GIAC GCFW Practical Version 4.1

A Hardware Solution to Buffer-Overflow Attacks

By Mark K. Miller

April 19, 2005

Table of Contents

Assignment 1: A Hardware Solution to Buffer- Overflow Attacks

Abstract

A fundamental flaw in the design of the PC, coupled with poor programming practices, has led to the success of countless numbers of buffer-overflow attacks. A solution could be found by developing a new personal computer bus subsystem that would no longer allow these malicious instructions to be executed by the central processing unit. Computers then would be invulnerable to buffer-overflow attacks, which are one of the major problems that cause the entire PC platform to be insecure.

This solution would apply to all PC-based systems that provide computing, networking, or security services. Major security advantages could be realized at the workstation, server, and perimeter security device levels.

Introduction

The original PC design followed a simple concept that became a standard in the days when the PC was first conceived and introduced. Initially, the idea was to build a personal computer that was affordable for the masses. The envisioned PC would enable home users and corporate employees alike to do their work in a much more productive and enjoyable fashion. Computers proved their value in large companies, universities, and government concerns. If they increased user productivity in those environments, why wouldn't they prove just as useful for everyone else?

It appears that, originally, all development efforts were devoted to functionality, which was the driving force that led to the popularity of the PC. No thought was given to the idea that someday security might become a major concern. Personal computers were in fairly widespread use for twelve to fifteen years before the concept of connecting them to a network was widely accepted. The notion of "function versus function securely" was seldom given a second look, let alone serious consideration.

The original designs allowed a simple operating system (OS) to boot the computer and then hand control of the system to a single application the operator had chosen. This design was brilliantly simple. The idea was to develop and sell a single piece of hardware that could perform a variety of chores by simply loading a particular program into memory. If the user wanted the computer to do something different, they just needed to locate and install the new program. It was assumed that the users' demand for new programs would create a whole new industry. The overwhelming success of the PC stands as a testament to the correctness of that assumption.

A second condition exists that explains why the original design flaw has never been fixed. That condition is the absolute necessity of backwards compatibility. As the speed of the PC improved and new functionalities were added, the producers of those better technologies understood that their products would be embraced only if they were compatible with the systems that were already in place. In other words, if a company wanted to produce the latest and most advanced hardware, it had to support all of the software a prospective customer may have already acquired. Similarly, any new software that was introduced had to function correctly on standard hardware. The hardware has undergone a vast array of changes and improvements, but due to the necessity of backwards compatibility, the fundamental way in which instructions are processed has not changed.

With the advent of nearly all computers now being connected to the Internet, systems that formerly required no real security mechanisms have suddenly been exposed to attackers from around the world. Of all the attack methods that have evolved, the buffer overflow has consistently resided near the top of the popularity list. Its popularity lies in the fact that nearly all applications and operating systems have vulnerabilities that make them the ideal target for buffer-overflow attack.

Buffer Overflow Defined

Most applications created by programmers have buffers, which are areas in memory where user-supplied data can be stored. In the process of writing a program, space must be reserved so this data will have a dedicated place to reside. The size of the space must be decided during the creation of the program. When the finished program is running, and the user is required to input data, a specific amount of space in memory will have already been set aside to hold the data. As long as the user enters data that is of the correct format and length, the program behaves as expected.

If a longer than expected data string is entered, the part that will fit in the buffer will be placed there. However, the remainder of the string will be stored in the adjoining memory locations that were not reserved for this data. Many times, that area contains instructions that will be processed. By overrunning the buffer with a data string that contains executable code, a situation develops that will allow very undesirable results. One popular result is the spawning of a reverse shell that will provide the attacker a command prompt that belongs to the victim system. The prompt will possess the same system access privileges as the program that was overwritten by the buffer. Many programs run with full system privileges, so any command entered from the prompt will likely be carried out. These commands could allow the stealing of sensitive files, creating a new user account with administrator privileges, or destroying the file system.

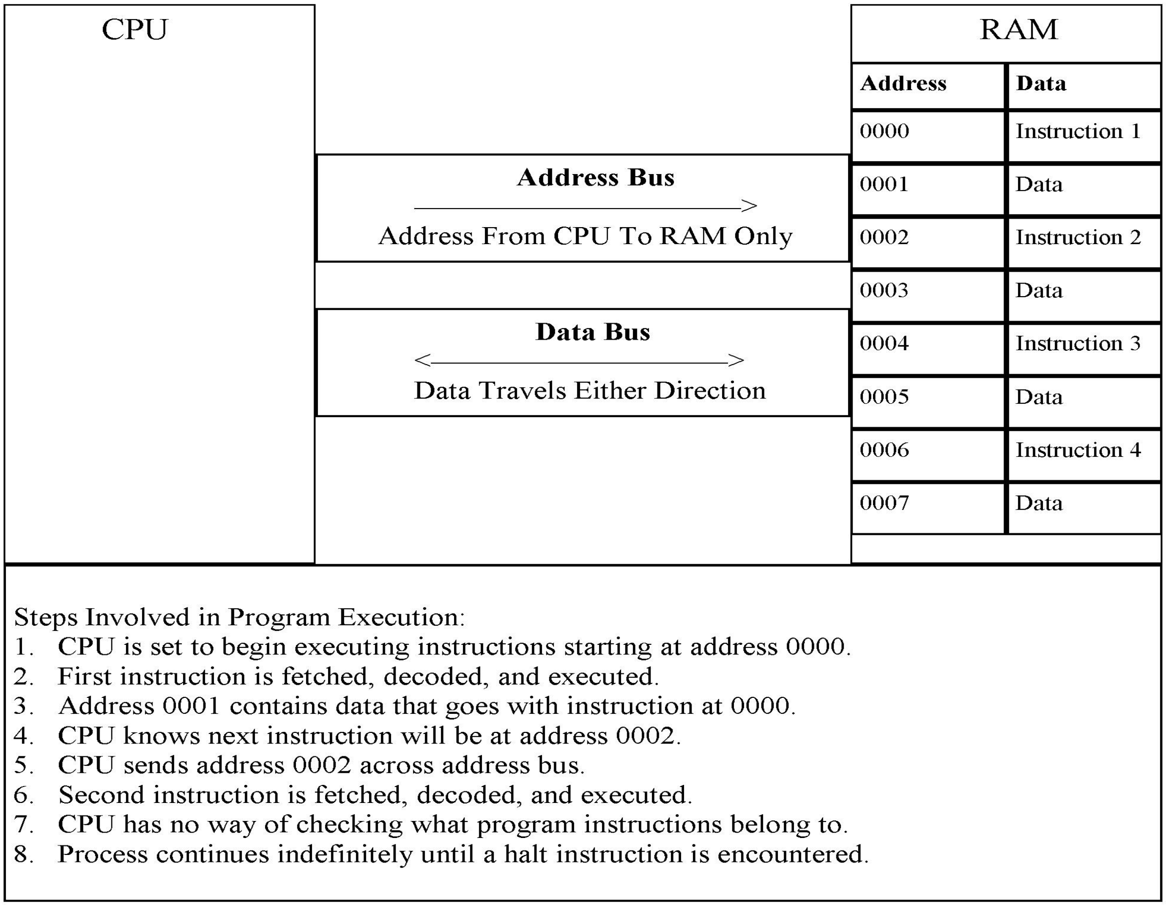
In the days before most computers were connected to the Internet, no thought was given to the necessity of checking the length of data provided by a user. For this reason, most programmers never developed the habit of forcing the program to analyze the data before writing it to memory. Some newer programming languages enforce checking the length of input data, but only work when they are used to create new programs. They provide no relief for programs that already exist or for new programs created with older programming languages.

The Flaw Exposed

A serious problem exists with our current method of dealing with buffer-overflow attacks. Vulnerabilities are discovered, and patches are developed that will alleviate the problem at hand. While a patch is being created and distributed, unethical internet citizens are hard at work building attack tools that will take advantage of the vulnerability. A high-stakes race ensues with each side hoping to finish first. If the production of a patch was the only requirement to secure all systems that possess the vulnerability, the side devoted to preventing the attacks would win a majority of the time.

The real problem with patches is not getting them developed, but rather getting them distributed and installed. In most environments, patches must be tested on non-production equipment to ensure they do not break other necessary functions. Because of the workload placed on many system administrators, it is not uncommon to hear about systems that are sixty days or more behind on patch installation [1]. There are patch management systems that can automatically install patches as they become available, but there is no way for these systems to test the patches before installing them.

