GIAC Certified Firewall Analyst (GCFW) Practical Assignment
Version 1.8

Ken Rode
Submitted March 23, 2003

Table of Contents

Abstract

This paper is offered to meet the requirements of the GCFW Practical assignment version 1.8. The assignment consists of four parts; Designing the Security Architecture, providing a Security Policy and Tutorial, verifying the firewall policy and "Design under Fire" (research and design attacks against a network design). The Security Architecture is addressed under the guise of a consultant brought in to address security issues at GIAC Enterprises. The remaining assignments are covered as separate sections.

Security Architecture Proposal (Assignment 1)

For: GIAC Enterprises, Inc.

Prepared by: Kenneth Rode
February 21, 2003

Introduction

GIAC Enterprises (GENT) is a startup company focused on the sale of fortunes to the fortune cookie industry. They were spun off from a larger graphics firm when it was decided that the fortune business was no longer profitable for that owner. The company's financial situation is currently stable but does not afford the luxury of significant capital investment.

Basic infrastructure equipment was provided as part of the spin off arrangement and GENT has been self sufficient for several months. However, several recent incidents, including defacement of their public web site, prompted GENT management to pursue outside assistance for security improvements. The following document offers a review of GENT business needs and detailed system designs to provide the required level of security within stated cost constraints.

Background

The existing GENT infrastructure consists of the basic elements required for operation but, as evidenced by the defacement, does not provide sufficient control for reasonably secure operation. Figure one depicts the existing infrastructure.

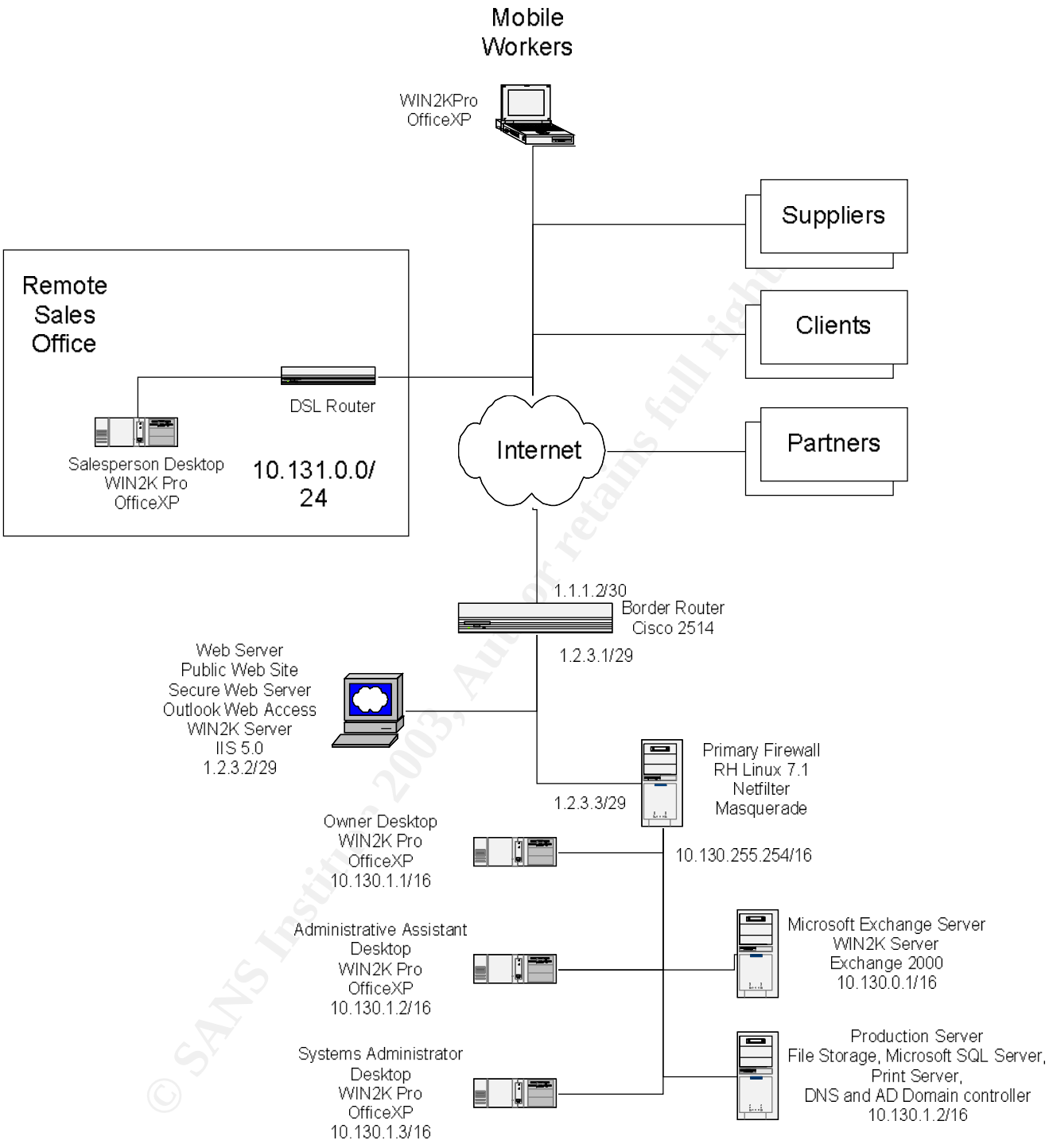


Figure 1

This network design is typical for many small businesses and does afford GENT all the functionality they need. Unfortunately, it does not sufficiently limit and control system access leaving GENT open to attack and compromise. Some of the most significant issues are;

1.) There are no specific protections installed for the remote office and this system is allowed terminal service access into the Production server. Compromise of the Remote Sales office will offer a ready path into the heart of the GENT network.
2.) No network protection is provided to the external Windows 2000 server. Windows was not designed for open operation and the number of vulnerabilities inherent in unprotected Windows servers is astronomical. Without constant patching and the addition of filtering, this machine is sure to be compromised again. Running Outlook Web Access on this machine simply adds to the number of potential problems.
3.) SMTP connections are opened directly to the internal Exchange server. While this may be preferred over exposing an Exchange server to the public network, there are improvements to this design that could greatly increase security.
4.) The use of Terminal Services for remote access is also a reasonable method to provide access to remote users. It can, however, be strengthened as we will show a bit later.

As we address these issues, there are many smaller adjustments and authentications that can be implemented to further deepen the GENT security posture. These will be addressed at the appropriate points in this proposal.

Each of the changes proposed must be designed in such a way that core business operations may continue unimpeded. To ensure this happens, a needs assessment for each business interest is provided below.

Business Operations

As mentioned during the Introduction, GENT sells fortunes to Fortune Cookie manufacturers. From meetings with the GENT management team we have come to learn that this is a commodity business. This mean that prices for the fortunes are essentially fixed and producers can only differentiate themselves through personal rapport with the client and exemplary service. The guiding principal for all of GENT's operations is to ensure every client gets the exact order they expect when needed.

Internal Users

Employees located at the main office have very basic system needs. They require full access to internal systems, access to basic Internet services (HTTP, HTTPS and FTP) and little else. DNS is provided by the ISP and all systems query these external servers directly. Email communications are provided via a Microsoft Exchange server and use of personal email accounts is discouraged. A review of existing firewall logs indicates that restricting access to the listed services will impact neither normal operations nor current user expectations.

Mobile Users

GENT mobile users are currently limited to one salesperson and the Owner. When traveling, these two users require access to company email, general network files and internal applications to run reports against production databases. Historically this has been offered via terminal services on the Production server and Outlook Web Access (OWA) from the Internet Server. The services offered to mobile users will be sustained while security is enhanced.

Remote Sales Office

When not traveling, the salesperson requires full access to internal systems from the remote sales office. The salesperson also requires the same basic Internet access protocols as internal users at the main office. All this has been offered via SDSL, terminal services and OWA. As with mobile users, the needs met by these services will be supported and better secured.

Suppliers

Fortune suppliers must be able to provide their products in a manner that verifies both the sender and the content. If errata are introduced into the product (i.e. an offensive fortune or gibberish) it may not be detected prior to receipt and use at a client. Such an occurrence has been known to end long-term relationships.

At present, all files are transferred via standard email. Once received, each file is automatically scanned for a list of offensive words and phrases prior to incorporation within the main GENT database. However, added checks to ensure the identity of the sender and the integrity of the file are warranted.

Clients

GENT Client contracts typically are renewed yearly and provide for regular product deliveries throughout the year. The clients' needs are simply to receive reliable data on the contracted schedule. Their main concern is to ensure that each fortune meets the needs of their consumers. Typically the files transmitted contain thousands of fortunes and they are fed directly into production systems. Consequently, an off-color, offensive or nonsensical fortune could make it into a cookie and result in hundreds of consumer complaints. Such an occurrence can be damaging to the cookie manufacturer and prompt them to seek a new fortune supplier.

The current procedure involves manually publishing data from the main GENT database to the secure web site once per month. This is working well and will not be altered beyond the addition of some rather transparent checks and balances.

Partners

All of GENT's fortunes are provided in English and the majority of GENT's dealings are with companies producing Fortune cookies in English speaking countries. However, on occasion, GENT will receive a request for products in another language. These requests are handled, for a premium price, by directing product delivery through one of several Business Partners. The general process is to send the completed order (in English) to one of several vetted partners for translation, verification of the results and return for final shipment to the client. As GENT is unable to personally verify the quality of the product after translation, partners must be approved prior to their first deal with GENT. This vetting is renewed yearly.

Partner communications have historically been handled via standard email. We must add checks to this process to ensure the identity of the sender and the integrity of the files. This is critical for communications both to and from business partners.

Summary of Operations

Figure 2 below provides a general summary of existing communication paths. As mentioned earlier, the most significant concerns are the number of unrestricted paths to the Internet and the fact that many communications and file transfers operate in a clear text format.

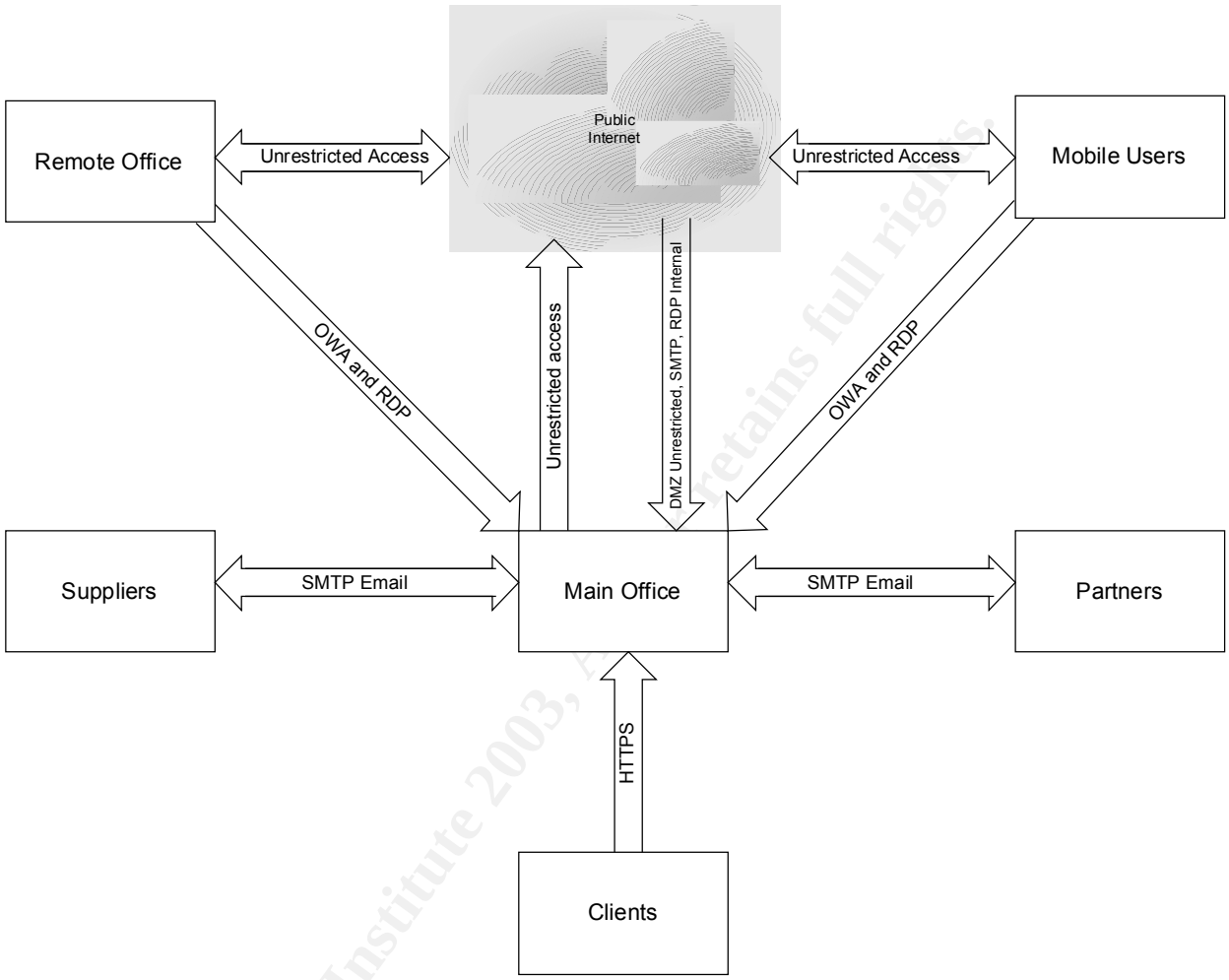


Figure 2

Budget Considerations

With a clear understanding of the security concerns and business requirements, the final item for immediate consideration is project cost. From discussions with GENT management, it has been learned that the inherent value of the fortunes themselves is rather insignificant. Loss of their entire database would not endanger the viability of the company. Even returning to the days of sending data to clients on CD ROM (for a brief period of time) would not be devastating. The focus of the current effort is to ensure the integrity of the data when it arrives at the client's facility and provide reasonable protection within a cost target that does not severely impact current GENT profitability figures. The general range provided by GENT management is $4,000 to $6,000.

New System Design

The new system design reuses all components in the existing GENT Network and minimizes additions as much as possible. Further, most business processes will continue without noticeable change. For example, the remote sales office will continue to have access to email and terminal services at the main office, these communications will simply now be encrypted and authenticated through the IPSEC tunnel. Our goal is to continue offering everything required for GENT personnel to do their jobs while limiting system access to only the required personnel and services. The resulting network design is shown in Figure two below.

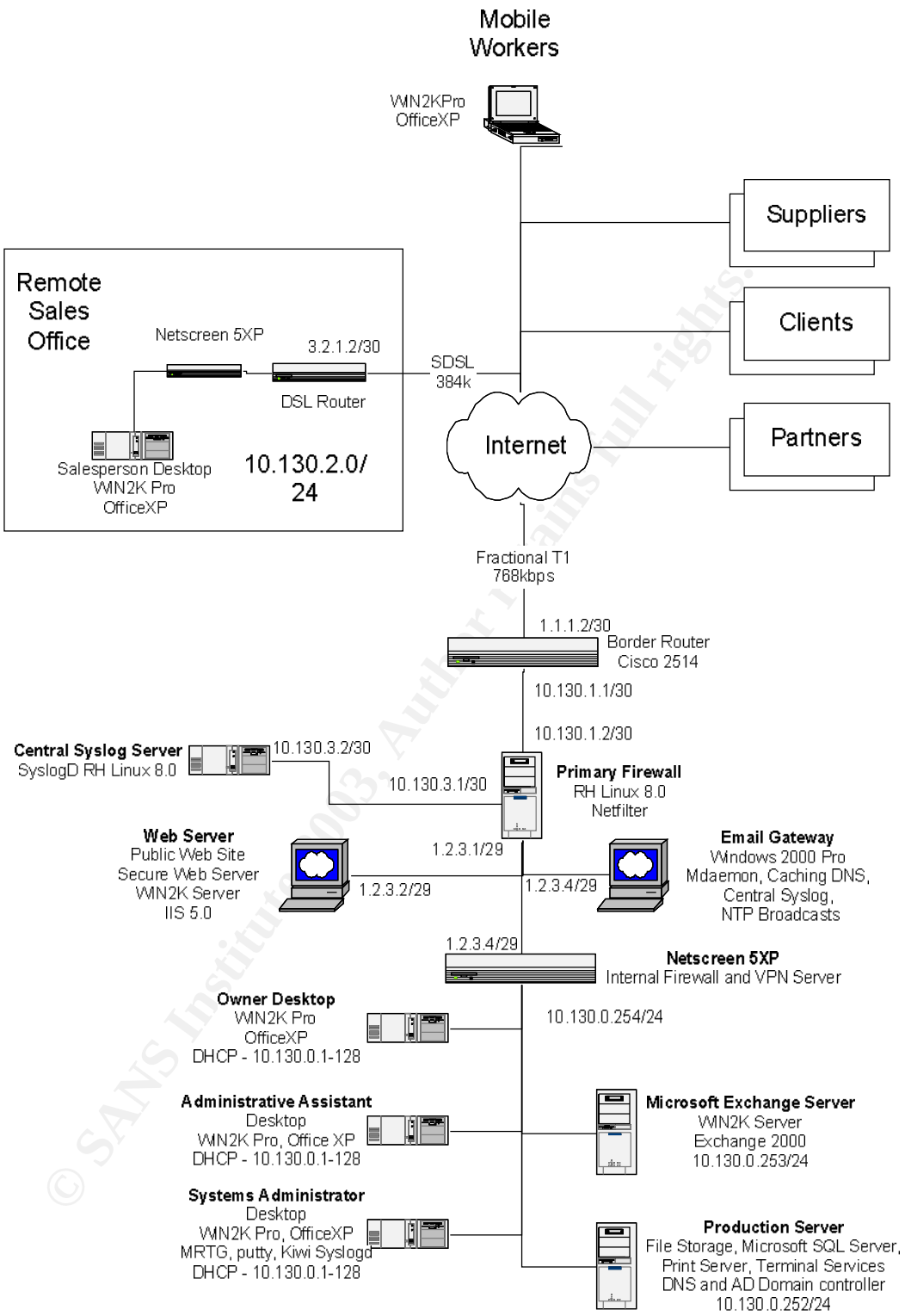


Figure 3

This design introduces several new components and significant changes to the configuration of others. Each of the changes will be explained in turn beginning with the IP Addressing scheme.

IP Addressing

The public IP addresses are static assignments leased from the various ISP's. As shown in table 1, the Main GENT offices have 6 available public IP addresses (1.2.3.1 to 1.2.3.6). Router communications with the ISP are controlled through IP blocks with only two hosts (ISP router and Client router). These subnets are shown in rows 2 and 3 of table 1. Client, supplier and partner addresses are shown for completeness. Assignment of these numbers is out of our control.

The Internal network at the main office has been compressed to a Class C address space (maximum of 254 hosts). This is still many more than will be required in the foreseeable future and affords clear delineation between DHCP assigned and static address ranges. The existing DHCP server will be modified to provide leases for addresses in the range 10.130.0.1 through 10.130.0.126. The upper end of the range will be reserved for static assignments to servers and general network devices.

As traffic from the Internal network passes through the internal firewall it is translated to appear to come from the firewall interface (1.2.3.4). With only limited free IP addresses (currently two) this is the only practical approach. Should one or two systems require Network Address Translation (NAT) at some future date, can be provided.

As no servers or other network components exist in the DMZ (the segment between the Primary firewall and the Border Router) this section is assigned a /30 address space (2 hosts).

The remote office also uses a private Class C address space for simplicity. To enable routing through the VPN it is critical for this office to reside on a different subnet from the main office. As shown in the table, this is the case.

Table 1 – IP Addressing scheme

Network	Type	IP Block	Gateway
Main – Services	Public	1.2.3.0/29	1.2.3.1
Main – ISP	Public	1.1.1.0/30	1.1.1.1
Remote - ISP	Public	3.2.1.0/30	3.2.1.2
Client1	Public	1.1.5.0/24	1.1.5.1
Client2	Public	1.1.6.0/24	1.1.6.1
Supplier	Public	1.1.7.12/30	1.1.5.13
Partner	Public	1.1.10.0/24	1.1.10.1
Main – Internal	Private	10.130.0.0/24	10.130.0.254
Main – DMZ	Private	10.130.1.0/30	10.130.1.1
Remote – Internal	Private	10.130.2.0/24	10.130.2.254

Main Office

Border Router (Cisco 2514 running IOS 12.2(1)) – The existing Cisco Router has years of usable service left in it. However, it does have a limited processor and low memory. Further, secure remote administration is not available from the router itself (ssh is not supported). Therefore, replacing this device with a more modern and robust model is warranted, when budgeting allows. At this time we

will mitigate the performance issues by limiting filtering to IP address only and mitigate Telnet security issues by only allowing administration from the Systems Administrator's workstation.

Primary Firewall (Compaq DL320 running RedHat 8.0 – 2.4.18-26.8.0) – The existing firewall will be relocated to a position just inside the Border router and configured to control all traffic to and from GENT systems. As the Border router offers only basic filtering, no systems will be deployed within the DMZ. If a need arises for a system on this segment, the Border router must be replaced with one that will allow a more robust rulebase and secure remote management.

The Primary Firewall will be configured to allow specific traffic and deny everything else. This basic tenet of good firewall design will be used throughout the GENT network. Any deviations will be noted and justified.

One addition to the existing firewall is a third network interface that connects to a central Syslog server. This allows additional protections and isolation for this critical service.

