Global Information Assurance Certification Paper

GCFW Practical Assignment
Firewalls, Perimeter Protection, and VPNs

Version 1.9

Mike Powell
06/27/2003

1

1 Security Architecture

1.1 Overview

GIAC Enterprises is a newly formed company with a bold business plan. That plan is to sell wholesale fortune cookie fortunes over the Internet. Business operations will be very streamlined. A small network of suppliers will sell fortunes to GIAC Enterprises. Partner companies will then translate those fortunes into multiple foreign languages. Finally, either GIAC Enterprises or one of their partners will sell the fortunes to customers over the Internet.

GIAC Enterprises' network must be able to support secured business operations with all of the groups mentioned above: suppliers, partners, and customers. It also must support GIAC Enterprises' employees. These employees will be located both internally, at the offices of GIAC Enterprises, and externally, as some of the sales force will work remotely as they travel to customers' and partners' locations.

Fortunately, GIAC Enterprises is a newly formed company. They are not trying to migrate an existing network into a system that will support their desired functionality and security. They are starting from square one. GIAC Enterprises has a number of business needs, but we are free to meet these needs in any way that we see fit, within the guidelines below.

1.2 Business Requirements

Whatever the final design for GIAC Enterprises, there are certain business functions that need to be supported. Certain groups will require access that allows them to perform certain functions.

- Customers must be able to buy fortunes
- Suppliers must be able to transfer fortunes to GIAC Enterprises
- Partners must be able to download fortunes and upload translations, along with downloading fortunes in bulk for resale to customers.
- Remote sales employees must be able to access fortune database to place orders for customers and download fortunes.

All of this access must take place in the most secure manner possible.

From these business requirements, we refined the following detailed business flow.

Customers
Customers will communicate with GIAC Enterprises by using a web-based application. Each customer will have a dedicated account. Customers must contact GIAC Enterprises to set up an account prior to purchasing and downloading fortunes. A strong password policy will be enforced for all customer account. Customers must log on with their unique username and password in order to purchase and download fortunes. Purchase information is transferred to the back-end database server through use of a database proxy. This proxy provides a second layer of data validation in addition to the web application. The proxy server also only exposes the functions that are needed for communicating with customers, suppliers, and partners. Once the purchase data is stored, fortunes are available for download to the customer. The database is configured so that the web server can only write to tables and fields needed to record purchase information, and can only read from tables and fields required to transfer fortunes to customers.

Suppliers
GIAC Enterprises has several suppliers that supply it with fortunes. It was determined that having suppliers access the fortune database using a web interface might be awkward and slow. Because of this, and because GIAC Enterprises has a very strong relationship with their suppliers, we have chosen to allow suppliers to connect to GIAC Enterprises using a VPN connection. Suppliers will then be able to use a custom application to record fortunes into the fortune database. This custom application connects to the database by IP address so does not need DNS resolution. Each supplier has a unique username and password to access the fortune database, and strong password policies are enforced. All supplier communication is also done through the database proxy. Suppliers only have write access to the tables that they need to be able to record their fortunes. They are unable to read fortunes from the database.

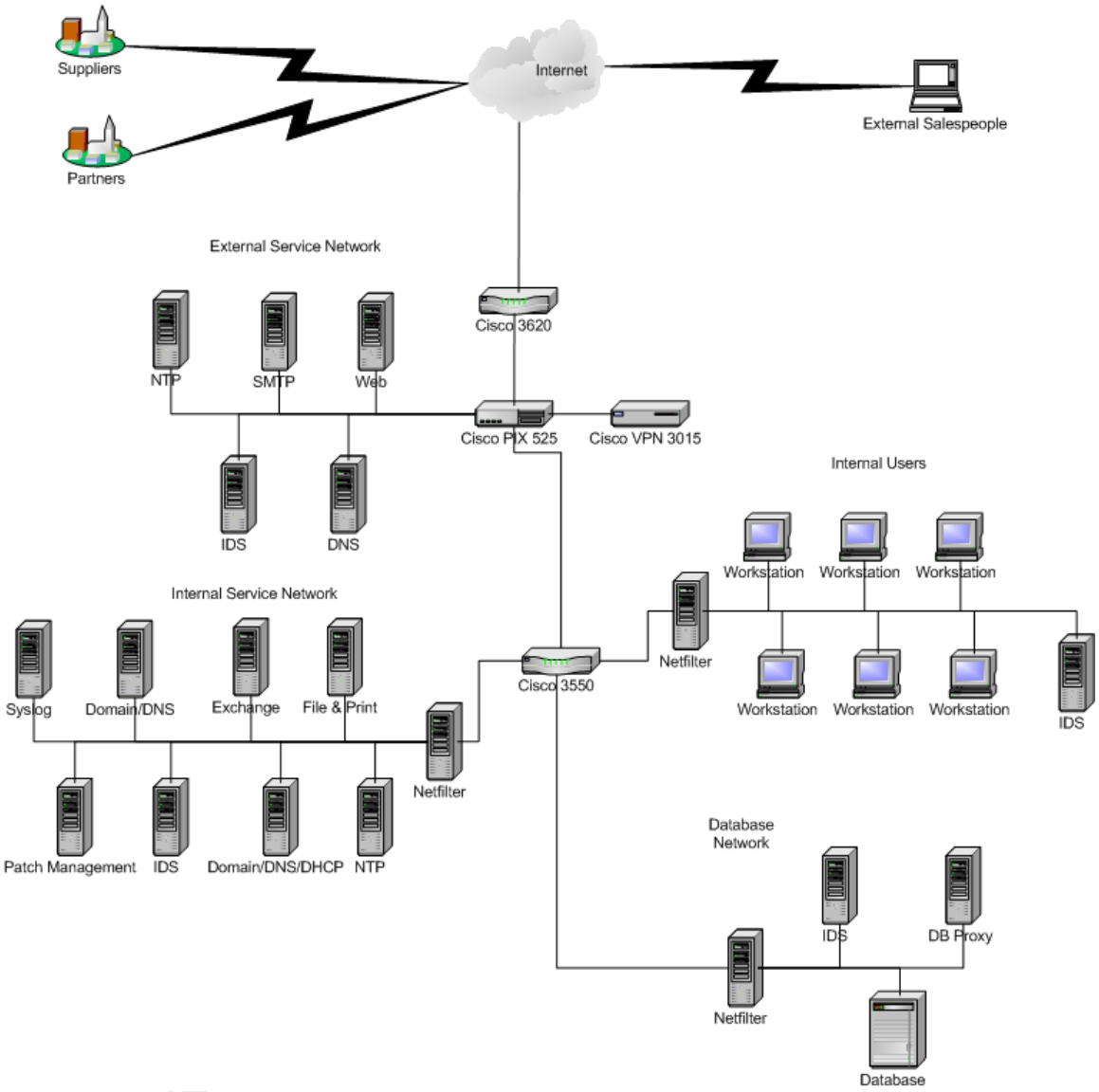
Partners
GIAC Enterprises has the strongest relationship of all with its partners. GIAC Enterprises' partners perform a number of functions to help GIAC Enterprises. Partners first download fortunes, translate them into other languages, and store the new translations in the database. They also download large volumes of fortunes to resell to their own customers. Like suppliers, partners will access the GIAC Enterprises fortune database over the VPN connection. Each partner has a unique username and password to access the fortune database. Using a custom application, they download fortunes for translation, upload fortune translations, and download fortunes to sell through the database proxy. This custom application connects to the database by IP address so does not need DNS resolution. Partners only have read and/or write access the tables and fields needed to perform their tasks.

Remote Sales Employees

In order to generate a higher volume of sales, GIAC Enterprises has a number of traveling salespeople. These salespeople must be able to access the GIAC Fortune database to report on sales volumes, customer and partner statistics, etc. To do this, they will first access the Internet by dialing in to GIAC Enterprises' national ISP dialup pool. Once connected, they use a VPN client to connect to GIAC Enterprises. They are now able to use a custom application to access the data needed for their reporting through the database proxy. This custom application connects to the database by IP address so does not need DNS resolution. Each salesperson has a unique username and password to access the database. Salespeople only have read permission to the tables and fields required for their reporting needs. Salespeople use email accounts provided by they ISP, so they do not need to access GIAC Enterprises' internal mail server over the VPN.

1.3 Network Design

Here is the final network design:



1.4 IP Addressing Scheme

Note that we have used the reserved address range of 70.0.0.0 – 77.0.0.0 (http://www.iana.org/assignments/ipv4-address-space) for GIAC Enterprises and its customers, suppliers, partners, and service providers.

1.4.1 Internal Addresses

Internal Networks

PIX to Internal Router	10.0.1.0/24
Service Firewall to Internal Router	10.0.2.0/24
Database Firewall to Internal Router	10.0.3.0/24
Workstation Firewall to Internal Router	10.0.4.0/24
Internal Service Network	10.0.5.0/24
Database Network	10.0.6.0/24
Workstation Network	10.0.7.0/24
External Service Network	10.1.0.0/24
VPN Network	10.2.0.0/24

Host Addresses

External Service Network

Cisco PIX External Service Interface	10.1.0.1
NTP	10.1.0.2
SMTP	10.1.0.3
Web	10.1.0.4
DNS	10.1.0.5
IDS	10.1.0.6

Internal Service Segment

Netfilter Interface	10.0.5.1
Domain/DNS/DHCP	10.0.5.2
Domain/DNS	10.0.5.3
Exchange	10.0.5.4
File & Print	10.0.5.5
Patch Management	10.0.5.6
NTP	10.0.5.7
Syslog	10.0.5.8
IDS	10.0.5.9

Database Segment

Netfilter Interface	10.0.6.1

Database Proxy	10.0.6.2
IDS	10.0.6.3
Database Server	10.0.6.4

Workstation Segment

Netfilter Interface	10.0.7.1
IDS	10.0.7.2
Workstations	10.0.7.x

VPN Segment

Cisco PIX VPN Interface	10.2.0.1
VPN Device External (Public)	10.2.0.2
VPN Device Internal (Private)	10.2.0.3

1.4.2 External Addresses

Addresses listed are static NAT translations for each device.

SMTP	70.0.0.3
Web	70.0.0.4
DNS	70.0.0.5
VPN	70.0.0.6

1.4.3 Other

Networking Devices

Cisco Layer 3 Switch		
	To PIX	10.0.1.2
	To Internal Service Net	10.0.2.1
	To Database Net	10.0.3.1
	To Workstation Net	10.0.4.1
Cisco PIX		
	Internet	70.0.0.2
	Intranet	10.0.1.1
	External Service	10.1.0.1
	VPN	10.2.0.1
Cisco 3620		
	Internal	70.0.0.1
	External	76.0.0.2
Internal Service Netfilter		
	To Service Net	10.0.5.1
	To Layer 3 Switch	10.0.2.1

Database Netfilter		
	To Database Net	10.0.6.1
	To Layer 3 Switch	10.0.3.1
Workstation Netfilter		
	To Workstation Net	10.0.7.1
	To Layer 3 Switch	10.0.4.1

Supplier & Partner Networks

Partner 1	71.0.0.0/24
Partner 2	72.0.0.0/24
Supplier 1	73.0.0.0/24
Supplier 2	74.0.0.0/24
ISP dialup provider	75.0.0.0/24

1.5 Technical Specifications

This section is comprised of the functionality and technical specifications of each device on the network. All servers are Dell model 1650 with 1.4 GHz Pentium III processors and 128 MB of RAM running Red Hat Linux 9.0 unless otherwise noted. Also, all hosts offering network services have been hardened, stripped of unneeded services and programs. We have chosen to use the benchmarks provided by the Center for Internet Security at http://www.cisecurity.org/ as guides to securing our systems. They have benchmarks for both Windows and Linux along with many other operating systems.

All non-user systems have been fingerprinted by Tripwire (http://www.tripwire.org/). Tripwire hashes are burned to CD whenever configuration changes are made. Comparing the current Tripwire signature to the stored signature is part of GIAC Enterprises' routine system audit procedure.

We will begin by working our way inwards, starting from the Internet.

1.5.1 Border Router (Cisco 3620)

I have chosen a Cisco 3620 as our border router. While there is a plethora of routers available on the market today, the Cisco 3620 stood out as an excellent choice for our network for a number of reasons.

First, Cisco routing equipment is as close to an industry standard as there is. There should not be any problems finding someone who is well qualified to administer this device.

Second, Cisco has a very comprehensive support network. Between service contracts, web-accessible information, and detailed manuals, there is no

shortage of information and assistance available for configuring and troubleshooting Cisco equipment.

Third, I feel that the 3620 will work very well with the Cisco PIX that we have specified for the external firewall. Any router could be installed to perform this function, but I feel that the similarities in configuration interfaces that are shared between Cisco's router and firewall products will make it easier for the administrators who come later to maintain these devices.

Fourth, the 3620 has a very wide range of features for controlling traffic. Even if these features aren't enough, the firewall feature set can be purchased from Cisco to enable even more filtering functionality. Since I plan to do some basic filtering with this router, these features played heavily in my decision.

Finally, the 3620 stood out as precisely the model that we were looking for. GIAC Enterprises is starting out with a single T1 (1.544 Mb/s) for Internet connectivity, but this router will easily scale up to handle the 3 Mb/s of traffic that GIAC plans on supporting in the future. This gives GIAC Enterprises room to grow as their business expands.

Since GIAC Enterprises relies entirely on the Internet for their business, we will be keeping a second, identically configured 3620 on hand in case of hardware failure.

1.5.2 External Firewall (PIX 525)

For our External Firewall, I have chosen to implement a pair of Cisco 525 routers configured in failover mode. As mentioned in the previous section, the similarities between this device and the Cisco 3260 will be a great benefit to maintaining a secure network.

Along with the 3620, the Cisco PIX family also has a very wide range of features for filtering and controlling traffic. It is beyond the scope of this document to outline all of the features of the Cisco PIX line, but I feel that its feature set compares favorably to any other firewall product on the market.

Another benefit of the Cisco 525 is that it is a dedicated firewall device. There are no extra services that might get turned on or installed as there might be when running a firewall on a Linux or Windows system. There is also a great deal less code since the firewall's software has to deal with a much more narrow range of functionality. Hopefully, less code means that there are fewer chances for bugs that might open up vulnerabilities in the firewall.