The core of the problem lies in the way data travels unchecked between the central processing unit (CPU) and Random Access Memory (RAM). When a program is opened, its executable code is first copied from the hard drive through the system bus into RAM. The instructions are then fetched, one or more at a time, and executed by the CPU, which knows nothing about the instructions. It only knows what memory addresses they came from and how to execute them. This means that if a section of code from a legitimate program happens to get replaced by malicious instructions, the CPU will execute those instructions as if they were part of the legitimate program. This may happen without the computer user realizing that anything abnormal has transpired. The possible result of the malicious instructions is limited only by the imagination of the programmer who created them. Figure 1 illustrates the current model by showing a brief section of code from a legitimate program. The paragraph below Figure 1 gives more specific details of the actual processing.



Address	Data
0000	Instruction 1
0001	Data
0002	Instruction 2
0003	Data
0004	Instruction 3
0005	Data
0006	Instruction 4
0007	Data

Steps Involved in Program Execution:

1. CPU is set to begin executing instructions starting at address 0000.
2. First instruction is fetched, decoded, and executed.
3. Address 0001 contains data that goes with instruction at 0000.
4. CPU knows next instruction will be at address 0002.
5. CPU sends address 0002 across address bus.
6. Second instruction is fetched, decoded, and executed.
7. CPU has no way of checking what program instructions belong to.
8. Process continues indefinitely until a halt instruction is encountered.

Figure 1
Current Model of Legitimate Program Execution

Simple Program Example:
Instruction 1 = move data at address 0001 into CPU
Instruction 2 = add data at address 0003 to data already in CPU
Instruction 3 = add data at address 0005 to data already in CPU
Instruction 4 = halt processing
Data at address 0007 may be the top end of space reserved for a buffer.

The program shown in Figure 1 shows only four instructions. In the real world, such a short program could exist, but it would be limited to perhaps adding three numbers together. Normal programs would generally consist of thousands or millions of instructions. Single instructions at this low level only have the ability to do simple acts. Examples of simple acts might include moving data from one place to another, adding two numbers together, or jumping to a different address that contains another part of the program. If some of the legitimate instructions were changed by malicious code via a buffer-overflow exploit, the entire functionality of the program could be changed. What if getting the malicious code into RAM on someone's computer was not the only step required in order to get the code executed? Perhaps there needs to be another layer of protection against this kind of attack. The current PC architecture performs no monitoring of the instructions waiting in RAM to be executed; therefore, a change in the basic design of the system is needed.

How much time does the average system administrator spend hardening the OS on all important systems and constantly monitoring security websites for news of recently discovered vulnerabilities and exploits? How much time do they devote to testing and installing patches to prevent the new exploits from being executed on their systems? Many of them are shaking their heads and saying, “There’s got to be a better way.”

Existing Partial Solutions

A product known as SecureStack [2] offers protection from buffer-overflow attacks by adding a kernel mode driver to the system. This driver has the ability to monitor all memory operations and detect when a buffer overflow has occurred. The main problem with this solution is the number of both false positives and false negatives. Microsoft Office 2000 uses a form of code that the SecureStack solution interprets as a buffer-overflow attack.

StackGuard [3] is another product that claims to offer this sort of protection, but it can only be used with applications that have the source code available. Since Microsoft does not allow source code to ship with their products, they cannot take advantage of StackGuard. StackGuard is utilized in the Linux variant “Trustix OS 2” [4].

A truly different approach to buffer-overflow-attack prevention is detailed in a paper titled Randomized instruction set emulation to disrupt binary code injection attacks [5]. This paper is the product of the University of New Mexico, Department of Computer Science. A working model of the Randomized Instruction Set Emulation (RISE) [6] solution has been produced and tested with favorable results.

The RISE system works by assigning a secret key to each application that is allowed to run on the machine. This key is used to encrypt the instructions as they are loaded from the hard drive to memory. As the instructions are called for by the CPU, they are decrypted using the secret key and passed to the processor. Any code that was injected by way of a buffer-overflow attack would miss the process of being encoded before it landed in RAM. The process of decrypting the malicious code that had never been encrypted has the effect of turning the instructions into random binary patterns. When the CPU attempts to process the random patterns, some of them will represent valid but random instructions while others will generate an error and cause the program to abort.

This outcome follows the security industry standard for programs that fail to enforce security. That standard states that if a program fails, it should fail to a safe condition. In the event of an attempted buffer-overflow attack, the program would halt, and it would be obvious to the user that something had gone wrong. Many times, in the absence of a protection mechanism, a well-developed attack can be executed without any visible symptoms. In that case, the attacker has the ability to install a rootkit [7], thereby taking ownership of the machine and using it to exploit other systems on the network, while giving no indication of the attack.

The RISE system works well without creating false positives or false negatives, and it is not necessary to possess the source code of the running programs in order to implement this solution. One major drawback to this solution is the reduction in processing performance. Since the solution exists solely in software, the processor is responsible for the full implementation. A second shortcoming is the proprietary nature of the software emulator that performs the encryption and decryption. This solution would need to be developed and recompiled to run on each specific operating system.

These partial solutions are all software-based, and each of them has unique problems or limitations that must be dealt with. Developing a hardware-based solution with similar functionality to the RISE system could solve a number of these problems.

Proposed Flaw Remediation

Why not fix the problem where it starts. The assumption that any code that gets into RAM must be suitable for execution is long outdated. This would be similar to giving anyone free access to your house to pilfer through anything including financial documents, safes, and medicine cabinets. What if a bank locked their door at night but left all the money lying around in bags on the floor instead of carefully locking it away in the safe?

The idea of redesigning the traditional system bus, so that it operates like a Personal Computer Packet Inspection Bus (PCPIB), offers a major innovation in how data is stored in RAM and moved to the CPU. By doing the encryption and decryption in dedicated hardware, no additional load would be placed on the CPU. Also, by developing a hardware solution, it would not need to be reinvented for each operating system and application. Furthermore, if a hardware solution became standard issue on all PC-based computing systems, there would be nothing else to buy or install. A true turnkey solution could exist, pertaining to buffer-overflow-attack prevention.

If data were loaded into RAM the same way data moves on a network, each packet could be encrypted and include headers with information such as Process ID numbers and checksums. If a group of instructions landed in RAM that was not part of an approved application, decrypting the instructions before sending them to the CPU would produce random binary patterns. It would be better to have these random patterns attempt execution than the specially crafted code the hacker had intended to execute.

A new subsystem would be placed between the CPU and RAM. The subsystem would act as a proxy on behalf of the CPU. Each time the CPU sent data to, or requested data from RAM, the data would be intercepted by the subsystem and either encrypted or decrypted. A proxy technique is also used everyday by highly secure systems that are connected to the Internet. Rather than allowing a user's computer to request information directly from the Internet, a request is made to a proxy server, which in turn, makes the actual connection to the system on the Internet.

This solution would offer a high degree of protection from buffer-overflow exploits. Since these attacks rely on the inputting of a larger-than-expected response in a data buffer, anything that was injected in an area of RAM without being properly encrypted would be decrypted into random binary patterns.

If a lockable, flash RAM-based utility was responsible for the encryption and decryption processes, it would be almost impossible to create a software-based exploit that could override the new controls.

With the operating systems and applications gaining more functionality with each revision, the amount of programming code that goes into them seems to increase exponentially. By the time most of the serious vulnerabilities of a given software release are discovered and patched, a new version becomes available, and the game starts all over. In light of this fact, it is reasonable to expect that the operating systems and applications will continue to have major security vulnerabilities.

Due to the complexity of the OS, and the fact that there are applications from many different vendors running on the average computer, there is no single software choke point that can be controlled. The only logical place that a choke point could exist is in hardware on the system bus.

In order for this concept to materialize, the entire standard of system busses will have to change. If the motherboard of the computer had the right hardware installed, along with the necessary code in a dedicated BIOS chip to oversee the operations, a computer could be designed to assign process identification numbers to each of the applications that were legitimately installed. As individual programs are opened and copied into RAM, the unique process ID numbers could be used to decide which secret key should be used for the encryption.

With the ongoing popularity of buffer-overflow activity, this type of functionality needs to become standard issue on all PC hardware, right out of the box.