1.) Allow Specific traffic from the Internal Firewall out – Egress rules applied to the Internal firewall will be mirrored here. This can introduce excessive complexity and inefficiency in large environments but will be manageable with the limited rulebase required in this instance. Specific details of the traffic allowed are provided within the Internal firewall discussion below.
2.) Allow Tunneling traffic to the Internal Firewall – Both site-to-site and client tunnels must be able to connect to the Internal firewall.
3.) Allow SMTP to/from the Email Gateway – This is the only system allowed to send or receive SMTP over external networks.
4.) Allow DNS from the Email Gateway (caching name server) to the ISP DNS Servers – DNS queries are strictly controlled within this design. Internal systems query the Internal DNS server, which forwards requests to the Email gateway for resolution. Only the email gateway is allowed to query external DNS servers and it is limited to contacting specific servers maintained by the ISP.
5.) Allow HTTP to the Web Server for the public web site – Incoming requests to the public web site must be allowed.
6.) Allow HTTPS from specific IP addresses (or address blocks) to the Web Server secure site – the secure site is provided for the sole use of specific clients. Therefore, no other Internet hosts are allowed incoming https access.
7.) Allow NTP traffic from the Email Gateway – This server provides NTP broadcasts to the rest of the systems on the network and needs to synchronize with external NTP servers to ensure accuracy.
8.) Allow Syslog traffic to the central syslog server from the appropriate security devices.
9.) Allow SSH from the Internal Firewall to the central syslog server for administrative management and monitoring.
10.) Allow Telnet from the Internal firewall to the Border router – This is required for administration by the Systems Administrator.
11.) Deny everything else

Central Syslog Server (Compaq Deskpro EP running RedHat Linux 8.0) – This is a very limited server acting as a central repository for syslog information from all GENT security devices. The only traffic allowed is Syslog messages from the various devices and ssh from the Internal firewall.

Web Server (Compaq Proliant DL380 running Windows 2000 and IIS5.0)– There is no plan to change the operation of the web server. The existing public and SSL enabled sites are working reliably and should continue to do so without immediate intervention. However, to prevent another compromise, we will be applying all current Microsoft patches and the Systems Administrator has agreed to implement a weekly schedule of patch updates. Further we will be implementing the Microsoft URLScan and IIS Lockdown tools to armor the web server. Finally, Outlook web access will be removed. Access to Microsoft Exchange will be provided via VPN connections directly to the Exchange server.

Email Gateway (Compaq Deskpro EN running Windows 2000 pro and Mdaemon 6.7.2) – A new addition is the email gateway. This is a Windows 2000 Pro system running MDaemon to isolate the Internal Microsoft Exchange server from the Internet. The gateway solution also offers SPAM filtering though the use of RBL services and the ability to scan and verify messages before they reach the internal Exchange server. To meet current cost targets, the scanning feature is not implemented within this proposal. This server will also offer NTP broadcasts to the service network and cache/forward DNS requests. It is typically preferable to isolate functions as much as possible but, in the interest of meeting cost targets, this combination is acceptable in this instance.

Internal Firewall (Netscreen 5XP running ScreenOS 4.0.0: r11) – The new Netscreen 5XP is recommended to replace the relocated Linux firewall. This device will provide the performance required and can offer VPN connectivity to mobile and remote users. Currently, GENT maintains a fairly lax security posture regarding Internet access by employees. While this will merit a reexamination as the firm grows, the firewall policies outlined below are designed to maintain all current services. In the policies below, each service is provided to all internal users except where noted.

1.) Allow HTTP and HTTPS from all Desktop systems out – GENT policies do not restrict web browsing by employees. To enable this, http and https traffic must be allowed out.
2.) Allow FTP from all Desktop systems and out – GENT policies also allow employees to download information from the Internet in the pursuit of business and personal interests. Web site updates are also applied using FTP transfers from the Administrative Assistant's workstation to the Web Server.
3.) Accept incoming VPN connections – There are currently no restrictions on traffic within the VPN tunnels. Once the current round of changes is complete, an analysis of this traffic to enable further restrictions is warranted.
4.) Allow ICMP from the Systems Administrator – The administrator must be able to verify system availability and connectivity using standard tools.
5.) Allow Telnet traffic from the Systems Administrator to the Border Router – The Border Router may only be administered from the System Administrator's workstation and requires Telnet.
6.) Allow SSH traffic from the Systems Administrator – The Administrator must be able to access the Netscreen and Central Syslog systems using ssh.
7.) Allow SMTP to/from the Email Gateway

8.) Allow DNS forwards from the internal DNS Server to the Email Gateway (DNS cache server) – Internal systems query the Internal DNS server, which must be able to forward external requests to the DNS cache server for resolution or continued forwarding to the ISP servers.
9.) Allow SNMP traffic to the System Administrator's workstation – SNMP is required to monitor bandwidth usage using MRTG.
10.) Deny everything else

Internal Network – Specific changes to the Internal network are limited to installing Anti-virus software. Sophos Antivirus will be added to the internal network and the remote sales office. This application is known to perform well and has been shown to have a minimal impact on system and network performance (http://www.infoworld.com/article/02/03/01/020304neantivirus_1.html?Template=/storypages/ctozone_story.html). While traveling, mobile users are required to connect to the Main Office network (via the VPN) a minimum of weekly to receive updates.

System Administrator's workstation – This machine is a key component of the new infrastructure. It is used to monitor the central log server and is the only system allowed management access to all security devices. It also monitors bandwidth usage through the router and both Netscreen firewalls using MRTG and SNMP.

While SNMP can introduce security concerns, the ability to monitor bandwidth usage at the various chokepoints is a critical security tool and worth the mitigated security risk. By monitoring bandwidth it is possible to discover a system compromise that might otherwise go undetected to the untrained eye. For example a large spike in bandwidth at 2am on a Sunday would clearly stand out from normal traffic patterns. A stealthy compromise of an ftp service to allow the uploading of Warez and other materials may be overlooked. Limiting access to the System Administrator's workstation and connecting to the remote Netscreen through the VPN tunnel offer mitigation of SNMP's insecurity.

Remote Sales Office

The main addition at the remote sales office is the new Netscreen 5XP firewall appliance (ScreenOS 4.0.0: r11). This is in place to protect the sales office from unwanted incoming traffic and to provide a network VPN connection into the main office. Limitations are similar to the internal firewall as shown below

1.) Allow HTTP and HTTPS out – GENT policies do not restrict web browsing by employees. To enable this, http and https traffic must be allowed out.
2.) Allow FTP out – GENT policies also allow employees to download information from the Internet in the pursuit of business and personal interests.
3.) Accept incoming VPN connections – There are currently no restrictions on traffic within the VPN tunnels. Once the current round of changes is complete, and analysis of this traffic to enable further restrictions is warranted.
4.) Allow SSH traffic from the Systems Administrator – The Administrator must be able to access the Netscreen systems through ssh.
5.) Allow DNS queries to the ISP DNS servers – restricting DNS queries to the tunnel or Email Gateway would be too inefficient and not offer a reasonable increase in security.
6.) Allow SNMP traffic to the System Administrator's workstation – SNMP is required to monitor bandwidth usage using MRTG.
7.) Deny everything else

Mobile Workers

Company laptops are protected with a personal firewall from Zone Labs (Zone Alarm), Sophos Anti-Virus (which must be updated weekly) and the Netscreen VPN client. Using the VPN client, mobile workers will have access to all internal systems, including terminal services on the production server.

Suppliers

Financial dealings with suppliers are handled through printed checks and postal mail. The only electronic communications are the sending of orders and the receipt of products. Historically this has been handled via email. To improve the security of this data transfer without redesigning the processes that have been developed, our proposal is to add PGP signatures and encryption to each message. With the limited number of suppliers and GENT personnel involved, this solution will provide a drastic increase in security for a minimal training investment.

Partners

Partner communications have also historically been via email. As with the Suppliers, the number of companies involved is limited and adding PGP to the existing process is the most viable option. Our recommendation is that this change be implemented immediately.

Clients

Clients typically contract for service yearly. During the term of the contract, clients are able to download the fortunes they require, at will, for a set monthly fee. An SSL enabled web site was originally developed for this purpose and continues to meet the need reliably. Data is currently pushed from the internal production server to the web site monthly. The only process change we propose, beyond weekly patch updates, is to refresh the data on the Web server every four hours rather than once a month. This will be an easy task to automate and can help guarantee that any data corruption is quickly resolved via a data refresh.

Summary of Services

Table 2 is provides a detailed list of the services offered at the main office, the direction of the initial traffic (from perspective of the primary firewall) and notes regarding the implementation. This is followed by Figure 4, which depicts the new communications paths.

Table 2 – Main Office Services

User	Service	Direction	Notes
Main – Internal, typical	FTP (TCP 20, 21) HTTP (TCP 80) HTTPS (TCP 443)	Outgoing	DNS and Email provided internally. Entering a request, supported with information regarding the business need, can open other protocols.
Main – Internal, Administrator	FTP (TCP 20, 21) HTTP (TCP 80) HTTPS (TCP 443) SSH (TCP 22) ICMP (Protocol 1) SNMP (UDP 161) TELNET (TCP 23)	Outgoing	As with typical users, further protocols may be opened for the Administrator if the business case can be made.

Mobil Users	IPSEC (Protocol 50, UDP 500)	Incoming	Currently there are no restrictions on the tunnel.
Remote Sales Office	HTTPS (TCP 443) VPN	Incoming	The remote sales office has access to the secure web site in addition to an unrestricted VPN.
Suppliers	None	None	All communications are via PGP encrypted email
Clients	HTTPS (TCP 443)	Incoming	All file transfers are from the secure web site. General communications are via standard email though PGP encryption can be offered as an option.
Partners	None	None	All communications are via PGP encrypted email
Service Network	DNS (TCP and UDP 53)	In/Out	Incoming requests are from the internal DNS server. Outgoing requests are forwards to the ISP DNS servers. Public DNS is not hosted on any internal servers.
	EMAIL (TCP 25)	In/Out	External email is received at the email gateway in the service network. Outgoing SMTP mail flows from the internal Exchange server to the gateway and then on to its final destination.
	HTTP (TCP 80)	Incoming	HTTP requests are allowed from anywhere to the public web server.
	HTTPS (TCP 443)	Incoming	HTTPS requests are allowed from approved addresses to the private web server
	NTP (TCP and UDP 123)	Outgoing	NTP from the Email server is allowed
	IPSEC (Protocol 50, UDP 500)	Incoming	IPSEC connections are allowed to the Internal firewall

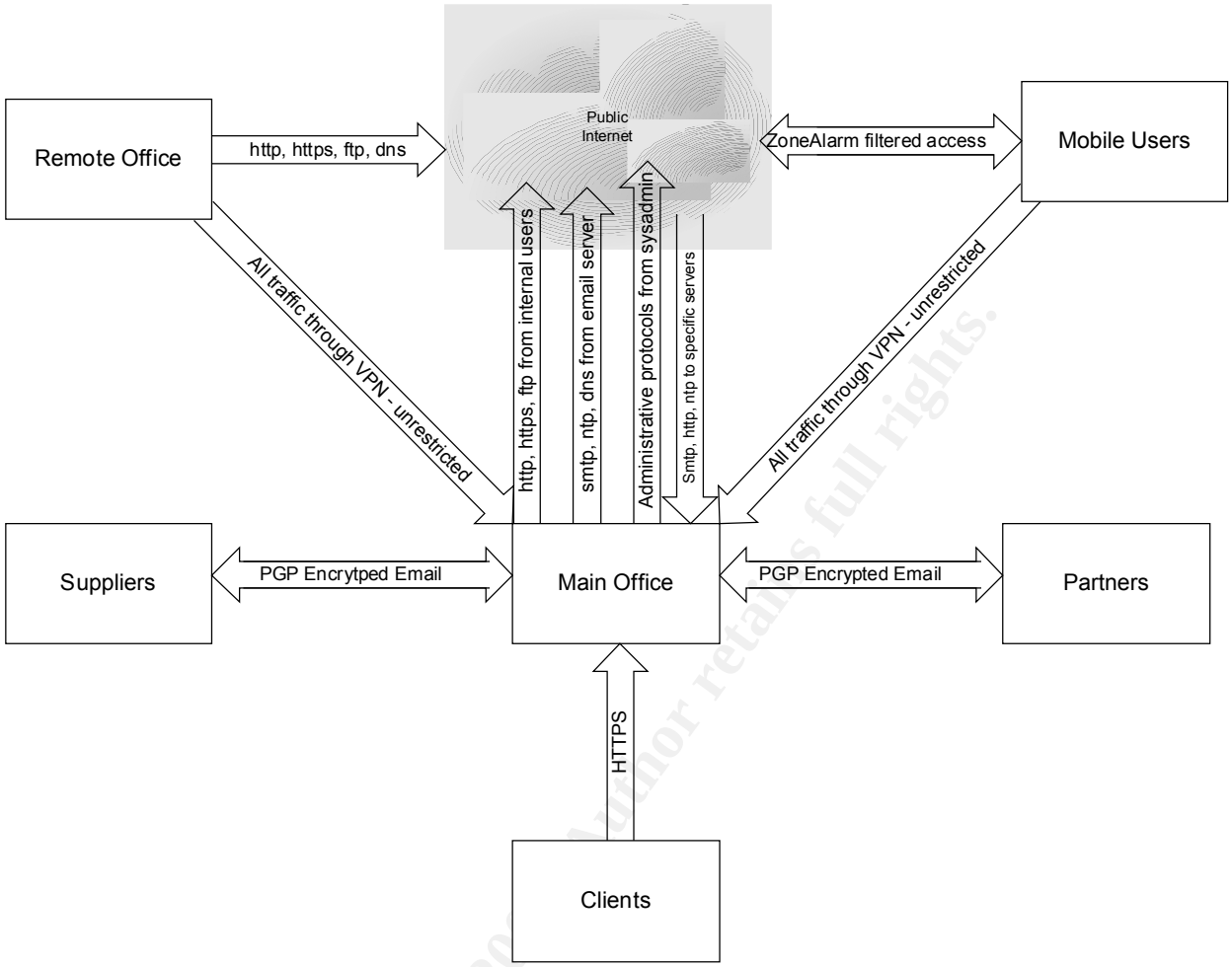


Figure 4

A Look Ahead

Several more items could have been included in this proposal but were omitted for various reasons. I would like to take this opportunity to review two of these and offer some guidance for the future.

Intrusion Detection Systems

Intrusion detection systems are an important part of a secure environment and would not be prohibitively expensive to incorporate into this network. However, without close management and monitoring, an IDS is not really worth the trouble. Once the GENT Administrator completes GSEC training, it is highly recommended this employee head up the design and installation of several SNORT sensors. Minimally placing sensors on the Services network, directly inside the internal firewall and at the Remote site will provide a good view of network operations. Of course, assistance for this project can be provided upon request.

Business Continuity Planning

Another critical item to address is business continuity planning. Due to budget constraints, there is no redundancy and only limited fault tolerance in most GENT systems. This is understandable in current circumstances but needs to be mitigated wherever possible. To achieve this, it is strongly recommended that a planning team be selected and the process commenced as soon as possible. We

have trained engineers to facilitate this process with an internal team and would be happy to provide a proposal on that project as well. Just remember, power outages and hardware failure can be as devastating to an unprepared company as a security incident.

Project Costs

When originally approached to design and install security improvements for GENT, cost was highlighted as a major consideration. It was clearly explained that the current client base only affords sustenance income and extensive protective measures would be neither cost effective nor actionable. Therefore, our design approach focuses on adding the minimum number of new components required for a reasonable security baseline. As future needs increase and expand, this new hardware can easily be retasked to new less demanding roles such as additional remote or home offices.

The new hardware/software consists of the email gateway, the internal firewall and the remote office firewall. Total cost for these three systems is estimated at $2,500. Configuration and implementation is anticipated to take approximately 10 hours for a total estimated cost of $3,750 plus travel and other miscellaneous expenses. The firewall audits will require approximately 10 man-hours both pre and post installation for an additional fee of approximately $1,250 and a project total of $5,000.

Proposal Summary

The preceding proposal has been offered per a request from GENT for practical security recommendations to decrease exposure and increase the reliability and integrity of data offered to their clients at minimal cost. To achieve this, we focused our recommendations on reusing all of the existing equipment while supplementing it with right-sized devices.

Once completed, the tasks outlined here will result in a much improved security posture. Some of the specific improvements are:

1.) Ingress/Egress filtering on the router reduces instances of IP spoofing and prevents some Denial of Service attacks.
2.) The Linux firewall will be hardened and relocated to a more appropriate position in the network. We will also add a third interface to this system for isolating Syslog traffic.
3.) Outlook Web Access (OWA) will be removed and replaced by VPN access directly to the Exchange Server. If OWA access becomes mission-critical, it can be reinstated at some future date with additional security controls.
4.) The IIS server will be locked down using available tools from Microsoft. Going forward, patches will be consistently maintained to reduce the possibility of compromise.
5.) An email gateway will be added to prevent direct SMTP communications into the internal network and offer some basic SPAM control. In the future, virus and content scanning may be easily added to this system.
6.) A new internal firewall has been added to deepen GENT's security posture and provide a ready means for remote access to internal systems.
7.) Sophos Antivirus has been added to protect the entire enterprise. The central installation files will be updated regularly by the Administrator and rolled out to client systems any time they connect to the internal network (whether locally or remotely).
8.) A new firewall will be installed at the remote sales office to both protect the systems installed there and provide a seamless VPN connection into the main office.

9.) All laptops will be configured with Zone Alarm personal firewalls and the Sophos Intercheck client.
10.) Supplier and Partner email communications will move to PGP encryption to ensure the authenticity and integrity of the materials transferred.
11.) Data will be refreshed to the web server on a more frequent basis to limit the impact of any potential future compromise.

These changes will immediately result in a much more secure environment. However, there is more that can be done. As business increases and requirements demand, further improvements can be implemented while the components specified could easily be integrated into expanded remote facilities and replaced with higher end products.

Security Policy and Tutorial (Assignment 2)

Detailed Security Policies

The detailed security policies are offered as appendices to this document. Links to the specific pages are provided below

Primary Linux Firewall Tutorial

Introduction

A complete tutorial on the installation and configuration of Red Hat Linux is beyond the scope of this document. Instead, we will provide basic guidelines for the Linux installation, a brief review of system hardening using Bastille-Linux and specific details on how to implement the firewall script from Appendix B.

Implementation of the Linux Firewall is reasonably straightforward. We begin by performing a custom installation of Red Hat Linux 8.0. Once the base operating system is in place, we can use Bastille-Linux 2.0.4 to harden the system. This hardening can also be done manually and there are many Internet resources to assist with this task. However, in this instance, using Bastille-Linux simplifies the process and can more quickly provide a reasonably secure system.

Red Hat Linux 8.0 Installation

Our main focus at this time is to install the minimum applications and services to operate and maintain this system as a firewall. Planning to save time or money by creating a firewall/DNS/Email/Web Server is asking for trouble. Ongoing maintenance tasks and the increased possibility of firewall compromise via one of those other servers will greatly overshadow the cost and time associated with multiple installations.

Red Hat has a nice installer that is easy to follow if you read the prompts and explanations offered. Therefore, I will not attempt to provide a step-by-step description of the process. Rather, the discussion below provides suggestions and guidance at key points during the installation.

As we begin, you must choose the user interface used by the installer. As mentioned previously, it is advisable to minimize software and services installed on any externally accessible machine. Therefore, we rarely install a mouse onto firewalls and must use the text mode installer. If desired, the graphical installer may be used (assuming a mouse is available) even if you choose not to install a graphical environment for later use (our recommendation).

The next major question is what packages to install. Again operating on the principle of a minimal installation, we select "Custom" rather than one of Red Hat's prepackaged choices. Continuing on we need to determine the partitioning scheme to use. If you have a large disk and no worries about

running out of log space, feel free to select "Autopartition". If you have limited disk space and/or concerns about the size of your log files, you can manually create the required partitions using Disk Druid. Partitioning for a specific need is dependent upon that need and not part of this discussion. In our case, a simple scheme of 2x physical memory for the swap partition, 98MB for the /boot partition and the remainder assigned to the root partition (/) using type ext3 is acceptable. Incidentally, this is the scheme provided by "Autopartition" for a custom installation.

In the next step, it is recommended to use the GRUB Boot Loader and set a password. While physical access to any machine will ensure eventual compromise, it is still critical to erect all reasonable barriers. Finally, for the boot configuration, install the boot loader in the Master Boot Record. Firewalls should be focused machines dedicated to that one task. Continue the installation by configuring your network cards (static IP addresses are required in this instance).

Next, while it might seem counterintuitive, do not install a default firewall configuration. This task will be managed through the custom scripts we will add later in the process. When prompted, be sure to install strong passwords. This firewall is the first line of defense and should not be weakened by poor or easily guessed passwords. Also do not disable the defaults of md5 and shadow passwords. The guidelines we use for password selection are shown below. Another reliable take on the subject is available at http://www.adpc.purdue.edu/BSC-Pete/ARIBA/passwrds.htm.

1. Dictionary words are never acceptable as passwords. The simplest password-cracking program will decode a dictionary word (or person's name) in under 1 minute. Even the addition of capitalization is no help for dictionary words.
2. Dictionary words plus several random characters are also very weak passwords and considered unacceptable. These types of passwords are easily cracked in less than 10-15 minutes using readily available (and free) tools.
3. The best passwords utilize multiple words with mixed capitalization and non-alphanumeric characters – i.e. This#1Good or use dictionary words with various letters replaced with numbers and punctuation – i.e. 7h!5isbetR. Using phrases also has the advantage of avoiding simple dictionary words while remaining memorable.
4. A random sequence of mixed alphanumeric and punctuation characters is technically the best password. Unfortunately, any advantages are typically offset by the fact that people must keep a written copy handy to remember what the password is.

The big decisions are now upon us with the choice of packages to install. For package groups, deselect everything except "Kernel Development" and "Text-based Internet". Then ensure "Select individual packages" is checked before choosing OK.

