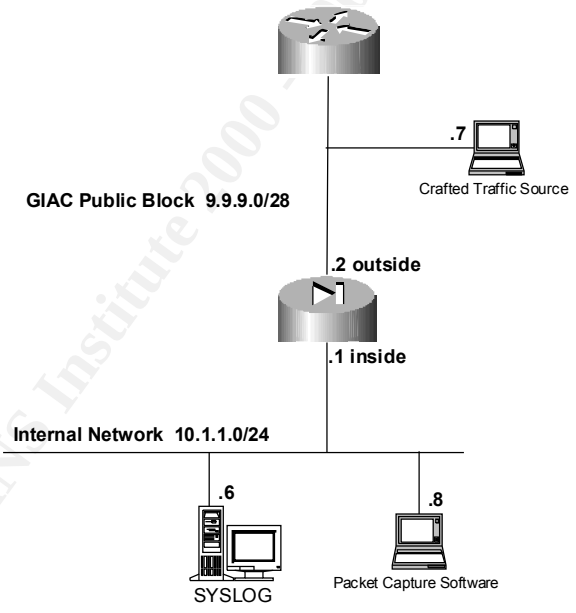


Figure 3.2: Test Setup 2

At this point the public address of the SYSLOG server (9.9.9.6) and the overloaded NAT address are the only possible targets of a scan (since the ping scan in item 4 came up blank for the private addresses). In this case we expect both the udp and tcp scans to come up blank even though the SYSLOG server does have an exception configured for the border router (9.9.9.1), because the source address of our scan will be the source laptop (9.9.9.7). Results confirm that the SYSLOG server is indeed invisible from 9.9.9.7, which we will take as sufficient proof that the conduit is

working correctly. A scan of the NAT address used by the internal users comes up blank as expected. A check of the SYSLOG files shows that the PIX is rejecting the packets as required.

```
------ begin output ------
c:>nmapnt -sT 9.9.9.6/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

All 65535 scanned ports on  (9.9.9.6) are: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 2806 seconds

c:>nmapnt -sU 9.9.9.6/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
All 65535 scanned ports on  (9.9.9.6) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1934 seconds

c:>nmapnt -sT 9.9.9.7/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

All 65535 scanned ports on  (9.9.9.7) are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 1988 seconds

c:>nmapnt -sU 9.9.9.7/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
All 65535 scanned ports on  (9.9.9.7) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2122 seconds

c:>
------ end output ------
```

Test Setup 3: Policy Items 3 and 5

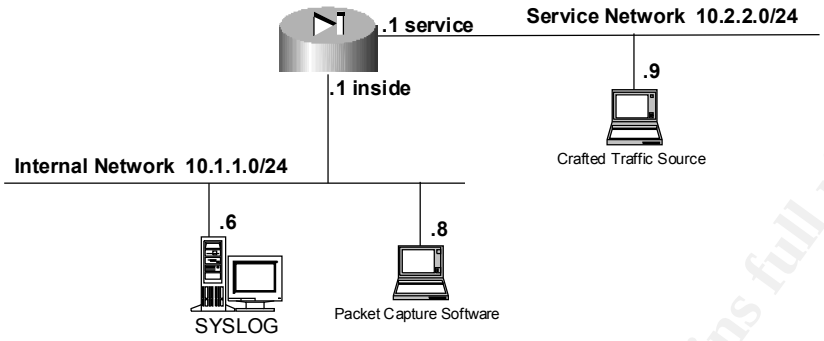


Figure 3.3: Test Setup 3

We now move the source laptop to the service network and attempt to map the addressing scheme on the internal network (item 5). As the results show, the PIX is protecting the internal network from the service network as expected. The packet capture laptop shows nothing. The SYSLOG records show the PIX dropping the pings.

```
------ begin output ------
c:>nmapnt -sP 10.1.1.0/24

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Nmap run completed -- 256 IP addresses (0 hosts up) scanned in 50 seconds

c:>
------ end output ------
```