Potential Implementation Problems

Certain problems exist that could prevent this new technology from being adopted on a large scale. The first is the issue of backwards compatibility. If a new hardware design is to gain popularity, it must be compatible with the software that is already in use. If anyone wanting to take advantage of the new system is forced to replace all of his or her current software, the technology is not likely to gain acceptance. On the other hand, if the existing applications would readily function on the new hardware, and the new system offered unprecedented protection from buffer-overflow attacks, the system would likely be an overnight success.

Another problem arises in the form of vendor-to-vendor compatibility. A single new standard is certainly called for, but it needs to be just that. If several companies all go their separate ways to develop the new standard, nothing will be compatible with anything else. This sort of problem could frustrate and confuse many, leaving the new technology with a bad name even before it had a chance to prove itself. To keep this from becoming a problem, a consortium of manufacturers would need to join forces to develop the new system.

The actual design and implementation of a working subsystem would not be a simple matter. It would require the coordinated efforts of hardware and software designers, and input would be needed from experts in the security, networking, and computer-related fields. Most of the individual technologies required to assemble this system already exist. The only missing ingredient would be a company to put the pieces of the puzzle together and apply the results to hardware.

This document presents an overview of what would be required to make this proposed solution a reality. The final product would be the culmination of a fair amount of research, insight, and ingenuity from an array of computer, networking, and security experts. If the best effort of the network security world could be applied to PC hardware, a new and much more secure computing platform could be the result.

Relevance to Perimeter Security and Defense-in-Depth

If this new concept was developed, it would be a successful weapon against buffer-overflow attacks. Its application to individual servers in the screened subnet is fairly obvious; if these systems are no longer vulnerable to buffer-overflow attacks, the risk of them being used to attack other systems will be reduced. That fact alone would make the technology worth investing in. The real benefit, however, lies in the application of Packet Inspection Bus technology to the systems that support the full functionality of the Internet.

How many of the systems that face the Internet use the same basic hardware architecture as the PC? The overwhelming majority of Web servers, DNS servers, and Mail servers that connect to the Internet are running on PC hardware, regardless of the operating systems and applications that control them. It should be noted that there are also huge numbers of software-based routers and firewalls using PC-based hardware.

The operating system and applications of most systems are vulnerable to buffer-overflow attacks because of poor programming practices and also because the hardware offers no form of protection. For this reason, many security administrators refuse to utilize a firewall that runs on a full-blown operating system. This fact keeps many perimeters from being protected by the widely available, extremely powerful, and in many cases, very affordable software-based firewalls.

Much of the effort that goes into designing a secure perimeter involves concerns about buffer-overflow attacks entering through the Internet. Many times, these attacks are launched against web servers that allow incoming HTTP data. A firewall that blocks packets based only on address and port information offers no defense against this kind of attack.

When designing a secure perimeter, some relevant factors actually reside outside of the perimeter. Consider the problems that arise from allowing remote users to connect to a corporate network through a virtual private network (VPN). The tunnel provided by the VPN allows sensitive data to safely cross the public Internet by using an encryption technique. Confidentiality and integrity of the data are ensured while it is in transit, but there is very little control over the system on the remote end. If the security of the remote system has been compromised, the VPN would then provide an encrypted tunnel that an attacker could use to access the internal network. If PCPIB technology was employed on the external systems, the risk of buffer-overflow attacks could be eliminated. The dangers associated with the use of a VPN would then be greatly reduced.

How much safer would the worldwide Internet be if most of the perimeter systems were using a hardware architecture that was no longer vulnerable to such threats? If all of the perimeter and internal systems included this new technology, the effects on defense-in-depth would be dramatic.

One of the first steps of a hacker is mapping out the systems that are visible to the Internet. The next step is to glean information about what operating system and applications are running on the hardware. With this data, the hacker can consider well-known exploits that can be used against those systems. A popular method of gaining access to the external systems is to get some form of malware installed there. Buffer-overflow attacks are a very popular early step in the process of installing malicious software. Once the hacker has found a way into one of the perimeter systems, he or she will often use that system for attacks that are aimed at either the internal network or other systems on the Internet.

Consider the implications of PCPIB for the hacker. By reducing the number of ways to compromise hosts that connect to the Internet, it will be much more difficult to launch attacks against the internal network.

The ultimate result of implementing this sort of new technology is hard to predict. Certain facts are hard to deny though. If a vast number of the present threats to information systems could be phased out over the next few years, the security practitioners of the world could devote more of their valuable time to other pursuits. Regardless of what we know today about how to make a system secure, there will be new weaknesses and vulnerabilities discovered in the future. The exploits designed to take advantage of them will start to appear almost immediately. If less time is needed to guard against buffer-overflow problems, more time will be available to defend against the multitude of other threats. Hopefully, this innovation will result in the overall landscape being reshaped into a much more pleasant view.

Summary

The original PC bus architecture was simple and straightforward. No mechanisms were put in place to monitor the validity of processes residing in memory. Due to the need for backwards compatibility, the basic functionality was never changed, even though it offered no immediate protection from malicious code execution. As more computers made the connection to the Internet, the amount of malware in existence has increased proportionally. For this reason, an overhaul of the system bus is in order. The technology already exists in other forms to alleviate this problem. These existing technologies could be adapted and applied to the bus to curtail the effectiveness of buffer-overflow attacks. Designing and developing such a system would require collaboration, ingenuity, and effort.

The effect on the computing industry, as a whole, and perimeter security, in particular, could be vast. Less time would be spent fixing compromised systems, and, therefore, more time could be spent implementing other forms of security measures. With the world economy and the infrastructure of all developed countries now dependent on the security of computing systems, it is obvious that a major leap forward is necessary to secure those systems. Let us hope that leap does not come too late.

Assignment 2: Security Architecture

Introduction

Assignment two proposes a secure network design for GIAC Enterprises, which is in the business of selling fortune cookie sayings. The customer base consists of bakeries from around the world. Most of the fifty GIAC employees work at the home office. Some employees work at one of four remote offices that are geographically dispersed, while others do their job from home computers or laptops. The Internet provides the means by which all sales are conducted. Since the fortunes are the stock-in-trade of GIAC, they are considered to be highly sensitive intellectual property and subject to theft or destruction. For this reason, their value is a major consideration when making design decisions about the network security posture. The overall design will provide confidentiality, integrity, and availability of all services and data that reside on the network. Threats to a successful implementation include hackers, malware, system failure, and data loss or theft.

The principle of least privilege will be enforced throughout the design by providing only the minimum services that are absolutely necessary to conduct business. This policy will certainly cause some activities to be less than fully automated, but the reward of having a more secure network while adding little extra cost will provide suitable justification.

The primary element providing perimeter security will be a Check Point Express Package, which provides both firewall and Virtual Private Network (VPN) capabilities. [8] Check Point provides a number of integrated functions, and many optional components can be added to enhance security and functionality. Check Point continues to add new products to their line on a regular basis. The major capabilities of the Check Point products will be discussed in the Firewall/VPN section.

Design Considerations

Security architecture design should start with the understanding that a network cannot be made totally secure. A great deal of time and money can be devoted to producing a highly secure system, but vulnerabilities will still exist. Armed with this knowledge, a security architect will naturally question how much security is enough. The answer comes in the form of one word, BALANCE.

It is not an easy task to find balance in a project that contains as many variables as a corporate network. A primary consideration is how to put a value on that which is being protected. The value of the stored data, the reputation of the company, and the cost to recover from a security breach must all be considered. The value of a company's assets must be more than that which is spent to protect them.

Any device that enhances the security stance must be evaluated for its cost versus effectiveness. Services that are allowed to breach perimeter security need to be scrutinized for the value they add to the company's financial bottom line. Many computer applications add convenient functionality to the overall system. When attempting to make the perimeter secure, these convenience items should be carefully considered. If they add very little or no profit, there is a strong case for disallowing them.

Many layers of security can be added, but at some point, the costs of implementation and ongoing maintenance will outweigh the benefits. Every server or workstation that is added to the network will need to be continuously patched and updated. Isolating network services so that each of them runs on a separate platform helps to keep compromised systems from affecting other services. However, this configuration also adds to the total number of items that need to be maintained. At some point, financial and personnel resources are wasted by trying to build perfect security.

Human labor is by far the most costly component of securing a network, and that expense needs to be heavily evaluated. Every component that is added to the basic network should be assessed not only by looking at the initial cost of adding the item but also by analyzing the ongoing maintenance and security resources it will consume. If intelligent decisions are made during the design phase, a satisfactory level of security will be more easily achieved, while the ongoing expenses are kept under control.