In the resulting individual package screen, I make the following changes:

1.) Add vim-enhanced – This is my favorite text editor – feel free to choose your own favorite
2.) Remove fetchmail – not needed
3.) Add lynx, indexhtml and perl-CGI – useful for access web pages
4.) Remove mutt, slrn, and isdn4k-utils – not needed
5.) Add tripwire – helpful to monitor for file changes
6.) Remove mouseconfig, raidtools, postfix and wvdial – not needed

Continue to follow the installation prompts through to completion. Once the system is up and running, the next step is to apply all current updates to the packages installed. The easiest way to complete this is to create a CD ROM containing the latest UPDATES folder for the version of the OS you are using. With this CD mounted, the required updates can be applied using the command "rpm –-freshen *.rpm". The "--freshen" switch tells rpm to only upgrade existing packages. One package I do not recommend upgrading in this manner is the kernel itself. I have never had a positive experience upgrading a kernel through rpm. Instead, I typically compile a new kernel version or recompile the existing version to further cull unnecessary items. As this is not a critical piece of this discussion, it will not be addressed here. Many good articles are available on compiling Linux Kernels. One source is http://www.tldp.org/HOWTO/Kernel-HOWTO-2.html

Once the basic installation is complete, it is good to run the Netstat –an command. While you might expect to see nothing open except ssh, this is frequently not the case. It is typical to see Sendmail (tcp port 25), the Portmapper (tcp port 111) and other services running. If so, it is critical to shut them down before proceeding. Many of these issues must be fixed by uninstalling packages (i.e. - rpm –ev portmap), while others (such as sendmail) can be remedied by adjusting configuration files.

Bastille- Linux Installation

Download files and instructions for Bastille-Linux are available at http://www.bastille-linux.org/. To ensure you do not expose the system to compromise prior to hardening, it is best to download the files from a secure system and transfer then to the new machine using a floppy disk. Please note that as we have not installed a graphical interface, we must use the Perl-Curses interface.

Once the installation is complete, run the application by typing bastille –c from the console prompt. Then follow the system prompts to complete the configuration. Our choices, with reasoning, are offered below.

1.) Accept the terms of the disclaimer to proceed.
2.) Choose yes to further tighten security on system administration utilities. In our installation, there are no normal users to worry about.
3.) As we have no local users, removing SUID status from all the prompted programs is acceptable.
4.) The "r-tools" have not been installed but it is still acceptable to tell Bastille-Linux to disable them.
5.) As with items 2 & 3, password aging is not an issue when using only the root login.
6.) Cron should be limited to admins per reasons cited earlier.
7.) Setting the most restrictive umask is the best choice for security.
8.) In a larger installation, it is frequently useful to use normal user accounts for the primary logon that can then use su for root privileges. In our case that is not required. Therefore we do not need to disable root login on tty's.
9.) While we already enable a GRUB password, it is best to say yes again and enter the same password used during installation. This is to ensure Bastille-Linux does not clear the prior setting.
10.) Our recommended settings provide for physically securing the server, including a Grub password, and protecting single user mode with a password. Disabling a potentially useful command such as CTRL-ALT-Delete is not required. Respond no

11.) Password protecting single user mode is a critical item and in no way limits access within our present situation.
12.) We will manually configure TCP Wrappers later in this process. We will not do it within Bastille.
13.) Disable telnet on the firewall. We use ssh.
14.) Disable ftp, it is not needed.
15.) An "Authorized Use" message is not expected to deter an intruder. It can however deflect an ignorance defense in the event someone does compromise the system and is caught. Select "Yes" for the question and enter your company name when prompted. If necessary, modify the default banner for your needs once the configuration is complete (/etc/issue).
16.) We will not restrict resource usage in this instance. The potential for inadvertent problems due to the restrictions is seen as greater than any benefits from this limitation.
17.) Again, as we have one user (root) restricting console access is a good idea.
18.) At the next prompt, we can increase logging and configure access to our central logging server. This is important and required. When prompted, enter the IP address 10.130.3.2
19.) Disable all the System Daemons as directed by Bastille.
20.) Again do not run the default packet filtering script. We will install custom scripts later.
21.) Finally, apply the changes and press TAB to complete the process.

Manual Configuration

With Bastille complete, we have a fairly secure system but we can still enhance security further.

First, within the /etc/hosts.deny file place the following line;

ALL: ALL

This disables access from all systems to all services.

Next, within the /etc/hosts.allow file place the next line;

Sshd: 1.2.3.4

This allows secure shell (ssh) access to the firewall from internal systems. The internal firewall itself limits this access to a single system.

At this point we have a system reasonably well locked down and ready for the firewall configuration. Before proceeding, reboot the firewall to activate all changes made by Bastille-Linux.

Firewall Script

The entire firewall script is annotated in Appendix B. Therefore, I will not reproduce it here. Rather, tips and suggestions for implementation will be offered alongside a basic tutorial on Netfilter operation and use.

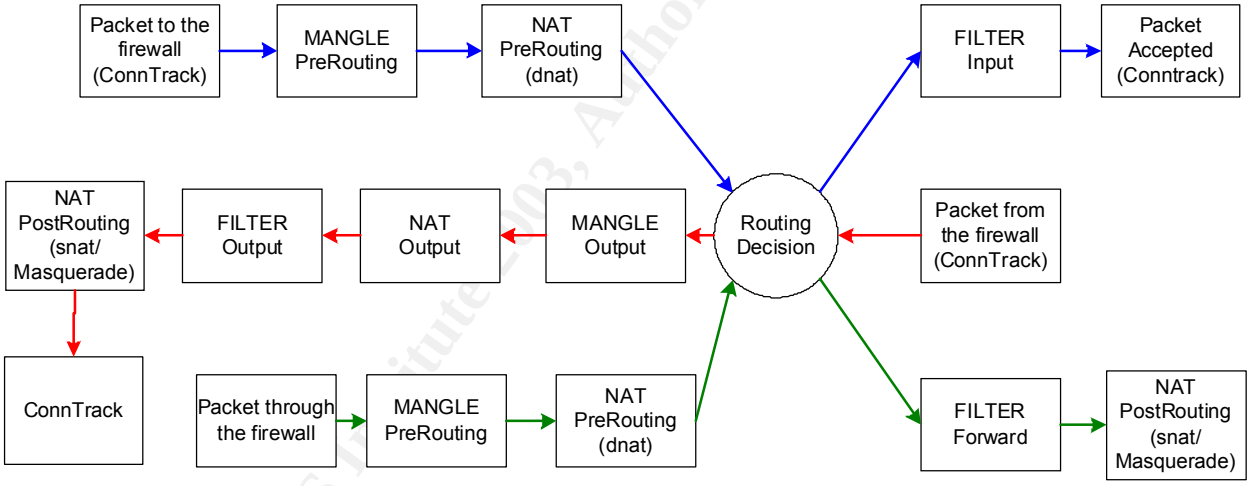
Understanding Tables

Netfilter operates on packets using rules that are ordered within chains that exist within tables. While this may sound a bit confusing at first, it is actual very logical and simpler to follow than the old IPChains flow.

Starting at the top, there are three standard tables in Netfilter. The first, and arguably most important, is the filter table. This is where the bulk of the work is performed and where the majority of our script will do its work. The Filter table contains the Input, Output, Forward and user-defined chains for filtering packets. Next (and also very important) is the Network Address Translation (nat) table. This table maintains information on packets that have been translated by snat (source nat), dnat (destination nat) and masquerade --Masquerade automatically translates packets using the current address for the particular interface the packet is exiting through. It is very useful for situations where an address is assigned dynamically (dial-up or DHCP) but is not as efficient as snat or dnat when static addressing is used--. The third table is mangle. This table is used to alter header fields and is not used within the implementation presented here. The nat and mangle tables also contain Prerouting and Postrouting chains.

Within the Filter table, the Input chain examines packets destined for the firewall itself, the Output chain checks packets from the firewall and the Forward chains tests packet arriving at one firewall interface and destined for an external system via a second interface.

Perhaps the best way to understand the various tables and chains is to track various packets as they are processed through the system. The diagrams below show a packet headed to the firewall (blue line), one generated by the firewall (red line) and one forwarding through (green line).



The first trace is a packet destined for the firewall itself. This packet begins with a check against the connection tracking table to see if it is part of an existing connection. It is then sent to the Mangle table where it is processed by the prerouting chain. Once finished, it is passed to the Nat table and its prerouting chain (this is where dnat is applied). Next, a routing decision is made. The routing decision portion is a significant change from the IPChains code and worth highlighting. Previously (in IPChains) all packets would traverse either the input or output filter chain. Packets destined to be forwarded would actually traverse the Input, Forward and Output chains. IPTables has made routing much more efficient by introducing code to route packets to only the required chain. As the packet we are following is destined for the firewall, it is routed to the Input Filter chain. Finally the connection information is stored (assuming the packet is accepted and a connection made).

The next trace shows a packet from the firewall itself. This journey also begins with a check to see if the packet is part of en existing connection, it then moves to the routing decision and traverses the

Output chain in the Mangle Table, the Output chain in the nat table (could be used to nat packets from the firewall) and the Output chain in the Filter table. Finally, the packet passes through the Postrouting chain in the nat table. This is where any Snat manipulations would occur. Finally, if needed, the connection is recorded in conntrack.

The final example is a packet forwarded through the firewall. This packet begins just like an Input packet by traversing conntrack, the prerouting chains in the Mangle and Nat tables. Then, after the routing decision, it is sent to the Forward chain in the filter table and, ultimately follows Output packets through the postrouting chain in the nat table and is recorded by conntrack.

Rule Order

When designing a Netfilter script, the order of each item has particular significance. First, within the script itself, each entry is executed in the order entered. Therefore, placing a command to flush all chains at the end of the script would obviously be a bad idea. Similarly enabling ip forwarding prior to setting default policies would open all traffic through the firewall (admittedly for an extremely brief period).

The order of the chains within the script is generally unimportant. As the routing code determines which chain is used for any particular packet, it does not matter which is specified first in the script. In fact, statements for different scripts could be intermixed - though the resulting rulebase would be confusing and difficult to maintain.

Within each chain the rules are examined in the order entered. Once a rule is matched, the packet proceeds as indicated by the –j directive (covered further under command syntax). Therefore, it is critical to ensure the rules within each chain are ordered such that the most frequently matched are placed as early in the chain as practical. Of course the overriding consideration is that each rule be placed to meet the requirements of the firewall policy.

Command Syntax

The best way to begin examining Netfilter command syntax is to simply offer examples from Appendix B and explain the particular parameters. Within Appendix B many variables are defined to simplify firewall maintenance. In this way future address changes only require modification to a single line of code. In the examples below, /sbin/iptables is represented by $IPTAB.
General Syntax for Netfilter rules is as follows;

- Call the binary ($IPTAB)
- Specify the table (Defaults to the filter table)
- Specify the action to perform (Append, Delete, Insert, et al)
- Specify the Chain (Input, Output, Forward, Prerouting, etc)
- Specify what to match against – i.e.
    - Protocol (tcp, udp, icmp etc) (-p)
    - Source address (-s)
    - Destination Address (-d)
    - Source Port (--sport)
    - Destination Port (--dport)
    - Specific tcp flags (--tcp-flags)

    - The incoming or outgoing interface for the packet (-i or –o)
    - The state of the packet (is it new or part of an existing connection?)
    - Many other packet details
- Specify what to do with the packet
    - Accept – accept the packet
    - Drop – refuse the packet without sending notification to the sender
    - Reject – refuse the packet and inform the sender
    - Log – log the packet to Syslog and return to the chain
    - Custom chain – jump to a custom chain and continue processing
    - Several other possibilities

A crucial item to note in the list above is that the result of most matches is to stop processing the packet. Only a few of the options result in continued processing (Log and "custom chain" in the list). This introduces a requirement to place the more specific chains before general ones to ensure they are processed. Below we will examine several statements from the script in Appendix B.

The first statement below flushes (-F) all rules from all built-in chains in the filter table. The second flushes the nat table (-t nat). These are needed to ensure we start at a known state for subsequent executions of the script.

```
$IPTAB -F
$IPTAB -t nat -F
```

The lines below set default policies on the specified chains to Drop. The last statement enables forwarding of packets.

```
$IPTAB -P INPUT DROP
$IPTAB -P FORWARD DROP
$IPTAB -P OUTPUT DROP
echo "1">/proc/sys/net/ipv4/ip_forward
```

Another item to address early in the script is what to do with return traffic (has an entry in ConnTrack). The statements below accept all traffic directly or indirectly (icmp messages for example) associated with entries in the state table.

```
$IPTAB -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTAB -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTAB -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Several general statements are shown below as further examples. The first operates on the Input chain of the Filter table (by default) and matches udp packets (-p udp) entering .on the internal interface (-i $INTIF) with a destination of the broadcast address (255.255.255.255) and a destination port of ntp (123). Any packets matching this criteria are accepted. Once the match is made, all processing of that packet ceases.

```
$IPTAB -A INPUT -p udp -i $INTIF -d $BCast --dport 123 -j ACCEPT
```

The next statement matches a packet in the Input chain which has a source ip address from the internal LAN ($INTLAN) that is not on the internal interface (-i ! $INTIF). Matching packets are logged to syslog with a prefix of "ALERT:". The second line matches the same packet and drops it.

```
$IPTAB -A INPUT -s $INTLAN -i ! $INTIF -j LOG --log-level info --log-prefix
"ALERT: "
$IPTAB -A INPUT -s $INTLAN -i ! $INTIF -j DROP
```

The lines above offer insight into two important aspects of NetFilter. The first is the negation indicated by the exclamation point. This modifier changes the indicated match to NOT coming in on the internal interface. This one switch allows us to use one line for this match rather than needing to repeat it for every interface except the internal. The second item of note is the ability to specify exact logging matches and introduce custom prefixes into the logs. As the matches to this rule are most likely an attempt to circumvent the firewall, we can use the prefix "ALERT". Then using log monitoring tools (such as SWATCH) we can generate custom responses based on what happened. Even without automatic monitors, standard log reviews can be greatly eased by the use of log prefixes.

A final item of note on these rules is that the LOG action continues processing the packet while DROP does not. Obviously, changing the order of the rules would mean that these packets are dropped but never logged.

The remaining rules in Appendix B operate similarly to those outlined here and should be fairly self explanatory with the annotations provided. For further explanation of NetFilter Syntax please refer to http://www.knowplace.org/netfilter/syntax.html or other online resources.

Script Creation

When creating a firewall script, care must be taken if it is created on a Windows machine for transfer to the firewall. Windows text editors frequently add non-viewable characters, which will cause the script to fail. To prevent this, it is sometimes easier to just create the script on the Linux machine using a text editor.

Once the script has been created in a text editor, you will need to make it executable by root. This may be accomplished with the simple statement chmod 700 firewall-script – assuming "firewall-script" is the filename of course. Once that is done, run the script with the command ./firewall-script and test the system to be sure it is performing to specification (Tips on how to audit the firewall are offered elsewhere in this document). After the audit is complete, set the script to run automatically by adding it to the end of the /etc/rc.d/rc.local script (i.e. /root/bin/firewall-script).

Changing rules "on the fly"

Thus far we have focused on creating and applying rules using a script file. While the script is definitely needed, there are times when rerunning a script may cause more harm than good. As explained earlier, one of the first things we do in the script is flush the tables. This essentially shuts down the firewall for a brief period. Any active connections are broken and must be reestablished. The disruptions caused by this can preclude the ability to rerun the script during business hours. So, what to do?

As you may have notice from the script, each line directly related to the rules begins by running the binary /sbin/iptables. This can also be done directly from the command line. The only complication is ensuring new statements are correctly located and changes or deletions are performed on the proper rule. An option that can help with this is –L (list). This option will list the rules in a specified chain or all chains currently active. The syntax is as simple as /sbin/iptables –L –line-numbers Output using this statement on our firewall is shown below.

```
[root@localhost root]# /sbin/iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    ACCEPT     all  --  anywhere             localhost
3    LOG        all  --  1.2.3.0/29           anywhere            LOG level info prefix `ALERT: '
4    LOG        all  --  10.130.1.0/30        anywhere            LOG level info prefix `ALERT: '
5    DROP       all  --  1.2.3.0/29           anywhere
6    DROP       all  --  10.130.1.0/30        anywhere
7    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-ns
8    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-dgm
9    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-ssn
10   DROP       udp  --  anywhere             anywhere            udp dpt:netbios-ns
11   DROP       udp  --  anywhere             anywhere            udp dpt:netbios-dgm
12   DROP       udp  --  anywhere             anywhere            udp dpt:netbios-ssn
13   DROP       icmp --  anywhere             anywhere
14   DROP       all  --  anywhere             224.0.0.0/4
15   LOG        tcp  --  1.2.3.4              anywhere            tcp dpt:ssh LOG level info prefix `ALERT: '
16   ACCEPT     tcp  --  1.2.3.4              anywhere            tcp dpt:ssh
17   ACCEPT     udp  --  anywhere             255.255.255.255     udp dpt:ntp
18   DROP       all  --  anywhere             255.255.255.255
19   LOG        all  --  anywhere             anywhere            LOG level info prefix `INPUT: '
20   DROP       all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
2    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-ns
3    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-dgm
4    DROP       tcp  --  anywhere             anywhere            tcp dpt:netbios-ssn
5    DROP       udp  --  anywhere             anywhere            udp dpt:netbios-ns
6    DROP       udp  --  anywhere             anywhere            udp dpt:netbios-dgm
7    DROP       udp  --  anywhere             anywhere            udp dpt:netbios-ssn
8    DROP       all  --  anywhere             255.255.255.255
9    DROP       all  --  anywhere             224.0.0.0/4
10   DROP       all  --  1.2.3.0/29           anywhere
11   ACCEPT     tcp  --  1.2.3.3              anywhere            tcp dpt:domain
12   ACCEPT     udp  --  1.2.3.3              anywhere            udp dpt:domain
13   ACCEPT     tcp  --  anywhere             1.2.3.2             tcp dpt:http
14   ACCEPT     tcp  --  anywhere             1.2.3.3             tcp dpt:smtp
15   ACCEPT     tcp  --  1.2.3.3              anywhere            tcp dpt:smtp
16   ACCEPT     udp  --  1.2.3.3              anywhere            udp dpt:ntp
17   LOG        tcp  --  anywhere             1.2.3.2             tcp dpt:https LOG level info prefix `HTTPS: '
18   ACCEPT     tcp  --  3.2.1.1              1.2.3.2             tcp dpt:https
19   ACCEPT     tcp  --  1.1.5.0/24           1.2.3.2             tcp dpt:https
20   ACCEPT     tcp  --  1.1.6.0/24           1.2.3.2             tcp dpt:https
21   ACCEPT     tcp  --  1.2.3.4              anywhere            tcp dpt:ftp
22   ACCEPT     tcp  --  1.2.3.4              anywhere            tcp dpt:ssh
23   ACCEPT     tcp  --  1.2.3.4              10.130.1.1          tcp dpt:telnet
24   ACCEPT     tcp  --  1.2.3.4              anywhere            tcp dpt:http
25   ACCEPT     tcp  --  1.2.3.4              anywhere            tcp dpt:https
26   ACCEPT     udp  --  1.2.3.4              anywhere            udp dpt:snmp
27   ACCEPT     ipv6-crypt--  anywhere         1.2.3.4
28   ACCEPT     udp  --  anywhere             1.2.3.4             udp dpt:isakmp
29   ACCEPT     udp  --  3.2.1.1              10.130.3.2          udp dpt:syslog
```

```
30   ACCEPT    udp  --  10.130.1.1        10.130.3.2       udp dpt:syslog
31   ACCEPT    tcp  --  1.2.3.4           10.130.3.2       tcp dpt:ssh
32   REJECT    tcp  --  anywhere          1.2.3.3          tcp dpt:auth reject-with icmp-port-unreachable
33   LOG       tcp  --  1.2.3.0/29        anywhere         tcp dpt:http LOG level info prefix `ALERT: '
34   LOG       tcp  --  1.2.3.0/29        anywhere         tcp dpt:ftp LOG level info prefix `ALERT: '
35   LOG       tcp  --  1.2.3.0/29        anywhere         tcp dpt:https LOG level info prefix `ALERT: '
36   ACCEPT    icmp --  1.2.3.4           anywhere
37   DROP      icmp --  anywhere          anywhere
38   LOG       all  --  anywhere          anywhere         LOG level info prefix `FORWARD: '
39   DROP      all  --  anywhere          anywhere

Chain OUTPUT (policy DROP)
num  target    prot opt source            destination
1    ACCEPT    all  --  anywhere          anywhere         state RELATED,ESTABLISHED
2    DROP      icmp --  anywhere          anywhere
3    ACCEPT    udp  --  anywhere          10.130.3.2       udp dpt:syslog
4    ACCEPT    tcp  --  anywhere          1.2.3.3          tcp dpt:smtp
5    ACCEPT    all  --  anywhere          localhost
6    LOG       all  --  anywhere          anywhere         LOG level info prefix `OUTPUT: '
7    DROP      all  --  anywhere          anywhere
```

Using the information above, it is rather simple to modify chains "on the fly". The key commands are Insert (-I), Delete (-D) and Replace (-R). Append is less useful as this would place the new rule at the end of the chain (after the drop all statement). To ensure the rules are placed where needed, each of the noted actions allows for the use of a line number. An example replacement is shown below.

```
/sbin/iptables –I OUTPUT 2 –p icmp –d 1.2.3.4 –j ACCEPT
```

This command will insert the rule shown as line 2 in the Output chain and move all other rules down one number as shown below.