Finally, we are purchasing two PIX 525 devices and configuring them to run in failover mode. In this mode, two PIX devices are connected with a dedicated Ethernet cable. This allows each of the devices to communicate with each other

in order to keep a current version of the state table. If the primary device should fail, the secondary unit can take over without even losing the state of existing connections. The failure would be completely transparent, even to users who were currently transferring data.

1.5.3 VPN Device

Incoming VPN connections will terminate at a Cisco 3015 VPN Concentrator. One reason for this is the expected level of VPN traffic. While the PIX 525 is capable of accepting VPN connections, GIAC Enterprises expects to utilize the VPN quite heavily. Adding the additional load of high volume VPN traffic to the firewall may max out its available processing power, slowing all Internet access. Separating the firewall and VPN functions will allow better management and tracking of each function.

Another reason that we have chosen a standalone VPN device is to simplify the configuration and management of the firewall. With the VPN device located on a dedicated interface, it becomes very easy to filter the incoming VPN traffic by creating rules for the firewall interface connected to the VPN. Terminating the VPN connections at the PIX would require the rules on both the external and internal interfaces to become much larger and more complex.

1.5.4 External Web Server

Our external web server is running Apache 2.0.44 (http://www.apache.org/). This host provides web services for customers who download fortunes from GIAC Enterprises over the Internet.

Several security measures should be mentioned here. This server has a certificate from Verisign to assure our customers that they are indeed communicating with our server. Apache's httpd process has also been configured to run as the user nobody, not as the root user. We have also modified the default http banner so that the server type and version number are given as useless information instead. This is to make an attacker work harder to get our server type and version information. It may not be much, but every little bit counts.

Once connected to our server, GIAC's customers are prompted for authentication. This is done using the username and strong password that has been assigned to them by GIAC Enterprises. This information is compared to the username and encrypted hash of their password that is stored in the database. Once we have confirmed the identity of our customer, they can proceed to the download section of the website where they can download fortunes.

When the web application retrieves fortunes for customers to download, it is done with an account that has only been granted access to the tables that are

necessary for the server's role. Read access is configured only where data actually needs to be read, and write access is only given for the tables that are used to record what fortunes customers have downloaded.

Also, we have had the code for this web application audited by a third party outside of GIAC Enterprises for security flaws and use of best practices. We want to be doubly sure that we catch any possible flaws in the application, since abused web applications are one of the leading causes of security compromises on the Internet today.

1.5.5 External SMTP Server

The external SMTP server is running Qmail 1.03 (http://www.qmail.org/) to function as an SMTP relay for the internal mail server. This may come as a surprise if you expected to see Sendmail used here, since Sendmail is the most popular SMTP server on the planet and comes bundled with just about every Linux and BSD distribution and many of the commercial Unix products as well.

Unfortunately, there are a few problems with Sendmail. The first is that it is very old. It was originally coded before we had the level of security awareness that we have today. This has led to the second problem, which is that Sendmail has had some severe vulnerabilities discovered in just the past few months.

Qmail is a very different program. It was coded very recently compared to Sendmail. The authors were very aware of Internet security issues when they were designing it. You could say that Qmail started out secure, while Sendmail is being made secure. Qmail is also open source, so it has had a great deal of code review. On top of all of this, Qmail is a very small, fast program. It can handle a higher volume of email with less hardware than Sendmail can.

We have taken two steps to keep as much information as possible from attackers. First, we have changed the default SMTP banner to read simply "220 ESMTP". Second, we have implemented Bruce Guenter's Qmail-queue (http://www.untroubled.org/qmail-qfilter/) to strip internal headers from outbound email before being relayed. Both of these actions help prevent information leakage about what type of mail servers we use (both internal and external), or if we even have an internal server, etc.

The reason that we have implemented this SMTP server is it will relay inbound email to the internal mail server, and outbound email to the Internet. It functions as an intermediary system so that the internal mail server is not exposed to the Internet. This is especially important when the internal mail server provides a great deal more functionality than simple mail queuing, such as our internal Exchange server. Exchange provides some wonderful functionality, but I wouldn't want to directly expose it to the Internet.

1.5.6 External DNS Server

ISC's BIND 9.2.2 (http://www.isc.org/products/BIND/) has been chosen to provide DNS services on the external service network. Since we are running a split-DNS architecture, BIND will be providing non-recursive lookups to external queries for giacenterprises.com along with recursive lookups for hosts on the external service network.

There are three measures that we are taking with this server in particular in order to enhance its security.

First, we have limited zone transfers so that only our ISP's DNS system may perform a zone transfer. The only other host that should normally be allowed to perform a zone transfer from this system would be an internal DNS server. Since our internal DNS will be Active Directory-integrated Windows DNS, we will not be doing zone transfers to our internal DNS server.

Second, we have configured BIND to return useless data in response to a version.bind request. It may not be much, but we are always happy to provide a little confusion to those who may be performing reconnaissance on our network.

Finally, BIND will be running under a specially created user account in a "chroot jail". This prevents BIND from accessing any files outside of the directory that we have given the BIND account access to. If an attacker manages to break in to our server via BIND, they will be limited to accessing only the files that the BIND account has access to.

We have also worked with our ISP to ensure that their DNS server does not allow Internet users to perform zone transfers of our DNS zone.

1.5.7 External NTP Server

To provide synchronized time to our network, we will be installing a Præcis Cntp server from End Run Technologies (http://www.endruntechnologies.com/ntp-server.htm). This device will allow GIAC enterprises to have their own stratum 1 NTP server.

The primary problem with most stratum 1 NTP servers that are commercially available is that they are based on the GPS system. This requires that an antenna be mounted external to the building. The Præcis Cntp server operates off of the cellular CDMA system. This signal will penetrate inside most buildings, allowing the server to operate with only an interior antenna.

Using this type of NTP server completely removes the possibility of an attacker somehow causing the NTP server to synchronize incorrectly through some type of network based attack. The only way to attack the time synchronization of this

server would be to either provide a rouge CDMA signal or hijack the CDMA system. While it may not be impossible to perform either of these attacks, it would likely require a very different skill set than what the average hacker possesses. Also, setting up a rouge signal would require the attacker or an accomplice to be physically present at the targeted location.

Additionally, since the CDMA-based NTP server does not need to access any external network resources to synchronize its clock, we will have one less hole in our external firewall.

If these advantages weren't enough, running a local NTP server that does not synchronize over the Internet also eliminates the variance factor that comes with packets transmitted across Internet. This should allow the Præcis Cntp server to be more accurate than an NTP server that synchronizes to an Internet source.

While the Præcis Cntp server has many additional services available (Telnet, FTP, etc.), all of these will be disabled following GAIC Enterprises' normal hardening procedures. The Præcis Cntp server runs a Linux kernel, so it should be very familiar to our administrators, and they should have no problems configuring it and locking it down.

1.5.8 Network Intrusion Detection System (Multiple)

Our NIDS system of choice is Snort. The fine folks at http://www.snort.org/ have just recently released version 2.0 of their excellent NIDS system. When we looked at the variety of software and hardware that are available to perform this crucial function, a number of advantages led us to choose Snort.

One advantage is that Snort has been battle-tested. Through our experience and the collective experience of a vast number of other users, Snort has proven that it can get the job done. Even though it is available free of charge, its robust design allows it to handle traffic volumes similar to that of many commercially available products.

Another advantage is its flexible rule generation language. This flexible configuration mechanism will allow us to choose exactly what traffic is important for us to monitor. Instead of being forced to simply check boxes next to names of known attacks, we can create our own rules to look for any type of traffic that we choose, including new attacks that a vendor may not have released a signature for, or traffic that might be benign on other networks but could be a warning sign on ours.

You will notice that we actually have quite a few NIDS machines spread around our network. Their locations have been specifically chosen in order to give us the most complete and detailed picture of malicious or abnormal traffic that might traverse any part of our network.

On the outermost front, we do not have a NIDS system outside of our firewall. If we were to operate a NIDS system there, we would risk overwhelming ourselves with alerts and potentially miss a successful attack due to the large volume of alerts we would be dealing with. Besides, most attacks should be dropped by the firewall and logged to the syslog server anyway. Any attacks that do get past the firewall will then be picked up by the internal NIDS systems. Placing a NIDS system outside of the external firewall only buys us extra work and two alarms for every attack instead of one.

Internally, we have placed a NIDS system on every network segment. This should alert us to any attack, whether it originates internally or comes through the firewall from the Internet.

1.5.9 Core Switch

Our core network connections will be to a Cisco 3550-12T. We will utilize the switch's layer 3 static routing features to connect our multiple internal subnets. This switch has 10 gigabit-capable Ethernet ports. This along with the gigabit network ports in the Dell servers running our internal Netfilter firewalls, will allow all internal subnets to be linked at very high speed.

For security, all unused ports will be shut down. We will also use the port-level security of the switch to allow connection to the port by only one media access control (MAC) address.

1.5.10 LAN Switches

Each subnet will be connected using Cisco 2950 12-port switches (2950G 24-port for workstation subnet). Each switch will use a spanning port to mirror all traffic that is passed over the switch to the NIDS system on that subnet. As with the core switch, all unused ports will be shut down. Active ports will be configured to only connect with the specified MAC address.

1.5.11 Internal Firewalls (Multiple)

Netfilter and IPTables version 1.2.8 (http://www.netfilter.org) will be providing our internal network protection. Even though all of our Red Hat systems already have Netfilter and IPTables, we have updated them with the latest release.

While we hope that our external firewall will block all attacks that are launched against GIAC Enterprises, we have to plan for the worst. The principle of "defense in depth" is that if an attacker gets past your first line of defense, hopefully you can stop them with your second line (or third, or fourth…). In this case, our internal firewalls are providing an additional line of defense for our internal network. They also provide internal traffic control to help protect from internal threats.

When choosing what type of firewall to use for this task, we immediately ruled out the Cisco PIX series. If an attacker could get past our external PIX, additional internal PIX devices would probably not pose much of an additional challenge. Our internal firewalls should be entirely different so that the internal firewalls are not vulnerable to the same exploits or weaknesses as the external firewall.

Looking at the other firewall systems available, we chose to go with Linux's built-in packet filter. The advantages are pretty obvious: It is simple, free, and has a feature list and a vulnerability record that compare well against any commercial firewall product.

1.5.12 Domain/DNS

Our two Domain/DNS server are running Windows 2000. These two systems are key to keeping the workstations and Windows servers configured securely. Active Directory is utilized to force secure configurations onto workstation systems and Windows servers.

We have also configured a static DNS entry for our external web server. This is the only server on our external service network that internal users need access to. This will prevent them from accessing the external DNS server to do lookups and is one more door in our firewall we can close.

1.5.13 Exchange

Microsoft's Exchange Server 2000 provides internal email. This system allows for internal communication among users. Email to or from external systems is relayed to the SMTP server on the external service network. All mail either inbound, outbound, or internal to GIAC Enterprises is scanned using McAffee Groupshield for Exchange (http://www.networkassociates.com/us/products/mcafee/antivirus/email/gs_exchange2000.htm).

1.5.14 Patch Management

We have chosen to use Patchlink Update from Patchlink, Inc. (http://www.patchlink.com/products/emanagement_services/patchlink_update.html). This system will regularly pull down new Microsoft patches and push them out to all Windows workstation machines. This will keep our Windows workstations as up-to-date as possible.

Server patches and updates will be individually tested and applied to all servers, regardless of operating system.

1.5.15 NTP

The latest release of XNTP (http://www.ntp.org/), currently 4.1.1, will provide time synchronization for our internal network. It will obtain time synchronization from the NTP server on our external service network.

1.5.16 Syslog

Our syslog server is running syslog-ng. The most recent version is 1.5.25 and can be found at http://www.balabit.com/products/syslog_ng/. This host will also run swatch to monitor the log files and alert systems personnel when needed.

We have chosen syslog-ng based on its advanced features compared to the native syslog service available with Red Hat Linux. Syslog-ng allows traffic from various hosts to have unique filtering applied. This will allow us to be more selective in the types of traffic that we alert and report on.

Swatch (http://swatch.sourceforge.net/) has been configured with a relatively tight alerting policy. If we find that systems personnel are overwhelmed with alerts generated by legitimate traffic, we will adapt our alerting policy based on our knowledge of normal traffic found on the GIAC Enterprises network.

This host also has specialized hardware. This system uses two RAID 1 (mirrored) disk arrays. One is for the operating system and one is for the syslog logs. This will prevent any data loss due to hard drive failure. This system also has the log files burned to CD and cleared as needed to provide a long-term record of traffic and activity on the GIAC Enterprises network.

1.5.17 Database Proxy

When the external web server or partners and suppliers access data from our database, they do not actually communicate with the database. They communicate with a custom application that functions as a database proxy. The database subnet firewall blocks all traffic except for traffic bound for the database proxy.

This proxy provides a second layer of data validation in addition to that of the web or custom application that is used to access the database. The proxy also only exposes the functions of the database that are needed by the end users of the database. If our applications do not need a “create table” function, the database proxy will not pass this function through. In this way we have an additional layer of protection from attacks such as SQL injection.

1.5.18 Database

The database server is a Dell PowerEdge 4600. It has a RAID-1 array and a RAID-5 array consisting of five drives plus one hot-spare. This server has a dedicated Sony AIT 3 backup drive. This prevents any of the fortune database data from ever traveling outside of the database subnet for backup purposes.

The database software used is Solaris 9i.

1.5.19 Workstations

User workstations are Dell Optiplex PCs or Dell Lattitude laptops running Windows 2000 Professional. All workstations use McAffee Virusscan Enterprise 7.0.(http://www.networkassociates.com/us/products/mcafee/antivirus/email/vs.htm) for virus protection. Virus signature updates are controlled by McAffee's ePolicy Orchestrator (http://www.networkassociates.com/us/products/mcafee/antivirus/fileserver/epo.htm). Remote users that do not enjoy the safety of our corporate network are equipped with McAffee Desktop Firewall 8.0 (http://www.networkassociates.com/us/products/mcafee/antivirus/desktop/desktop_firewall.htm).

1.5.20 Backup (not shown)

All hosts requiring backup on the internal service network will be backed up onto an HP direct attached fiber channel tape backup (http://h18006.www1.hp.com/storage/entrystorage.html). The domain controllers, Exchange server, and File & Print server will all have fiber channel cards connecting them to the fiber channel switch. This switch will allow communication with the fiber channel backup system.