Most open-source products offer high quality performance while costs are kept to a minimum. They also have a track record of having patches developed quickly when new vulnerabilities are discovered. For these reasons, priority is given to the selection of open-source products whenever possible. Red Hat Enterprise Linux ES v.4 has been chosen for all Linux systems [9]. The requirements for other components are straightforward; any product chosen must be widely deployed and must have a good track record. The product must also provide a good value to the overall project thereby reducing the total cost of ownership.

The ideal system will be a synthesis of managerial support, technological controls, effective and enforceable security policies, user awareness, ongoing education of staff, and consistent audits of total system security. The leverage that each of these components provides will ultimately result in a balanced system that is affordable, manageable, secure, functional, and, most importantly, profitable.

Access Requirements

Customers

Customers of GIAC consist of large and small bakeries that insert the sayings into the fortune cookies they produce. At present, thirty-two large bakeries and ninety-seven small bakeries worldwide are purchasing fortunes from GIAC. For the bakeries to do business with GIAC, they need the ability to exchange e-mail and download fortunes in bulk. Access for this group will be limited to the Web and mail proxies, which both reside in the screened subnet. Neither direct nor indirect access to the GIAC database will be allowed. When a customer places an order via e-mail, an internal employee will query the database, and the results of the query will be converted into an ASCII text file. Then this file will be placed on the Web server for the customer to download by going through the reverse proxy. Anyone purchasing fortunes from GIAC will be able to establish a secure connection by using a modern, SSL-enabled browser. The tunnel created with SSL will be terminated at the Check Point firewall/VPN.

Suppliers

This group supplies fortune sayings in bulk to GIAC. By giving all the suppliers their own dedicated storage area on the Web server, their files will be protected from damage by other suppliers. Files that have been uploaded to these areas will be moved as quickly as possible to the internal database server and deleted from the Web server. No direct access to systems on the internal network will be provided to the suppliers. A secure connection to GIAC will be made possible with a Check Point product named SecuRemote [10], which utilizes a full-function IPSEC tunnel. SecuRemote is free to download from Check Point and must be installed on all supplier computers that connect to GIAC. The advantage of this product can be fully realized when used in conjunction with SecureClient [11]. When a SecuRemote computer connects to the main firewall that has SecureClient installed, both a centrally managed personal firewall and corporate security policy will be pushed out to the remote computer. By enforcing the use of a properly configured personal firewall, the risks associated with connecting remote systems to the corporate network are reduced.

Sales

The sales force will work both from the remote offices and from their homes or customer sites. Their only requirements will be to send and receive secure e-mail and to exchange files with the home office in a secure fashion. For clarity, the sales group has been divided into the two subgroups listed below.

Remote Office

For secure communications between the remote offices and the GIAC corporate network, a Check Point appliance called VPN-1 Edge [12] will be installed in each remote office. This device allows the construction of a site-to-site IPSEC tunnel between the remote office and the home office. Additional measures include the mandatory installation and maintenance of antivirus and personal firewall software on all machines that connect to GIAC. VPN-1 Edge also supplies stateful inspection firewall functionality.

Traveling Sales/Laptops

Of utmost importance will be the security concerns of connecting potentially insecure laptops or home computers to the corporate network through an encrypted tunnel. If this arrangement is handled in a careless manner, the remote computers could introduce a vast array of security problems. The solution lies in both the SecureClient functionality at the home office and the SecuRemote software on the remote computer.

SecuRemote establishes an IPSEC tunnel with the home office and enforces the installation of a personal firewall, which is provided by SecureClient. Properly maintained antivirus software is also mandatory.

Partners

The function of partners is to download fortune sayings to their site, translate them to other languages, and resell them. Their access requirements and restrictions are identical to those of the Customer group. Partners will establish an IPSEC tunnel using SecuRemote.

Employees

Everyone included in this group will work only from inside the GIAC facility and will have varying degrees of access to the internal resources, depending on their job requirements. These users will be granted only the absolute minimum access needed to fulfill their obligations to GIAC. In addition to their level of access to internal resources, they will all have external access to the Internet (HTTP, HTTPS, and E-Mail).

General Public

The general public will have very limited access to the GIAC network. They will be able to connect to the Web server through the proxy to view information about the company. The address where they can send e-mail, should they wish to correspond, will be included on the Web page.

Secondary ISP

Numerous older protocols fail to include any native security mechanisms. FTP is a good example of an insecure protocol that is still very popular in spite of its lack of security. FTP is occasionally called for in the GIAC environment, but it is not allowed to pass through the GIAC perimeter to any part of the internal network or screened subnet. In an effort to grant external FTP access to internal employees, a secondary network has been instituted. This system has its own ISP, firewall, and three workstations. Each of the workstations has a CD/DVD recorder, which allows employees to download data via FTP, copy the data to a CD or DVD, and manually move the data to the internal system. The computers all have current anti-virus, anti-spyware, and personal firewalls installed. The use of air gaps such as this one are discussed in the book Inside Network Perimeter Security by Northcutt, Zeltser, Winters, Frederick and Ritchey [13].

The ISP provides a DSL connection, which enables GIAC to outfit this network with an affordable Linksys broadband router [14]. This device also includes a four-port switch, Network Address Translation (NAT), and a wireless access point. At this time, the wireless function is prohibited, and therefore, is disabled. The combination of router, personal firewall, and anti-virus software provides adequate protection and bandwidth for this network. In addition to its primary functionality, this arrangement also serves as a backup Internet connection in the event of a denial-of-service attack or failure of the primary ISP, router, or firewall. Furthermore, since it behaves like an external system, it provides additional troubleshooting and security auditing abilities for the main network.

Data Flow Tables

Tables 2.1 and 2.2 show the data flows for the user groups.

Source	Destination	Protocol - Port	Description
Customers	Reverse Proxy	HTTP - TCP/80	Customer Access to Web Server for Casual Browsing
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Customer - HTTP Proxy to Internal Web Server
Customers	Reverse Proxy	HTTPS - TCP/443	Customer SSL Access to Web Server to Download Sayings
Reverse Proxy	Internal Web Server	HTTPS - TCP/443	Customer - HTTPS Proxy to Internal Web Server
Customers	Mail Relay	SMTP - TCP/25	Customer Access to Mail Server
Mail Relay	Internal Mail Server	SMTP - TCP/25	Customer - SMTP Relay to Internal Mail Server
Customers	External DNS	DNS - UDP/53	Customer Access to External Name Server
Suppliers to VPN	Check Point Firewall/VPN	ISAKMP - UDP/500	IKE Key Negotiation with VPN
Suppliers to VPN	Check Point Firewall/VPN	ESP - IP/50	IPSEC Tunnel from Supplier to VPN
VPN (Suppliers)	Reverse Proxy	HTTP - TCP/80	Supplier Access to Web Server for Browsing & Uploading
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Supplier - HTTP Proxy to Actual Server
VPN (Suppliers)	Mail Relay	SMTP - TCP/25	Supplier Access to Mail Relay
Mail Relay	Internal Mail Server	SMTP - TCP/25	Supplier - SMTP Relay to Internal Mail Server
VPN (Suppliers)	External DNS	DNS - UDP/53	Supplier Access to External Name Server
Sales / Remote Offices to VPN	Check Point Firewall/VPN	ISAKMP - UDP/500	IKE Key Negotiation with VPN
Sales / Remote Offices to VPN	Check Point Firewall/VPN	ESP - IP/50	IPSEC Tunnel from Supplier to VPN
VPN (Remote Offices)	Reverse Proxy	HTTP - TCP/80	Remote Office Access to Web Server for Browsing & Downloading
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Remote Office - HTTP Proxy to Actual Server
VPN (Remote Offices)	Mail Relay	SMTP - TCP/25	Remote Office Access to Mail Relay
Mail Relay	Internal Mail Server	SMTP - TCP/25	Remote Office - SMTP Relay to Internal Mail Server
VPN (Remote Offices)	External DNS	DNS - UDP/53	Remote Office Access to External Name Server