```
Chain OUTPUT (policy DROP)
num  target    prot opt source            destination
1    ACCEPT    all  --  anywhere          anywhere         state RELATED,ESTABLISHED
2    ACCEPT    icmp --  anywhere          1.2.3.4
3    DROP      icmp --  anywhere          anywhere
3    ACCEPT    udp  --  anywhere          10.130.3.2       udp dpt:syslog
4    ACCEPT    tcp  --  anywhere          1.2.3.3          tcp dpt:smtp
5    ACCEPT    all  --  anywhere          localhost
6    LOG       all  --  anywhere          anywhere         LOG level info prefix `OUTPUT: '
7    DROP      all  --  anywhere          anywhere
```

To delete this unnecessary rule we can use the command;

```
/sbin/iptables –D OUTPUT 2
```

This will return the chain to the original configuration.

Two items of note when acting directly on the chains are;

1.) Changes are active immediately. Therefore care must be taken to ensure connectivity problems are not introduced (particularly problems that might affect remote administration)

2.) Changes made directly on the chains are not reflected in the script until manually added. While this might sound silly, it is easy to create or modify a live chain with the expectation to script it after it has run for a few days only to forget.

Monitoring and adjusting your script for performance

Another key issue for firewalls is to monitor performance and adjust settings to increase efficiency as required. The tool for this is also the –L switch. Using the statement; /sbin/iptables –L –v –x provides an exact listing of the traffic that has matched each rule within each chain. Reviewing these counters will show which rules are matching the most packets. To increase firewall efficiency, the most active matches should be moved as high within the chain as possible (without sacrificing security).

Another use for monitoring counters is to locate filters which are never matching packets. It may be that these are catch-all or other rules not expected to match any packets or, this may be due to placing the rule incorrectly in relation to other rules in the chain. Whichever is the case, unmatched rules should always be investigated and explained.

Tripwire

A detailed examination of configuring Tripwire is beyond the scope of this document. However, once the firewall configuration is complete, this is an excellent tool to monitor unauthorized file changes. A good resource on tripwire configuration is available at http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/ref-guide/ch-tripwire.html.

Log Monitors

The final recommendation is to install a log monitor utility such as Swatch. This application will run as a Daemon and can be used to monitor the log files for specific activity. Personally, I like to configure Swatch to monitor for log in attempts and send an email to my pager when this occurs. With a limited number of administrators accessing the system, unscheduled logins are most likely a sign of a serious compromise. More information on Swatch is available at http://swatch.sourceforge.net/.

Summary

While I do not have sufficient space to cover all the possible settings, statements and switches available in Netfilter, the previous review of the key settings should provide sufficient basis for the reader to understand the script in Appendix B and continue their education using the reference material and Internet search engines. The key is to plan, research, assemble and test. Earlier sections of this paper have explored the planning and research sections. This tutorial should help with the assembly and the next section will discuss how to test the result.

Firewall Audit (Assignment 3)

Many people feel that once the firewalls are installed and all services seem to be working that the project is complete and can be declared a success. This is not the case. Without performing a controlled and detailed audit of the systems put in place, there is no way to be certain that all the desired protections are functional and no holes were inadvertently created. For this level of assurance, an audit is required.

The audit design detailed herein is a good baseline for most situations. The goal is to balance thoroughness with cost constraints. For example, it would be possible to scan all 65,535 ports during every test and use every scan type available in NMap for every potential communication path and IP address combination. However, the length of time required to complete this level of scan would not typically be justified by the level of information collected. Therefore, this audit focuses on testing the firewall rules themselves and does not wander too far (we must remember the $5,000 limit from the original proposal).

(For further discussion of firewall auditing, please see an excellent paper by Lance Spitzner at http://www.spitzner.net/audit.html.

Audit Design

A well-designed audit has several phases. First, we must examine the firewall itself to ensure it is not vulnerable to attack. This is particularly important for the Linux-based primary firewall, as it is relatively easy for misconfiguration of the underlying OS to open the entire system to compromise. The audit is completed by reviewing the services running on the firewall, verifying the version and patch levels for major components, port scanning each Ethernet interface and running a vulnerability analysis against the firewall itself.

Once the firewall is deemed secure, we must verify that the firewall rulebase correctly provides all of the services desired (and nothing more). This is confirmed through the use of a port scanner directed from each of the interfaces through to every other interface. Verification of which packets pass and which are blocked is confirmed through firewall logs and the use of packet capturing on the far side of the firewall. Further, any required authentication or encryption must be verified to work where expected and prevent access when required.

Finally, all firewall logs must be examined to ensure that every blocked attempt was logged as expected and verify that no unplanned system errors or denials occurred.

The specific steps for our audit will be as follows.

1.) OS and patch levels
    As this is a new installation, the OS and rpm levels are known and have been updated to the latest versions. Simply record this information on the Audit log for future reference.
2.) Internal firewall interface
    a. Perform a tcp connect scan against the firewall internal interface from each valid internal address. Verify that only the desired ports are open.

b. Perform a tcp connect scan against the firewall internal interface from an invalid internal address. Verify that no ports are open.
c. Perform a udp scan against the firewall internal interface from each valid internal address. Verify that only the desired ports are open.
d. Perform a brief connect scan on the internal interface from an external IP address. Verify proper blocking and logging of the event.
e. Within the firewall logs, verify logging of invalid IP address/Interface combinations and verify no logging of general windows traffic (as detailed in the firewall script).

3.) External firewall interface
   a. Perform a syn scan against the firewall external interface from each valid remote address. Verify that only the desired ports are open.
   b. Perform a brief connect scan on the external interface from an internal IP address. Verify proper blocking and logging of the event.
   c. Within the firewall logs, verify logging of invalid IP address/Interface combinations and verify no logging of general windows traffic (as detailed in the firewall script).

4.) Syslog firewall interface
   a. Perform a syn scan against the firewall syslog interface. Verify that only the desired ports are open.
   b. Perform a brief connect scan on the syslog interface from an internal IP address. Verify proper blocking and logging of the event.
   c. Within the firewall logs, verify logging of invalid IP address/Interface combinations and verify no logging of general windows traffic (as detailed in the firewall script).

5.) Vulnerability Scanning of the Firewall
   a. Using Nessus, perform a complete vulnerability scan against the internal interface. Verify there are no actionable issues.
   b. Using Nessus, perform a complete vulnerability scan against the external interface. Verify there are no actionable issues.
   c. Using Nessus, perform a complete vulnerability scan against the syslog interface. Verify there are no actionable issues

6.) Verification of the firewall rulebase – Internal to External

   Injection of the test packets will proceed in a similar manner to when we verified the firewall itself. However, we cannot use the firewall logs and nmap results alone to determine success or failure. Instead, we will place a sniffer on the far side of the firewall and verify which packets are passed using the sniffer logs.

   a. Perform a syn scan to a random external address from each valid internal address. Verify that only the desired ports are open.
   b. Perform a syn scan to the syslog address from each valid internal address. Verify that only the desired ports are open.
   c. Perform a udp scan to a random external address from each valid internal address. Verify that only the desired ports are open.
   d. Perform a udp scan to a the syslog address from each valid internal address. Verify that only the desired ports are open.
   e. Using Hping2, verify that the required protocols are forwarded through the firewall to sample remote addresses.
   f. Within the firewall logs, verify logging of invalid IP address/Interface combinations and verify no logging of general windows traffic (as detailed in the firewall script).

7.) Verification of the firewall rulebase – External to Internal

a. Perform a syn scan from a random external address to the internal IP address range. Verify that only the desired ports are open.
b. Perform a syn scan from a random external address to the syslog ip address. Verify that only the desired ports are open.
c. Perform a udp scan from a random external address to the internal IP address range. Verify that only the desired ports are open.
d. Perform a udp scan from a random external address to the syslog ip address. Verify that only the desired ports are open.
e. Using Hping2, verify that the required protocols are forwarded through the firewall from sample remote addresses.
f. Within the firewall logs, verify logging of invalid IP address/Interface combinations and verify no logging of general windows traffic (as detailed in the firewall script).

8.) Verification of the firewall rulebase – Syslog to External
   a. Perform a syn scan from the Central Syslog server to a random external ip address. Verify that only the desired ports are open.
   b. Perform a udp scan from the Central Syslog server to a random external ip address. Verify that only the desired ports are open.
   c. Using Hping2, verify that the required protocols are forwarded through the firewall.
   d. Within the firewall logs, verify logging

9.) Verification of the firewall rulebase – Syslog to Internal
   a. Perform a syn scan from the Central Syslog server to the internal IP address range. Verify that only the desired ports are open.
   b. Perform a udp scan from the Central Syslog server to the internal IP address range. Verify that only the desired ports are open.
   c. Using Hping2, verify that the required protocols are forwarded through the firewall.
   d. Within the firewall logs, verify logging

10.) Repeat a similar Audit procedure for each of the firewalls within the planned architecture and use a similar procedure for the Border Router.

11.) Once the entire infrastructure has been installed, operation of VPN encryption will be confirmed through the use of a sniffer and controlled communication packets.

Completing the Audit

Completing a detailed firewall audit is disruptive to normal operations and can consume significant network resources. In extreme cases, it is possible that unforeseen issues could even result in complete loss of network access for a period of time. It is also more complicated to monitor traffic on busy networks. As many of the audit steps included capturing packets, it is important to complete the audit when non-test traffic is extremely light. In consideration of these issues, it is best to perform the audit procedure at various stages during the assembly of the components and schedule the final audit during system installation.

Audit Details and Results

The audit was performed using the following open-source tools:

- Nmapwin is a port of the classic nmap port scanner to windows. Nmapwin is available at http://sourceforge.net/projects/nmapwin (nmap is available from http://www.insecure.org/nmap/). This tool is used to provide the various port scans needed.

- Hping2 (http://www.hping.org/) is used to craft the non-standard packets required to verify some of the rules. Hping2 is a very versatile tool and uses beyond the basic things we do here are detailed throughout the Internet.

- The last tool we use is ethereal from http://www.ethereal.com/. This is a packet sniffer and is useful to verify much of our rulebase. For UDP scanning through the firewall it is required. In this instance neither filtered nor passed packets will return information to NMAP for logging. Without capturing packets on the far side of the firewall, we would be limited to reviewing the firewall logs and assuming that anything not logged was passed. Obviously, monitoring traffic on the far side to record exactly what gets through is more precise.

Operating System and Patch levels
The firewall was developed on a new system using Red Hat Linux 8.0 with all patches current as of 3/5/03. Bastille –Linux 2.0.4 was used to lock down the server.

Scans to the internal firewall interface
These scans are designed to verify that no errant tcp or udp ports are accessible on the firewall, scans are properly logged and alerting will operate as desired. As shown in the results below, all of these goals have been met. The only unfiltered port is tcp 22 when tested from the internal firewall IP address. This is required for ssh access to the firewall itself.

Each scan was performed using the standard NMap port list (1601 tcp and 1468 udp ports). Scans to all 65,535 ports averaged 5 hours each and were determined to be cost prohibitive. Tests a through d were run using NmapWin v1.3.1 available at http://www.nmapwin.org. Test e was run using Hping2 to send a spoofed IP address to the wrong interface and, finally, test f was run using Nessus to determine if any significant firewall vulnerabilities exist.

As an interesting note, UDP and FIN scans can return confusing results when all icmp is rejected and all unaccepted traffic is simply dropped. Both UDP and FIN scans work a bit backwards, UDP expects to receive no reply for an open port and an icmp reject message for a closed port. Our firewall will not respond in either case. FIN scans expect a RST for a closed port and no response for an open port. Again, our firewall will not respond in either case. Therefore, all ports seem to be open for both UDP and FIN scans. However, nmap interprets the results differently based upon the number of ports scanned. Scanning one port reports it as open. Scanning 25 ports reports them all as open. Scanning 250+ ports reports them all as filtered (even the ones actually open). This really doesn't have much to do with the audit itself, but it is good to note. Also, this is why we need to use a sniffer and review the firewall logs in order to see exactly what is going on.

a. **TCP connect scans**
From 1.2.3.4 – This shows port 22 as the only one open. This is a planned opening to allow firewall administration.
nmap (V. 3.00) scan initiated Sat Mar 15 07:48:02 2003 as: nmap -sT -P0 -vv -T 3 1.2.3.1
Interesting ports on (1.2.3.1):
(The 1600 ports scanned but not shown below are in state: filtered)

Port State Service
22/tcp open ssh
Nmap run completed at Sat Mar 15 08:19:26 2003 -- 1 IP address (1 host up) scanned in 1884 seconds

From 1.2.3.3 – no ports open

nmap (V. 3.00) scan initiated Sat Mar 15 09:45:14 2003 as: nmap -sT -P0 -vv -T 3 1.2.3.1
All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 11:55:04 2003 -- 1 IP address (1 host up) scanned in 7790 seconds

From 1.2.3.2 – no ports open

nmap (V. 3.00) scan initiated Sat Mar 15 11:57:49 2003 as: nmap -sT -P0 -vv -T 3 1.2.3.1
All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 14:07:38 2003 -- 1 IP address (1 host up) scanned in 7789 seconds

The ethereal results below show how the DROP target provides no reply packets to a probe. You should also note the response to tcp port 22 from 1.2.3.4. Please note that the full ethereal trace from each probe total almost 16,000 lines. Therefore, the results have been abbreviated.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.4	1.2.3.1	TCP	3451 > 353 [SYN] Seq=1812386009 Ack=0 Win=16384 Len=0
2	0.000039	1.2.3.4	1.2.3.1	TCP	3452 > pop3 [SYN] Seq=1812421026 Ack=0 Win=16384 Len=0
3	0.000062	1.2.3.4	1.2.3.1	TCP	3453 > 313 [SYN] Seq=1812477329 Ack=0 Win=16384 Len=0
4	0.000084	1.2.3.4	1.2.3.1	TCP	3454 > 154 [SYN] Seq=1812540182 Ack=0 Win=16384 Len=0
5	0.000106	1.2.3.4	1.2.3.1	TCP	3455 > 49 [SYN] Seq=1812576115 Ack=0 Win=16384 Len=0
6	3.064453	1.2.3.4	1.2.3.1	TCP	3456 > 353 [SYN] Seq=1814136906 Ack=0 Win=16384 Len=0
7	3.064831	1.2.3.4	1.2.3.1	TCP	3457 > pop3 [SYN] Seq=1814189569 Ack=0 Win=16384 Len=0
8	3.065159	1.2.3.4	1.2.3.1	TCP	3458 > 313 [SYN] Seq=1814228606 Ack=0 Win=16384 Len=0
9	3.065620	1.2.3.4	1.2.3.1	TCP	3459 > 154 [SYN] Seq=1814292276 Ack=0 Win=16384 Len=0
10	3.065967	1.2.3.4	1.2.3.1	TCP	3460 > 49 [SYN] Seq=1814353858 Ack=0 Win=16384 Len=0
11	6.008360	1.2.3.4	1.2.3.1	TCP	3456 > 353 [SYN] Seq=1814136906 Ack=0 Win=16384 Len=0
2460	1295.876215	1.2.3.4	1.2.3.1	TCP	4543 > 22 [SYN] Seq=2189720387 Ack=0 Win=16384 Len=0
2461	1295.876735	1.2.3.1	1.2.3.4	TCP	22 > 4543 [SYN, ACK] Seq=89866511 Ack=2189720388 Win=5840 Len=0
2462	1295.876773	1.2.3.4	1.2.3.1	TCP	4543 > 22 [ACK] Seq=2189720388 Ack=89866512 Win=17520 Len=0
2463	1295.899922	1.2.3.1	1.2.3.4	TCP	22 > 4543 [PSH, ACK] Seq=89866512 Ack=2189720388 Win=5840 Len=23
2464	1295.922289	1.2.3.4	1.2.3.1	TCP	4543 > 22 [RST] Seq=2189720388 Ack=89866512 Win=0 Len=0

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.3	1.2.3.1	TCP	3803 > 146 [SYN] Seq=3809811318 Ack=0 Win=16384 Len=0
2	0.000043	1.2.3.3	1.2.3.1	TCP	3804 > 136 [SYN] Seq=3809859416 Ack=0 Win=16384 Len=0
3	0.000067	1.2.3.3	1.2.3.1	TCP	3805 > 171 [SYN] Seq=3809916212 Ack=0 Win=16384 Len=0
4	0.000089	1.2.3.3	1.2.3.1	TCP	3806 > 618 [SYN] Seq=3809980422 Ack=0 Win=16384 Len=0
5	0.000112	1.2.3.3	1.2.3.1	TCP	3807 > 776 [SYN] Seq=3810023428 Ack=0 Win=16384 Len=0
6	3.034725	1.2.3.3	1.2.3.1	TCP	3808 > 146 [SYN] Seq=3811571741 Ack=0 Win=16384 Len=0
7	3.035170	1.2.3.3	1.2.3.1	TCP	3809 > 136 [SYN] Seq=3811618367 Ack=0 Win=16384 Len=0
8	3.035499	1.2.3.3	1.2.3.1	TCP	3810 > 171 [SYN] Seq=3811673332 Ack=0 Win=16384 Len=0
9	3.035829	1.2.3.3	1.2.3.1	TCP	3811 > 618 [SYN] Seq=3811713261 Ack=0 Win=16384 Len=0

No.	Time	Source	Destination	Protocol	Info
32	10.354882	1.2.3.2	1.2.3.1	TCP	4362 > 933 [SYN] Seq=1822424320 Ack=0 Win=16384 Len=0
33	10.354924	1.2.3.2	1.2.3.1	TCP	4363 > 722 [SYN] Seq=1822468038 Ack=0 Win=16384 Len=0
34	10.354947	1.2.3.2	1.2.3.1	TCP	4364 > 356 [SYN] Seq=1822501922 Ack=0 Win=16384 Len=0
35	10.354970	1.2.3.2	1.2.3.1	TCP	4365 > 747 [SYN] Seq=1822567369 Ack=0 Win=16384 Len=0
36	10.354992	1.2.3.2	1.2.3.1	TCP	4366 > 715 [SYN] Seq=1822601984 Ack=0 Win=16384 Len=0
37	13.389918	1.2.3.2	1.2.3.1	TCP	4367 > 849 [SYN] Seq=1824156799 Ack=0 Win=16384 Len=0
38	13.390213	1.2.3.2	1.2.3.1	TCP	4368 > 525 [SYN] Seq=1824209352 Ack=0 Win=16384 Len=0

b. TCP connect from 1.2.3.7 (unused)

Tcp Connect scan against all ports – no ports open

nmap (V. 3.00) scan initiated Mon Mar 17 05:32:23 2003 as: nmap -sT -P0 -vv -T 3 1.2.3.1
All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 14:07:38 2003 -- 1 IP address (1 host up) scanned in 7790 seconds

c. UDP Scans – Traffic was monitored using ethereal and no response traffic was recorded.

From 1.2.3.4 – no ports open

nmap (V. 3.00) scan initiated Sat Mar 15 08:33:55 2003 as: nmap -sU -P0 -vv -T 3 1.2.3.1
All 1468 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 09:03:32 2003 -- 1 IP address (1 host up) scanned in 1777 seconds

From 1.2.3.3 – no ports open
nmap (V. 3.00) scan initiated Sat Mar 15 09:08:31 2003 as: nmap -sU -P0 -vv -T 3 1.2.3.1
All 1468 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 09:38:04 2003 -- 1 IP address (1 host up) scanned in 1773 seconds
From 1.2.3.2 – no ports open
nmap (V. 3.00) scan initiated Sat Mar 15 14:21:34 2003 as: nmap -sU -P0 -vv -T 3 1.2.3.1
All 1468 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 15 14:51:07 2003 -- 1 IP address (1 host up) scanned in 1773 seconds

d. **Packet from 10.130.1.1 on the internal interface**
Hping2 is used to verify the denial and alerting of a connection attempt from a known IP address on the wrong interface. The following command is used to run this test
Hping2 –I eth1 –a 10.130.1.1 –p 22 1.2.3.1
This results in the following log message – confirmation that the logging filter is operational.
Mar 22 13:18:13 localhost kernel: ALERT: IN=eth0 OUT= MAC=00:50:8b:08:95:64:80:0
2:04:06:08:0a:08:00 SRC=10.130.1.1 DST=1.2.3.1 LEN=40 TOS=0x00 PREC=0x00 TTL
=64 ID=53175 PROTO=TCP SPT=1833 DPT=22 WINDOW=512 RES=0x00 URGP=0

e. **Log Files**
The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Scans to the external firewall interface

Nothing is allowed to connect to the firewall external interface. To verify this, scans were run from 10.130.1.1. For the external interface it was determined that the cost of a full portcan was justified.

a. **SYN Scan from 10..130.1.2**
nmap (V. 3.00) scan initiated Sat Mar 15 20:47:39 2003 as: nmap -sS -P0 -p 1-65535 -vv -T 3 10.130.1.1
All 65535 scanned ports on (10.130.1.1) are: filtered
Nmap run completed at Sun Mar 16 02:27:20 2003 -- 1 IP address (1 host up) scanned in 20381 seconds

b. **Spoofed packet from 1.2.3.4**
Hping2 is used to verify the denial and alerting of a connection attempt from a known IP address on the wrong interface. The following command is used to run this test
Hping2 –I eth1 –a 1.2.3.4 –p 22 10.130.1.1
This results in the following log message
Mar 22 13:28:36 localhost kernel: ALERT: IN=eth1 OUT= MAC=00:a0:c9:69:4d:1e:80:0
2:04:06:08:0a:08:00 SRC=1.2.3.4 DST=10.130.1.2 LEN=40 TOS=0x00 PREC=0x00 TTL
=64 ID=43440 PROTO=TCP SPT=1819 DPT=22 WINDOW=512 RES=0x00 URGP=0

c. **Log Files**
The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Scans to the syslog firewall interface

Nothing is allowed to connect to the firewall syslog interface. To verify this, scans were run from 10.130.3.2. Again, as time was available to run the scan overnight, it was determined that a full portcan was justified.

a. **SYN Scan from 10.130.3.2**

nmap (V. 3.00) scan initiated Thu Mar 13 15:44:41 2003 as: nmap -sS -P0 -p 1-65535 -vv -T 3 10.130.3.1
All 65535 scanned ports on (10.130.3.1) are: filtered
Nmap run completed at Sun Mar 13 21:24:22 2003 -- 1 IP address (1 host up) scanned in 20381 seconds

b. Spoofed packet from 1.2.3.4

Hping2 is used to verify the denial and alerting of a connection attempt from a known IP address on the wrong interface. The following command is used to run this test

Hping2 –I eth1 –a 1.2.3.4 –p 22 10.130.3.1

This results in the following log message

Mar 13 15:28:36 localhost kernel: ALERT: IN=eth1 OUT= MAC=00:a0:c9:69:4d:1e:80:0 2:04:06:08:0a:08:00 SRC=1.2.3.4 DST=10.130.31.1 LEN=40 TOS=0x00 PREC=0x00 TTL =64 ID=43440 PROTO=TCP SPT=1819 DPT=22 WINDOW=512 RES=0x00 URGP=0

c. Log Files

The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Firewall Vulnerability scans

a. Internal Interface

Nessus requires two components for operation. There is a server daemon (nessusd) that only runs on *nix systems (Linux, Unix, etc) and a client, which is available for *nix or windows systems. For our audit we ran the daemon under Red Hat Linux 8.0 and NessusWx (a windows client available from http://nessuswx.nessus.org/) under Windows XP.

As can be seen, nothing concerning was found. After completing our series of port scans above, the general/tcp informational message may be ignored and there isn't much we can do about the IP id sequencing issue at this time.