2 Security Policy and Tutorial

2.1 Overview

In planning the security between the Internet and GIAC Enterprises, we have come up with a design that relies on three Cisco devices. The device we address first will be the border router, a Cisco 3620 router. next we will detail the configuration of our primary firewall, a Cisco PIX 525. Finally we will discuss the VPN device, a Cisco 3015 VPN concentrator.

The Cisco 3620 will perform preliminary filtering. It will filter some basic types of traffic that we never want to see coming into or going out of our network. This includes filtering non-routable addresses, NetBIOS ports, etc. We will leave the heavy lifting of the more complex filtering to the PIX since its stateful inspection is a more powerful tool than what is available on the 3620.

The Cisco PIX 525 will be our primary filtering device. It sits between the Internet, external service, internal, and VPN networks. It is where the buck stops when it comes to traffic filtering. All filtering on the PIX will be done using a "deny all except…" policy. All traffic will be dropped unless it is specifically allowed, and the only traffic allowed will be traffic that we feel is needed for continuing business operations and does not pose a security threat.

Finally, the VPN device will provide an endpoint for encrypted traffic to and from our suppliers and partners. Having the VPN device connect to the PIX on its own interface allows us to terminate the VPN inside of GIAC Enterprises' protected network, but still examine the unencrypted traffic before it travels to a network segment that contains GIAC Enterprises servers.

2.2 Cisco 3620 Router Configuration Tutorial

Note: There is a vast array of resources on the Internet for configuring Cisco IOS devices. Two that I have used here are the NSA's "Router Security Configuration Guide" at http://www.nsa.gov/snac/cisco/guides/cis-2.pdf and Cisco's IOS documentation at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/index.htm.

When first logging in to an unconfigured Cisco device, the command prompt will be `>`. This indicates unprivileged mode. Unprivileged mode has a small number of available functions that allow the user to view some settings.

Using the `enable` command changes to privileged mode. Privileged mode is indicated by a command prompt of `#`. This brings a greater set of commands that actually allow some configuration of the device. "Disable" will return to unprivileged mode.

`configure terminal` is used to enter the highest level of global access on the Cisco 3620, configuration mode. The command prompt changes to `(configure) #`, indicating that the system is currently in configuration mode. This mode has access to all global commands on the Cisco 3620.

Finally, interface configuration mode can be entered by typing interface <hardware_id>. Interface configuration mode is used to set interface-specific settings such as IP address.

The commands `exit` or `quit` are used to log out of the current configuration mode and to end a terminal session or terminate the connection for a telnet or SSH session.

Note: all routers and firewalls should fully configured before they are connected to the live network. All configuration is performed using the terminal connection, and initial testing is done in a test lab environment.

2.2.1 Global Configuration

The first configuration changes that need to be made to the router will take place in configuration mode.

First we need to give our router a name. It is unlikely that an attacker will ever discover the name of our router. Just in case, we will give our router a plain name that does not invite attention.

```
hostname pc463
```

Next we will want to set a password for privileged mode. We also want to enable this password to be stored with the best encryption the router has.

```
service password-encryption
enable secret <password>
```

Since it will be important for us to know what kind of traffic the router is dropping or passing, we will configure the router to log to our syslog server. We will also set up a small buffer for local logging, which can be useful for troubleshooting. We have also configured warnings to be displayed on the console.

```
logging buffered 1600
logging console warnings
logging source eth0/0
logging <logserver>
```

Our final global configuration will be to set a warning banner. In many jurisdictions, someone who illegally accesses a system without a warning banner can avoid conviction.

```
banner login AUTHORIZED GIAC ENTERPRISES USERS ONLY
```

2.2.2 Disable Unneeded Functions

One method attackers have used to circumvent some types of network security was to use either loose or strict source routing. This is another functionality that was useful in the early days of the Internet, but is now mainly a nuisance. There is no reason that Internet hosts or internal users should send us packets with source routing.

```
no ip source-route
```

We have chosen to disable DNS name resolution on the router, since it should not ever need to do DNS lookups.

```
no ip domain-lookup
```

We do not allow directed broadcasts through our router.

```
no ip directed-broadcast
```

Another “stupid IP trick” is ICMP redirect. This allows a router to communicate that traffic to a certain host should be redirected through a router other than the default router. It is seldom used anymore, and should never come across the Internet.

```
no ip redirects
```

We will disable ICPM unreachable and ICMP netmask response since they can be used to disclose information about our network to an attacker.

```
no ip unreachables
no ip mask-reply
```

Our router will not be using proxy ARP.

```
no ip proxy-arp
```

Since administrators will only be logging in to the router through the terminal interface, we will disable the virtual terminals by setting up an access list that allows only console access

```
access list 5 deny any
```

```
line vty 0 4
access class 5 in
```

We will enable the use of “subnet zero” (e.g. 10.10.0.0/24).

```
ip subnet zero
```

2.2.3 Disable Unneeded Services

Our next task is to ensure that a number of services that the Cisco 3620 can run are disabled. Running a finger deamon or web server for a web-based administration interface are invitations for denial of service attacks and possible backdoors into the system. Remember, if you don’t absolutely need it, disable it.

Some of these services are already turned off by default, but we feel it is better to get into the habit of turning them off instead of relying on the defaults. One day, a firmware update might ship that changes the default, and using this method we will still be safe.

We do not need our router to provide IP addresses or other boot-time information to any hosts. Therefore, we will disable the bootp service.

```
no ip bootp server
```

We need the so-called “small services” like we need a hole in our head. Many of these services were used in the early days of the Internet, but there is absolutely no reason that they should even be included in a router’s firmware, much less enabled.

```
no service tcp-small-servers
no service udp-small-servers
```

The finger service was originally used to let users find out who was logged on to a system at any given time. This is information that would be very dangerous in the hands of an attacker.

```
no ip finger
```

Identd is another service that allows users to gain information about a system. We do not want to allow attackers to access this type of information

```
no ip identd
```

We are not using SNMP at GIAC Enterprises. It can be useful, but it has not been considered secure for some time.

```
no snmp-server
```

Cisco has included a web-based administration interface for the 3620 router. Again, we would rather deal with the command line than expose a service that gives an attacker one more door to try to break in to.

```
no ip http server
```

We are not running a large Cisco network that requires routers to be able to automatically discover each other and communicate routing information. Since this is the function of Cisco Discovery Protocol, we disable it.

```
no cdp run
```

We will now disable the router's ability to load its configuration from over the network.

```
no boot network
no service config
```

2.2.4 Interface Configuration

Now we will assign addresses to our interfaces. Still in configuration mode, we enter

```
interface ethernet 0
```

to configure the Ethernet (internal) interface. We set its IP address and subnet mask.

```
ip address 70.0.0.1 255.255.255.0
```

Exiting from the internal interface, we now configure the serial (external) interface.

```
quit
interface serial0
ip address 76.0.0.2 255.255.255.0
```

For the Internet (external) interface only, we will block NTP. While our router will send NTP requests to the NTP server on the external service network, it will never send or receive NTP traffic over the Internet.

```
ntp disable
```

We now return to global configuration mode.

```
quit
```

We must also set up the routing table so the router knows onto which interface to route packets with a given address. First, we set the default route. If the packet doesn't match any of the other entries we make, this rule will route it to the Internet. In this way, we only have to set up rules for our IP addresses, and anything else goes to the Internet. Unlike Access Control Lists, routing is performed by using the most restrictive rule first instead of the order in which they were entered. That is why it is common to enter the default route first.

```
ip route 0.0.0.0 0.0.0.0 76.0.0.2
```

Now we will create an entry to route packets destined for GIAC Enterprises' public IP range over the correct interface.

```
ip route 70.0.0.0 0.0.0.255 70.0.0.1
```

2.2.5 Access Control Lists (ACLs)

By default, a Cisco router will forward all traffic. To be able to filter traffic, an Access Control List, or ACL, must be applied to an interface. Once an ACL is applied to an interface, the behavior immediately changes from forward all to drop all. This is called the implicit deny rule. Also, only one ACL can be applied to a given interface in a given direction, e.g. inbound on the Internet interface.

Cisco routers have three types of ACLs. Standard, Extended, and Reflexive. A Standard ACL looks at only at source IP address. An Extended ACL can filter based on source IP and port, destination IP and port, protocol, TCP flags (SYN, ACK, RST, FIN, etc.) and ICMP type. Reflexive ACLs have all of the functionality of an Extended ACL, and additionally they can maintain a state table. The reason that Extended ACLs were added is that a Cisco router will not pass packets related to an existing connection like a stateful firewall.

Here is the syntax for both Standard and Extended ACLs:

Standard ACL

```
access-list number action source [wild card] | any
```

Options:
Number: 1-99
Action: permit or deny
Source: source IP
Wild Card: inverse of subnet mask (e.g. 0.0.0.255 specifies any host on same class C as source)
Any: matches to any address

Extended ACL

```
access-list number action protocol source [wild card] [src-
port] destination [wild card] [dest-port] [other-options]
```

Options:
Number: 100-199
Action: permit or deny
Protocol: protocol number or name
Source: source IP
Wild Card: inverse of subnet mask (e.g. 0.0.0.255 specifies any host on same class C as source)
Src-port: source port
Destination: destination IP
Wild Card: inverse of subnet mask (e.g. 0.0.0.255 specifies any host on same class C as source)
Dest-port: destination port

The naming conventions for each type of filter vary. Standard ACLs are numbered 1-99, Extended ACLs can range from 100-199, and Extended ACLs are only referred to by name only. Each ACL type also requires a different level of processing power. Standard ACLs use the least power, with Extendended ACLs requiring more power, and Reflexive ACLs taking the most processing power.

We want to minimize the load on our router so that we will not overload it. We do not need the router to duplicate the filtering of the PIX, only augment it. However, since we do need to do some filtering based on ports, we will be using Extended ACLs.

It is also important to note that we will be applying our ACLs in the inbound direction. This saves processing power and memory on the router. We will not waste resources to examine and route the packet, only to drop it on the way out. There are cases where outbound filters are needed, but our simple router configuration with only two interfaces does not have such needs.

2.2.6 ACL Input

2.2.6.1 Internet Interface ACL

The ACL for our Internet interface will be ACL 101. We are configuring all rules on our border router to log all dropped traffic to the syslog server. This means we will see a large number of “script kiddie” or simplistic attacks. We may disable this logging in the future if we find that it is not useful.

The first rules that we will add to our ACL will be to drop traffic that comes from non-routable ranges. Many attacks spoof these addresses as their source

addresses. Also, sometimes a NAT device will leak and allow privately addresses traffic onto the Internet. Since this traffic cannot be replied to at best and is an attack at worst, we will drop and log it.

We will start by dropping traffic with a source address in the private non-routable IP ranges (http://www.faqs.org/rfcs/rfc1918.html).

```
access-list 101 deny 10.0.0.0 0.255.255.255 log
access-list 101 deny 172.16.0.0 0.15.255.255 log
access-list 101 deny 192.168.0.0 0.0.255.255 log
```

Deny traffic with source address of loopback address.

```
access-list 101 deny 127.0.0.0 0.255.255.255 log
```

Drop packets that appear to come from an address in the multicast range

```
access-list 101 deny 224.0.0.0 15.255.255.255 log
```

There are a number if IP blocks that are theoretically routable, but have not been assigned by IANA (http://www.iana.org/assignments/ipv4-address-space). If we see traffic with these addresses in the source, we are seeing spoofed traffic. This list should be monitored for periodic changes.

```
access-list 101 deny 0.0.0.0 0.255.255.255 log
access-list 101 deny 1.0.0.0 0.255.255.255 log
access-list 101 deny 2.0.0.0 0.255.255.255 log
access-list 101 deny 5.0.0.0 0.255.255.255 log
…
access-list 101 deny 255.0.0.0 0.255.255.255 log
```

We should never see traffic coming into the Internet interface with GIAC Enterprises' own public IP addresses.

```
access-list 101 deny 100.0.0.0 0.255.255.255 log
```

The only types of ICMP traffic that we will allow into our network are source quench and packet too big messages. Almost every other type of ICMP traffic can by used by attackers in one way or another. It is important to place the allow rule before the deny rule because Cisco devices process rules in top-down order. If the deny rule was first, all ICMP traffic including source quench would match the deny rule and be dropped.

```
access-list 101 permit icmp any any source-quench
access-list 101 permit icmp any any packet-too-big
```

```
access-list 101 deny icmp any any
```

Now that we have blocked source addresses and ICMP types that we never want to enter our network, we will block ports that we never want to enter our network, even if they are from valid addresses. We will not bother to block every single port in existence, but rather the ports that are commonly probed for and correspond to services that we do not make available to the Internet. A great aid to compiling this list is at http://www.iana.org/assignments/port-numbers.