Finally, we test policy item 3. After the failure to map the internal address space in the last step, our targets are reduced to the service network addresses of the internal Mail Server (10.2.2.6), SYSLOG Server (10.2.2.7) and Database Server (10.2.2.8). The results show that the only ports open are the ones we expect. Note that nmap reports the sql-net port 1521 as ncube-lm. I believe this is an old assignment that Oracle has "taken over".

```
------ begin output ------
c:>nmapnt -sT 10.2.2.6/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Interesting ports on  (10.2.2.6):
(The 65534 ports scanned but not shown below are in state: closed)
```

```
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 2397 seconds

c:>nmapnt -sU 10.2.2.6/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
All 65535 scanned ports on  (10.2.2.6) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2083 seconds

c:>nmapnt -sT 10.2.2.7/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

All 65535 scanned ports on  (10.2.2.7) are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 1866 seconds

c:>nmapnt -sU 10.2.2.7/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

No entries in arp cache.
Interesting ports on  (10.2.2.7):
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
514/udp    open        syslog

Nmap run completed -- 1 IP address (1 host up) scanned in 2438 seconds

c:>nmapnt -sT 10.2.2.8/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Interesting ports on  (10.2.2.8):
(The 65534 ports scanned but not shown below are in state: closed)
Port       State       Service
1521/tcp   open        ncube-lm

Nmap run completed -- 1 IP address (1 host up) scanned in 2200 seconds

c:>nmapnt -sU 10.2.2.8/32 -p 1-65535

Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )
```

```
No entries in arp cache.
All 65535 scanned ports on  (10.2.2.8) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2083 seconds

c:>
------ end output ------
```

3.3 Analysis

All of the tests went well, which says that the PIX is doing the jobs it is supposed to be doing. My suggestions for improving this perimeter design are as follows:

1. Address the fact that the PIX is a single point of failure for most of the critical network services.

If the PIX were to fail, this network would be in very bad shape! As soon as GIAC can afford it, I would strongly suggest buying a duplicate PIX as a failover unit. These devices have a hot failover capability built in that keeps both devices in perfect synch (including mirrors of all state tables). If one device fails the other takes over without interruption (using the same IP address).

2. Add a secondary firewall (something other than a Cisco product) between the PIX and internal LAN.

A multi-vendor solution is much less likely to be compromised because vulnerabilities tend to be product specific. If GIAC's admin staff includes someone with Linux experience a secondary stateful firewall could be set up very cheaply using IP Tables and an old Pentium no one wants anymore.

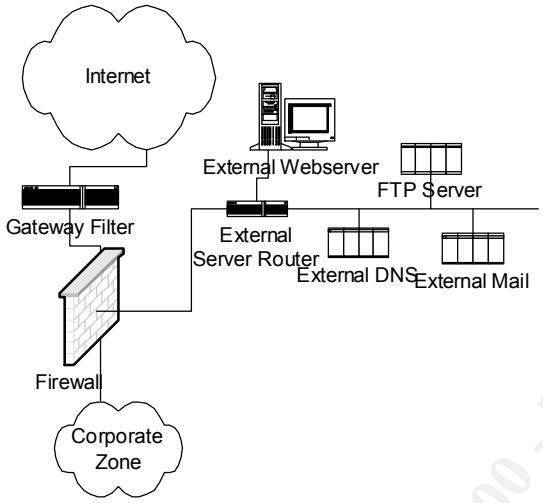
3. Add a more sophisticated IDS system to the design.

SYSLOG is great as far as it goes, but a good IDS system will go much further, especially in terms of identifying patterns. As GIAC grows so should the expertise of its security staff. Money should be budgeted each year for books, time and training to enable the staff to stay current.

4. Design Under Fire

4.1 The Target Network

For this section I have chosen to attack the network designed by Jeremy Browns (http://www.sans.org/y2k/practical/Jeremy_Browns_GCFW.zip). The parts of Jeremy's design that these attacks will target are specified in the following partial network diagram copied from his paper.



The important configuration files are those of the border router (called a "Gateway Filter" in the diagram) and the firewall. Jeremy specifies the border router as a "Cisco 4000" without specifying a particular build of IOS. He specifies the firewall as a Cisco PIX (no model number or OS version specified). Therefore I have to make some assumptions. For purposes of this section I am going to assume that the IOS software version is at least 12.0 and the PIX OS version is 5.1(0). This represents a network configuration that is out of date, but not dramatically so.

Here are the two configurations (copied from Jeremy's paper):

Boundary Router Configuration:

```
No ip direct-broadcasts
No ip source-route
No service Finger
No ip http
No ip bootp
Banner //// Warning Unauthorized Usage of Fortune Maker's Network, Will
Result in
Arrest and Possible Conviction /////
Access-list extended egress permit ip 192.71.21.0 255.255.255.0
Access-list extended egress deny ip any any logging

Interface serial 1/0
```

```
Ip address 192.21.71.1 255.255.255.0
access-group 100 in
access-list 100 deny udp any any eq 137
access-list 100 deny udp any any eq 138
access-list 100 deny tcp any any eq 139
access-list 100 deny tcp any any eq 109
access-list 100 deny tcp any any eq 110
access-list 100 deny tcp any any eq 111
access-list 100 deny udp any any eq 111
access-list 100 deny tcp any any eq 143
access-list 100 deny udp any any eq 520
access-list 100 deny tcp any any eq 389
access-list 100 deny udp any any eq 389
access-list 100 permit tcp any any
access-list 100 permit udp any any
```