Table 2.1 - Data Flows for Customers, Suppliers, and Remote Office Sales

Source	Destination	Protocol - Port	Description
Sales / Traveling to VPN	Check Point Firewall/VPN	ISAKMP - UDP/500	IKE Key Negotiation with VPN
Sales / Traveling to VPN	Check Point Firewall/VPN	ESP - IP/50	IPSEC Tunnel from Supplier to VPN
VPN (Traveling Sales)	Reverse Proxy	HTTP - TCP/80	Traveling Sales Access to Web Server for Browsing & Downloading
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Traveling Sales - HTTP Proxy to Actual Server
VPN (Traveling Sales)	Mail Relay	SMTP - TCP/25	Traveling Sales Access to Mail Relay
Mail Relay	Internal Mail Server	SMTP - TCP/25	Traveling Sales - SMTP Relay to Internal Mail Server
VPN (Traveling Sales)	External DNS	DNS - UDP/53	Traveling Sales Access to External Name Server
Partners to VPN	Check Point Firewall/VPN	ISAKMP - UDP/500	IKE Key Negotiation with VPN
Partners to VPN	Check Point Firewall/VPN	ESP - IP/50	IPSEC Tunnel from Supplier to VPN
VPN (Partners)	Reverse Proxy	HTTP - TCP/80	Partners Access to Web Server for Browsing & Downloading
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Partners - HTTP Proxy to Actual Server
VPN (Partners)	Mail Relay	SMTP - TCP/25	Partners Access to Mail Relay
Mail Relay	Internal Mail Server	SMTP - TCP/25	Partners - SMTP Relay to Internal Mail Server
VPN (Partners)	External DNS	DNS - UDP/53	Partners Access to External Name Server
Internal Employees	Web Proxy	HTTP - TCP/80	Internal Employees Access to Internet
Web Proxy	Internet	HTTP - TCP/80	Internal Employees - Web Proxy to Internet
Internal Employees	Web Proxy	HTTPS - TCP/443	Internal Employees Access to Secure Internet Sites
Web Proxy	Internet	HTTPS - TCP/443	Internal Employees - Web Proxy to Internet - SSL
Internal Mail Server	Mail Relay	SMTP - TCP/25	Employees - Mail Server to Mail Relay to send External Mail
Mail Relay	Internet	SMTP - TCP/25	Employees - Mail Relay to Internet to send External Mail
Public	Reverse Proxy	HTTP - TCP/80	Public access to Reverse Proxy
Reverse Proxy	Internal Web Server	HTTP - TCP/80	Public – Reverse Proxy to Internal Web Server
Public	Mail Relay	SMTP - TCP/25	Public access to Mail Relay
Mail Relay	Internal Mail Server	SMTP - TCP/25	Public - Mail Relay to Internal Mail Server
Public	DNS Server	DNS - UDP/53	Public access to External Name Server

Table 2.2 - Data Flows for Traveling Sales, Partners, Internal Employees, and Public Architecture Components

Router

Routers exist primarily for the purpose of routing data between networks. They can also do a good job of basic packet filtering because of the way they operate and because they are installed at the perimeter of a network. While they lack the ability to inspect the data in a packet for malicious content, they are ideally suited to filter out the packets that should absolutely not be allowed into the network.

A Cisco router has been chosen as the first line of defense. Static packet filtering will be employed, because it is fast and efficient for removing absolutes from the incoming packets. Absolutes are defined as packets that should never need to enter the network. Examples include packets with a source IP that equals a private address (RFC 1918) [15], non-assigned public address, internal address, and source-routed packets. By removing these items from the incoming stream, the load on the firewall is reduced, and the firewall will be protected from various forms of attack.

On the Cisco router, these packets will be filtered using extended Access Control Lists (ACLs), which limit the traffic based on source and destination addresses as well as destination port numbers. For instance, if a packet is directed to port 80 (HTTP) and includes a destination IP address the Web server, the packet will be allowed to pass. A packet that is destined for port 80 on the mail server, however, can be blocked. Extended ACLs provide an extra level of protection for the overall network and the firewall.

Since the firewall has application intelligence, it will have ports listening for the services that are allowed to pass. When the router provides filtering based on header information, and the firewall makes its decisions by looking at the data in the packet, a stronger form of protection will result.

The Cisco Multiservice Access Router, model 3725 [16] with IOS version 12.3 Mainline [17], was chosen because it has the necessary processing power to handle the extended ACLs, while avoiding the creation of a bottleneck in the network. This router supports up to 100,000 packets per second, and this level of performance will allow for future growth. Cisco products are in extensive use routing data on the Internet, and the technical support is considered well above average in the industry.

Bugtraq lists eight known vulnerabilities for IOS 12.3 [18], all of which could potentially lead to a denial-of-service attack. Cisco offers workarounds or system patches to mitigate each of them. Most of the problems involve services that would normally be disabled on the router, so they would not present a problem in most installations. No known vulnerabilities exist for the hardware of the 3725.

Firewall/VPN

A Check Point Express NG with Application Intelligence R55 [19] has been selected. This package includes Firewall-1, VPN-1, SmartDefense, SmartCenter, and unlimited SecuRemote clients [20]. The Express package that was chosen includes licensing that allows up to one hundred hosts to reside behind the firewall in the protected zone. Components that were added to the base package include four VPN-1 Edge appliances for the four remote offices, SmartClient, and a yearly contract for upgrades to the SmartDefense utility. Check Point Express is a very popular product and thus is considered a "proven" solution. Since it has a very large install base, its strengths and weaknesses are well documented.

The Express package was chosen for the following reasons:

- Integrated firewall and VPN.
- Seamless operation with VPN-1 Edge appliances.
- Provides both stateful inspection and application awareness.
- VPN supports SSL and IPSEC tunnels.
- Allows clientless SSL connections.
- SmartDefense upgrades keep attack signatures updated.
- SmartClient pushes personal firewall and security policy out to clients.
- Performs Network Address Translation (NAT).
- Provides strong authentication – DES, (3DES and AES are optional).
- Built-in X.509 digital certificates.
- Unlimited SecuRemote client software downloads.
- Centralized management of complete package, including VPN-1 Edge.

Check Point Express provides a graphical, centralized management interface known as SmartCenter. From this easy-to-use console, all Check Point components can be configured including the VPN-1 Edge appliances at the remote offices. The convenience of SmartCenter is a major purchasing factor. Since everything can be configured from a graphical user interface, less time is spent on original setup and ongoing maintenance. Over time, this feature will lower the total cost of ownership.

The Application Intelligence facet of the firewall recognizes proper behavior for over 150 popular applications, protocols, and services, plus it includes a database of known-attack patterns. If a known protocol such as HTTP attempts to access a port that is not normally associated with HTTP, the firewall will recognize this as improper behavior. SmartDefense is the add-on that allows the system to keep the attack patterns updated automatically. This service needs to be renewed on a yearly basis.

The Check Point software will be installed on a hardened server rather than a preconfigured appliance because faster hardware will be available in the future at a reasonable price. The Check Point software can be migrated to new hardware without losing all of the configuration information. Improved hardware performance will allow the firewall/VPN to process more packets without sacrificing reliability.

A secondary consideration is the fact that hardware does fail occasionally. A decision was made early on by GIAC to develop a standard hardware platform, which is used for every server. It is quite economical to keep a cold spare on hand, and if there is a catastrophic hardware failure of any server, it can be replaced with very little down time. This arrangement is more efficient than keeping both a spare hardware appliance and a spare server on hand.

Having the integrated VPN terminate at the firewall allows the data to be decrypted before it passes through the firewall. This configuration prevents potentially malicious encrypted data from passing the firewall unchecked. The integration of the overall package also makes it easy to configure. The downside of this configuration is the creation of a single point of failure for the network.

Bugtraq lists four known vulnerabilities for Check Point NG-AI, R55 [21], and the vendor offers a patch for all but one of the problems. The remaining issue can allow an attacker to detect the version of the software plus the capabilities of the firewall. With this information, a determined attacker might eventually find a way further into the system. Since the other three vulnerabilities have a workaround or fix, a suitable mitigation strategy is to monitor the system logs for suspicious behavior and continue to watch for a solution from the vendor. Additionally, any new vulnerabilities of the firewall should be patched immediately.

nIDS

A combination of NetOptics Gigabit Copper taps (# 96298) [22] and Snort version 2.3.2 [23] has been selected for the network Intrusion Detection System. Snort was chosen because it is an open-source product, and because it has proven its usefulness in the intrusion detection arena. The NetOptics devices provide a passive connection that will not interrupt traffic flow in the event of tap failure. The gigabit model was chosen because it will be installed in high-bandwidth locations between the firewall and the switches. The nIDS offers no direct protection for the network. Its strength lies in its ability to alert administrators when malicious activity is detected. If an alert is acted upon quickly, the intrusion can be blocked. These systems need current attack signature files in order to offer protection from new threats, so they add to the list of items that need ongoing maintenance to remain effective.