Service	**Protocol and Port**
telnet	TCP 23
SSH	TCP 22
FTP	TCP 21
rlogin	TCP 512-514
Portmap/rpcbind	TCP & UDP 111
NFS	TCP & UDP 2049
lockd	TCP& UDP 4045
NetBIOS	TCP & UDP 135
	TCP & UDP 137-139
Windows Directory	TCP & UDP 445
X Windows	TCP 6000-6255
LDAP	TCP & UDP 389
HTTP Proxy	TCP 8000, 8080, 8888
Small Services	TCP & UDP < 21
Time	TCP & UDP 37
TFTP	UDP 69
finger	TCP 79
NNTP	TCP 119
NTP	TCP 123
LPD	TCP 515
syslog	UDP 514
SNMP	TCP & UDP 161 & 162
BGP	TCP 179
SOCKS	TCP 1080
IRC	TCP & UDP 194 & 6667
MSN Messenger	TCP 1863
AOL Instant Messenger	TCP 5190
Back Oriface	TCO 31337& 31338
SQL Slammer block	UDP 1434
Sub-seven	TCP 27374

```
access-list 101 deny tcp any any range 1 23 log
```

```
access-list 101 deny udp any any range 1 23 log
access-list 101 deny tcp any any eq 37 log
access-list 101 deny udp any any eq 37 log
access-list 101 deny upd any any eq 69 log
access-list 101 deny tcp any any eq 79 log
access-list 101 deny tcp any any eq 111 log
access-list 101 deny udp any any eq 111 log
access-list 101 deny tcp any any eq 119 log
access-list 101 deny tcp any any eq 123 log
access-list 101 deny tcp any any eq 135 log
access-list 101 deny udp any any eq 135 log
access-list 101 deny tcp any any range 137 139 log
access-list 101 deny udp any any range 137 139 log
access-list 101 deny tcp any any range 161 162 log
access-list 101 deny udp any any range 161 162 log
access-list 101 deny tcp any any eq 179 log
access-list 101 deny tcp any any eq 194 log
access-list 101 deny tcp any any eq 389 log
access-list 101 deny udp any any eq 389 log
access-list 101 deny tcp any any eq 445 log
access-list 101 deny udp any any eq 445 log
access-list 101 deny tcp any any range 512 515 log
access-list 101 deny upd any any eq 514 log
access-list 101 deny tcp any any eq 1080 log
access-list 101 deny udp any any eq 1434 log
access-list 101 deny tcp any any eq 1863
access-list 101 deny tcp any any eq 2049 log
access-list 101 deny udp any any eq 2049 log
access-list 101 deny tcp any any eq 4045 log
access-list 101 deny udp any any eq 4045 log
access-list 101 deny tcp any any eq 5190 log
access-list 101 deny tcp any any range 6000 6255 log
access-list 101 deny tcp any any eq 6667 log
access-list 101 deny tcp any any eq 8000 log
access-list 101 deny tcp any any eq 8080 log
access-list 101 deny tcp any any eq 8888 log
access-list 101 deny tcp any any eq 27374 log
access-list 101 deny tcp any any range 31337 31338 log
```

Now we will add rules to allow approved traffic into the network. First we will allow traffic with the TCP ACK bit set to pass the router.

```
access-list 101 permit tcp any any established
```

It is important to note that this is very different from how a stateful firewall works. This will allow more traffic through than a stateful firewall. Fortunately, we are not relying on our border router to filter every single bad packet.

The first point of contact on our network for most Internet hosts will be our DNS server. We need to allow Internet clients to do DNS lookups for our external service hosts.

```
access-list 101 permit tcp any 100.0.0.12 eq 53
access-list 101 permit udp any 100.0.0.12 eq 53
```

Once an Internet host has performed a DNS query, they will start communicating with one of the servers on our external service network. We allow Internet hosts to speak SMTP to our SMTP server,

```
access-list 101 permit tcp any 100.0.0.14 eq 25
```

HTTP and HTTPS to our web server,

```
access-list 101 permit tcp any 100.0.0.10 eq 80
access-list 101 permit tcp any 100.0.0.10 eq 443
```

and IPsec to our VPN device. Since we will hopefully be adding partners and suppliers on a regular basis, we will allow any host through the router on the IPsec ports. We will do filtering for individual addresses on the PIX firewall. This is will ease configuration changes in the future.

```
access-list 101 permit udp any 100.0.0.20 eq 500
access-list 101 permit esp any 100.0.0.20
```

Finally, we drop all traffic that didn't match one of the above rules.

```
access-list 101 deny any any log
```

2.2.6.2 Intranet Interface ACL

We will start a new ACL for the inbound direction of our Intranet interface. Even though we have placed a restrictive filter on our Internet interface, that does not stop any traffic that is traveling from our internal network out to the Internet. While GIAC Enterprises' Security Policy allows users open access to the Internet, there are a few types of traffic that we absolutely must block for security reasons.

Drop any traffic generated or claiming to be generated by the internal interface.

```
access-list 111 deny ip host 100.0.0.2 log
```

ICMP time exceeded messages can be useful, especially to attackers. They are often used to map systems behind a firewall. We don't want to let this information out.

```
access-list 111 deny icmp any any time exceeded
```

We never want ICMP echo replies to go out to the Internet. We have already blocked incoming ICMP echo requests, but we would rather be safe than sorry

```
access-list 111 deny icmp any any echo-reply
```

ICMP host unreachable can also disclose useful information to an attacker.

```
access-list 111 deny icmp any any host-unreachable
```

Drop any traffic using a dangerous service and log it so we can track down the source.

TFTP
```
access-list 111 deny udp any any eq 69 log
```
NetBIOS
```
access-list 111 deny tcp any any eq 135 log
access-list 111 deny udp any any eq 135 log
access-list 111 deny tcp any any range 137 139 log
access-list 111 deny udp any any range 137 139 log
access-list 111 deny tcp any any eq 445 log
```
SNMP
```
access-list 111 deny udp any any range 161 162 log
```
syslog
```
access-list 111 deny upd any any eq 514
```

Now that we have blocked any dangerous traffic that we do not want allowed onto the Internet, we will allow out any other traffic as long as the source address is within GIAC Enterprises' public IP range.

```
access-list 111 permit ip 100.0.0.0 0.0.0.255 any
```

Drop any traffic that hasn't matched a rule so far.

```
access-list 111 deny any any log
```

2.2.7 Apply ACLs

The last ACL task is to apply our ACLs to the appropriate interface. We start with the Ethernet interface.

```
interface ethernet0
```

To apply the ACL to the interface, we enter

```
ip access-group 101 in
```

After typing

```
quit
```

to exit from interface configuration mode, the Ethernet interface is configured. Now we simply repeat the same steps for our external interface and ACL.

```
interface serial0
ip access-group 111 in
quit
```

The last step is to copy the running configuration into the router's memory to be used at the next startup.

copy running-config startup-config

Our Cisco 3620 is now configured!

2.3 Security Policy Listings

2.3.1 Cisco 3620

See section 2.2.1 for a complete description of each command and rule.

hostname pc463
service password-encryption
enable secret <password>
logging buffered 1600
logging console warnings
logging source eth0/0
logging <logserver>
banner login AUTHORIZED GIAC ENTERPRISES USERS ONLY
no ip source-route
no ip domain-lookup
no ip directed-broadcast
no ip redirects
no ip unreachables
no ip mask-reply
no ip proxy-arp
access list 5 deny any
line vty 0 4
access class 5 in
ip subnet zero

```
no ip bootp server
no service tcp-small-servers
no service udp-small-servers
no ip finger
no ip identd
no snmp-server
no ip http server
no cdp run
no boot network
no service config
interface ethernet 0
ip address 70.0.0.1 255.255.255.0
interface serial0
ip address 76.0.0.2 255.255.255.0
ntp disable
ip route 0.0.0.0 0.0.0.0 76.0.0.2
ip route 70.0.0.0 0.0.0.255 70.0.0.1
access-list 101 deny 10.0.0.0 0.255.255.255 log
access-list 101 deny 172.16.0.0 0.15.255.255 log
access-list 101 deny 192.168.0.0 0.0.255.255 log
access-list 101 deny 127.0.0.0 0.255.255.255 log
access-list 101 deny 224.0.0.0 15.255.255.255 log
access-list 101 deny 0.0.0.0 0.255.255.255 log
access-list 101 deny 1.0.0.0 0.255.255.255 log
access-list 101 deny 2.0.0.0 0.255.255.255 log
access-list 101 deny 5.0.0.0 0.255.255.255 log
…
access-list 101 deny 255.0.0.0 0.255.255.255 log
access-list 101 deny 100.0.0.0 0.255.255.255 log
access-list 101 permit icmp any any source-quench
access-list 101 permit icmp any any packet-too-big
access-list 101 deny icmp any any
access-list 101 deny tcp any any range 1 23 log
access-list 101 deny udp any any range 1 23 log
access-list 101 deny tcp any any eq 37 log
access-list 101 deny udp any any eq 37 log
access-list 101 deny upd any any eq 69 log
access-list 101 deny tcp any any eq 79 log
access-list 101 deny tcp any any eq 111 log
access-list 101 deny udp any any eq 111 log
access-list 101 deny tcp any any eq 119 log
access-list 101 deny tcp any any eq 123 log
access-list 101 deny tcp any any eq 135 log
access-list 101 deny udp any any eq 135 log
access-list 101 deny tcp any any range 137 139 log
access-list 101 deny udp any any range 137 139 log
```

```
access-list 101 deny tcp any any range 161 162 log
access-list 101 deny udp any any range 161 162 log
access-list 101 deny tcp any any eq 179 log
access-list 101 deny tcp any any eq 194 log
access-list 101 deny tcp any any eq 389 log
access-list 101 deny udp any any eq 389 log
access-list 101 deny tcp any any eq 445 log
access-list 101 deny udp any any eq 445 log
access-list 101 deny tcp any any range 512 515 log
access-list 101 deny upd any any eq 514 log
access-list 101 deny tcp any any eq 1080 log
access-list 101 deny udp any any eq 1434 log
access-list 101 deny tcp any any eq 1863
access-list 101 deny tcp any any eq 2049 log
access-list 101 deny udp any any eq 2049 log
access-list 101 deny tcp any any eq 4045 log
access-list 101 deny udp any any eq 4045 log
access-list 101 deny tcp any any eq 5190 log
access-list 101 deny tcp any any range 6000 6255 log
access-list 101 deny tcp any any eq 6667 log
access-list 101 deny tcp any any eq 8000 log
access-list 101 deny tcp any any eq 8080 log
access-list 101 deny tcp any any eq 8888 log
access-list 101 deny tcp any any eq 27374 log
access-list 101 deny tcp any any range 31337 31338 log
access-list 101 permit tcp any any established
access-list 101 permit tcp any 100.0.0.12 eq 53
access-list 101 permit udp any 100.0.0.12 eq 53
access-list 101 permit tcp any 100.0.0.14 eq 25
access-list 101 permit tcp any 100.0.0.10 eq 80
access-list 101 permit tcp any 100.0.0.10 eq 443
access-list 101 permit udp any 100.0.0.20 eq 500
access-list 101 permit esp any 100.0.0.20
access-list 101 deny any any log
access-list 111 deny ip host 100.0.0.2 log
access-list 111 deny icmp any any time exceeded
access-list 111 deny icmp any any echo-reply
access-list 111 deny icmp any any host-unreachable
access-list 111 deny udp any any eq 69 log
access-list 111 deny tcp any any eq 135 log
access-list 111 deny udp any any eq 135 log
access-list 111 deny tcp any any range 137 139 log
access-list 111 deny udp any any range 137 139 log
access-list 111 deny tcp any any eq 445 log
access-list 111 deny udp any any range 161 162 log
access-list 111 deny upd any any eq 514
```

```
access-list 111 permit ip 100.0.0.0 0.0.0.255 any
access-list 111 deny any any log
interface ethernet0
ip access-group 101 in
interface serial0
ip access-group 111 in
```

2.3.2 Cisco PIX

This section details the configuration of the Cisco PIX 525.

First we name each interface with a descriptive name. At this point we also assign security levels for the interfaces. Since a PIX will be default pass any packets sourced from a higher level interface that are bound for a lower level interface, it is important to set security levels that match the relative security of each network. The exact numbers are unimportant, it is what the number is relative to the others that matters.

```
nameif ethernet0 inside security 100
nameif ethernet1 vpn security 25
nameif ethernet2 extsrv security 10
nameif ethernet3 outside security 0
```

Here are the encrypted passwords for logging into the PIX and accessing configuration mode

```
enable password GyK/NN.eKzdi2PAv encrypted
passwd v/1tSs9e3ttp5yRU encrypted
```

We have named our firewall with an innocuous name.

```
hostname pc 462
```

Set timezone and auto adjust for daylight savings time

```
clock timezone PST -8
clock summer-time PDT recurring

Configure how many lines our terminal can display

pager lines 22
```

Set all interfaces to auto-detect TX rate and duplex

```
interface ethernet0 auto
interface ethernet1 auto
```

```
interface ethernet2 auto
interface ethernet3 auto
```

Set IP addresses and netmasks for each interface

```
ip address inside 10.0.1.1 255.255.255.0
ip address vpn 10.2.0.1 255.255.255.0
ip address extsrv 10.1.0.1 255.255.255.0
ip address outside 70.0.0.2 255.255.255.0
```

Set up a default route, and a route for all of our private IP ranges

```
route outside 0 0 100.0.0.2 1
route inside 10.0.0.0 255.255.248.0
```

Turn on logging, set the syslog and specify UDP to by used (Cisco can do TCP syslog), buffer logs on the firewall (for easy debugging), timestamp logs that are sent to syslog.

```
logging on
logging host 10.0.5.8 udp
logging buffered
logging timestamp
```

Don't pop up log messages on the console

```
no logging console
```

Shut off snmp

```
no snmp server
```

Disable web server (no pdm)

```
no http server enable
```

Telnet command commented out to disable telnet logins

```
!telnet 0.0 inside
```

Enable application protocol inspection for services that we use.

```
fixup protocol http 80
fixup protocol domain 53
fixup protocol ntp 123
fixup protocol sqlnet 1521
fixup protocol smtp 25
```

Set static NAT translations for hosts on external service network that need to be reachable

```
static (extsrv,outside) 70.0.0.3 10.1.0.3 netmask
255.255.255.255
static (extsrv,outside) 70.0.0.4 10.1.0.4 netmask
255.255.255.255
static (extsrv,outside) 70.0.0.5 10.1.0.5 netmask
255.255.255.255
static (extsrv,outside) 70.0.0.6 10.1.0.6 netmask
255.255.255.255
```

Set hide NAT groups for internal networks that need Internet access

```
nat (outside) 1 10.0.5.0 255.255.255.0
nat (outside) 2 10.0.7.0 255.255.255.0
```

Assign external IP ranges to use for hide NAT groups. We are not using port address translation because we have more addresses than internal systems that need Internet access

```
global (outside) 1 70.0.0.65 – 70.0.0.126 netmask
255.255.255.0
global (outside) 2 70.0.0.129 – 70.0.0.190 netmask
255.255.255.0
```

This ACL will be applied to the external interface in the inbound direction in the inbound direction.