Pix configuration:

```
hostname pixfirewall
interface ethernet0 auto
interface ethernet1 auto
ip address outside 192.71.21.2 255.255.255.0
ip address inside 10.1.1.1 255.255.255.0
ip address services 192.71.28.1 255.255.255.0
ip address ipsec 10.2.1.1 255.255.255.0
static(outside, services) 192.71.21.2 192.71.28.1 netmask 255.255.255.0
syslog level 5
conduit permit tcp host 192.71.21.2 eq www any
conduit permit tcp host 192.71.21.2 eq 1080 any
conduit permit tcp host 192.71.21.2 eq ftp any
conduit permit tcp host 192.71.21.2 eq ssl any
conduit permit tcp host 192.71.21.2 eq dns any
conduit permit udp host 192.71.21.2 eq dns any
static(outside, inside) 192.71.21.2 10.1.1.1  netmask 255.255.255.0
syslog level 7
conduit permit tcp
conduit permit tcp host 192.71.21.2 eq tcp established-port
conduit permit udp host 192.71.21.2 eq udp establish-port

route outside 10.0.0.0 255.255.255.240 172.17.11.3 1
static(outside,ipsec) 192.71.21.2 10.2.1.1 netmask 255.255.255.0
crypto isakmp policy 2
authentication pre-share
hash md5
crypto isakmp key inside address 10.2.1.1
crypto isakmp key outside address 192..0.20


no rip outside passive
no rip outside default
no rip inside passive
no rip inside default
no rip services passive
no rip services default
no snmp-server location
no snmp-server contact
```

4.2 An Attack Against the Firewall Itself

One of the most important features of the PIX firewall can be attacked by way of a TCP reset vulnerability made public by Cisco on July 11, 2000. In order to understand this vulnerability it is necessary to understand both the normal lifecycle of a TCP connection and how the PIX uses a connection table to provide stateful security to internal users.

Under normal operation TCP connections go through a three-stage lifecycle of establishment, data transfer and termination. The establishment phase begins when one host (the "client", identified by a source address/port pair) sends a TCP packet with the SYN bit set to another host (the "server", identified by a destination address/port pair). The server responds to the client with a TCP packet acknowledging the sequence number sent and requesting acknowledgment of its own sequence number (both SYN and ACK bits set). The "three-way handshake" is finally completed by the client when it sends a packet acknowledging the server's sequence number (ACK bit set). The second "data transfer" phase is then perpetuated for as long as need be and is kept in synch by the constant exchange and verification of sequence numbers – in other words, each side always knows what sequence number to expect next from the other side. Normally, the termination phase is accomplished with a four-way exchange using the FIN and ACK bits, but there is another way to end a connection that concerns us here. If either side of the exchange decides it is necessary (due to receipt of a message with a bad sequence number, for example) to end the connection abruptly, it can sends a TCP packet with the RST bit set. The design of TCP requires that a host receiving a valid RST request must terminate the connection immediately with no response sent to the requesting side. [4-1]

A "stateful" firewall like the PIX does two main things to improve security for the hosts behind it. First, it does not allow the three-way handshake to be initiated from an outside host (aside from specifically configured exceptions). Second, it maintains a "connection table" of all connections initiated through it by inside hosts, and checks all incoming TCP traffic against it for validity. In this way it can block any incoming TCP packets that are not part of permitted ongoing connections. Cisco describes it this way in a white paper:

> "Each time a TCP connection is established from an inside host accessing the Internet through the PIX Firewall, the information about the connection is logged in a stateful session flow table. The table contains the source and destination addresses, port numbers, TCP sequencing information, and additional flags for each TCP connection associated with that particular host. This information creates a connection object in the PIX Firewall. Thereafter, inbound packets are compared against session flows in the connection table and are permitted through the Cisco PIX Firewall only if an appropriate connection exists to validate their passage. This connection object is temporarily set up until the connection has been terminated." [4-2]

From this model it is clear that even though the PIX is not a participant in these connections (it is neither the client nor the server) it must nevertheless make changes in its connection table based on the content of the traffic flows. For example, it must eliminate entries from its connection table when it sees a valid RST request. A RST request should only be considered valid if two conditions

are met: 1) its two address/port pairs match an existing connection 2) it has the sequence number expected next by the receiving side.