Additional Layers

Nothing from the external world is allowed to directly communicate with the internal network. A combination of Postfix and Sendmail are employed to handle all mail transactions. The Sendmail server, which is located in the screened subnet, is running a free anti-virus product known as ClamAV [24]. This e-mail security product was the 2004 winner of Linux Journal's Editor's Choice Award.

ClamAV will be configured to scan all incoming e-mail for virus content. Sendmail is configured to strip the outbound header of information about the internal network. By using different applications for the server and relay, another layer of defense is added.

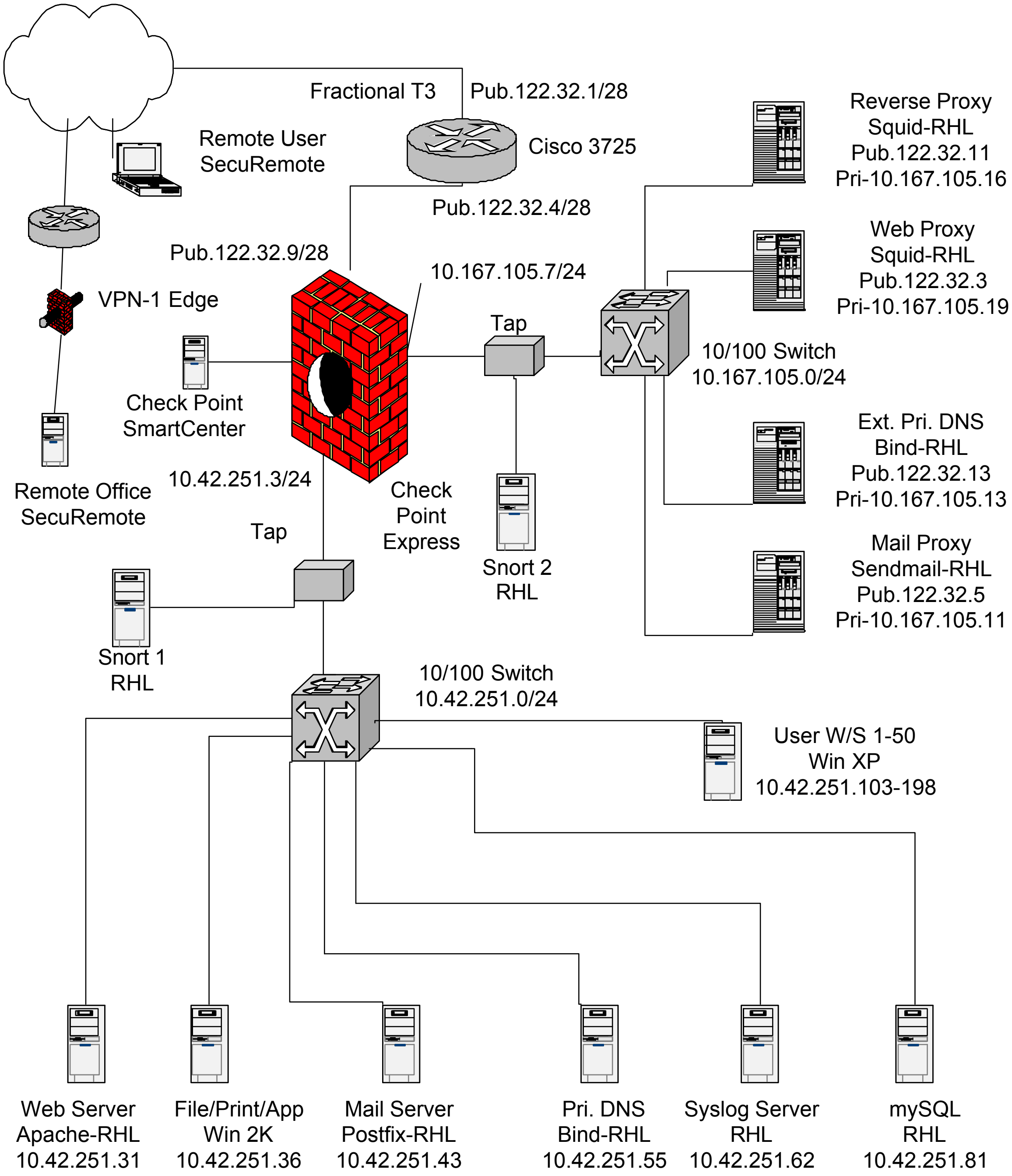
Squid version 2.5 [25] will function in the screened subnet for both ingress and egress flows of Web content. A standard Web proxy will filter all outgoing requests while a reverse proxy will handle those that are incoming. By caching Web pages from both internal and external sources, the overall load on the firewall is reduced. Using Apache as the actual Web server and having Squid as a proxy adds another security layer. The vulnerabilities in Apache are different from those in Squid, and this fact will prevent a single exploit from affecting both systems.

A split DNS will further prevent the system from leaking internal network information out to the Internet. The internal DNS will serve only the internal systems, and the DNS in the screened subnet will serve the general public. Neither server will be allowed to pass zone transfers to the Internet. For simplicity, the secondary DNS server in each location has been omitted from the network diagram.

A Syslog server on the internal network collects log entries from the router, the firewall, and all Linux systems. A unified log collection system makes monitoring the logs much easier, while preventing an intruder from erasing his or her tracks.

All hosts in the screened subnet are hardened by applying current patches and removing unnecessary services and applications.

Network Diagram



Note: All systems have current patch levels. All Linux systems are Red Hat Enterprise Linux ES v.4.

IP Addressing

GIAC is currently using eight of the fourteen public IP addresses in the range of PUB.122.32.1-14/28. All systems inside the firewall rely on Network Address Translation (NAT), which is provided by the Check Point system. NAT allows an organization to provide Internet access to many private internal IP addresses while using only a few of the routable, public addresses. The servers that face the Internet will each externally display one public address via static NAT while actually using a private IP address internally. Everything else on the network uses private addresses by way of hide NAT. Hide NAT provides an additional layer of protection by never showing externally the actual IP address of the internal device.

The address scheme shown on the network diagram is not a typical, sequential scheme. Attackers expect the internal address of a router to be one address higher or lower than the external address. By adding a degree of randomness to the scheme, a slight security advantage is realized.

Defense-In-Depth

Many layers of security-related mechanisms will provide Defense-In-Depth. The router and firewall will provide the first two layers. Every device on the network that runs a full operating system will include antivirus, anti-spyware, and a personal firewall. They will also have all unnecessary services disabled.

Employees will receive short monthly training sessions about the potential hazards they should guard against. These hazards include social engineering, leakage of sensitive information, importance of using good passwords, installation of approved software, and unexpected e-mail attachments. The training will clearly spell out what constitutes acceptable use and what consequences can be expected if the rules are not followed. Everyone connecting to the internal network will be required to sign an acceptable use agreement, and the document will require a yearly re-signing to remind employees that they are liable if they cause a breach of security. Data traversing the network will be limited to only that which is necessary. The principle of least privilege will be enforced.

Other considerations include physical security of the premises. This facet relates to having all networking devices and servers installed in secure areas with access only being granted to those persons deemed necessary. The only allowance for business continuity/disaster recovery will be the storing of backup tapes in an offsite facility. In the event of a disaster, the tapes could be used to build a new system and restore the business with a minimum of downtime. To expedite the process of rebuilding the business, there is a specific, written plan that will be followed in the event of a disaster.

Dial-up modems and wireless access points are not allowed in this environment, and regular checks will be made to detect their existence. Because of the dangers associated with packet sniffers, network hubs are also prohibited in this environment.

Policy dictates that all newly installed systems must meet the current standards of security before they are connected to the network, which includes having updated anti-virus, anti-spyware, personal firewall, and OS patches. By enforcing this rule, new systems will be protected from numerous threats and will not present a hazard to the other systems on the network. It is also required that all default passwords be replaced with strong passwords that are known only to the proper members of the GIAC team.

GIAC strives to retain all current employees, but special effort is devoted to keeping a low turnover rate among the network support team. If any of them should leave GIAC, their accounts on all systems will be removed immediately, and the administrative passwords on all key systems will be changed. Any former employee can present a risk if these measures are not enforced.