Drop all traffic claiming to come from private IP ranges (RFC 1918)

```
access-list from_ext deny ip 10.0.0.0 255.0.0.0 any
access-list from_ext deny ip 172.16.0.0 255.240.0.0 any
access-list from_ext deny ip 192.168.0.0 255.255.0.0 any
```

Allow any host (limited to internet hosts since list will be applied to external interface) to talk to our DNS server on port 53 UDP & TCP…

```
access-list from_ext permit udp any host 70.0.0.5 eq 53
access-list from_ext permit tcp any host 70.0.0.5 eq 53
```

…and our SMTP server on TCP 25…

```
access-list from_ext permit tcp any host 70.0.0.3 eq 25
```

…and our web server on TCP 80 and 443

```
access-list from_ext permit tcp any host 70.0.0.4 eq 80
access-list from_ext permit tcp any host 70.0.0.4 eq 443
```

Allow suppliers, partners, and remote users to access the VPN device on UDP 500 and using ESP protocol

```
access-list from_ext permit udp 75.0.0.0 255.255.255.255
host 70.0.0.6 eq 500
access-list from_ext permit esp 75.0.0.0 255.255.255.255
host 70.0.0.6
access-list from_ext permit udp 74.0.0.0 255.255.255.255
host 70.0.0.6 eq 500
access-list from_ext permit esp 74.0.0.0 255.255.255.255
host 70.0.0.6
access-list from_ext permit udp 73.0.0.0 255.255.255.255
host 70.0.0.6 eq 500
access-list from_ext permit esp 73.0.0.0 255.255.255.255
host 70.0.0.6
access-list from_ext permit udp 72.0.0.0 255.255.255.255
host 70.0.0.6 eq 500
access-list from_ext permit esp 72.0.0.0 255.255.255.255
host 70.0.0.6
access-list from_ext permit udp 71.0.0.0 255.255.255.255
host 70.0.0.6 eq 500
access-list from_ext permit esp 71.0.0.0 255.255.255.255
host 70.0.0.6
```

Allow border router to send syslog traffic to our syslog server on UDP 514

```
access-list from_ext permit udp host 70.0.0.1 host 10.0.5.8
eq 514
```

Deny and log all other traffic

```
access-list from_ext deny ip any any log-input
```

This ACL will be applied to the external service network interface in the incoming direction.

Allow DNS server to do lookups to any internet DNS host over TCP & UDP 53

```
access-list from_extsrv permit tcp host 10.1.0.5 any eq 53
access-list from_extsrv permit udp host 10.1.0.5 any eq 53
```

Allow our SMTP server to contact any Internet SMTP host over TCP 25

```
access-list from_extsrv permit tcp host 10.1.0.3 any eq 25
```

Allow web server to communicate with database server on TCP 1521

```
access-list from_extsrv permit tcp host 10.1.0.4 host
10.0.6.2 eq 1521
```

Allow external service network hosts to log to internal syslog server on UDP 514

```
access-list from_extsrv permit udp host 10.1.0.2 host
10.0.5.8 eq 514
access-list from_extsrv permit udp host 10.1.0.3 host
10.0.5.8 eq 514
access-list from_extsrv permit udp host 10.1.0.4 host
10.0.5.8 eq 514
access-list from_extsrv permit udp host 10.1.0.5 host
10.0.5.8 eq 514
access-list from_extsrv permit udp host 10.1.0.6 host
10.0.5.8 eq 514
```

Deny and log all other traffic

```
access-list from_extsrv deny ip all all log-input
```

This ACL will be applied to the VPN interface in the incoming direction

When suppliers, partners, and external sales force connect to the VPN device, their source addresses will be changed to a private IP range by the VPN device using NAT. These rules allow traffic to get to the database server on TCP 1521.

```
access-list from_vpn permit tcp 192.168.1.0 255.255.255.0
host 10.0.6.4 eq 1521
access-list from_vpn permit tcp 192.168.2.0 255.255.255.0
host 10.0.6.4 eq 1521
access-list from_vpn permit tcp 192.168.3.0 255.255.255.0
host 10.0.6.4 eq 1521
access-list from_vpn permit tcp 192.168.4.0 255.255.255.0
host 10.0.6.4 eq 1521
access-list from_vpn permit tcp 192.168.5.0 255.255.255.0
host 10.0.6.4 eq 1521
```

Allow VPN traffic to perform DNS lookups on our internal DNS. Required for looking up address of database server.

```
access-list from_vpn permit tcp 192.168.1.0 255.255.255.0
host 10.0.6.4 eq 53
access-list from_vpn permit tcp 192.168.2.0 255.255.255.0
host 10.0.6.4 eq 53
access-list from_vpn permit tcp 192.168.3.0 255.255.255.0
host 10.0.6.4 eq 53
access-list from_vpn permit tcp 192.168.4.0 255.255.255.0
host 10.0.6.4 eq 53
access-list from_vpn permit tcp 192.168.5.0 255.255.255.0
host 10.0.6.4 eq 53
```

Allow VPN device to communicate with VPN devices at suppliers and partners networks. Since the VPN will never initialize communication with external sales force, we do not need to allow traffic out for that as it will be handled by the PIX state table functionality.

```
access-list from_vpn permit udp host 10.2.0.2 host
74.0.0.18 eq 500
access-list from_vpn permit esp host 10.2.0.2 host
74.0.0.18
access-list from_vpn permit udp host 10.2.0.2 host
73.0.0.25 eq 500
access-list from_vpn permit esp host 10.2.0.2 host
73.0.0.25
access-list from_vpn permit udp host 10.2.0.2 host
72.0.0.10 eq 500
access-list from_vpn permit esp host 10.2.0.2 host
72.0.0.10
access-list from_vpn permit udp host 10.2.0.2 host 71.0.0.5
eq 500
access-list from_vpn permit esp host 10.2.0.2 host 71.0.0.5
```

Allow VPN device to sync up to NTP server on UDP 123

```
access-list from_vpn permit udp host 10.2.0.2 host 10.0.5.7
eq 123
```

Allow VPN device to send syslog on UDP 514

```
access-list from_vpn permit udp host 10.2.0.2 host 10.0.5.8
eq 514
```

Deny and log all other traffic

```
access-list from_vpn deny ip any any log-input
```

This ACL will be applied to the internal interface in the inbound direction.

Allow Exchange server to transfer mail to SMTP server on TCP 25

```
access-list from_internal permit tcp host 10.0.5.4 host
10.1.0.3 eq 25
```

Allow internal DNS servers to to DNS lookups to the Internet

```
access-list from_internal permit udp host 10.0.5.2 any eq
53
access-list from_internal permit tcp host 10.0.5.2 any eq
53
access-list from_internal permit udp host 10.0.5.3 any eq
53
access-list from_internal permit tcp host 10.0.5.3 any eq
53
```

Allow our patch management system to access the internet to download updates. The patch management update is an HTTP connection on TCP 80.

```
access-list from_internal permit tcp host 10.0.5.6 77.0.0.0
255.255.255.0 eq 80
```

Allow systems on workstation subnet to connect to the external service web server on TCP 80 and 443. This rule must come before the next rule, or this traffic will be blocked based on that rule and never be evaluated against this rule.

```
access-list from_internal permit tcp 10.7.0.0 255.255.255.0
host 10.1.0.4 eq 80
access-list from_internal permit tcp 10.7.0.0 255.255.255.0
host 10.1.0.4 eq 443
```

Drop all other traffic from the workstation subnet to the external service network and log it. Internal users have no need to access these systems. This rule needs to be here since the internal PIX interface is a more secure interface than the external service PIX interface, and the PIX will be default pass traffic from a more secure interface to a less secure interface. It also needs to be before the next rule or this traffic will be passed based on that rule and never get evaluated by this rule.

```
access-list from_internal deny ip 10.7.0.0 255.255.255.0
10.1.0.0 255.255.255.0 log-input
```

Allow workstation network machines to connect to all other addresses. This would be the normal behavior, but we have place a deny all rule at the end of this ruleset, so we must specify that this traffic should be passed.

```
access-list from_internal permit ip 10.7.0.0 255.255.255.0
any
```

Allow internal NTP server to sync with external service NTP server on UDP 123

```
access-list from_internal permit udp host 10.0.5.7 host
10.1.0.2 eq 123
```

Drop and log all other traffic

```
access-list from_internal deny ip any any log-input
```

Apply ACL to appropriate interface in the inbound direction.

```
access-group from_ext in interface outside
access-group from_internal in interface inside
access-group from_vpn in interface vpn
access-group from_extsrv in interface extsrv
```

Set the timeout for the ARP cache.

```
arp timeout 14400
```

Disable snmp

```
no snmp-server location
no snmp-server contact
no snmp-server enable traps
```


2.3.3 Cisco 3015 VPN

Note: Sample screenshots and instructions in this section based on Cisco's VPN 3000 series configuration examples found at http://www.cisco.com/warp/public/471/config_vpn_3k_site.html.

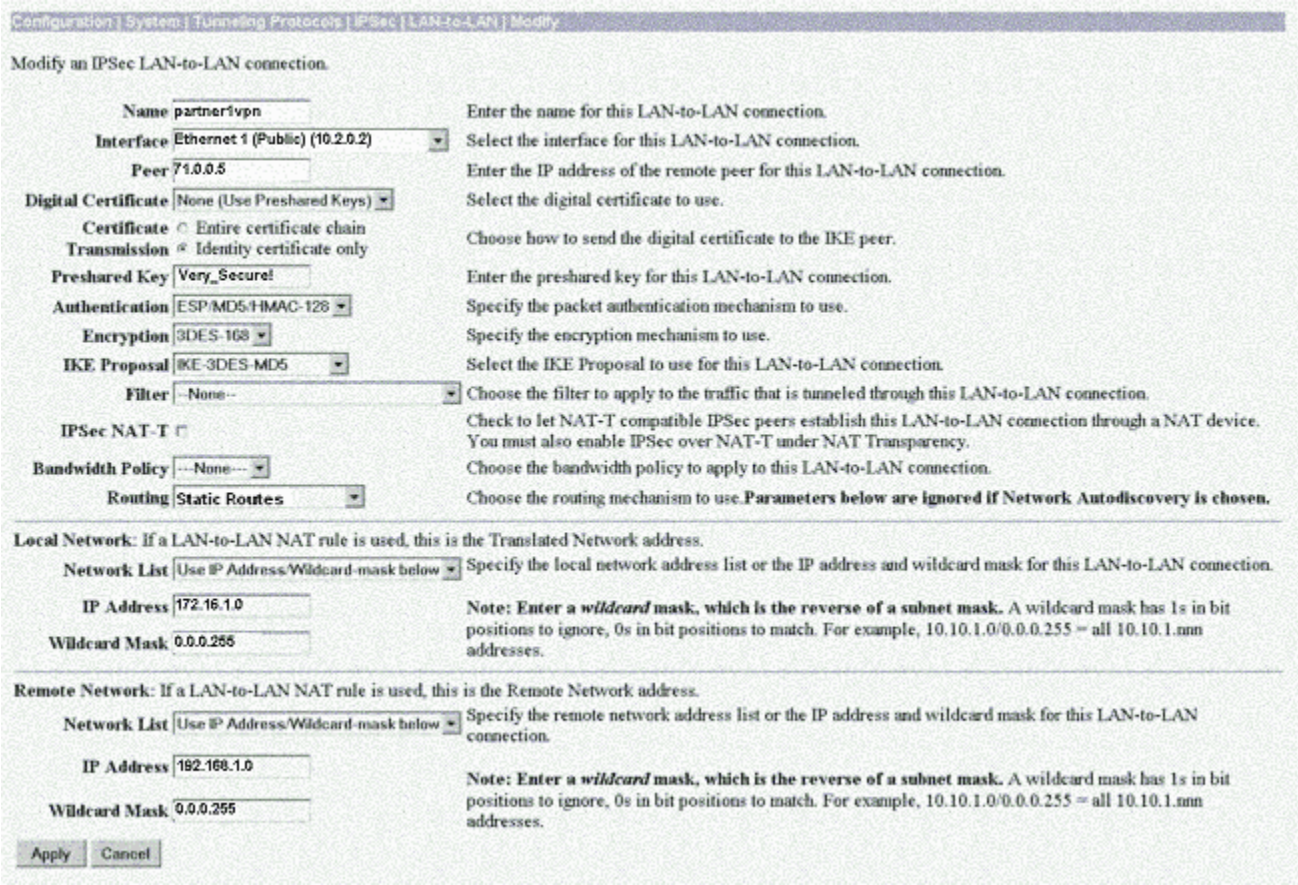
This table shows the actual VPN device IP (entire subnet for ISP dialup provider) and what we will translate their internal addresses into when they enter our network.

Subject	VPN Device	Translated to

Partner 1	71.0.0.5	192.168.1.0/24
Partner 2	72.0.0.10	192.168.2.0/24
Supplier 1	73.0.0.25	192.168.3.0/24
Supplier 2	74.0.0.18	192.168.4.0/24
ISP dialup provider	75.0.0.0/24	192.168.5.0/24

We will now step through the configuration for the Cisco VPN device as used for the VPN tunnel to Partner company 1.

First, we must access the LAN-to-LAN IPsec setup. This is found at *Configuration* > *System* > *Tunneling Protocols* > *IPSec* > *LAN-to-LAN* > *Modify* in the VPN configuration interface.



Input fields are addressed in order.

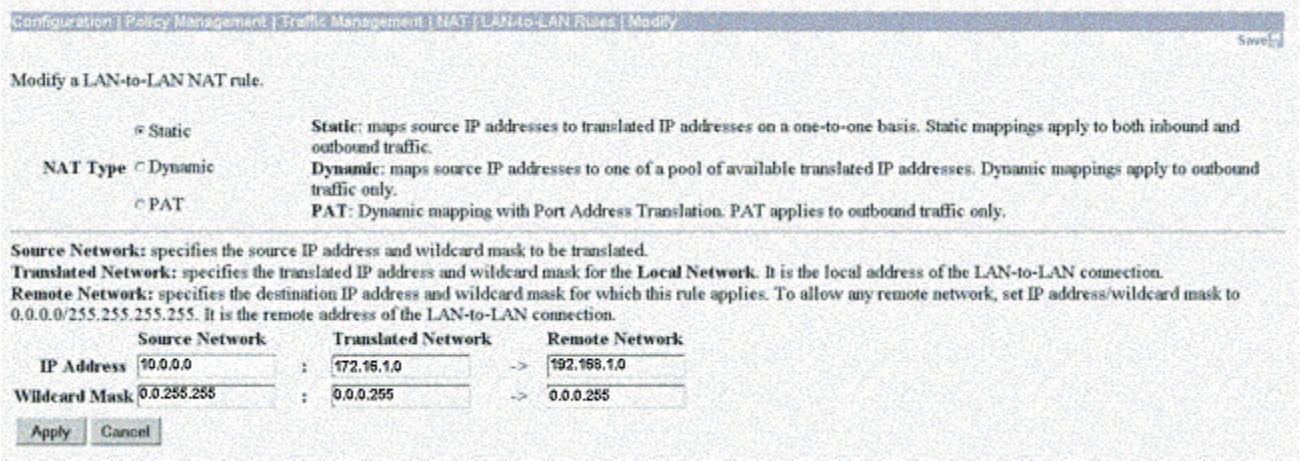
Name: We must name this connection. Here, we have used a descriptive name.
Interface: Choose the interface that will be the external interface. We have chosen Ethernet 1.