The PIX software up through version 5.1(1) checks only the first of these conditions, leaving it's connection table open to abuse by forged TCP reset packets. Cisco describes this vulnerability as follows:

> "The Cisco Secure PIX Firewall cannot distinguish between a forged TCP Reset (RST) packet and a genuine TCP RST packet. Any TCP/IP connection established through the Cisco Secure PIX Firewall can be terminated by a third party from the untrusted network if the connection can be uniquely determined. This vulnerability is independent of configuration. There is no workaround." [4-3]

So far so good, but the number of possible combinations of source and destination address/port pairs makes a brute force approach impractical. The possibilities will have to be reduced dramatically in order for an attack to make sense. The best idea I can come up with is to target a specific external TCP-based service that is critical to the organization behind Jeremy's firewall and deny service to it. An amusing choice might be Cisco's FTP site, thus preventing Jeremy from downloading the upgrade he needs to fix the problem.

Once we know what we want to deny service to, the only thing left to learn is the public address or addresses the PIX is using for NAT. (I confess I am making the assumption here that the PIX checks incoming TCP packets against the connection table before checking them against the translation table. I couldn't find documentation to confirm or deny this, but that is the pattern that IOS would have if an ingress ACL with the 'established' keyword were in use. It would also follow the general design principle of the PIX of optimizing the processing order at every point in order to improve throughput.) Since this information is visible from any destination an internal user might browse to, it should be possible to discover the NAT addresses by watching the traffic on a website we expect Jeremy's users to visit. Maybe we lure them in with an email advertising free beer at a site we have set up just for this purpose. But I digress…

Forged TCP reset packets ("sourced" from Cisco's ftp site, for example) can then be sent to the NAT addresses in waves across the entire range of high port numbers that could possibly be in use as source ports. One tool we could use to accomplish this is Salvatore Sanfilippo's hping. Hping can be used to send TCP packets with any combination of flag bits set, and its use could be scripted into the particular patterns we will need. [4-4]

4.3. A Distributed Denial of Service Attack

Jeremy's border router ingress ACL does not include any anti-spoofing rules, and his PIX configuration does not specify a non-zero embryonic connection limit for the 'static' commands, so the hosts on the external services network (DNS, Mail, Web, etc.) are open to some very effective denial of service attacks. [4-5] For example, the DNS server could be flooded with TCP SYN requests that would appear to be sourced from the Mail server. This would take out two services

directly (Mail and DNS) and the other two indirectly (Web and FTP, since Internet users will not be able to resolve to find them).

A good tool to use for this attack is TFN (Tribal Flood Network). In his analytical paper on the subject, David Dittrich describes it this way:

> "TFN is made up of client and daemon programs, which implement a distributed network denial of service tool capable of waging ICMP flood, SYN flood, UDP flood, and Smurf style attacks, as well as providing an "on demand" root shell bound to a TCP port." [4-6]

The best measure that could be taken to prevent this attack is setting a non-zero embryonic connection limit on the static translations in the PIX (see the command reference for the 'static' command for syntax). Normal traffic patterns on the service network should be analyzed so that the limit can be set just high enough to allow for the maximum normal traffic load. Rate limiting might also be configurable on the device labeled "External Server Router" in the diagram.

4.4. Compromise of an Internal Host

The hosts most exposed in Jeremy's design are the ones on the external services network. He doesn't specify what web software he is running so let's assume that it's Microsoft's IIS 5 on top of Windows 2000. If Jeremy has not yet patched this server to address the new buffer overflow vulnerability announced by CERT just a few days ago we are going to take full control of it! The CERT advisory says:

> "Windows 2000 includes support for the Internet Printing Protocol (IPP) via an ISAPI extension. According to Microsoft, this extension is installed by default on all Windows 2000 systems, but it is only accessible through IIS 5.0. The IPP extension contains a buffer overflow that could be used by an attacker to execute arbitrary code in the Local System security context, essentially giving the attacker compete control of the system." [4-7]

Microsoft's description of the problem makes it clear that this attack is not the sort that can be stopped by perimeter policies at all.