```
NESSUS SECURITY SCAN REPORT
Created 22.03.2003                    Sorted by host names
Session Name : ExtScan
Start Time   : 22.03.2003 14:41:28
Finish Time  : 22.03.2003 15:14:57
Elapsed Time : 0 day(s) 00:23:29
Total security holes found : 2
          high severity : 0
           low severity : 1
          informational : 1
Scanned hosts:
Name                    High  Low   Info
-----------------------------------------------
1.2.3.1                 0     1     1
Host: 1.2.3.1
Open ports:
  general/tcp
Service: general/tcp
Severity: Low

The remote host uses non-random IP IDs, that is, it is
possible to predict the next value of the ip_id field of
the ip packets sent by this host.
An attacker may use this feature to determine if the remote
host sent a packet in reply to another request. This may be
used for portscanning and other things.
```

Solution : Contact your vendor for a patch
Risk factor : Low

b. External Interface

As expected, identical results were received on the external interface.

```
NESSUS SECURITY SCAN REPORT
Created 22.03.2003                    Sorted by host names
Session Name : ExtScan
Start Time   : 22.03.2003 13:51:36
Finish Time  : 22.03.2003 14:15:03
Elapsed Time : 0 day(s) 00:23:27
Total security holes found : 2
          high severity : 0
           low severity : 1
          informational : 1
Scanned hosts:
Name                    High  Low   Info
-------------------------------------------------
10.130.1.2               0     1     1
Host: 10.130.1.2
Open ports:
  general/tcp
Service: general/tcp
Severity: Low

The remote host uses non-random IP IDs, that is, it is
possible to predict the next value of the ip_id field of
the ip packets sent by this host.
An attacker may use this feature to determine if the remote
host sent a packet in reply to another request. This may be
used for portscanning and other things.
Solution : Contact your vendor for a patch
Risk factor : Low
```

c. Syslog Interface

As expected, identical results were received on the other interfaces.

```
NESSUS SECURITY SCAN REPORT
Created 22.03.2003                    Sorted by host names
Session Name : LogScan
Start Time   : 22.03.2003 15:21:36
Finish Time  : 22.03.2003 15:45:03
Elapsed Time : 0 day(s) 00:23:27
Total security holes found : 2
          high severity : 0
           low severity : 1
          informational : 1
Scanned hosts:
Name                    High  Low   Info
-------------------------------------------------
10.130.3.1               0     1     1
Host: 10.130.3.1
Open ports:
  general/tcp
Service: general/tcp
Severity: Low
```

The remote host uses non-random IP IDs, that is, it is possible to predict the next value of the ip_id field of the ip packets sent by this host.
An attacker may use this feature to determine if the remote host sent a packet in reply to another request. This may be used for portscanning and other things.
Solution : Contact your vendor for a patch
Risk factor : Low

Scans through the internal interface

As we are also filtering outgoing traffic, it is important that we verify only the ports desired are open in this direction as well. Further, as different systems operate under different rules, the scans were performed for each active internal IP address. Finally, to receive feedback to the nmap scans, an active system was placed at the external ip address.

a. SYN Scan

From 1.2.3.2 to external

In its functional configuration, the firewall does not allow the Web server to initiate any communications. The scan was run to confirm all packets were dropped but, there was no resultant ethereal output to present.

From 1.2.3.3 to external

nmap (V. 3.00) scan initiated Sun Mar 16 13:01:50 2003 as: nmap -sS -P0 -vv -T 3 10.130.1.1
Interesting ports on (10.130.1.1):
(The 1599 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	closed	smtp
53/tcp	closed	domain

Nmap run completed at Sun Mar 16 13:10:26 2003 -- 1 IP address (1 host up) scanned in 516 seconds

The associated Ethereal traces are shown below

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.3	10.130.1.1	TCP	50748 > 25 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
4	6.007817	1.2.3.3	10.130.1.1	TCP	50749 > 25 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
5	535.148438	1.2.3.3	10.130.1.1	TCP	50748 > 53 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
8	541.197271	1.2.3.3	10.130.1.1	TCP	50749 > 53 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
9	571.246099	1.2.3.3	10.130.1.1	TCP	50751 > 53 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
10	577.285170	1.2.3.3	10.130.1.1	TCP	50752 > 53 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
11	1226.962897	1.2.3.3	10.130.1.1	TCP	50751 > 25 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
14	1232.970608	1.2.3.3	10.130.1.1	TCP	50752 > 25 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0

From 1.2.3.4 to external

As mentioned earlier, nmap scanning can return incorrect answers when there are no responses to any packets. As shown below, nmap assumes all ports are filtered while Ethereal clearly shows our allowed traffic making it through.

nmap (V. 3.00) scan initiated Sun Mar 16 13:24:16 2003 as: nmap -sS -P0 -S 1.2.3.2 -vv -T 3 10.130.1.1
All 1601 scanned ports on (10.130.1.1) are: filtered
Nmap run completed at Sun Mar 16 13:52:54 2003 -- 1 IP address (1 host up) scanned in 1718 seconds

The associated Ethereal traces are shown below

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.4	10.130.1.1	TCP	58274 > 80 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
4	6.009791	1.2.3.4	10.130.1.1	TCP	58275 > 80 [SYN] Seq=1190158642 Ack=0 Win=4096 Len=0
5	318.763671	1.2.3.4	10.130.1.1	TCP	58273 > 443 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
8	324.773458	1.2.3.4	10.130.1.1	TCP	58274 > 443 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
9	667.517603	1.2.3.4	10.130.1.1	TCP	58273 > 21 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
12	673.556643	1.2.3.4	10.130.1.1	TCP	58274 > 21 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
13	739.695318	1.2.3.4	10.130.1.1	TCP	58273 > 22 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
16	745.703111	1.2.3.4	10.130.1.1	TCP	58274 > 22 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
17	1173.095708	1.2.3.4	10.130.1.1	TCP	58276 > 22 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
18	63.544933	1.2.3.4	10.130.1.1	TCP	58827 > 23 [SYN] Seq=1067015888 Ack=0 Win=4096 Len=0
19	63.545016	10.130.1.1	1.2.3.4	TCP	23 > 58827 [RST, ACK] Seq=0 Ack=1067015889 Win=0 Len=0
20	1179.105493	1.2.3.4	10.130.1.1	TCP	58277 > 22 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0

21	1269.322289	1.2.3.4	10.130.1.1	TCP	58276 > 21 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
24	1275.332038	1.2.3.4	10.130.1.1	TCP	58277 > 21 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
25	1576.138692	1.2.3.4	10.130.1.1	TCP	58276 > 443 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
28	1582.144540	1.2.3.4	10.130.1.1	TCP	58277 > 443 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
29	1702.402392	1.2.3.4	10.130.1.1	TCP	58276 > 80 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
32	1708.412125	1.2.3.4	10.130.1.1	TCP	58277 > 80 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0

From 1.2.3.2 through 1.2.3.4 to the Syslog interface

With the three interfaces, it is important to ensure that tests are run to and from each interface. This test resulted in traffic identical to that allowed out the external interface. This is a violation of our stated policy and will be addressed later in the audit. In the interest of brevity, only the ethereal traces are included to verify the statements above.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.3	10.130.3.2	TCP	50748 > 25 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
4	6.007817	1.2.3.3	10.130.3.2	TCP	50749 > 25 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
5	535.148438	1.2.3.3	10.130.3.2	TCP	50748 > 53 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
8	541.197271	1.2.3.3	10.130.3.2	TCP	50749 > 53 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
9	571.246099	1.2.3.3	10.130.3.2	TCP	50751 > 53 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
10	577.285170	1.2.3.3	10.130.3.2	TCP	50752 > 53 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
11	1226.962897	1.2.3.3	10.130.3.2	TCP	50751 > 25 [SYN] Seq=538916868 Ack=0 Win=1024 Len=0
14	1232.970608	1.2.3.3	10.130.3.2	TCP	50752 > 25 [SYN] Seq=1820536778 Ack=0 Win=1024 Len=0
No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.4	10.130.3.2	TCP	58274 > 80 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
4	6.009791	1.2.3.4	10.130.3.2	TCP	58275 > 80 [SYN] Seq=1190158642 Ack=0 Win=4096 Len=0
5	318.763671	1.2.3.4	10.130.3.2	TCP	58273 > 443 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
8	324.773458	1.2.3.4	10.130.3.2	TCP	58274 > 443 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
9	667.517603	1.2.3.4	10.130.3.2	TCP	58273 > 21 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
12	673.556643	1.2.3.4	10.130.3.2	TCP	58274 > 21 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
13	739.695318	1.2.3.4	10.130.3.2	TCP	58273 > 22 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
16	745.703111	1.2.3.4	10.130.3.2	TCP	58274 > 22 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
17	1173.095708	1.2.3.4	10.130.3.2	TCP	58276 > 22 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
20	1179.105493	1.2.3.4	10.130.3.2	TCP	58277 > 22 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
21	1269.322289	1.2.3.4	10.130.3.2	TCP	58276 > 21 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
24	1275.332038	1.2.3.4	10.130.3.2	TCP	58277 > 21 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
25	1576.138692	1.2.3.4	10.130.3.2	TCP	58276 > 443 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
28	1582.144540	1.2.3.4	10.130.3.2	TCP	58277 > 443 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0
29	1702.402392	1.2.3.4	10.130.3.2	TCP	58276 > 80 [SYN] Seq=1749893517 Ack=0 Win=4096 Len=0
32	1708.412125	1.2.3.4	10.130.3.2	TCP	58277 > 80 [SYN] Seq=2012874650 Ack=0 Win=4096 Len=0

b. UDP Scan

From 1.2.3.2 to external

In its functional configuration, the firewall does not allow the Web server to initiate any communications. The scan was run to confirm all packets were dropped but, there was no resultant ethereal output to present.

```
# nmap (V. 3.00) scan initiated Sun Mar 16 13:55:48 2003 as: nmap -sU -P0 -S 1.2.3.2 -vv -T 3  10.130.1.1
All 1468 scanned ports on  (10.130.1.1) are: filtered
# Nmap run completed at Sun Mar 16 14:25:17 2003 -- 1 IP address (1 host up) scanned in 1769 seconds
```

From 1.2.3.3 to external

Again NMap is a bit confused by the response received and reports all ports except the ones we allow through as open. The Ethereal trace shows the true story. Note how the active system (without open DNS or NTP ports) sends an icmp unreachable message back.

```
# nmap (V. 3.00) scan initiated Sun Mar 16 14:29:26 2003 as: nmap -sU -P0 -S 1.2.3.3 -vv -T 3 10.130.1.1
Interesting ports on  (10.130.1.1):
(The 2 ports scanned but not shown below are in state: closed)
# Nmap run completed at Sun Mar 16 14:34:49 2003 -- 1 IP address (1 host up) scanned in 323 seconds
```

The associated Ethereal traces are shown below

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.3	10.130.1.1	NTP	NTP
4	0.002093	10.130.1.1	1.2.3.3	ICMP	Destination unreachable
5	7.228498	1.2.3.3	10.130.1.1	DNS	[Malformed Packet]
6	7.228585	10.130.1.1	1.2.3.3	ICMP	Destination unreachable

From 1.2.3.4 to external

As mentioned earlier, nmap scanning can return incorrect answers when there are no responses to any packets. As shown below, nmap assumes all ports are filtered while Ethereal clearly shows our allowed traffic making it through.

nmap (V. 3.00) scan initiated Sun Mar 16 14:39:28 2003 as: nmap -sU -P0 -S 1.2.3.4 -vv -T 3 10.130.1.1
Interesting ports on (10.130.1.1):
(The 1 port scanned but not shown below is in state: closed)
Nmap run completed at Sun Mar 16 15:04:58 2003 -- 1 IP address (1 host up) scanned in 1529 seconds

The associated Ethereal traces are shown below

No.	Time	Source	Destination	Protocol	Info
3	0.000378	1.2.3.4	10.130.3.2	SNMP	Source port: 59942 Destination port: 161[Malformed Packet]
4	6.039071	1.2.3.4	10.130.3.2	SNMP	Source port: 59943 Destination port: 161[Malformed Packet]

From 1.2.3.2 through 1.2.3.4 to the Syslog interface

With the three interfaces, it is important to ensure that tests are run to and from each interface. This test resulted in traffic identical to that allowed out the external interface. This is a violation of our stated policy and will be addressed later in the audit. In the interest of brevity, only the ethereal traces are included to verify the statements above.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.2.3.3	10.130.3.2	NTP	NTP
4	6.009801	1.2.3.3	10.130.3.2	NTP	NTP
5	625.636701	1.2.3.3	10.130.3.2	DNS	[Malformed Packet]
8	631.646487	1.2.3.3	10.130.3.2	DNS	[Malformed Packet]

c. Specific Services

There are no outgoing services or protocols that are limited to specific destinations.

d. Log Files

The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Scans through the external interface

To verify connections through the firewall we used Nmap to input packets on one side and ethereal to monitor what came through. In these scans we used the default Nmap tcp connect ports. The results below show proper operation of our firewall rulebase.

a. SYN scans

From external to 1.2.3.2 through 1.2.3.4

To verify packets passed through the firewall, TCP Nmap scans were run through the firewall while the internal interface was monitored using ethereal. As expected, only HTTP (tcp port 80) to the Web Server and SMTP (tcp port 25) on the Email gateway were detected during these scans. The nmap traces and ethereal results are shown below.

nmap (V. 3.00) scan initiated Sun Mar 16 09:05:10 2003 as: nmap -sS -P0 -vv 1.2.3.2-5
Interesting ports on (1.2.3.2):
(The 1600 ports scanned but not shown below are in state: filtered)

Port	State	Service
80/tcp	closed	http

Interesting ports on (1.2.3.3):
(The 1600 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	closed	smtp

All 1601 scanned ports on (1.2.3.4) are: filtered
All 1601 scanned ports on (1.2.3.5) are: filtered
Nmap run completed at Sun Mar 16 10:25:12 2003 -- 4 IP addresses (4 hosts up) scanned in 4802 seconds

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.130.1.1	1.2.3..2	TCP	3129 > 80 [SYN] Seq=3202850653 Ack=0 Win=5840 Len=0
2	2.999268	10.130.1.1	1.2.3..2	TCP	3129 > 80 [SYN] Seq=3202850653 Ack=0 Win=5840 Len=0
5	8.998019	10.130.1.1	1.2.3..2	TCP	3129 > 80 [SYN] Seq=3202850653 Ack=0 Win=5840 Len=0
14	1352.298548	10.130.1.1	1.2.3.3	TCP	3265 > 25 [SYN] Seq=352134587 Ack=0 Win=5840 Len=0

- 15 1358.315847 10.130.1.1 1.2.3.3 TCP 3285 > 25 [SYN] Seq=354318580 Ack=0 Win=5840 Len=0
- 22 2116.374839 10.130.1.1 1.2.3.3 TCP 1939 > 25 [SYN] Seq=1142445312 Ack=0 Win=5840 Len=0
- 23 2119.358073 10.130.1.1 1.2.3.3 TCP 1939 > 25 [SYN] Seq=1142445312 Ack=0 Win=5840 Len=0
- 26 2125.536635 10.130.1.1 1.2.3.3 TCP 1939 > 25 [SYN] Seq=1142445312 Ack=0 Win=5840 Len=0
- 27 2125.538635 10.130.1.1 1.2.3.3 TCP 1964 > 25 [SYN] Seq=1149236903 Ack=0 Win=5840 Len=0

From external to 10.130.3.2

nmap (V. 3.00) scan initiated Sat Mar 22 21:45:14 2003 as: nmap -sS -P0 -vv -T 3 10.130.3.2
All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 22 23:55:04 2003 -- 1 IP address (1 host up) scanned in 7790 seconds

b. UDP scans

From external to 1.2.3.2 through 1.2.3.4

To verify packets passed through the firewall, Nmap scans were run through the firewall while the intemal interface was monitored using ethereal. As expected, the only open port was 500 to the internal firewall for IPSEC connections. As above, the ethereal summary output is shown below. In this output as well a number of ARP requests and Name queries were removed. They were not required to verify the firewall script and cluttered the report.

nmap (V. 3.00) scan initiated Sun Mar 16 10:27:46 2003 as: nmap -sU -P0 -vv 1.2.3.2-5
All 1468 scanned ports on (1.2.3.2) are: filtered
All 1468 scanned ports on (1.2.3.3) are: filtered
All 1468 scanned ports on (1.2.3.4) are: filtered
All 1468 scanned ports on (1.2.3.5) are: filtered

No.	Time	Source	Destination	Protocol	Info
3	0.000324	10.130.3.2	1.2.3.4	ISAKMP	[Malformed Packet]
4	0.380555	1.2.3.4	10.130.3.2	ESP	ESP (SPI=0x00000000)
5	6.010482	10.130.3.2	1.2.3.4	ISAKMP	[Malformed Packet]
6	6.010702	1.2.3.4	10.130.3.2	ESP	ESP (SPI=0x00000000)

From external to 10.130.3.2

nmap (V. 3.00) scan initiated Sun Mar 23 07:35:21 2003 as: nmap -sU -P0 -vv -T 3 10.130.3.2
All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 23 09:45:11 2003 -- 1 IP address (1 host up) scanned in 7790 seconds

c. Remote sites to Internal Services

Several remote sites are offered additional access into GENT systems. These filters are confirmed through the use of Hping2 to craft the packets and ethereal to monitor successful receipt.

Remote systems to port 443 on the web server (extraneous entries removed):

Hping2 –I eth1 –a <spoofed ip address> -p 443 1.2.3.2

3	0.000320	3.2.1.1	1.2.3.2	TCP	3020 > https [] Seq=1554009239 Ack=959156604 Win=512 Len=0
4	0.000382	1.2.3.2	3.2.1.1	TCP	https > 3020 [RST, ACK] Seq=0 Ack=1554009239 Win=0 Len=0
5	0.997901	3.2.1.1	1.2.3.2	TCP	3021 > https [] Seq=1326456283 Ack=508355707 Win=512 Len=0
6	0.997971	1.2.3.2	3.2.1.1	TCP	https > 3021 [RST, ACK] Seq=0 Ack=1326456283 Win=0 Len=0
7	1.997706	3.2.1.1	1.2.3.2	TCP	3022 > https [] Seq=1601201840 Ack=731233054 Win=512 Len=0
8	1.997776	1.2.3.2	3.2.1.1	TCP	https > 3022 [RST, ACK] Seq=0 Ack=1601201840 Win=0 Len=0
9	2.997506	3.2.1.1	1.2.3.2	TCP	3023 > https [] Seq=154470189 Ack=1253369664 Win=512 Len=0
10	2.997575	1.2.3.2	3.2.1.1	TCP	https > 3023 [RST, ACK] Seq=0 Ack=154470189 Win=0 Len=0
11	3.997283	3.2.1.1	1.2.3.2	TCP	3024 > https [] Seq=550993156 Ack=1082499246 Win=512 Len=0
12	3.997351	1.2.3.2	3.2.1.1	TCP	https > 3024 [RST, ACK] Seq=0 Ack=550993156 Win=0 Len=0
13	4.997105	3.2.1.1	1.2.3.2	TCP	3025 > https [] Seq=1000848843 Ack=1858479018 Win=512 Len=0
14	4.997174	1.2.3.2	3.2.1.1	TCP	https > 3025 [RST, ACK] Seq=0 Ack=1000848843 Win=0 Len=0
15	25.857216	1.1.5.1	1.2.3.2	TCP	2915 > https [] Seq=984763498 Ack=1743261170 Win=512 Len=0
16	25.857287	1.2.3.2	1.1.5.1	TCP	https > 2915 [RST, ACK] Seq=0 Ack=984763498 Win=0 Len=0
17	26.855956	1.1.5.1	1.2.3.2	TCP	2916 > https [] Seq=1925306434 Ack=1016788071 Win=512 Len=0
18	26.856025	1.2.3.2	1.1.5.1	TCP	https > 2916 [RST, ACK] Seq=0 Ack=1925306434 Win=0 Len=0
19	27.855744	1.1.5.1	1.2.3.2	TCP	2917 > https [] Seq=879209087 Ack=596010532 Win=512 Len=0
20	27.855814	1.2.3.2	1.1.5.1	TCP	https > 2917 [RST, ACK] Seq=0 Ack=879209087 Win=0 Len=0
21	28.855553	1.1.5.1	1.2.3.2	TCP	2918 > https [] Seq=666222219 Ack=614044826 Win=512 Len=0
22	28.855621	1.2.3.2	1.1.5.1	TCP	https > 2918 [RST, ACK] Seq=0 Ack=666222219 Win=0 Len=0
23	36.908270	1.1.6.1	1.2.3.2	TCP	2373 > https [] Seq=1083912956 Ack=310829631 Win=512 Len=0

24	36.908345	1.2.3.2	1.1.6.1	TCP	https > 2373 [RST, ACK] Seq=0 Ack=1083912956 Win=0 Len=0
25	37.906472	1.1.6.1	1.2.3.2	TCP	2374 > https [] Seq=1446527242 Ack=1947882685 Win=512 Len=0
26	37.906541	1.2.3.2	1.1.6.1	TCP	https > 2374 [RST, ACK] Seq=0 Ack=1446527242 Win=0 Len=0
27	38.906250	1.1.6.1	1.2.3.2	TCP	2375 > https [] Seq=339086557 Ack=307644613 Win=512 Len=0
28	38.906320	1.2.3.2	1.1.6.1	TCP	https > 2375 [RST, ACK] Seq=0 Ack=339086557 Win=0 Len=0
29	39.906042	1.1.6.1	1.2.3.2	TCP	2376 > https [] Seq=1971839744 Ack=156886308 Win=512 Len=0
30	39.906111	1.2.3.2	1.1.6.1	TCP	https > 2376 [RST, ACK] Seq=0 Ack=1971839744 Win=0 Len=0

Protocol 50 from any external address to the internal firewall

Hping2 -0 –I eth1 –H 50 1.2.3.4

33	121.493808	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]
34	122.492166	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]
35	123.491950	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]
36	124.491762	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]
37	125.491558	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]
38	126.491337	10.130.1.1	1.2.3.4	ESP	[Malformed Packet]

Syslog traffic (UDP 514) from the various syslog devices to the central syslog server

Hping2 -2 –I eth1 –a 10.130.1.1 –p 514 10.130.3.2
Hping2 -2 –I eth1 –a 3.2.1.1 –p 514 10.130.3.2

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.130.1.1	10.130.3.2	Syslog	[Malformed Packet]
14	4.997174	3.2.1.1	10.130.3.2	Syslog	[Malformed Packet]

d. Log Files

The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Scans through the syslog interface

To verify connections through the firewall we used Nmap to input packets on one side and ethereal to monitor what came through. In these scans we used the default Nmap tcp connect ports.

The results for the syslog interface were identical to the other two. By design, this interface should be more restrictive. This issue will be addressed in the discussion to follow.

a. SYN scans

From syslog to 1.2.3.2 through 1.2.3.4

nmap (V. 3.00) scan initiated Sun Mar 16 20:30:21 2003 as: nmap -sS -P0 -vv 1.2.3.2-4
Interesting ports on (1.2.3.2):
(The 1600 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp closed http
Interesting ports on (1.2.3.3):
(The 1600 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp closed smtp
All 1601 scanned ports on (1.2.3.4) are: filtered
Nmap run completed at Sun Mar 16 21:18:39 2003 -- 3 IP addresses (3 hosts up) scanned in 2898 seconds

No.	Time	Source	Destination	Protocol	Info
3	0.001017	10.130.3.2	1.2.3.2	TCP	59036 > 80 [SYN] Seq=1009898547 Ack=0 Win=2048 Len=0
4	0.001075	1.2.3.2	10.130.3.2	TCP	80 > 59036 [RST, ACK] Seq=0 Ack=1009898548 Win=0 Len=0
7	288.721805	10.130.3.2	1.2.3.3	TCP	59036 > 25 [SYN] Seq=2680528342 Ack=0 Win=2048 Len=0
8	288.721859	1.2.3.3	10.130.3.2	TCP	25 > 59036 [RST, ACK] Seq=0 Ack=2680528343 Win=0 Len=0

From syslog to 10.130.1.1

nmap (V. 3.00) scan initiated Sat Mar 22 19:45:14 2003 as: nmap -sS -P0 -vv -T 3 10.130.1.1

All 1601 scanned ports on (1.2.3.1) are: filtered
Nmap run completed at Sat Mar 22 21:55:04 2003 -- 1 IP address (1 host up) scanned in 7790 seconds

b. UDP scans

From syslog to 1.2.3.2 through 1.2.3.4

nmap (V. 3.00) scan initiated Sun Mar 16 18:25:01 2003 as: nmap -sU -P0 -vv 1.2.3.2-5
All 1468 scanned ports on (1.2.3.2) are: filtered
All 1468 scanned ports on (1.2.3.3) are: filtered
All 1468 scanned ports on (1.2.3.4) are: filtered
All 1468 scanned ports on (1.2.3.5) are: filtered
Nmap run completed at Sun Mar 16 20:24:13 2003 -- 4 IP addresses (4 hosts up) scanned in 7152 seconds

No.	Time	Source	Destination	Protocol	Info
3	0.000324	10.130.3.2	1.2.3.4	ISAKMP	[Malformed Packet]
4	0.380555	1.2.3.4	10.130.3.2	ESP	ESP (SPI=0x00000000)
5	6.010482	10.130.3.2	1.2.3.4	ISAKMP	[Malformed Packet]
6	6.010702	1.2.3.4	10.130.3.2	ESP	ESP (SPI=0x00000000)

From syslog to 10.130.1.1

nmap (V. 3.00) scan initiated Sun Mar 16 21:26:34 2003 as: nmap -sU -P0 -vv 10.130.1.1
All 1468 scanned ports on (10.130.1.1) are: filtered
Nmap run completed at Sun Mar 16 21:55:35 2003 -- 1 IP address (1 host up) scanned in 1741 seconds

c. Log Files

The firewall logs were examined and entries for denied packets generally verified. We then used grep to verify no logging of packets purposely dropped to prevent log entries.

Summary of Results

Firewall Interfaces

The only open port to any firewall interfaces is tcp 22 (ssh) from the internal firewall to the internal interface. This is required for management and allowed per the policy. Logging was verified to perform per policy and no other problems were noted.

Vulnerability Scans

Each firewall interface was scanned for vulnerabilities using Nessus. No actionable issues were discovered.

Internal to External

Per the design, only specific ports are open through the firewall from internal systems. The following policies were confirmed;

- SMTP (tcp 25) NTP (udp 123) and DNS (tcp and udp 53) from the Email Gateway
- HTTP (tcp 80) HTTPS (tcp 443) FTP (tcp 21) SSH (tcp 22) telnet (tcp 23) IKE (udp 500) and SNMP (udp 161) from the Internal firewall.
- No other ports are open

Internal to Syslog

SSH from the internal firewall is the only protocol that should be open for this scan. However, due to an oversight, all Internal to External services are available to the syslog interface as well. This must be altered as discussed in the conclusions.

External to Internal
Per the design, only specific ports are open through the firewall from external systems. The following policies were confirmed;

- HTTP (tcp 80) to the web server
- SMTP (tcp 25) to the email gateway
- HTTPS (tcp 443) from Client sites and the Remote office to the web server
- IKE (udp 500) and ESP (protocol 50) to the internal firewall

External to Syslog
All but approved ports were successfully shown to be blocked to the Syslog server. Only Syslog (udp 514) from the remote office and the border router to the central syslog server was detected.

Syslog to Internal
As with the Internal to Syslog tests, an oversight allows the same ports opened from "External to Internal" to pass traffic here. This must be remedied for compliance with policy.

Syslog to External
As required by policy, no traffic is passed from the Syslog to the External interface.

Conclusions and Recommendations

In general the audit may be deemed a success. No unexpected firewall ports are open and all required traffic will flow to its destination. There was, however one issue which must be addressed. Communications to and from the syslog interface must be tightly controlled to ensure the data remains secure. Unfortunately, when this interface was added at the last minute, we overlooked these special needs and did not sufficiently isolate traffic. This can easily happen when moving from a simple, dual interface firewall to a system with multiple ports.

There are a couple of ways to close down the syslog interface. During the initial design, it is usually a good idea to set up custom chains for multi-interface firewalls. In this configuration the forward chain is simply used to steer traffic to the correct custom chain as shown below. (note – variables are used as shown in Appendix B)