Peer: This is the IP address of the remote system that will be the other end of the VPN tunnel.
Digital Certificate: For ease of use, we have chosen to go with pre-shared keys. The keys will be either shared in person or transmitted via email using PGP or GPG. We will likely move to digital certificates once we are familiar with them.
Certificate transmission: Not used without digital certificates.
Preshared Key: This is our pre-shared key. A pre-shared key should follow the same rules as a strong password.
Authentication: We have chosen ESP (Encapsulating Security Protocol), MD5, and HMAC-128 as the possible methods for providing data integrity.
Encryption: Her we use true triple-DES encryption with a 168-bit key for encrypting traffic.
IKE Proposal: These are the methods that the device will use during the separate phases of IKE key negotiation.
Filter: We can filter traffic leaving the VPN device, but we have chosen to do that with our PIX as the VPN does not provide as many options as the PIX.
IPSec NAT-T: Only needed if partner's VPN device is itself behind a NAT device.
Bandwidth Policy: We are not managing the bandwidth with the VPN device.
Routing: We will use static routing to direct traffic flow.

Local Network
These settings configure how our traffic is presented to the remote network. Here, we have chosen to use NAT to change our addressing scheme to the 172.16.1.0/24 range. This is to prevent information disclosure.

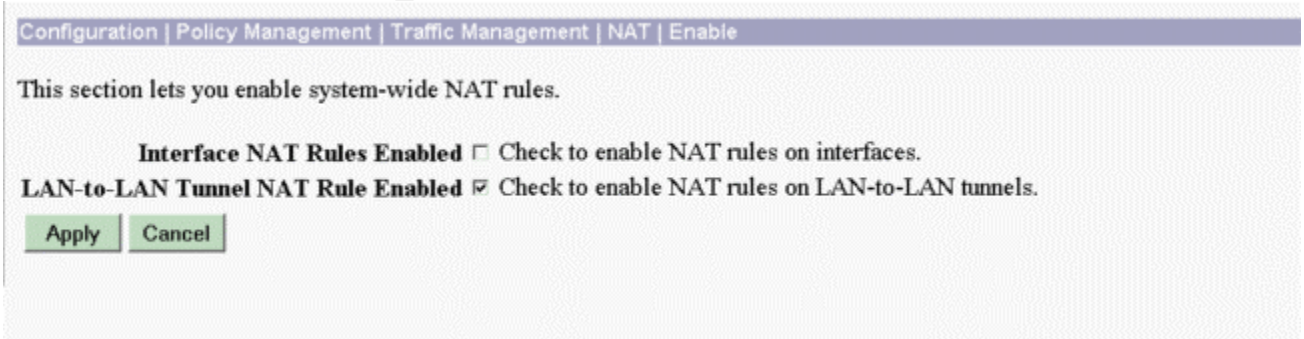
Remote Network
We are translating the remote network addresses into the private 192.168.1.0/24 range. This prevents possible address collision. It also allows our partner to change their internal addressing with minimum hassle on our end. None of the filtering rules on our PIX would have to be re-written due to our partner's addressing change.

Next, we must set up a static NAT rule. This is accessed under *Configuration* > *Policy Management* > *Traffic Management* > *NAT* > *LAN-to-LAN Rules* > *Modify*.



NAT Type: We have chosen static NAT mapping.
Source Network: This is our local network that the VPN device will see traffic from. Since our partner will be accessing hosts on two internal subnets (DNS server and database proxy are on different subnets) we have used a 0.0.255.255 bitmask to allow traffic from both systems to be sent over the VPN tunnel.
Translated Network: This is the address range that our partner will see our machines inhabiting (regardless of their actual addressing).
Remote Network: The addresses that we will see for our partner's systems (regardless of their actual addressing).

Finally, we must enable the LAN-to-LAN NAT translation. These settings can be found by following *Configuration > Policy Management > Traffic Management > NAT > Enable*.



3 Verify the Firewall Policy

3.1 Overview

We have now planned out and deployed what we believe is a highly secure network. Unfortunately, this is where many security practitioners stop. What we need to do now is verify that the configurations that we have put in place are actually performing as we have intended them to. This is a security audit.

For the purposes of demonstration, we will be auditing our perimeter firewall. In a production environment, an audit should be performed on each unique configuration on the network i.e. only one workstation needs to be audited if all workstations are deployed using the same secure configuration standard.

An excellent guide to firewall auditing can be found at http://www.spitzner.net/audit.html. Here, Lance Spitzner has outlined a very thorough guide to firewall auditing that we will use as a base for our firewall audit. It goes beyond simply verifying the security policy, and we feel that everything in the guide should be used instead of just verifying that the firewall is managing traffic according to the policy.

One note to make that is not mentioned in Lance's guide is that auditing any system can cause very bad things to happen. When you start sending crafted packets at systems, the results can often be unexpected. Systems may lock up, crash, or spontaneously restart. Business activities may come to a screeching halt.

Because of this, we recommend obtain understanding and support of your activities from a high level of management. This may be a CIO, CTO, CEO, or General Manager. Whoever it is, make sure to get their ok, and get it in writing if possible. People have lost their jobs and gone to jail for performing this type of activity.

3.2 Planning

3.2.1 Time of Day

The time of day that the scanning is performed is important. For penetration testing, it is often best to scan during periods of peak business in order to hide activity amongst legitimate traffic. For auditing, it is usually best to scan during periods of low business activity. We are not trying to hide our scanning, and it may even prove educational for some of our junior employees to see what scanning traffic looks like.

Once GIAC Enterprises is running live business on their network, we will perform audit scans and configuration changes on Sundays at 10:00 p.m. This will disrupt

the least amount of business for GIAC Enterprises. We will make sure to warn business partners and place notices on both our internal and external web sites.

For this first audit, we will not have to take any of these precautions. We have created a new network for GIAC Enterprises, so we have the luxury of testing whenever we like before the system has gone live.

3.2.2 System Backups

A normal part of the auditing process will also be to make sure that we have backups of all configuration files from our firewalls and routers, and backups of all of our mission critical systems. If our scanning should cause any data loss to any of our systems after we are live, we want to be able to recover as quickly as possible.

3.2.3 Technical Plan

The first part of our audit will be to test the physical security of the firewall. A firewall must be located in a secure location so that no unauthorized person can tamper with it.

Next, we will verify that the firewall has been correctly hardened. In our case, this mainly consists of verifying that services such as finger or echo are not running. Compared to a firewall running on a Linux or Windows host, a Cisco PIX is close to hardened out of the box, with only a few services to disable.

Now we will scan the firewall itself to ensure that the firewall does not respond to any traffic directed directly at it. Since our policy is that all firewall and router changes are managed through the local console connection, the firewall should not respond to any traffic from any address on any interface. Also, the firewall should log this activity to the syslog server.

To test this, we will scan all ports of the firewall on all interfaces by generating our own packets using a tool called nmap. If we have configured the firewall correctly, we will receive no replies to any of our traffic

Finally, we need to verify that the various rules that we have configured into our firewall are performing as we expect them to. To do this, we will use nmap in conjunction with tcpdump, a network sniffing tool. By placing a host running nmap on a network connected to one firewall interface and hosts running tcpdump to the networks connected to the other firewall interfaces, we can generate various types of traffic and watch to see where the traffic goes (or doesn't go). By using nmap to send traffic at each firewall interface in turn, we can determine exactly what traffic will pass through the firewall and what traffic won't. We will also be able to verify that the firewall is correctly logging traffic.

By comparing our actual results to the results that we expected our ruleset to accomplish, we can determine if any changes need to be made to our ruleset.

3.2.4 Time/Cost Estimate

Task	Hours	Comments
Planning	5	
Physical Security	1	
Host Security (hardening)	1	
Port scan Firewall	4	(1 hour per interface)
Rulebase verification	8	(2 hours per interface
Report generation	5	
Total hours	24	
20% allowance for unexpected events	28.8	
Final Cost	**4320**	**# hours x $150 labor rate**

3.3 Physical Security

The first step to security is physical security. The firewall should be physically secure. This means it is located in a locked room where only a minimum of authorized users have access. A system that logs each physical access is nice to have here as well.

Although it violates many company's internal policies, it is a good idea to leave the door to the network room or server room unlabeled. If an attacker is going to mount a physical intrusion, at least make them guess which door they need to go through. Another common item that is overlooked is raised ceilings. Often, getting into a supposedly secure server room only requires finding a ladder and pushing up a few ceiling tiles.

The reason behind all of these precautions is that once someone gains physical access to your systems, nothing else matters. For instance, an attacker with console access to a Cisco PIX firewall can halt the normal boot sequence and cause it load a factory-fresh configuration from a network location. They can then gain privileged mode on the firewall and configure it however they like. For firewalls running on standard computers, an attacker can simply remove the hard drive from the system and mount it in another computer, reading and modifying the contents as desired.

3.4 Firewall Defense

3.4.1 Harden the Firewall

Once we have verified that our firewall and other network systems have adequate physical security, the next step is to verify that the operating system

has been hardened as required for a system that operates in such a potentially hostile networked environment.

For some systems, such as Cisco PIX and some other dedicated firewall devices, this work has mostly been done since the built-in operating system has few services to enable. However, even hardware-based firewalls have the ability to run some network services. We have verified that the PIX firewall has had all non-essential services disabled.

We have also verified that we are running the most up-to-date release of our particular firewall firmware. This is analogous to verifying that the latest patches have been installed.

3.4.2 Firewall Port Scan

Next on the list is to scan the firewall itself from both the internal network and external to the firewall. This is to ensure that the firewall is not responding to any packets that are being sent at it from any system other than authorized system management stations. If a firewall is incorrectly configured so that it would allow a plaintext telnet session from an unauthorized computer, an attacker can spend all day trying to guess your login password.

Since we are using a filtering router, we will have to connect between the router and the firewall for our external testing in order to make sure that the router is not blocking some traffic that we want the firewall to be exposed to for testing purposes. This can easily be done with a simple network hub.

For this type of testing, nmap (http://www.insecure.org/nmap) is generally considered to be the best tool around. It will allow you to send almost any type of packets toward your target, and then it will give you the results of any returned packets. If that wasn't enough, there is also a Windows version (http://www.nmapwin.org/) and has graphical front ends for both operating systems. The options for nmap are listed below (from nmap.usage.txt included with the nmap tarball at http://download.insecure.org/nmap/dist/?M=D):

```
Nmap 3.28 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
```

-F Only scans ports listed in nmap-services
-v Verbose. Its use is recommended. Use twice for greater effect.
-P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
-6 scans via IPv6 rather than IPv4
-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
-n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
-oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
-iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
--interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'

For most of our scanning, we will be using the stealth TCP SYN scan (option -sS). In this mode, nmap will send out a TCP SYN packet attempting to open a connection. If it receives a TCP SYN/ACK, it will list the port as open. Receipt of an ICMP administratively prohibited will cause the port to be listed as closed. Failure to receive any packet will list the port as filtered. No final TCP ACK will be sent. We must then check to see if our firewall has correctly logged this "half connection". This provides us with the most information, along with being how many black hats scan networks when performing reconnaissance.

We connect our laptop to the hub between the border router and the firewall and configure with a valid IP from our available external range. Nmap is run against our firewall, scanning all ports with a stealth TCP SYN scan. The command that we will use is: nmap –sS –P0 –p 1.65535 –n –v –T 3 100.0.0.1.

The –sS option specifies a TCP SYN stealth or "half-open" scan where the final TCP ACK is never sent. Some firewalls will not log dropped packets if the connection is never successfully set up. The –P0 option prevents nmap from pinging the host to first see if it is up. Since our firewall drops ICMP echo requests, this will prevent nmap from halting the scan when it thinks that the host is down. –p 1-65535 will cause the scan to scan every port from 1 through 65535 (inclusive). –n tells nmap not to do a DNS lookup to resolve the host's name. –v is verbose mode, which returns more information about the scan. –vv can be used to get the most information. –T 3 selects the frequency with which nmap sends out packets. –T 0 is Paranoid (long time between packets) and –T 5 is insane (almost no timing between packets). 100.0.0.1 is the host we are targeting.

Output from nmap:

```
nmap –sS –P0 –p 1-65535 –n –v –T 3 100.0.0.1
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Host (100.0.0.1) appears to be up … good.
Initiating SYN stealth scan against (100.0.0.1)
The SYN Stealth Scan took 9265 seconds to scan 65535 ports
All 65535 scanned ports on (100.0.0.1) are: filtered
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in
9265 seconds
```

As you can see, this scan took a long time. This type of scan will typically take from one to several hours. This is due to configuring our firewall to drop traffic instead of reject traffic. Rejecting traffic will send out an ICMP administratively prohibited, which informs an attacker in short order that a port is closed. Dropping traffic will force an attacker to send out multiple packets to each port and wait for the full timeout before the port can be considered not open. With 65535 ports, all of the timeouts add up. An attacker may decide to move on to a softer target rather than wait a few hours per host for their scan to complete.

We are happy to note that we did not see any return traffic on any of our scans. Checking our syslog server, we found that the firewall logged all of the packets that it dropped.

We must now perform the same scan from our internal network and our external service network. Fortunately, the results were the same for the additional scans. Since this is our first audit of this firewall, we have also chosen to perform a UDP port scan (identical from above except for –sU instead of –sS) and an ICMP scan (–sP option replaces –sS). Again, no returns from the firewall were noted and everything was logged correctly.

3.5 Verify the Rules

We now consider the firewall itself secure. This is only part of the process since a firewall must do much more than simply sit there and be secure. It must monitor and selectively pass traffic. Now we must verify that the firewall is passing traffic that we expect it to pass, and dropping traffic that we expect it to drop.

This step will again use nmap, along with another common tool called tcpdump (http://www.tcpdump.org/). For those auditing from Windows systems, tcpdump has been ported to Windows as Windump (http://windump.polito.it/).