> "The attacker could exploit the vulnerability against any server with which she could conduct a web session. No other services would need to be available, and only port 80 (HTTP) or 443 (HTTPS) would need to be open. Clearly, this is a very serious vulnerability, and Microsoft strongly recommends that all IIS 5.0 administrators install the patch immediately." [4-8]

I think this makes it a great example to include here, because it illustrates that network security encompasses more than perimeter defense. In fact, from what I have seen this pattern of "in-band"

attacks is becoming more common. Here are more details about this specific attack quoted from the original announcement by eEye Digital Security:

> "It turns out the latest development code of Retina was able to find a buffer overflow within the .printer ISAPI filter C:\WINNT\System32\msw3prt.dll) which provides Windows 2000 with support for the Internet Printing Protocol (IPP) which allows for the web based control of various aspects of networked printers.
>
> The vulnerability arises when a buffer of aprox. 420 bytes is sent within the HTTP Host: header for a .printer ISAPI request.
>
> Example:
> GET /NULL.printer HTTP/1.0
> Host: [buffer]
>
> Where [buffer] is aprox. 420 characters.
>
> At this point an attacker has sucessfully caused a buffer overflow within IIS and has overwritten EIP. Now normally the web server would stop responding once you have "buffer overflowed" it. However, Windows 2000 will automatically restart the web server if it notices that the web server has crashed. While the feature is nice to help create a longer period of "up time" it is actually a feature that makes it easier for remote attacks to execute code against Windows 2000 IIS 5.0 web servers.
>
> As we stated earlier our overflow is able to overwrite the EIP register with whatever we want. That basically means we can overwrite EIP with a location in memory that jumps to our "exploit" code, in memory, and then executes our code with SYSTEM level access." [4-9]

This attack serves as a great gateway attack to other hosts on Jeremy's network as well. With a privileged command prompt available on the web server we may now have access to the other external servers or even to privileged e-commerce data.

Endnotes

[1-1] http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/tech/nat_wp.htm

[1-2] An additional interoperability tip: I have seen BlackICE Defender block the startup of SoftPK if both are starting automatically. The answer is to start Defender in a disabled state and include a batch file in the Startup folder that sends a single ping to the secure network (causing SoftPK to set up the IPSec SAs) and then calls 'blackd' to enable Defender. Also see http://www.soft-pk.com/

[1-3] http://www.cisco.com/warp/customer/707/overload_private.html

[1-4] http://www.content-security.com/home.asp

[2-1] all of the router hardening suggestions came from Spitzner, Advanced Perimeter Protection and Defense in Depth, Jan. 30 2001 pp.50-58.

[2-2] http://www.cert.org/tech_tips/packet_filtering.html

[3-1] http://www.eeye.com/html/Databases/Software/nmapnt.html

[4-1] Stevens, Richard. TCP/IP Illustrated Vol. 1 – The Protocols. Pp. 229-248 cover the whole process. pp 246-248 talk about the RST abort method in particular.

[4-2] http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/tech/nat_wp.htm

[4-3] http://www.cisco.com/warp/public/707/pixtcpreset-pub.shtml

[4-4] http://security-archive.merton.ox.ac.uk/bugtraq-199812/0023.html

[4-5] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v50/config/commands.htm#xtocid459365

[4-6] http://staff.washington.edu/dittrich/misc/tfn.analysis

[4-7] http://www.cert.org/advisories/CA-2001-10.html

[4-8] http://www.microsoft.com/technet/security/bulletin/MS01-023.asp

[4-9] http://marc.theaimsgroup.com/?l=bugtraq&m=98874912915948&w=2

General References

Brenton, Chris. VPNs and Remote Access. 1-31-01. Textbook for SANS New Orleans 2001, Course 2.4.

Doraswamy, Naganand and Harkins, Dan. IPSec – The New Security Standard for the Internet, Intranets, and Virtual Private Networks. Prentice Hall PTR, 1999.

Held, Gil and Hundley, Kent. Cisco Access Lists Field Guide. McGraw Hill Technical Expert Series, 1999.

Northcutt, Stephen. TCP/IP for Firewalls and Intrusion Detection. 1-28-01. Textbook for SANS New Orleans 2001, Course 2-1.

Spitzner, Lance. Firewalls 101: Perimeter Protection with Firewalls. 1-29-01. Textbook for SANS New Orleans 2001, Course 2.2.

Spitzner, Lance. Advance Perimeter Protection and Defense in Depth. 1-30-01. Textbook for SANS New Orleans 2001, Course 2.3.

Spitzner, Lance. Auditing Firewalls. 2-1-01. Textbook for SANS New Orleans 2001, Course 7.5 PM.

Stevens, W. Richard. TCP/IP Illustrated: the Protocols. Addison-Wesley Professional Computing Series, 1994.

SANS GCFW Practicals: Bob Hockensmith, Chris Olson, Larry Coons, Gavin Vallance, Elton Wright.

Tripod, Mark. Cisco Router Configuration and Troubleshooting – Second Edition. New Riders Publishing, 2000.