```
$IPTAB -A FORWARD -s $Any -d $INTLAN -i $EXTIF -o $INTIF -j ext-int
$IPTAB -A FORWARD -s $Any -d $LOGLAN -i $EXTIF -o $LOGIF -j ext-log
$IPTAB -A FORWARD -s $INTLAN -d $Any -i $INTIF -o $EXTIF -j int-ext
$IPTAB -A FORWARD -s $INTLAN -d $LOGLAN -i $INTIF -o $LOGIF -j int-log
$IPTAB -A FORWARD -s $LOGLAN -d $EXTLAN -i $LOGIF -o $EXTIF -j log-ext
$IPTAB -A FORWARD -s $LOGLAN -d $INTLAN -i $LOGIF -o $INTIF -j log-ext
```

Then rules can be isolated within each custom chain for the specific traffic. Splitting up rules like this also helps manage the system as it becomes more complex and can make a large firewall more efficient by reducing the number of rules each packet must traverse.

Another way to remedy this problem is to simply make each forward filter a bit more specific. This remedy is shown below (again using the variables from Appendix B).

```
$IPTAB -A FORWARD -p tcp --dport 137 -j DROP
```

```
$IPTAB -A FORWARD -p tcp --dport 138 -j DROP
$IPTAB -A FORWARD -p tcp --dport 139 -j DROP
$IPTAB -A FORWARD -p udp --dport 137 -j DROP
$IPTAB -A FORWARD -p udp --dport 138 -j DROP
$IPTAB -A FORWARD -p udp --dport 139 -j DROP
$IPTAB -A FORWARD -d $BCast -j DROP
$IPTAB -A FORWARD -d $MCAST -j DROP
$IPTAB -A FORWARD -s $INTLAN -i ! $INTIF -j DROP
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $MAILServerIP --dport 53 -j ACCEPT
$IPTAB -A FORWARD -p udp -o $EXTIF -s $MAILServerIP --dport 53 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $INTIF -d $WEBServerIP --dport 80 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $INTIF -d $MAILServerIP --dport 25 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $MAILServerIP --dport 25 -j ACCEPT
$IPTAB -A FORWARD -p udp -o $EXTIF -s $MAILServerIP --dport 123 -j ACCEPT
$IPTAB -A FORWARD -p tcp -d $WEBServerIP --dport 443 -j LOG --log-level info --
log-prefix "HTTPS: "
$IPTAB -A FORWARD -p tcp-o $INTIF -s $ROffice -d $WEBServerIP --dport 443 -j
ACCEPT
$IPTAB -A FORWARD -p tcp -o $INTIF -s $Client1 -d $WEBServerIP --dport 443 -j
ACCEPT
$IPTAB -A FORWARD -p tcp -o $INTIF -s $Client2 -d $WEBServerIP --dport 443 -j
ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP --dport 21 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP --dport 22  -j LOG --log-level
info --log-prefix " LOGON: "
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP --dport 22 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP -d $RouterIP --dport 23 -j
ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP --dport 80 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $EXTIF -s $INTServerIP --dport 443 -j ACCEPT
$IPTAB -A FORWARD -p udp -o $EXTIF -s $INTServerIP --dport 161 -j ACCEPT
$IPTAB -A FORWARD -p 50 -o $INTIF -d $INTServerIP -j ACCEPT
$IPTAB -A FORWARD -p udp -o $INTIF -d $INTServerIP --dport 500 -j ACCEPT
$IPTAB -A FORWARD -p udp -o $LOGIF -s $ROffice -d $CentLogIP --dport 514 -j ACCEPT
$IPTAB -A FORWARD -p udp -o $LOGIF -s $Router -d $CentLogIP --dport 514 -j ACCEPT
$IPTAB -A FORWARD -p tcp -o $LOGIF -s $INTServerIP -d $CentLogIP --dport 22  -j
LOG --log-level info --log-prefix " LOGON: "
$IPTAB -A FORWARD -p tcp -o $LOGIF -s $INTServerIP -d $CentLogIP --dport 22 -j
ACCEPT
$IPTAB -A FORWARD -p tcp --s $ServiceNet --dport 80 -j LOG --log-level info --log-
prefix "ALERT: "
$IPTAB -A FORWARD -p tcp -s $ServiceNet --dport 21 -j LOG --log-level info --log-
prefix " ALERT: "
$IPTAB -A FORWARD -p tcp -s $ServiceNet --dport 443 -j LOG --log-level info --log-
prefix " ALERT: "
#$IPTAB -A FORWARD -p tcp -o $EXTIF -s $ServiceNET --dport 80 -j ACCEPT
#$IPTAB -A FORWARD -p tcp -o $EXTIF -s $ServiceNET --dport 21 -j ACCEPT
#$IPTAB -A FORWARD -p tcp -o $EXTIF -s $ServiceNET --dport 443 -j ACCEPT
$IPTAB -A FORWARD -p icmp -s $INTServerIP -j ACCEPT
$IPTAB -A FORWARD -p icmp -j DROP
$IPTAB -A FORWARD -j LOG --log-level info --log-prefix "FORWARD: "
$IPTAB -A FORWARD -j DROP
echo "FORWARD Done"
```

Which approach to take is dependent upon the specific circumstances of any situation. If time allows, I would always go with custom chains. This solution is much more scaleable and positions the rulebase for the future. In this particular situation however, I have chosen the second plan. This is

because the possibility of rapid growth in the rulebase is extremely remote and the project is already behind schedule. In this instance, we can proceed with the more explicit rules. Adding a comment to the beginning of the script can also act as a reminder that this issue must be addressed prior to (or during) the next significant change to the rulebase.

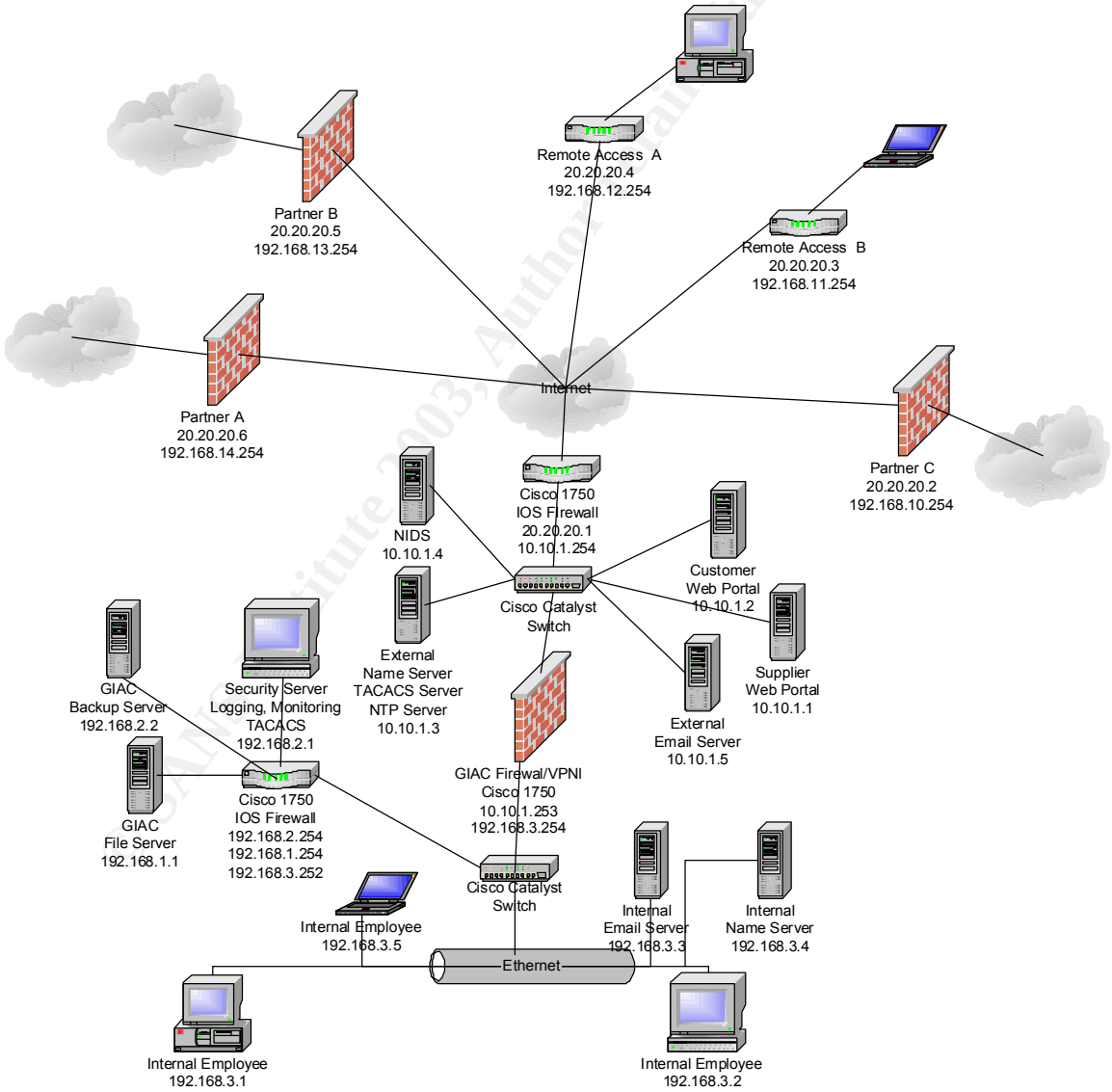
With the above changes in place, each of the "forwarding" tests (internal to external, internal to syslog, etc.) was run to verify the rulebase changes. The results were successful. At this point the primary firewall audit is deemed complete. The remaining systems may now be evaluated and installed at the client site for system-level auditing.

Design Under Fire (Assignment 4)

Introduction

This assignment requires the student to explore security from the other side. A network design must be selected from previously posted practical assignments and pasted within this document. We then research and design a series of attacks against that architecture.

The architecture shown in the figure below was selected from the practical completed by Keith Pachulski and posted at http://www.giac.org/practical/Keith_Pachulski_GCFW.doc



An Attack on the Primary Firewall

Over the years there have been a many attacks developed for Cisco IOS. For the most part (as seems typical for firewall appliances in general) these vulnerabilities focus on denial of service. On occasion, however, an exploit will arise that can provide an attacker full access to the system. This type of issue arose with many Cisco products in July of 2001 as detailed by CIAC at the following location http://www.ciac.org/ciac/bulletins/l-106.shtml and by Cisco at http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html

This vulnerability is admittedly a bit dated but it seems quite possible that anyone who would activate an http server on their primary firewall might also neglect to monitor security alerts and fail to upgrade their IOS software regularly.

The Exploit

For the exploit to succeed, the firewall must have the http server enabled, use local authentication and be running IOS 11.3 to 12.1 inclusive. If that is the case, the attacker simply needs to send an http request to http://<device_address>/level/**xx**/exec/....

Where <device_address> is the address of the Cisco router and xx is a number between 16 and 99. The exact number required can only be found through trial and error but 84 attempts are not seen as any deterrent or even as a delay. If successful, access is provided at level 15 (enable) and any available IOS commands may be used.

The Attack

The first thing we need to do to attack GIAC Enterprises is to locate them. A quick lookup on http://www.networksolutions.com/cgi-bin/whois/whois reveals name server ip addresses that we can then use at ARIN (http://ws.arin.net/cgi-bin/whois.pl) to collect further information. From our searches on ARIN, we see that GIAC Enterprises has leased a full Class C address block (10.10.1.0/24) for their use. This gives us the starting point we require to locate vulnerable systems. FYI, if all registered DNS servers belong to the ISP, these steps will not reveal the correct IP block. In that case we would need to work a little harder by using nslookup to determine web, email and other server addresses until we locate the correct addresses.

With the GIAC IP addresses in hand, we return to our old friend nmap. The only difference now, is we want to ensure we are not immediately detected as we scan the GIAC network. The simplest way to prevent detection is to move slow and steady. First, we must quietly probe for active http servers. The following nmap command will do exactly that.

```
nmap -sT -P0  -p 80 -vv -T 0 -oN "patience.txt" 10.10.1.1-254
```

This command will scan 1 IP address at a time and wait at least 5 minutes in between each probe. It is highly unlikely that any functional IDS can detect correlation between scans spaced this far apart. The down side is it will take about 24 hours for this scan to complete. Of course it is unlikely that the IOS will be updated in a day so we can proceed cautiously.

Once we have our list of active http servers, we need to determine whether any might be a Cisco router. Again proceeding slowly, connect to them one at a time using a standard web browser; active web sites can be crossed off the list. Any questionable addresses can be used for further probes.

With our limited number of possible http servers in hand we can begin to try the exploit. At this point, it is quite possible that our actions will be detected and recorded by an IDS system. To ensure our actions are not traced back to us, we need to work through a public proxy server. Many proxys are readily available and can be located from http://www.publicproxyservers.com/index.html. With our IP address protected, scan the addresses recorded, cycling the numeric entries from 16 to 99. If any one of the routers is vulnerable, we will be able to enter any commands desired.

Once we know the vulnerable routers, we need to plan our best approach to obtaining access quickly. If we are able to access multiple routers (as might be the case in a one vendor/one product environment) we could easily configure our own VPN connection into the heart of the GIAC network. If we want to move a bit quicker, changing each of the acls to a simple "permit any any" statement will allow us to quickly assess the other machines on the network for vulnerabilities. The goal with this approach would be to quickly compromise as many systems as possible before disconnecting (the assumption being that our actions will be caught by IDS or log file reviews shortly). At some future date, we can slowly begin to review the systems previously compromised (from another compromised host) in the hope that at least one was overlooked in the cleanup effort.

The Result

While our reconnaissance went well, we were unable to locate any vulnerable routers and the rest of our attack was aborted. As this is a newer network installation, it is using IOS 12.2 and the security engineer is wise enough to use the CLI interface instead of enabling an http server. Of course, this is merely a temporary setback. To work out some frustration at being foiled, we decide to explore options for a DDOS attack.

DDOS Attack

While our previous attack was unsuccessful for system compromise, Denial of Service is frequently easier to achieve. In fact, even if our earlier compromise attempt was successful, a follow-on DDOS attack could help thin recovery resources and increase chances that a compromised system would not be fully rebuilt.

The Exploit

The attack we have chosen actually became a DDOS attack after Cisco repaired a system compromise issue. It is called ssh crc32 and is described in detail at http://www.kb.cert.org/vuls/id/945216. Essentially what happens is that a specially crafted packet can create a hash table set to zero. When sshd attempts to hash values into this null table, the values can be used to "modify the return address of the function call" and execute arbitrary code as the sshd process (typically run as root). However, the key for our discussion is not the exploit itself, simply the fact that probing for it can cause the ssh process in Cisco devices to consume excessive cpu cycles. It can even go as far as rebooting the device (see http://www.cisco.com/warp/public/707/SSH-scanning.shtml). A good analysis of the exploit itself may be found at http://www.linuxsecurity.com/articles/intrusion_detection_article-4002.html

The Attack

Our only limitation on using this exploit against GIAC is the fact that the exploit code only runs under Linux. Fortunately, we have 50 cable modem/DSL systems running VNC

(http://www.uk.research.att.com/vnc/) . Using the client utility, we upload the ssh crc32 exploit code to each of the 50 systems.

The next step is to program each of drones to commence staggered scans against the GIAC network. Our goal here is not to have all 50 systems attack at once, as the number of simultaneous scans is not a factor in this exploit. Rather we want CPU "stalls" and router reboots to occur randomly but frequently throughout the GIAC network. A limitation of the version of the exploit we have is it will not scan a block of addresses. Therefore, we must schedules probes to each desired address. To do this, CRON should work well. A sample Crontab file is shown below. Obviously, to create the "staggered" effect, the specific trigger times will vary for each of the 50 systems.