To keep the system secure, ongoing maintenance is an absolute essential. Operating system and application patch levels, as well as anti-virus and IDS signatures, all need to be kept updated. Security administration will need to be kept current on new threats and mitigation tactics. Log files need to be monitored on a daily basis. The overall system will continue to be a work-in-progress, using system tuning to keep performance satisfactory while maintaining a high level of security. Penetration tests will be performed on a quarterly basis utilizing the secondary ISP/network. Third-party auditors will be contracted to test the system yearly.

By leveraging the best of affordable technologies with human intelligence to oversee the entire operation, a multi-layer solution has been developed. However, each individual piece of the security structure has vulnerabilities that could be exploited. By applying the defenses in various layers, the overall system will certainly slow the progress of any intrusion attempts. Monitoring the logs on a daily basis will keep the security technicians abreast of any attempted malicious action. By responding in a timely manner to abnormal behavior of the system, the attempted intrusion can be stopped before serious damage results.

Assignment 3: Firewall and Router Policies

General Security Stance

The general GIAC security policy stance is intended to deny all which is not explicitly allowed. The only place this policy is not suitable is the ingress filters of the router. Legitimate E-mail and Web server requests could come from virtually any valid public IP address. An explicit ingress-deny stance on the router would prohibit most public users from reaching GIAC. For that reason, the ingress filters allow all that is not explicitly denied. In this manner, the router can deny traffic that is never desired inside the network, yet admit traffic that might be desired.

The router is the first line of defense, and helps to reduce the incoming load on the firewall. The firewall has services listening on open ports because it is inspecting the data portion of the packets. The router helps to defend these ports by eliminating some of the traffic that constitutes a security risk. Examples include incoming traffic with unassigned or private IP source addresses, and packets that claim to have come from the internal network. GIAC public IP addresses should only be generated from the GIAC network. Anything entering the external interface of the router with a GIAC source address is obviously spoofed. Unassigned and private IP addresses should never appear on the Internet. If they attempt to traverse the incoming border, they constitute malicious or undesirable traffic. For these reasons, packets with unassigned, private, or GIAC-registered addresses will be dropped.

Layered security is accomplished in this configuration by filtering absolutes from the incoming stream using the router, then applying the stateful inspection abilities and the application intelligence of the firewall. Packets that are allowed to enter through the router only need the right address and port number in their header. The router sees a packet labeled with the IP address and port 80 of the reverse proxy as allowable. Many known exploits take advantage of vulnerabilities on Web servers and E-mail servers. If the only layer of defense is the router, these malicious packets can easily deliver a dangerous payload to one of the servers. The firewall, however, has the ability to identify those packets that are carrying malicious code because it inspects the data portion of packets. By keeping the attack signatures updated on the firewall, popular attack methods can be blocked. The only negative impact of using layered security is the added expense of initial equipment purchase, configuration, and ongoing maintenance.

Router Policies

Commercial Cisco routers offer the choice of static or reflexive ACLs. Static filters allow the router to process packets very quickly. Reflexive ACLs, however, force the router to check both a state table and an access control list, effectively doubling the workload on the router. Since the firewall is providing stateful inspection, the router will only be using static ACLs. This arrangement helps keep the load on the router at a manageable level.

Extended ACLs allow the router to filter packets based on source and destination IP addresses and destination ports, which gives more granular control than using standard ACLs. Since there are more criteria being evaluated in each packet, the router needs additional processing power to handle the extra load. Trying to save money by installing a less powerful router creates a bottleneck and can overload the device. Anytime a bottleneck exists, the excess traffic will either be dropped or allowed to pass unchecked. The risk of allowing malicious packets to enter the network at a time of peak traffic load is not a good tradeoff for the small amount of money that could be saved by buying an underpowered router.

Ingress Rules

Table 3.1 includes the ingress ACLs, applied incoming to interface.

#	Act	Proto	Source IP	Dest IP	Dest Port	Log ?	Description
1	Deny	IP	0.0.0.0/8	Any	Any		Deny Unassigned IP Addresses
2	Deny	IP	1.0.0.0/8	Any	Any		Deny Unassigned IP Addresses
3	Deny	IP	2.0.0.0/8	Any	Any		Deny Unassigned IP Addresses
4	Deny	IP	5.0.0.0/8	Any	Any		Deny Unassigned IP Addresses
5	Deny	IP	7.0.0.0/8	Any	Any		Deny Unassigned IP Addresses
6	Deny	IP	10.0.0.0/8	Any	Any		Deny Private IP Addresses
7	Deny	IP	172.16.0.0/12	Any	Any		Deny Private IP Addresses
8	Deny	IP	192.168.0.0/16	Any	Any		Deny Private IP Addresses
9	Deny	IP	127.0.0.0/8	Any	Any		Deny Loopback (Test) Addresses
10	Deny	IP	Pub.122.32.1/28	Any	Any		Deny GIAC Internal Public Addresses
11	Deny	UDP	Any	Any	69	Log	Deny Incoming TFTP Services
12	Deny	TCP	Any	Any	445	Log	Deny Incoming SMB Services
13	Deny	UDP	Any	Any	514	Log	Deny Incoming Syslog Services
14	Deny	TCP	Any	Any	135-139	Log	Deny Incoming Microsoft Services
15	Deny	UDP	Any	Any	135-139	Log	Deny Incoming Microsoft Services
16	Deny	UDP	Any	Any	161-162	Log	Deny Incoming SNMP Services
17	Deny	TCP	Any	Any	6000-6255	Log	Deny Incoming X-Windows Services
18	Deny	ICMP	Any	Any	Echo-request, Host-redirect		Deny Incoming ICMP Hazards
19	Permit	Any	Any	Any	Any		Allow Everything Else

Table 3.1 – Router Ingress ACLs – Applied to interface Pub.122.32.1

Rules 1 through 5 prevent packets with an Internet Assigned Numbers Authority (IANA) unassigned IP address from entering the GIAC network. Since these addresses have not been allocated to anyone, their use indicates a high likelihood that the packets have been crafted. IANA lists approximately one hundred other blocks of unassigned addresses, but adding all of them to the ACL would slow the router to an unacceptable performance level.

Rules 6 through 8 protect the network from RFC 1918 private addresses. These packets should never be routed on the Internet, but an improperly configured router could allow them to appear at the GIAC perimeter. When attackers want their behavior to be anonymous, they can craft malicious packets using a private IP address. These packets can be a security risk if they are allowed into the network. It is also possible that they are harmless packets, which are the result of an improperly configured system. Therefore, their entry attempts will not be logged.

Rule 9 blocks packets from the loopback (test) address. A Microsoft exploit exists in which the computer is tricked into seeing crafted packets from the loopback address, telling the system to reduce its data transfer rate. This exploit can cause all Microsoft systems on the network to slow down.

Rule 10 prevents packets with a source address that belongs to GIAC from entering. These packets are obviously crafted, and therefore, must be blocked and logged. Watching for this sort of activity can give personnel an advanced warning that someone is attempting to attack the network.

Rules 11 through 17 pertain to services that are running on the GIAC network. These services are of a sensitive nature and are not intended to be accessible from the Internet. Some of these could give out information about the network, while others have many known security flaws. Blocking them at the firewall gives another layer of protection. This action will, however, prevent the system administrators from doing certain diagnostics or maintenance from a remote computer.

Rule 18 prevents persons on the Internet from pinging systems that are internal to the router. Packets that reply to a ping request contain information about the network that an attacker could use to gain additional access to the system. Also, a Denial-of-Service attack exists in which a router can be told to redirect traffic for certain hosts to a different router. For this reason, ICMP host-redirect packets are not accepted [26].

Traffic that does not meet any of the previous criteria is allowed to enter the network via Rule 19. The first eighteen rules greatly limit the malicious and undesirable traffic, while creating only a moderate processing load for the router. The firewall is located internal to the router and uses an explicit deny policy. In the event that either the router or firewall is improperly configured, there is always a second layer of protection.

Rule order is crucial to a successful implementation. Cisco routers process rules from the top down, which means that as soon as a match is found, processing stops. If Rule

19 was placed at the top of the list, every packet entering the router would be allowed to pass, negating the effect of any rules that come later in the list.

The first ten rules only check the source address of the packet, which requires very little processing power from the router. Any packet with an undesirable source address gets quickly filtered out. These packets could have been prevented by using a standard ACL, but Cisco prevents the application of more than one list to a given interface. Standard ACLs lack the ability to inspect destination port information, and their use would allow in all packets that fit the criteria for Rules 11 through 18.