3.5.1 Scanning from the Internet

We will start this portion of the testing with our nmap system connected between the filtering router and the firewall, but with three additional systems. One system running tcpdump will be connected to the monitoring port of the external service switch. Another will also be running tcpdump, but will be connected between the firewall and our internal router using a basic hub. The third system will connect between the firewall and the VPN device, again with a basic hub.

Using this setup, we can generate a wide variety of traffic from the system running nmap, and monitor to see what traffic gets through by viewing the output

from tcpdump. The goal now is to run nmap against every public IP address that routes to GIAC Enterprises.

Our first scans with nmap are very similar to our previous scans:

```
nmap -sS -P0  -p 1-65535 -n -v -T 3 100.0.0.0/24
nmap –sU –p0  -p 1-65535 –n –v –T 3 100.0.0.0/24
nmap –sP –p0  -p 1-65535 –n –v –T 3 100.0.0.0/24
```

The only difference is that our target host is now the entire public subnet that belongs to GIAC Enterprises. Monitoring the systems running tcpdump, we found that the only traffic that is allowed through is traffic that matches the expected behavior of our ruleset. This included TCP port 25 to our SMTP server, TCP port 80 and 443 to our web server, UDP port 500 to the VPN server, and port 53 over both TCP and UDP to our DNS server. We were not able to reach the syslog server over UDP 514 since the source address was not that of the border router. A special scan to UDP port 514 using the source address of the border router verified that this one address was able to get through to the syslog server. We are also able to see return traffic from those internal systems back to our scanning host. Additionally, the firewall logged all of the traffic to the syslog server.

Note: The hosts that are accessible using publicly routable IP addresses are all low in the range of addresses. We are not going to actually scan every address from xxx.xxx.xxx.1 through xxx.xxx.xxx.254. We don't have several days to perform scans! Once we have scanned every address that corresponds to an existing host plus a small range of addresses that are in the range used for both static and hide NAT, we will terminate the scan. If we cannot pass traffic to a small group of the addresses in the NAT pool, we should not be able to pass traffic to any of them.

3.5.2 Scanning from the External Service Network

For our next scan, we will swap the laptop running tcpdump on the external service network with the laptop running nmap, so that we can now test the behavior for traffic coming from the external service segment. We will again run the nmap scans, but this time we will target both our internal IP address range and also the Internet. We will also have to change either the IP address of the nmap host or instruct nmap to change the source address since we expect different traffic to pass to the internet or internal network based on what address the traffic is coming from.

We have chosen to shut down the hosts on the external service network and change the IP address of the laptop running nmap. This will prevent traffic from being returned to the hosts on the external service network, which then might result in additional traffic. Shutting down the hosts on the external service

network means that we will only have to deal with the minimum amount of traffic as we verify the firewall rules.

Note: Again, we will not scan every IP in our internal address range, VPN address range, and the Internet. A small range of hosts from each subnet should tell us what we need to know.

Checking our tcpdump output, we find that the following traffic was passed by the firewall:

IDS Server IP: Able to access syslog server on internal network via UDP 514.

SMTP Server IP: Able to access both internal and external email servers using TCP 25.

DNS Server IP: Able to send traffic to any host over TCP and UDP 53.

Web Server IP: Able to access the database proxy server on the internal network via TCP 1521, but unable to send any traffic to the Internet or VPN subnet.

3.5.3 Scanning from the VPN

Bringing the hosts on the external service network back up, I again re-arrange machines. The tcpdump laptop is moved from the VPN network to the external service network and the laptop running nmap is moved to the VPN network.

Scanning from the VPN subnet, we were unable to get any packets through to the Internet, our external service network, or our internal network. After shutting down the VPN device and scanning using the VPN device's IP address, we were able to pass packets over UDP port 500 to any address.

Changing to the private IP range of 192.168.1.0/24, we were able to pass traffic on TCP port 1521 to the database proxy server and to the internal DNS server using TCP and UDP port 53.

3.5.4 Scanning from Internal Network

Our final scans will originate from our internal network. With a laptop running tcpdump again connected to the VPN network, we begin to scan from inside GIAC Enterprises' LAN. We have temporarily connected our laptop running nmap between the internal router and the firewall so that we can send out packets with a source address from any of our internal subnets.

We scan the external service network, VPN network, and Internet using a source address from our workstation subnet. As per GIAC Enterprises' policy, we are able to send any traffic to the Internet, but only TCP port 80 to the web server's

address on the external service network. No traffic can be passed to the VPN network.

Using an address from the database subnet, we are unable to pass any traffic to anywhere. The database proxy will only be responding to other systems' connections. The PIX will automatically allow responses to connections that have been initiated by using its state table functionality.

Finally, we use an address from our internal service network. We are unable to send any traffic to the Internet, VPN network, or external service network. Using the source address of each internal server in turn, we are able to pass the following traffic:

NTP server IP: UDP port 123 to external NTP server

Domain/DNS server IP: TCP and UDP port 53 to any Internet host

Exchange server IP: TCP port 25 to external SMTP server

Domain/DNS/DHCP server IP: TCP and UDP port 53 to any Internet host

Patch Management server IP: TCP port 80 to software vendor's IP block.

3.6 Analysis

To analyze whether the traffic that we could pass conformed with our security policy, we review the traffic and the rule that allowed the traffic to pass. We then asked ourselves whether the traffic passed was what the rule should do and what we expected the rule to do. Then we must determine if the allowed traffic is the absolute minimum that is needed to pass on order to support continued business operations.

Here is our comparison:

External Network:

	Expected?	Minimum Needed?
TCP port 25 to our SMTP server	X	X
TCP port 80 to our web server	X	X
TCP port 443 to our web server	X	X
UDP port 500 to the VPN server	X	X
port 53 over TCP to our DNS server	X	X
port 53 over UDP to our DNS server	X	X
UDP port 514 to syslog from border router	X	X
External network IDS to Syslog UDP 514	X	X

External SMTP to any TCP 25	X	X

External Service Network

	Expected?	Minimum Needed?
IDS to syslog UDP 514	X	X
SMTP to any TCP 25	X	
External DNS to any Internet IP UDP 53	X	X
External DNS to any Internet IP TCP 53	X	X
Web server to datbase proxy TCP 1521	X	X

VPN Network

	Expected?	Minimum Needed?
192.168.1.0/24 to database proxy TCP 1521	X	X
192.168.1.0/24 to internal DNS TCP 53	X	X
192.168.1.0/24 to internal DNS UDP 53	X	X
VPN device to any IP UDP 500	X	
VPN device to any IP ESP	X	

Internal Network

	Expected?	Minimum Needed?
Internal NTP to external NTP UDP 123	X	X
Exchange to SMTP TCP 25	X	X
Internal DNS to any host TCP and UDP 53	X	X
Patch server: Port 80 to vendor	X	X

3.7 Recommendations

Based on our findings that some traffic was not the minimum possible, we have recommended several rule changes based on the rules that did not provide the highest level of security. First, a change in the rule allowing the external DNS server to connect to any IP over TCP and UDP 53. The rules should be re-ordered and modified to allow traffic to only Internet DNS servers. Second, the external SMTP server can now send traffic to any host over port 25. This should be modified to allow communication to any Internet SMTP server along with the internal Exchange server. Finally, the VPN device can now send traffic to any host over UDP 500 and ESP. This should be modified to only allow traffic to the addresses of partner and supplier VPN devices and the IP range of GIAC's dial-up ISP.

4 Design Under Fire

4.1 Overview

At this point, I switch hats from a consultant for GIAC Enterprises to a hacker with the notorious underground hacking group Black Hats, Ink. As a member of this group, I often target networks on the Internet with multi-phase attacks in order to prove my "mad skillz" to the world. Today, I have chosen to attack the network of Chong KahSing. His network design is located at http://www.giac.org/practical/GCFW/Chong_KahSing_GCFW.pdf, and was published on April 30, 2003.

4.2 Attack the Firewall

My first attack is aimed at the firewall itself. Chong has chosen to use Checkpoint Firewall-1 NG as his external firewall. While Firewall-1 NG is a good firewall product, the most insidious flaw of all still can bite it: misconfiguration.

I am hoping that Chong has forgotten to disable the default ruleset that Checkpoint Firewall-1 is configured with. Checkpoint Firewall-1 has a large list of implied rules that allow certain traffic to pass through the firewall without being logged. This is true even if filters are configured to specifically block this traffic, since most of the implied rules are processed before the user-configured rules.

To view these rules, an administrator must open the Firewall-1 policy editor and select *View* | *Implied Rules*. This will show the long list of traffic that will pass through a Checkpoint Firewall-1 unless the implied rules are disabled.

4.2.1 Research

Since I am hoping that a great deal of traffic can be passed either to or through Chong's firewall, all I need is a known Checkpoint Firewall-1 vulnerability to take advantage of. Searching through the archives of several security mailing lists and security vendors, I found an interesting vulnerability report. Internet Security Systems (http://www.iss.net/) reported on July 7th, 2000, that sending a continuous stream of packets to a Checkpoint Firewall-1 on port 264 resulted in CPU utilization spiking to 100%, causing a denial of service condition that prevents any local or remote users from accessing the management GUI (http://www.iss.net/security_center/static/7368.php). This bulletin was later updated to further state that the issue had not been resolved as of June 2002.

If Checkpoint did not fix this issue for two years, is it possible that they have not fixed it to this day? As this is the only serious Checkpoint Firewall-1 vulnerability that has not been definitively resolved, I have decided to use this vulnerability as my vector of attack.

4.2.2 Execution

Exploitation of this bug is trivial. Any program capable of crafting a packet will do the job. I will use hping2 to send out my packets.

```
hping2 202.185.236.3 --fast -a www.thisisfake.com -p 264 -d
64
```

Details: hping2 (binary executable) 202.185.136.3 (target host) --fast (10 packets/sec.) -a www.thisisfake.com (spoofs source address with host listed) -p 264 (target host port 264) -d (packets will contain 64 bytes of data)

I have chosen to launch this attack during the middle of the day to give myself the most cover and cause the most embarrassment to my target. At 12:00 pm, I enter the command in and wait. Not knowing how much data must be sent to be successful, I allow hping to run for 15 minutes. At that point, I terminate the program and hope that I accomplished my goal.

4.2.3 Analysis

It turns out that my attack was not successful. First, testing for this vulnerability on Checkpoint Firewall-1 NG with feature pack 3 has shown that it is not vulnerable. Although Chong has opted to stick with feature pack 2 instead of feature pack 3, I still believe that the vulnerability has been fixed. In this case, however, it didn't matter. Even if Checkpoint had not fixed this bug, Chong would have been safe. Why was he protected from a known vulnerability on a device that he has deployed? Defense in depth.

First, Chong disabled the implied rules when deploying his Checkpoint Firewall-1 so that it would drop the traffic that I sent at it. Not only that, but he has implemented filtering on his border router. He is only allowing traffic past his router that he knows is required for his business operations. The packets that I was sending to his Checkpoint Firewall-1 never even reached it. Due to exercising defense in depth, Chong would have been defended even if this vulnerability had not been fixed.

4.3 Denial of Service

An attack with little visible effect to the outside world is not worth very much when it comes to credibility in the hacking underground. What I need to do is knock Chong offline. That will show him and everyone else just what I am made of.

For this phase of my attack, I will be subjecting Chong to a distributed denial of service (DDOS) attack. I will perform this attack using 50 cable modem systems that I have compromised.

4.3.1 Research

I have two choices to make when launching a DDOS attack. First, what tool I will use, and second, what type of attack I will launch. Obviously, my second choice will be limited by the capabilities of my first choice (or vice versa if I want to find a tool to perform a specific type of attack).

In researching the first choice, there are really only two options. The vast majority of cable modem users are running some flavor of Microsoft Windows, so my tool of choice must at least have a Windows agent, whether the handler runs on Windows or not. The two main contenders that fall within this category are WinTrinoo, a Windows version of the well-known Trinoo DDOS agent, and Tribe Flood Network 2000 (TFN2K), the latest and greatest from the author of the original Tribe Flood Network software.

When comparing the two, TFN2K is the obvious winner. It is much more difficult to detect than WinTrinoo since it can be configured to accept communications over a variety of protocols, it encrypts all communication data, and both the agent and handler can spoof the source address in all packets. Additionally, the agent accepts commands silently without sending confirmation back to the handler.

Not only is TFN2K more difficult to detect, it also has a much greater feature set than WinTrinoo. Where WinTrinoo is limited to UDP floods, TFN2K can create UDP, TCP SYN, and both ICMP and broadcast ICPM floods.

Due to both of these advantages, I have chosen to infect my cable modem users with TFN2K. Getting the agent onto these computers is trivial, as most Windows home computer users rarely, if ever, keep up-to-date with patches from Microsoft. I have a variety of pre-coded tools available to exploit a vulnerability and get my agent installed.

Once I have my agents installed, I must decide what type of attack to perform. Again, there are two main choices for a DDOS attack. The older method of attack is designed to exhaust the resources of a target machine. Typically referred to as a SYN flood, an attacker sends a large volume of TCP SYN packets with spoofed source addresses at a target. The target returns a TCP SYN/ACK packet for each one and allocates resources for the pending connection. Most operating systems will wait for 30 to 90 seconds or even longer before allowing the connection to time out. Since most firewalls on the Internet are configured to silently drop unsolicited TCP SYN/ACK packets, it is unlikely that any hosts that correspond to the spoofed addresses will return a TCP RST to clear the connection, the standard response to an unsolicited TCP SYN/ACK.

Eventually, the target host will run out of resources for new incoming connections and either crash or remain in a state unable to accept further connections until existing connections time out. In this manner, even an attacker on a 56K modem

sending out a mere 10 packets per second can quickly exhaust the resources of a powerful server.

However, this attack is also the easier of the two to defend against as many firewalls now come with some type of SYN flood prevention capability.

A newer form of attack is largely a result of the proliferation of always-on high-bandwidth Internet connections in the home. An attack can be launched that sends such a high volume of traffic at the target that the target's Internet connection is simply filled up. This traffic can range from UDP traffic; to TCP SYN, SYN/ACK, or ACK traffic; to ICMP traffic, especially ICMP echo replies received from misconfigured networks used as "smurf amplifiers". All that is needed is a handful of compromised cable modem or DSL users. Although many cable and DSL systems cap the upload speed of users at 128 Kbs, some simple math shows that a full T1 connection at 1.544 Mbs can be overwhelmed by only 13 DSL or cable users (128 Kbs x 13 = 1664 Kbs or 1.664 Mbs). If compromised users are on a system that has a higher upload cap, even fewer are needed.