```
[crontab file]
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

#run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
03 * * * * root run-parts /root/daily.update # This is our script (given an innocuous name in
case the owner of the machine takes a look at root's crontab). The next one will be set to run at
3 minutes after the hour on up. Just be certain to skip times already used (01, 02, 22, 42).
```

The attack script is common to all systems and simply attempts connection to each external address. As the only requirement for this exploit is the connection attempt, we do not need to wait for responses or introduce any complexity into the script. We simply need to initiate the connection attempt and move on.. The exploit will not work against the devices we are attacking, so each connection should simply time-out and reset.

```
[daily.update]
#!/bin/sh
/bin/ssh 10.10.1.1 -p 22 -t 0
/bin/ssh 10.10.1.2 -p 22 -t 0
/bin/ssh 10.10.1.3 -p 22 -t 0
/bin/ssh 10.10.1.4 -p 22 -t 0
/bin/ssh 10.10.1.5 -p 22 -t 0
/bin/ssh 10.10.1.6 -p 22 -t 0
.
.
.
.
/bin/ssh 10.10.1.254 -p 22 -t 0 # These commands are repeated for each of the 4 scan types (0-
3) to ensure we will hit all vulnerabilities)
```

It is very likely that many of our 50 drones will be lost as part of this attack but the satisfaction is worth it. I am certain there are many more high school kids with a DSL line, cdrom burner and an old system they want to throw on the Internet with a default Red Hat 7.2 installation.

Protection against this DDOS Attack
There are two ways that GIAC can prevent this attack from occurring. The most obvious is to update their IOS software to the latest revision however; this is not necessarily the best or only step that should be taken. As GIAC has a VPN in place, they could deny all communications with the external interface on their routers and only allow ssh communications from trusted IP addresses or disallow ssh from all external systems. Either of these changes would be beneficial in addition to updating the Cisco IOS. Any of these changes will render the exploit noted ineffective.

Attack via a Perimeter System

Well, our frontal assault was thwarted by a current IOS and proper router configuration; our DOS attack was moderately successful but short-lived. This indicates a well-protected core network. Rather than beating on these gates until we are caught, it is time to explore other avenues for accessing GIAC systems. Our next target is one of the remote sites.

The Exploit
In our experience, unless specifically addressed by corporate policy, remote offices tend to utilize laxer security measures. This may be because their access into core systems is limited, because they have insufficient local resources to support a strong infrastructure or simply due to the old adage "Out of sight, Out of mind". Whatever the cause, this is seen as our best remaining chance to access GIAC systems.

The Attack
The biggest impediment to attacking a specific remote system is finding it. For example, a mobile user might be open to a wide range of attacks but, if they are using a dial-up account and not connecting frequently, the odds of intentionally locating them are slim. Fortunately, in this case, remote employees use local ISP's and local email addresses. Through a web search on GIAC employees and several phone calls (using information collected from whois and the company web site) we are able to determine the location of one remote office. As this office is using SDSL with a static IP block (/30), the game is on.

Our next lucky break (we were due for on don't you think?) is that the remote office is using a Linksys Router with remote management enabled. Using information discovered by Corelabs (http://www1.corest.com/common/showdoc.php?idx=276&idxseccion=10) we now have a clear opening into this device. I can only assume this is due to a lack of local IT talent and a need to control/monitor this connection point. As an aside, Corelabs was able to craft an http request that would cause the router to open up remote access to the Internet. As mentioned in their discussion, we could send such a link to the remote office in an email disguised as coming from the main office. Getting them to run the link would open up the router to the rest of our attack. Finally, the router firmware was updated during installation but has not been touched since.

With access to the Linksys established, we simply need to designate the remote computer (behind the router) as the DMZ Host. This designation fully exposes this system for attack and compromise. Our

next lucky break is that the remote system was installed without a local administrator password. (This is the default for the automatic Compaq workstation setup and many systems are out there with blank administrator passwords). To fully control the system, we map a drive to the administrative share on the c: drive (\\ipaddress\c$) and logon using the user machine-name\administrator with a blank password. Completing the full compromise of the remote workstation is accomplished by placing a custom subseven executable in the C:\Documents and Settings\All Users\Start Menu\Programs\Startup folder. The next time the system is restarted, the Trojan will be installed and full control available to us. Once full control is available, we can go back into the Linksys configuration to remove the DMZ Host entry, add port forwarding for access to the Subseven server and disable remote administration. Finally, we will add a password to the local Adminstrator account. All these changes are designed to prevent someone else from compromising this system and stealing control from us.

With the remote workstation fully under our control, our entry into the GIAC network has been completed. However, as vpn access is limited to the File Server, we could continue escalation of privileges by now attacking this core server.

Accessing administrative privileges on the main file server will not be as simple as the remote workstation. This system is under direct control of IT security and does not have blatant holes. However, as we currently have control of a system on the network, there is much we can do. Our first goal is to obtain access to an administration account for the internal domain.

With full control of the remote system we can install and run applications at will. Our first step is to install Lophtcrack from @stake (http://www.atstake.com/research/lc/index.html). Then with the packet capture routing running, we send emails to the most likely accounts (tech support email address are most useful) as noted in the LC4 faq at http://www.atstake.com/research/lc/faq25.html#hashes_sniffing_4. Once this email is opened we will have the password hash of the user and can begin running a crack against it. With some luck we will be able to crack the password before it expires and is changed. We can then access the File server (with an even higher level of privilege) and continue our incursion into GIAC systems.

Our initial target was chosen because it is located away from the main GIAC offices but has privileged network access into systems at the main office. It is typically difficult for IT security to properly control, manage and monitor these systems so our attacks are more likely to be successful and go undetected. From this system, our attacks proceeded in an effort to escalate privilege via whatever target seems the most opportune. As the only internal system we could access is the file server that was the next target of choice.

Attacking the File server using exploits and probes is dangerous as this system is most likely closely monitored by knowledgeable people and Intrusion detection systems. Therefore we chose a more passive approach to gain a username and password for access. Once we have administrative privileges it is much easier to continue exploration without setting off alarms.

Protection against this Attack

The main issue that enabled this attack is poor management of remote security systems and services. This is very difficult to address due to physical distance, local ownership of systems or services and a lack of expertise at the remote sites. However, when possible IT should get as close to the following recommendations as possible;

- All Security hardware must be specified, purchased and configured by trained IT personnel.
- Consideration must be given to how remote management will be accomplished securely on all remote devices.
- Any remote security device must be able to provide logging back to a central facility.
- All remote computers must be configured by IT to meet the same requirements as systems on the internal network.
- All passwords must meet specific guidelines and be changed at a schedule designed to prevent brute-force compromise.

Any of these recommendations would have prevented the attack outlined above. The key is to ensure that remote systems have security postures comparable to machines at the main office.

Conclusions

From this review, it seems that the GIAC network investigated is pretty well locked up. The most likely avenue for a significant compromise appears to be via the remote offices. As their configuration is not discussed in detail and they use local ISP's for email and FTP, it is possible that they are not secured as tightly and would offer better opportunities. In general though, this is a tight ship.

Appendix A
Cisco 2514 Router Configuration

The border router configuration is designed to provide basic filtering in addition to controlling communications through the ISP to the Internet. Within the design below, many of the disabled servers and services are off by default in current IOS versions. However, as router configurations are frequently used as templates and specifically shutting them off again incurs no penalty, the necessary steps are included.

The actual router commands are shown in black text. Annotations and comments are shown in blue.

!Enable date and time stamping of logs using the local timezone
service timestamps debug uptime
service timestamps log datetime localtime
!
!Disable unneeded servers and services
no boot network *– In conjunction with "no service config" this disables loading configuration data from the network.*
no service config
no cdp enable *– Disables the "Cisco Discovery Protocol". The service is not needed.*
no ip http server *– Disables the unneeded http server.*
no service tcp-small-servers *– Disables unneeded servers (Echo, Chargen, et al).*
no service udp-small-servers *– Disables unneeded servers (Echo, Chargen, et al).*
no ip boot server *– Disables the unneeded bootp server.*
no ip finger *- Disables the unneeded finger command.*
no ip domain-lookup *- Disable unneeded DNS resolution on the router itself.*
no ip source-route *- Disable unneeded source routing.*
no snmp-server enable traps *- In conjunction with "no snmp-server" this disables the unneeded snmp server.*
no snmp-server
no proxy-arp *– Proxy arp can reveal internal addresses and we do not require it.*
no ip classless *– Disable routing of packets without clearly defined subnets.*
ip route 0.0.0.0 0.0.0.0 1.1.1.1 *– Sets the default gateway to the ISP router.*
ip route 1.2.3.0 255.255.255.248 10.130.1.2 *– Specifies the route to the service network through the primary firewall.*
ip route 10.130.3.2 255.255.255.255 10.130.1.2 *– Specifies the route to the central syslog server through the primary firewall.*
!
hostname border-router *- Sets the hostname of the router.*
process-max-time 200
!
! Configure password settings
service password-encryption *– Encrypt all passwords*
enable secret 1PGC4$SeBHeTAaSnUa5T4DxaftY1 *Configures the password required to enter enable mode on the router. Using "enable secret" stores the password in an encrypted format - "enable password" would store it as plain text.*
!
!Configure Interfaces

interface Ethernet0
 ip address 10.130.1.1 255.255.255.252 - *This sets an internal ip address scheme for the Ethernet connection to the primary firewall.*
ip access-group 102 in *– Filter all outgoing packets using access list 102*
no ip directed-broadcast *– Prevent forwarding of directed broadcasts. These have no use in the network and can be abused in DDOS attacks (specifically Smurf attacks)*.
!
interface Ethernet1 *– This interface is not in use so, it is disabled*
 no ip address
 shutdown
!
interface Serial0
 ip address 1.1.1.2 255.255.255.252 *– This sets the external ip address of the router.*
 ip access-group 101 in *– Filter all incoming packets using access list 101*
no ip unreachable *– Certain ICMP messages can be used in network mapping attempts. This statement disables one method.*
 no ip redirect *– Certain ICMP messages can be used in network mapping attempts. This statement disables one method.*
 no ip mask-reply *– Certain ICMP messages can be used in network mapping attempts. This statement disables one method.*
 no ip directed-broadcast *– Prevent forwarding of directed broadcasts. These have no use in the network and can be abused in DDOS attacks (specifically Smurf attacks)*.
ntp disable *– ntp is used on the internal network to synchronize all system clocks. As the router obtains its setting from an internal system, this service is not needed on the external interface.*
!
interface Serial1 *– This interface is not used so, it is disabled*
 no ip address
 shutdown
!
logging 10.130.3.2 *– Send all syslog messages to the Central Syslog server.*
!
!Access list 101 blocks illegal and unnecessary source ip addresses from entering the external interface of the router.
access-list 101 deny ip 0.0.0.0 0.255.255.255 any log
access-list 101 deny ip 10.0.0.0 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 169.254.0.0 0.0.255.255 any log
access-list 101 deny ip 172.16.0.0 0.31.255.255 any log
access-list 101 deny ip 192.0.2.0 0.0.0.255 any log
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
access-list 101 deny ip 224.0.0.0 15.255.255.255 any log
access-list 101 deny ip 240.0.0.0 7.255.255.255 any log
access-list 101 deny ip 248.0.0.0 7.255.255.255 any log
access-list 101 permit ip any 1.2.3.0 0.0.0.7
access-list 102 deny ip any any

!
!Access list 102 prevents all but acceptable internal IP addresses from entering the internal interface.

```
access-list 102 permit ip 1.2.3.0 0.0.0.7 any
access-list 102 deny   ip any any
!
Access list 103 prevents telnet access to the router from anywhere except the internal network (NAT
will result in all packets appearing to come from the external interface of the internal firewall).
access-list 103 permit tcp 1.2.3.4 0.0.0.0  10.130.1.1 0.0.0.0 eq 23
access-list 103 deny   ip any any
!
line con 0 – Console port configuration
 transport input none – No remote transports enabled
 password 7 133712010E0D162923047A676C – Set the access password
 login – Require a login for access
line aux 0 – Auxiliary port configuration
 transport input none - No remote transports enabled
 no exec – No exec interface available (essentially this port is port is disabled)
line vty 0 3 – Shutdown 4 of the 5 virtual terminals
 login – Require a login for access
 transport input none – Don't allow any transports access
line vty 4 – Configure a single virtual terminal for access
 access-class 103 in – Apply Access-list 103 to incoming connections
 password 7 133712010E0D162923047A676C – Set the access password
 session-limit 1 – limit access to a single connection
 login – Require a login for access
 exec-timeout 5 0 – Timeout inactive sessions after 5 minutes
 transport input telnet – Only allow telnet connections

!
end
```

Appendix B

Primary Firewall Configuration

The primary firewall is designed to provide more specific filters than the border router. It is at this point that we begin blocking specific services. Within the instructions below annotations and comments are shown in blue.

Please note, use of any software-based firewall requires proper, secure configuration of the base server. Guidance for installing and configuring Red Hat Linux 8.0 is offered within the body of this document. For further reading, many good Linux hardening references are available online.

The order of the rules within each chain below is based upon assumptions regarding expected usage. Within netfilter, each packet must traverse its particular chain until it meets a matching rule. To speed the filtering process, the most frequently matched rules should be placed near the top of each chain. If specific traffic is considered high priority, it too may be placed high on the list. Of course the last two rules must be to log and deny any packets that have not matched previous rules.

```
#!/bin/sh
#
# Interfaces and Common Variables
#
# All IP Addresses, ports and other data are specified as variables.  In this way
any future changes to this information can be quickly incorporated into the
firewall.
# Define the interfaces, networks and firewall ip addresses
Any="0.0.0.0/0"
LoopIF="lo"
LoopIP="127.0.0.1"
EXTIF="eth0"
EXTIP="10.130.1.2"
EXTLAN="10.130.1.0/30"
INTIF="eth1"
INTIP="1.2.3.1"
INTLAN="1.2.3.0/29"
LOGIF=”eth2”
LOGIP=”10.130.3.2”
LOGLAN=”10.130.3.0/30”
#
# Define the Service network system ip addresses
ServiceNET=”1.2.3.0/29”
WEBServerIP="1.2.3.2"
MAILServerIP="1.2.3.3"
INTServerIP="1.2.3.4"
RouterIP=”10.130.1.1”
CentLogIP=”10.130.0.3”
#
# Define ip addresses and networks for the Remote Office and Clients.
ROffice=”3.2.1.1”
Client1=”1.1.5.0/24”
Client2=”1.1.6.0/24”
```

```
#
# Define other general IP addresses and variables
BCast="255.255.255.255/32"
MCAST="224.0.0.0/4"
IPTAB="/sbin/iptables"
#
echo statements are used after each major section to assist in isolating
typographical or other errors in the completed script.
echo "variables set"
#
# Flush chains, set policies, allow forwarding
If multiple interfaces or complex rule sets are used, it is frequently helpful to
break the rule base down into smaller increments.  In this specific case, this is
not required but, if it were, the additional chains would be created here as well.
#
$IPTAB -F   This statement flushes (erases) all the existing chains.
$IPTAB -t nat -F   This statement flushes (erases) the nat table.
The -P statements set the default policy for each chain to "DROP".  This will
cause any packets not expressly accepted to be ignored (no reply or error packet
sent).
$IPTAB -P INPUT DROP
$IPTAB -P FORWARD DROP
$IPTAB -P OUTPUT DROP
echo "1">/proc/sys/net/ipv4/ip_forward  This statement configures the system to
forward traffic between interfaces.  It is set after the Policy statements to
prevent creating wide open connections (even momentarily).
#
echo "Policies Set"
#
# Accept established connections - These lines allow reply traffic for connections
that are currently listed in the state tables.  In this way, the server can "self
manage" the return ports required for this traffic.  These are placed on top of
each chain in the interest of performance.  As each packet must traverse the chain
until it matches, the busiest statements should be evaluated first.
#
$IPTAB -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTAB -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTAB -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
#
echo "State set"
#
#
#Configure Rules for the Input Chain - The Input chain controls all traffic
destined for the firewall itself.
#
$IPTAB -A INPUT -d $LoopIP -j ACCEPT - This accepts any traffic from the local
loop IP address.  This is required for internal server processes.
#
The following two statements log and then drop any traffic which uses an IP
address from one interface but arrives on another.  The log entry is highlighted
with "Alert" as this should not occur in normal use.
$IPTAB -A INPUT -s $INTLAN -i ! $INTIF -j LOG --log-level info --log-prefix
"ALERT: "
$IPTAB -A INPUT -s $EXTLAN -i ! $EXTIF -j LOG --log-level info --log-prefix
"ALERT: "
$IPTAB -A INPUT -s $LOGLAN -i ! $LOGIF -j LOG --log-level info --log-prefix
"ALERT: "
```

```
$IPTAB -A INPUT -s $INTLAN -i ! $INTIF -j DROP
$IPTAB -A INPUT -s $EXTLAN -i ! $EXTIF -j DROP
$IPTAB -A INPUT -s $LOGLAN -i ! $LOGIF -j DROP
#
# Drop common packets that we don't want to log - These are common packets that we
expect on the internal interface.  They will quickly fill our logs if not dropped
prior to the log statement.
#
$IPTAB -A INPUT -p tcp --dport 137 -j DROP
$IPTAB -A INPUT -p tcp --dport 138 -j DROP
$IPTAB -A INPUT -p tcp --dport 139 -j DROP
$IPTAB -A INPUT -p udp --dport 137 -j DROP
$IPTAB -A INPUT -p udp --dport 138 -j DROP
$IPTAB -A INPUT -p udp --dport 139 -j DROP
$IPTAB -A INPUT -p icmp -j DROP
$IPTAB -A INPUT -d $MCAST -j DROP
#
# Allow ssh traffic from internal systems
#
$IPTAB -A INPUT -p tcp -s $INTServerIP --dport 22 -j LOG --log-level info --log-
prefix "Alert: " - All ssh traffic should be from the System Administrator.
Logging this traffic with an alert will ensure notification if this protection is
breached.
$IPTAB -A INPUT -p tcp -s $INTServerIP --dport 22 -j ACCEPT
#
# Allow NTP broadcasts on the Internal interface
#
$IPTAB -A INPUT -p udp -i $INTIF -d $BCast --dport 123 -j ACCEPT
#
$IPTAB -A INPUT -d $BCast -j DROP - This is placed later in the INPUT chain to
allow receipt of NTP broadcasts from the Mail gateway.
# Log and Drop everything else - Technically the final drop should be handled by
the policy for the chain but adding an additional drop incurs no penalty.
#
$IPTAB -A INPUT -j LOG --log-level info --log-prefix "INPUT: "
$IPTAB -A INPUT -j DROP
#
echo "INPUT done"
#
# Configure Rules for the Output Chain - The Output chain controls all packets
originating from the firewall.  The only traffic planned is syslog data sent to
the central server, SMTP to the Email gateway for reporting and local loop traffic
for internal processes.
#
$IPTAB -A OUTPUT -p icmp -j DROP - We do not want to log excessive icmp rejects
$IPTAB -A OUTPUT -p udp -d $CentLogIP --dport 514 -j ACCEPT
$IPTAB -A OUTPUT -p tcp -d $MAILServerIP --dport 25 -j ACCEPT
$IPTAB -A OUTPUT -d $LoopIP -j ACCEPT
#
# The following statements are used for upgrading the firewall itself.  Normally
they are disabled.
#$IPTAB -A OUTPUT -p tcp --dport 21 -j ACCEPT
#$IPTAB -A OUTPUT -p tcp --dport 53 -j ACCEPT
#$IPTAB -A OUTPUT -p udp --dport 53 -j ACCEPT
#$IPTAB -A OUTPUT -p tcp --dport 80 -j ACCEPT
#$IPTAB -A OUTPUT -p tcp --dport 443 -j ACCEPT
#
```

```
# Log and Drop everything else
#
$IPTAB -A OUTPUT -j LOG --log-level info --log-prefix "OUTPUT: "
$IPTAB -A OUTPUT -j DROP
#
echo "OUTPUT done"
#
# Configure Rules for the FORWARD Chain - The Forward chain controls traffic that
arrives on one interface destined for another.  This is where all of our detailed
controls are placed.
#
# Again, drop the traffic we do not wish to log and traffic not on the correct
interface.
$IPTAB -A FORWARD -p tcp --dport 137 -j DROP
$IPTAB -A FORWARD -p tcp --dport 138 -j DROP
$IPTAB -A FORWARD -p tcp --dport 139 -j DROP
$IPTAB -A FORWARD -p udp --dport 137 -j DROP
$IPTAB -A FORWARD -p udp --dport 138 -j DROP
$IPTAB -A FORWARD -p udp --dport 139 -j DROP
$IPTAB -A FORWARD -d $BCast -j DROP
$IPTAB -A FORWARD -d $MCAST -j DROP
$IPTAB -A FORWARD -s $INTLAN -i ! $INTIF -j DROP
#
# Set the specific traffic allowed
# The order of the statements is based on anticipated traffic load and priority of
the traffic (highest load/priority are checked first).
$IPTAB -A FORWARD -p tcp -s $MAILServerIP --dport 53 -j ACCEPT DNS from the mail
server
$IPTAB -A FORWARD -p udp -s $MAILServerIP --dport 53 -j ACCEPT DNS from the mail
server
$IPTAB -A FORWARD -p tcp -d $WEBServerIP --dport 80 -j ACCEPT HTTP to the web
server
$IPTAB -A FORWARD -p tcp -d $MAILServerIP --dport 25 -j ACCEPT SMTP to the mail
server
$IPTAB -A FORWARD -p tcp -s $MAILServerIP --dport 25 -j ACCEPT SMTP from the mail
server
$IPTAB -A FORWARD -p udp -s $MAILServerIP --dport 123 -j ACCEPT NTP requests from
the mail server
# Accept https traffic from designated locations to the web server
$IPTAB -A FORWARD -p tcp -d $WEBServerIP -dport 443 -j LOG --log-level info --log-
prefix "HTTPS: " - As all incoming HTTPS traffic is restricted, logging all
traffic can be of use in tracking problems or resolving customer disputes.
$IPTAB -A FORWARD -p tcp -s $ROffice -d $WEBServerIP --dport 443 -j ACCEPT
$IPTAB -A FORWARD -p tcp -s $Client1 -d $WEBServerIP --dport 443 -j ACCEPT
$IPTAB -A FORWARD -p tcp -s $Client2 -d $WEBServerIP --dport 443 -j ACCEPT
# Allow approved communications out from the internal firewall
$IPTAB -A FORWARD -p tcp -s $INTServerIP --dport 21 -j ACCEPT
$IPTAB -A FORWARD -p tcp --s $INTServerIP --dport 22 --dport 22 -j LOG --log-level
info --log-prefix " LOGON: "
$IPTAB -A FORWARD -p tcp -s $INTServerIP --dport 22 -j ACCEPT
$IPTAB -A FORWARD -p tcp -s $INTServerIP -d $RouterIP --dport 23 -j ACCEPT
$IPTAB -A FORWARD -p tcp -s $INTServerIP --dport 80 -j ACCEPT
$IPTAB -A FORWARD -p tcp -s $INTServerIP --dport 443 -j ACCEPT
$IPTAB -A FORWARD -p udp -s $INTServerIP --dport 161 -j ACCEPT
# Allow IPSEC tunnels to the internal firewall
$IPTAB -A FORWARD -p 50 -d $INTServerIP -j ACCEPT
$IPTAB -A FORWARD -p udp -d $INTServerIP --dport 500 -j ACCEPT
```

```
# Syslog traffic from the remote office and the router is allowed to the syslog
server
$IPTAB -A FORWARD -p udp -s $ROffice -d $CentLogIP --dport 514 -j ACCEPT
$IPTAB -A FORWARD -p udp -s $Router -d $CentLogIP --dport 514 -j ACCEPT
# Connections to the Syslog server are allowed from the Administrator's
workstation and are logged
$IPTAB -A FORWARD -p tcp -o $LOGIF -s $INTServerIP -d $CentLogIP --dport 22 -j LOG
--log-level info --log-prefix " LOGON: "
$IPTAB -A FORWARD -p tcp -o $LOGIF -s $INTServerIP -d $CentLogIP --dport 22 -j
ACCEPT
#
# Any approved http, https and ftp traffic from the Service network should have
already matched a rule and been accepted.  The logging below is included to detect
unapproved connection attempts from systems on the service network and generate an
alert.  These will fire when systems are being configured but, as that is a
scheduled event, it can be managed.
$IPTAB -A FORWARD -p tcp -s $ServiceNet --dport 80 -j LOG --log-level info --log-
prefix "ALERT: "
$IPTAB -A FORWARD -p tcp -s $ServiceNet --dport 21 -j LOG --log-level info --log-
prefix " ALERT: "
$IPTAB -A FORWARD -p tcp -s $ServiceNet --dport 443 -j LOG --log-level info --log-
prefix " ALERT: "
# These statements may be activated while systems are being installed or updated
within the service network.  However, they must be disabled again once that work
is complete.
#$IPTAB -A FORWARD -p tcp -s $ServiceNET --dport 80 -j ACCEPT
#$IPTAB -A FORWARD -p tcp -s $ServiceNET --dport 21 -j ACCEPT
#$IPTAB -A FORWARD -p tcp -s $ServiceNET --dport 443 -j ACCEPT
# ICMP traffic is allowed from the Administrator workstation
$IPTAB -A FORWARD -p icmp -s $INTServerIP -j ACCEPT
$IPTAB -A FORWARD -p icmp -j DROP - Drop all other icmp traffic without logging it
#
$IPTAB -A FORWARD -j LOG --log-level info --log-prefix "FORWARD: "
$IPTAB -A FORWARD -j DROP
#
echo "FORWARD Done"
```

Appendix C
Internal Firewall/VPN Gateway Configuration

The internal firewall is provided to block all unauthorized access to the internal network and control outgoing traffic. The Netscreen product has been chosen both for its attributes and the simple fact that it is different than the primary firewall. As they are connected serially, this inclusion of disparate technologies requires a hacker to find vulnerabilities in both the Linux/Netfilter and the Netscreen firewall to obtain internal system access. The roles of this firewall are;

1.) Block all externally-initiated, non-VPN access.
2.) Allow internal access via authenticated VPN tunnels.
3.) Restrict use of unauthorized services from internal machines.

The Netscreen devices offer an html-based GUI configuration tool. This can simplify configuration and management but, is difficult to document as part of this appendix. Therefore, the discussion below depicts all configuration commands using the Command Line Interface (CLI). A good reference for Netscreen configuration using CLI version 3.0 may be found at http://www.sans.org/rr/firewall/netscreen.php. A resource for hardening CLI version 3.1 is available at http://www.qorbit.net/documents/screenos-hardening-appnote.htm and source reference for all current CLI commands is available at http://www.netscreen.com/support/downloads/CLI_4_0_0_RevG.pdf

The instructions below use many recommendations from the references cited above and, as with the other appendixes, annotations and comments are shown in blue.

First, set a custom administrator name and password. This should be completed prior to connecting the firewall to any network.
Set admin name gentmain
Set admin password 2Xsdm@!n0fS
Enable SSH communications
Set scs enable
SNMP is used by MRTG (http://people.ee.ethz.ch/~oetiker/webtools/mrtg/) to report on bandwidth utilization on the Systems Administrator's workstation. The following commands enable SNMP, ensure the default settings are not used and limit connections to the address 10.130.0.3 (Systems Administrator's workstation).
Set snmp name Janus
Set snmp community c0nNx2@mN6oX Read-Only Trap-on
Set snmp host C0nNx2@mN6oX 10.130.0.3
Configure the domain name and dns servers
Set dns host dns11.2.3.3
Set domain gent.com
Direct logging to the central syslog server, set it to log all traffic and log access to the firewall itself
Set syslog config 10.130.3.2 local0 local0 debug
Set syslog enable
Set syslog traffic
Set log-self
Set log-self ike

Set log-self snmp
Configure the trusted Ethernet interface for the internal (trusted) network.
Set interface trusted zone trust
Set interface trusted ip 10.130.0.254/24
Set interface trusted route
Set interface trusted manage ping
Set interface trusted manage scs
Set interface trusted manage snmp
Unset interface trusted manage telnet
Unset interface trusted manage global
Unset interface trusted manage global-pro
Unset interface trusted manage ssl
Unset interface trusted manage web
Unset interface trusted manage ident-reset
Configure the second interface for the external (untrusted) network
Set interface untrusted zone untrust
Set interface untrusted ip 1.2.3.4/29
Unset interface untrusted manage ping
Unset interface untrusted manage scs
Unset interface untrusted manage snmp
Unset interface untrusted manage telnet
Unset interface untrusted manage global
Unset interface untrusted manage global-pro
Unset interface untrusted manage ssl
Unset interface untrusted manage web
Unset interface untrusted manage ident-reset
Configure NTP Time settings so that all security logs operate on a common time signal. The email server synchronizes externally to the gent network and all other systems synchronize from it.
Set ntp timezone 0
Set ntp server 1.2.3.3
Set clock timezone 0
Set clock ntp
Configure the VPN to the Remote Sales office – the configuration settings shown are the defaults and considered completely adequate for this situation. The proposal settings indicate that we are using a Preshared key (pre-) Diffie-Hellman (D-H) Group 2 (g2-), 3DES encryption (3des-), SHA-1 integrity/authentication (sha)). The name of the gateway is "roffice".
Set ike gateway roffice ip 3.2.1.1 main preshare M@ke7hisareallyHardpassw0rd proposal pre-g2-3des-sha
The proposal-2 settings are similar, calling for Diffie-Hellman (D-H) Group 2 (g2-), ESP-3DES encryption (3des-) with MD5 integrity/authentication (md5)
Set vpn roffice gateway roffice tunnel proposal g2-esp-3des-md5
Name the Interfaces
Set address trust Internal 10.130.0.0 255.255.255.0 "Internal"
Set address untrust Service 1.2.3.0 255.255.255.248 "Service"
Set address untrust Remote 10.130.2.0 255.255.255.0 "Remote"
Configure policies to allow traffic between the two offices
Set policy from trust to untrust Internal Remote any tunnel vpn roffice

Set policy from untrust to trust Internal Remote any tunnel vpn roffice
Configure remote connection settings for the salesman and the owner
First create the users using Fully Qualified Domain Names (fqdn)
Set user sales ike-id u-fqdn sales@gent.com share-limit 1
Set the user type and enable the user
Set user sales type ike
Set user sales enable
Set user owner ike-id u-fqdn owner@gent.com share-limit 1
Set user owner type ike
Set user owner enable
Add the new users to the dialup group
Set dialup ireclients + sales
Set dialup ireclients + owner
Set the initial authentication parameters in a similar manner to the remote office. Some additions are to ensure operation if they are behind a NAT device (nat-traversal)
Set ike gateway remoteusers dialup ireclients main preshare @n0therReal14hAr6oNe proposal pre-g2-3des-sha
Set ike gateway remoteusers nat-traversal udp-checksum
Set ike gateway remoteusers net-traversal keepalive-frequency 5
Set vpn remoteusers gateway remoteusers no-replay tunnel proposal g2-esp-3des-md5
Set policy from untrust to trust "Dial-Up VPN" Internal any tunnel vpn-dialup
Set ike respond-bad-spi 1
Set ike id-mode subnet
Unset ike policy-checking
Unset ike accept-all-proposal
Block all traffic by default
Set zone untrust block
Set zone trust block
Allow NTP requests from the internal domain controller
Set address trust domcon 10.130.1.2 255.255.255.255 "Domain Controller"
Set policy from trust to untrust domcon out-any ntp nat permit
Allow DNS requests from the internal DNS server (Domain Controller) to the DNS Cache server in the Service network
Set address untrust mailserve 1.2.3.3 255.255.255.255 "Mail Server"
Set policy from trust to untrust domcon mailserve dns nat permit
Allow internal traffic as listed in the body of the paper and the established network policies
Set policy from trust to untrust Internal out-any http nat permit
Set policy from trust to untrust Internal out-any https nat permit
Set policy from trust to untrust Internal out-any ftp nat permit
Set address trust admin 10.130.1.3 255 255 255 255 "Admin Workstation"
Set policy from trust to untrust admin out-any ping nat permit
Set policy from trust to untrust admin out-any telnet nat permit log
Set policy from trust to untrust admin out-any ssh nat permit log
Set policy from trust to untrust admin Service snmp nat permit log
Set address trust xchng 10.130.0.1 255.255.255.255 "Exchange Server"
There is a custom route on the service network email gateway to direct smtp traffic through the internal firewall.

Set policy from trust to untrust xchng mailserve smtp permit
Set policy from untrust to trust mailserve xchng smtp permit
The next section sets the firewall to block known bad traffic. The options available are much more extensive than included here but, for a low traffic, low visibility site, these are a cost-effective assortment.
Set zone untrust screen fin-no-ack - *Reject illegal flags*
Set zone untrust screen icmp-fragment - *Reject fragmented icmp packets*
Set zone untrust screen icmp-large - *Reject icmp packets >1024*
Set zone untrust screen ip-bad-option - *Reject frames with malformed or incomplete options*
Set zone untrust screen ip-filter-src - *Reject packets with Source Routing enabled*
Set zone untrust screen ip-loose-src-route - *Reject packets with Loose Source routing set*
Set zone untrust screen ip-spoofing - *Reject false source ip addresses*
Set zone trust screen ip-spoofing - *Reject false source ip addresses – this is valid on all interfaces*
Set zone untrust screen ip-strict-src-route - *Reject packets with Strict Source routing set*
Set zone untrust screen ping of death - *Reject packets with Loose Source routing set*
Set zone untrust screen port-scan – *Block further access from a remote IP that scans an excessive number of ports*
Set zone untrust screen syn-ack-ack-proxy - *Reject syn-ack-ack attacks*
Set zone untrust screen syn-fin - *Reject illegal flag combinations*
Set zone untrust screen syn-frag - *Reject excessive fragmented syn packets*
Set zone untrust screen tcp-no-flag - *Reject illegal packets with missing or malformed flags field*
Set zone untrust screen tear-drop – *Block the Tear drop attack*
Set zone untrust screen unknown-protocol - *Reject protocol id's > 100*
Set zone untrust screen winnuke – *Detect winnuke attacks*

Appendix D

Remote Sales Office Firewall/VPN Gateway Configuration

The Remote Office firewall is very similar to the internal firewall at the main office. The main differences are that it does not need to accept incoming ad-hoc VPN tunnels and it does not have additional security layers.

First, set a custom administrator name and password. This should be completed prior to connecting the firewall to any network.
Set admin name gentremote
Set admin password 4aw@yfr0mhom3
Enable SSH communications
Set scs enable
SNMP is used by MRTG (http://people.ee.ethz.ch/~oetiker/webtools/mrtg/) to report on bandwidth utilization on the Systems Administrator's workstation. The following commands enable SNMP and ensure the default settings are not used.
Set snmp name Gabriel
Set snmp community c0nNx2@mN6oX Read-Only Trap-on
Set snmp host C0nNx2@mN6oX 1.2.3.4 - *SNMP connections from the admin station will be "natted" to the external ip of the Internal firewall.*
Configure the domain name and dns servers
Set dns host dns11.2.3.3.224 – *This is an ISP dns server*
Set domain gent.com
Set route 10.130.3.2 255.255.255.255 gateway 1.1.1.2 – *route syslog traffic to the Border router, it knows the next hop to the Central Syslog server*
Direct logging to the central syslog server, set it to log all traffic and log access to the firewall itself
Set syslog config 10.130.3.2 local0 local0 debug
Set syslog enable
Set syslog traffic
Set log-self
Set log-self ike
Set log-self snmp
Configure the first Ethernet interface for the internal (trusted) network.
Set interface trusted zone trust
Set interface trusted ip 10.130.2.0/24
Set interface trusted route
Set interface trusted nat
Set interface trusted manage ping
Set interface trusted manage scs
Unset interface trusted manage snmp
Unset interface trusted manage telnet
Unset interface trusted manage global
Unset interface trusted manage global-pro
Unset interface trusted manage ssl
Unset interface trusted manage web
Unset interface trusted manage ident-reset

Configure the second interface for the external (untrusted) network
Set interface untrusted zone untrust
Set interface untrusted ip 3.2.1.2/29
The advantage for the admin to be able to ping and administer the remote firewall from the home office outweighs the desire for it to be invisible in this instance.
Set interface untrusted manage ping
Set interface untrusted manage scs
Set interface untrusted manage snmp
Unset interface untrusted manage telnet
Unset interface untrusted manage global
Unset interface untrusted manage global-pro
Unset interface untrusted manage ssl
Unset interface untrusted manage web
Unset interface untrusted manage ident-reset
Configure NTP Time settings so that all security logs operate on a common time signal.
Set ntp timezone 0
Set ntp server 1.2.3.3
Set clock timezone 0
Set clock ntp
Configure the VPN to the Main office identically to the settings at the Main Office (Appendix C)
Set ike gateway roffice ip 3.2.1.2 main preshare secret1 proposal pre-g2-3des-sha
Set vpn roffice gateway roffice tunnel proposal g2-esp-3des-md5
Set address trust Internal 10.130.0.0 255.255.255.0 "Internal"
Set address untrust Service 1.2.3.0 255.255.255.248 "Service"
Set address untrust Remote 10.130.2.0 255.255.255.0 "Remote"
Set policy from trust to untrust Internal Remote any tunnel vpn roffice
Set policy from untrust to trust Internal Remote any tunnel vpn roffice
Set ike respond-bad-spi 1
Set ike id-mode subnet
Unset ike policy-checking
Unset ike accept-all-proposal
Block all traffic by default
Set zone untrust block
Set zone trust block
Allow internal traffic as listed in the body of the paper and the established network policies
Set policy from trust to untrust Internal out-any http permit
Set policy from trust to untrust Internal out-any https permit
Set policy from trust to untrust Internal out-any ftp permit
Set policy from trust to untrust Internal out-any dns permit
The next section sets the firewall to block known bad traffic. The options available are much more extensive than included here but, for a low traffic, low visibility site, these are a cost-effective assortment.
Set zone untrust screen fin-no-ack - *Reject illegal flags*
Set zone untrust screen icmp-fragment - *Reject fragmented icmp packets*
Set zone untrust screen icmp-large - *Reject icmp packets >1024*
Set zone untrust screen ip-bad-option - *Reject frames with malformed or incomplete options*
Set zone untrust screen ip-filter-src - *Reject packets with Source Routing enabled*

Set zone untrust screen ip-loose-src-route - *Reject packets with Loose Source routing set*
Set zone untrust screen ip-spoofing - *Reject false source ip addresses*
Set zone trust screen ip-spoofing - *Reject false source ip addresses – this is valid on all interfaces*
Set zone untrust screen ip-strict-src-route - *Reject packets with Strict Source routing set*
Set zone untrust screen ping of death - *Reject packets with Loose Source routing set*
Set zone untrust screen port-scan – *Block further access from a remote IP that scans an excessive number of ports*
Set zone untrust screen syn-ack-ack-proxy - *Reject syn-ack-ack attacks*
Set zone untrust screen syn-fin - *Reject illegal flag combinations*
Set zone untrust screen syn-frag - *Reject excessive fragmented syn packets*
Set zone untrust screen tcp-no-flag - *Reject illegal packets with missing or malformed flags field*
Set zone untrust screen tear-drop – *Block the Tear drop attack*
Set zone untrust screen unknown-protocol - *Reject protocol id's > 100*
Set zone untrust screen winnuke – *Detect winnuke attacks*

References

Product Pages

Netscreen Firewall appliances - http://www.netscreen.com/products/fw_vpn_appliances.html
Cisco Routers - http://www.cisco.com/en/US/products/hw/routers/ps233/index.html
Zone Alarm - http://www.zonelabs.com/store/application;JSESSIONID_ZL_STORE_COMMERCE=2Z1jhIN0OoiZCohSi13OmslVXJ2uPQ17a67fjgXluAAmzbpkG65J!1149884956!173109956!7511!7512?namespace=zls_main&origin=global.jsp&event=link.catalogHome&&zl_catalog_view_id=201
Sophos Antivirus - http://www.sophos.com/products/software/
MDaemon - http://www.altn.com/Products/Default.asp?product_id=MDaemon
Microsoft – http://www.microsoft.com
PGP - http://www.pgpi.org/
Bastille-Linux - http://www.bastille-linux.org/
Tripwire - http://www.tripwire.org/
Swatch - http://swatch.sourceforge.net/
Linksys – http://www.linksys.com
MRTG - http://people.ee.ethz.ch/~oetiker/webtools/mrtg
VNC - http://www.uk.research.att.com/vnc/

Tools Used

NmapWin - http://www.nmapwin.org/
Ethereal - http://www.ethereal.com
Hping2 - http://www.hping.org/
URLScan/IIS Lockdown - http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/urlscan.asp
NMap - http://www.insecure.org/nmap/
Nessus - http://www.nessus.org/
http://nessuswx.nessus.org
Snort - http://www.snort.org/
L0phtcrack - http://www.atstake.com/research/lc/index.html
http://www.atstake.com/research/lc/faq25.html#hashes_sniffing_4

Miscellaneous

Linux Kernel How to – http://www.tldp.org/HOWTO/Kernel-HOWTO-2.html
Tripwire configuration - http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/ref-guide/ch-tripwire.html.
Firewall Auditing - http://www.spitzner.net/audit.html.
GCFW Practical by Keith Pachulski - http://www.giac.org/practical/Keith_Pachulski_GCFW.doc
CIAC Security Bulletin - http://www.ciac.org/ciac/bulletins/l-106.shtml
Cisco Security Bulletins - http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html
http://www.cisco.com/warp/public/707/SSH-scanning.shtml
Network Solution whois - http://www.networksolutions.com/cgi-bin/whois/whois
ARIN whois - http://ws.arin.net/cgi-bin/whois.pl
CoreLabs Security Bulletin - http://www1.corest.com/common/showdoc.php?idx=276&idxseccion=10
Public Proxy Servers - http://www.publicproxyservers.com/index.html

Cert Security Bulletin - http://www.kb.cert.org/vuls/id/945216
Netfilter Syntax - http://www.knowplace.org/netfilter/syntax.html
Password Policies - http://www.adpc.purdue.edu/BSC-Pete/ARIBA/passwrds.htm
InfoWorld Article - http://www.infoworld.com/article/02/03/01/020304neantivirus_1.html?Template=/storypages/ctozone_story.html
Netscreen Configuration - http://www.sans.org/rr/firewall/netscreen.php
Netscreen Hardening - http://www.qorbit.net/documents/screenos-hardening-appnote.htm
Netscreen CLI Documentation - http://www.netscreen.com/support/downloads/CLI_4_0_0_RevG.pdf
Ssh crc32 exploit analysis - http://www.linuxsecurity.com/articles/intrusion_detection_article-4002.html