This type of attack is much more difficult to defend against than the resource attack, especially if the attacker is smart and sends traffic that mimics traffic the target normally allows or needs. If the target runs a web server as part of their business, it will be very difficult to filter out the attacker's traffic from random source addresses on port 80 from legitimate traffic on port 80.

Even if the target is able to discern a signature in the attacker's traffic, the filtering must take place upstream from the target's Internet connection. Filtering the traffic at the border router will stop the traffic from entering the target's network, but the traffic will continue to clog the target's Internet connection. If the target is able to find a way to filter the traffic, the time it takes to uniquely identify the attacker's traffic and work with an ISP to get filters in place my take several hours. If the target is not able to do this, the attack will continue until the compromised hosts running the DDOS agents are offline or the attacker stops the attack.

4.3.2 Execution

For my attack, I have chosen to employ several different methods. Since I have 50 cable modem users at my disposal, I have chosen to split them into different groups, each performing a different attack. My hope is that this will make my attack more difficult to understand and will cause Chong to spend more time before he is able to mitigate it.

The first group of cable modem users will consist of 10 hosts sending ICPM echo requests (ping) to a list of "ping amplifiers" or "smurf amplifiers". The list that I will be using is found at http://www.powertech.no/smurf/. These are networks that are incorrectly configured to allow ICMP echo requests to the broadcast address

of their network, which are then broadcast to every host on their network. The result is a large number of ICMP echo replies for each ICMP echo request sent out. By spoofing the source address, these ICPM echo replies can be directed to any host on the Internet. My hope is that this traffic alone will be enough to overwhelm Chong's Internet connection.

I will then configure the remaining 40 hosts at my disposal to send TCP SYN packets with random source addresses at port 80 of Chong's web server. If I am lucky, this will overwhelm the resources of Chong's web server. If Chong has systems in place to prevent his web server from receiving my TCP SYN traffic, the sheer volume should also be more than enough to overwhelm his Internet connection, even without the ICMP traffic figured in. This traffic should also be much more difficult to filter out than the ICPM traffic above.

I will launch this attack on a Monday at approximately 8:00 am Pacific time (11:00 am East Coast time, 4:00 pm in the UK). Since my goal is to publicly knock Chong offline, this gives the best window for customers and partners worldwide to be unable to connect. Also, any traffic created by customers and partners attempting to connect will only aid my efforts.

Since autonomous agents carry out my attack, I am done once I have triggered the attack. The only thing that I will do after kicking off the attack is try to connect to Chong's website every 10 minutes to monitor the progress of my attack.

4.3.3 Analysis

Even though my attack did not work exactly as I had hoped, it was still incredibly successful. Unfortunately, none of my TCP SYN traffic ever reached Chong's web server. Chong has chosen to enable the SYN Relay feature on his Checkpoint Firewall-1. Using this feature, the Checkpoint Firewall-1 completes the TCP 3-way handshake for incoming connections on behalf of the target host without the host (in this case, the web server) ever seeing a packet. If a connection is successfully set up, the Checkpoint Firewall-1 then performs a 3-way handshake with the web server on behalf of the Internet client, and then allows traffic to flow between the two hosts. While this does introduce a short delay when initially setting up connections, it is effective in shielding servers from SYN floods.

Although I was not able to overload the web server, I was very successful in overwhelming the Internet connection. With just 10 cable modem hosts sending out ICMP echo requests to several smurf amplifier sites, I was able to generate 12.8 Mbs of ICMP traffic (assuming 10 ICMP echo replies for every ICMP echo request) going to Chong's network. Additionally, the 40 hosts sending TCP SYN packets generated 5.12 Mbs of traffic. This is nearly 18 Mbs of network traffic in total. Since Chong is using a single T1 with 1.544 Mbs of capacity, I have generated vastly more traffic than his connection can handle.

How long is Chong knocked off the Internet by my attack? To determine this, I will have to estimate how long it will take Chong to filter the traffic from my attack. When he first realizes that he is under attack, he will likely check his firewall logs to see what he is being hit with. Here is where one weakness of Chong's network shows through. He has not set his Checkpoint Firewall-1 to rotate its log files. The volumes of traffic that I am sending at his network will cause his log files to grow very quickly. The Checkpoint Firewall-1 log viewer can have performance issues with large log files, and most people choose to use the built-in functionality of Checkpoint Firewall-1 to rotate the log files.

Despite being slowed, I would estimate that Chong would be able to realize that he is seeing a SYN flood very quickly. The only problem is determining a signature to filter the legitimate traffic from the malicious traffic. My estimate on recognizing the attack, determining a unique signature to the traffic, and working with his ISP to block the traffic will take at least one hour.

After filtering the TCP SYN traffic, Chong will still be under attack. Since he has configured his border router to block ICMP traffic, Chong will not have seen the ICPM packets in his firewall logs. He will not know that he is on the receiving end of a smurf amplification attack until he checks the logs on his border router.

Logging in to the router remotely may be problematic as the large volume of traffic may cause high CPU utilization on the router. Whether Chong is able to log on remotely or locally, it will be pretty obvious that there is a large volume of ICMP traffic being sent at his network. At this point, it should only take 30 minutes or less to again contact his ISP and have ICMP traffic blocked upstream.

In total, I knocked Chong off of the Internet for (a conservative estimate of) 1.5 hours. Although it would be nice to point out a flaw with Chong's network design at this point, in reality there was no flaw that was exploited. This is what makes the new DDOS tools so dangerous. There is really nothing you can do short of having your ISP filter the offending traffic, but you probably won't be able to do that until you are already under attack and have an idea about what traffic to filter.

4.4 Internal Host Compromise

For my final attack, I will compromise a host inside Chong's network. I have chosen to compromise his web server. The reason for this is that it is stated in his paper that the "…web server is powered by Navigator engine 3.6". Since I am not clear what software package Chong refers to by "Navigator engine 3.6", I will assume that he is running Netscape Enterprise Server version 3.6. I would normally have correlated to a software package with a version that more closely matched his version statement, but it appears that all available Netscape products are currently at versions in the 6.x – 7.x range (Netscape Enterprise

Server 6.1 SP3 at http://enterprise.netscape.com/docs/enterprise/index.html and Netscape Directory Server 6.11 at http://enterprise.netscape.com/docs/directory/index.html).

4.4.1 Research

Identifying the address of Chong's web server does not present a problem. A simple nslookup or dig query will provide the address for the server since it is a publicly available host. However, I would like to also identify what software is in use. Normally, I would launch a scan using a scanner such as Nessus or by simply grabbing the banner from the web server, since he has not modified the banner of his web server.

Today, however, I would like to avoid sending traffic to Chong's network directly from mine until the last minute. Instead, I visit the website http://www.netcraft.com/. This site allows you to scan a remote web server using the Netcraft site. It will report on OS version, web server package and version, and many other interesting details. From Netcraft, I find that Chong's web server is running Netscape Enterprise Server version 3.6.

Searching the Security Focus vulnerability archives at http://www.securityfocus.com/bid/vendor/, I find a number of vulnerabilities listed for Netscape Enterprise Server 3.6. Two recently disclosed vulnerabilities jump out at me: a buffer overflow vulnerability (http://www.securityfocus.com/bid/6792/info/) and a file/directory disclosure vulnerability (http://www.securityfocus.com/bid/7621/info/).

4.4.2 Execution

At this point, it should be easy to compromise the web server. I will first try the file/directory disclosure vulnerability. By simply access the target web server using http://www.example.com/?PageServices, the server should list the contents of the web root directory. In many cases, web developers leave sample code or other files in the web root that are potentially dangerous.

If I cannot compromise the server in this manner, I will use the buffer overflow vulnerability. Sample exploit code is provided in the advisory, but it only crashes the web server. Testing will have to be done to determine what memory gets overwritten when the web services crash and what executable code can be inserted to give me control of the machine.

4.4.3 Analysis

There was little that could be done to protect this particular web server. Everything that happened during the attack would have gone through the firewall

since it looked just like normal HTTP traffic on port 80. The IDS may have alerted on this attack, but not until after the damage had been done.

The only real solution in this case is to make sure that the web server you are running is the most current version and is updated with all available patches. Netscape Enterprise Server 6.1 SP3 would be a likely candidate for this installation.

5 References

Internet Assigned Numbers Authority "INTERNET PROTOCOL V4 ADDRESS SPACE" 04/04/2003 http://www.iana.org/assignments/ipv4-address-space 06/20/2003

Center for Internet Security 2003 http://www.cisecurity.org/ 05/8/2003

Qmail mirror site 05/31/2003 http://www.qmail.org/ 05/31/2003

Internet Software Consortium "ISC Bind" 2003 http://www.isc.org/products/BIND/ 05/18/2003

End Run Technologies "Stratum 1 NTP Server" 2002 http://www.endruntechnologies.com/ntp-server.htm 05/05/2003

Snort.org "Snort – The Open Source Network Intrusion Detection System" 05/21/2003 http://www.snort.org/ 05/23/2003

Netfilter.org "Netfilter – firewalling, NAT, and packet mangling for Linux 2.4" 05/07/2003 http://www.netfilter.org 05/15/2003

Network Associates "Groupshield for Exchange 2000" 2003 http://www.networkassociates.com/us/products/mcafee/antivirus/email/gs_exchange2000.htm 05/08/2003

Patchlink "Patchlink Update – The Standard in Patch Management" 2002 http://www.patchlink.com/products/emanagement_services/patchlink_update.html 05/12/2003

NTP.org "Home of the Network Time Protocol" 06/08/2003 http://www.ntp.org/ 06/18/2003

BalaBit "Security Begins When You Know What is Happening NOW" 2003 http://www.balabit.com/products/syslog_ng/ 06/23/2003

Swatch "The Simple WATCHer of Logfiles" 2003 http://swatch.sourceforge.net/ 06/09/2003

Network Associates "VirusScan Enterprise" 2003 http://www.networkassociates.com/us/products/mcafee/antivirus/email/vs.htm 05/08/2003

Network Associates “ePolicy Orchestrator” 2003 http://www.networkassociates.com/us/products/mcafee/antivirus/fileserver/epo.htm 05/08/2003

Network Associates “Desktop Firewall 2003 http://www.networkassociates.com/us/products/mcafee/antivirus/desktop/desktop_firewall.htm 05/08/2003

HP “HP Storage Works” 2003 http://h18006.www1.hp.com/storage/entrystorage.html 06/20/2003

National Security Agency “Router Security Configuration Guide” 9/27/2002 http://www.nsa.gov/snac/cisco/guides/cis-2.pdf 06/09/2003

Cisco Systems “Cisco IOS Release 12.1” 2003 http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/index.htm 6/18/2003

Y. Rekhter et. al. “RFC 1918” 02/1996 http://www.faqs.org/rfcs/rfc1918.html 06/02/2003

Internet Assigned Numbers Authority “INTERNET PROTOCOL V4 ADDRESS SPACE” 04/05/2003 http://www.iana.org/assignments/ipv4-address-space 06/15/2003

Internet Assigned Numbers Authority “PORT NUMBERS” 06/15/2003 http://www.iana.org/assignments/port-numbers 06/15/2003

Cisco Systems “Configuring the Cisco VPN 3000 Concentrator in a Site-to-Site IPSec VPN with Overlapping Private Networks” 10/30/2002 http://www.cisco.com/warp/public/471/config_vpn_3k_site.html 06/18/2003

Lance Spitzner “Auditing Your Firewall Setup” 12/12/2000 http://www.spitzner.net/audit.html 06/20/2003

Fydor “Nmap” 05/03/2003 http://www.insecure.org/nmap 06/17/2003

Nmapwin.org “Project: Nmapwin: Summary” 06/14/2003 http://www.nmapwin.org/ 06/18/2003

Nmap.org “Index of /nmap/dist” 06/15/2003 http://download.insecure.org/nmap/dist/?M=D 06/15/2003

Tcpdump.org “tcpdump/libcap” 02/28/2003 http://www.tcpdump.org/ 06/16/2003

Windump.polito.it “Windump: tcpdump for Windows” 08/08/2002 http://windump.polito.it/ 06/05/2003

Chong Kasing “SANS GCFW Practical Assignment v1.8” 02/08/2003 http://www.giac.org/practical/GCFW/Chong_KahSing_GCFW.pdf 06/05/2003
Internet Security Systems “Check Point FireWall-1 port 264 Denial of Service” 07/07/2000 http://www.iss.net/security_center/static/7368.php 06/10/2003

Powertech “Smurf Amplifier Registry (SAR)” 06/03/2003 http://www.powertech.no/smurf/ 06/03/2003

America Online “Netscape Enterprise Server” 2003 http://enterprise.netscape.com/docs/enterprise/index.html 06/08/2003

America Online “Netscape Directory Server” 2003 http://enterprise.netscape.com/docs/directory/index.html 06/08/2003

Netcraft 2003 http://www.netcraft.com/ 06/12/2003

Security Focus “Netscape Enterprise Server HTTP Method Name Buffer Overflow Vulnerability” 02/07/2003 http://www.securityfocus.com/bid/6792/info/ 06/05/2003

Security Focus “Netscape Enterprise Server PageServices Information Disclosure Vulnerability” 05/16/2003 http://www.securityfocus.com/bid/7621/info/ 06/05/2003

Matt Briddell “A Comprehensive Perimeter Security Architecture for GIAC Enterprises” 4/12/2002 http://www.giac.org/practical/Matt_Briddell_GCFW.zip 05/03/2003

Emily Gladstone “SANS GCFW Practical Assignment” 4/30/2002 http://www.giac.org/practical/Emily_Gladstone_GCFW.zip 05/03/2003

Toby Kohlenburg “A design and review of a firewall infrastructure for GIAC Enterprises” 11/30/2002 http://www.giac.org/practical/Toby_Kohlenberg_GCFW.zip 05/03/2003

Matt Dubinsky “GIAC Certified Firewall Analyst Practical Assignment” 12/30/2002 http://www.giac.org/practical/GCFW/Mark_Dubinsky_GCFW.pdf 05/03/2003