For the sake of processing, Rules 1 through 18 could be handled in any order, but it is more logical to group similar rules together. When checking the ACL configuration, it would be easy to miss an important entry if there was no logical order. The rules that inspect source addresses are grouped together, and the ones that look at the destination port form another group. Logging everything could be temporarily enabled in order to tune the processing speed of the router. By discovering the nature of the rules that are being used most often, those rules can be moved closer to the top of the list. If a given rule is being used by much of the traffic entering the router, it makes sense to have those packets processed very early on. A packet that is denied by the first rule causes no further burden on the router.

Egress Rules

Table 3.2 includes the egress ACLs, applied incoming to interface.

#	Act	Proto	Source IP	Dest IP	Dest Port	Log ?	Description
1	Deny	UDP	Any	Any	69	Log-input	Deny Outgoing TFTP Services
2	Deny	TCP	Any	Any	445	Log-input	Deny Outgoing SMB Services
3	Deny	UDP	Any	Any	514	Log-input	Deny Outgoing Syslog Services
4	Deny	TCP	Any	Any	135-139	Log-input	Deny Outgoing Windows Services
5	Deny	UDP	Any	Any	135-139	Log-input	Deny Outgoing Windows Services
6	Deny	UDP	Any	Any	161-162	Log-input	Deny Outgoing SNMP Services
7	Deny	TCP	Any	Any	6000-6255	Log-input	Deny Outgoing X-Windows Services
8	Deny	ICMP	Any	Any	Echo-reply, Unreachable		Deny Outgoing ICMP Hazards
9	Deny	IP	Pub.122.32.0/28	66.151.158.183	Any	Log-input	Deny Outgoing access to gotomypc.com
10	Allow	ICMP	Any	Any	Echo-request		Testing - Allow Outgoing Ping from GIAC Systems
11	Allow	IP	Pub.122.32.0/28	Any	Any		Permit GIAC Public Addresses to Internet
12	Deny	Any	Any	Any	Any	Log-input	Deny & Log Everything Else

Table 3.2 – Router Egress ACLs – Applied to interface Pub.122.32.4

All packets that leave the GIAC network should include a public source IP address that has been assigned to GIAC. Packets attempting to exit with any other address are either malicious or the result of an improper configuration of the firewall. For this reason, rules that regulate outgoing traffic can be much more restrictive than those that apply to incoming traffic.

The rules in table 3.2 are very similar to those that are applied to the ingress filter with only a few exceptions. Since legitimate incoming requests for access to the Web server or E-mail server could be generated by any computer in the world, a deny-all rule would not work at the end of the ingress filter. Having a deny-all rule at the end of the egress filter though, works because there is a small range of addresses that can generate outgoing traffic. This explicit deny is called a clean-up rule because it blocks everything that has not been explicitly allowed. Because of the clean-up rule, it is unnecessary to individually block outgoing unassigned, private, or test addresses. This fact results in a much shorter ACL for the exiting traffic.

Rules 1 through 7 apply to the same sensitive internal services as the ones listed in the ingress rules. Even though connections to those services are prohibited from entering the network, they can still pose a security problem. A system on the GIAC network could become infested with malware, which was designed to exploit weakness in those services, and attempt to return sensitive information to the Internet. By preventing those services from leaving, malware of this nature poses much less of a threat. Logging those attempts with log-input not only makes administrators aware of the situation but also records the MAC address of the offending internal system. Having the MAC address will allow that machine to be quickly disconnected.

Rule 8 prevents ICMP from answering ping requests, which can be used in the initial scanning stages of an attack. Depending on the answer to a ping request, an attacker can ascertain specific information about the systems in question. Armed with the ping replies, the attacker can make better decisions about how to proceed. By blocking them at the router, no reply is returned, which gives to attacker no information.

Rule 9 is especially critical, as it pertains to the firewall subversion scheme that was devised by www.gotomypc.com. If this software is installed on GIAC computers, it will attempt to contact the gotomypc Web site at regular intervals. Since this appears to the firewall as a legitimate outgoing HTTP request, the data is allowed to pass. Once the connection is established, the employee who installed the software can connect from his or her home computer to the work computer by going through the gotomypc Web site. This arrangement allows the user to work from home or use the GIAC computer to attack other systems. In either case, data is allowed to traverse the firewall with gotomypc as the man-in-the-middle. The security implications of this “tool” are staggering. Log-input is again used to record the MAC address of the offending workstation. Behavior of this nature is strictly forbidden in the GIAC acceptable use policy, and this action clearly cannot be tolerated.

Rule 10 allows outgoing ping requests, which helps network-troubleshooting efforts.

Rule 11 allows all GIAC systems with a valid public IP address to access the Internet. Those systems that use a private IP address pass their packets through the NAT function of the firewall where the private IP is replaced with a public address. The only condition checked with this rule is the source IP address. While the packets could also be filtered based on the protocol, at some point, a decision must be made about the balance between security and performance of the router.

Rule 12, the clean-up rule, denies any other packets that don't get filtered out by the previous rules. It is much more efficient for the router to process this single, general rule that blocks everything than to process an assortment of specific rules that accomplish the same thing.

Egress rule order is important for the list because if Rule 11 gets placed above all of the others, all traffic that had a GIAC public source address would be allowed out. Rules 1 through 9 could be in any order within that group, but they must come before the allow rule (# 11). If Rule 12 was placed anywhere else on the list, all rules below it would be overridden. This order is especially crucial as it relates to Rule 11. If logging everything was temporarily enabled, the rules could be arranged so the ones that get used most often are closer to the top. This rearranging would only be practical if it retained the integrity of the desired filtering.

Firewall Policies

The firewall offers the second line of defense for GIAC by being installed internal to the router. All network traffic that enters or leaves the perimeter is forced to flow through both devices. The four Ethernet interfaces that are installed allow the following connections:

Eth0 is the external interface to the router (Pub.122.32.9).
Eth1 is the Screened Subnet interface (10.167.105.7).
Eth2 is the internal network interface (10.42.251.3).
Eth3 is the SmartCenter control station.

The only device connected to Eth3 is the workstation that is used to configure all of the Check Point equipment. This computer is physically located in the secure server area, and only network administrators are allowed there. The only connection to this computer is the direct connection to the administrative interface of the firewall.

Table 3.3 controls data that is entering from the external (Eth0) interface.

#	Act	Proto	Source	Destination	Dest Port	Log ?	Description
1	Permit	TCP	Any	10.167.105.16	80		Incoming requests to reverse proxy for GIAC HTTP pages
2	Permit	TCP	Any	10.167.105.16	443		Incoming requests to reverse proxy for GIAC HTTPS pages
3	Permit	UDP	Any	10.167.105.13	53		Incoming requests for GIAC DNS info - UDP Only
4	Permit	TCP	Any	10.167.105.11	25		Incoming E-mail to Mail Proxy
5	Permit	UDP	Any	Firewall Eth0	500	Log	IKE Key Negotiation for IPSEC
6	Permit	IP	Any	Firewall Eth0	50	Log	ESP Protocol for IPSEC Tunnel
7	Permit	UDP	Pub.122.32.4	10.42.251.62	514		Syslog Data from Router to Syslog Server
8	Drop	Any	Any	Any	Any	Log	Clean-up Rule to Drop Everything Else & Log Those Attempts

Table 3.3 - Rules - Incoming Data to Firewall Interface Eth0

Rules 1 and 2 allow anyone to access GIAC Web pages via HTTP and HTTPS by going to the reverse proxy. With Rules 3 and 4, anyone can also send E-mail via the mail proxy, or get GIAC screened subnet DNS information. Zone transfers are disallowed on all GIAC DNS servers, and this policy is reinforced by blocking TCP 53 access.

Rules 5 and 6 allow anyone to attempt connection to the VPN. This is not the ideal arrangement from a security perspective, but the traveling sales group needs access from virtually any IP address. If the firewall was very selective about the IP addresses that could connect to the VPN, access for this group would be blocked. The Check Point package provides strong authentication; therefore, there is still a layer of protection. All VPN connection attempts are logged, so this situation can be monitored.

Rule 7 allows the Cisco router to send Syslog data to the internal Syslog server.

Rule 8 is the clean-up rule, which drops all traffic that has not been explicitly allowed. All packets that hit this rule will be logged, which will help monitor the behavior of the router and the nature of packets that unsuccessfully attempt to access the GIAC network.

Table 3.4 controls data that is entering from the screened subnet (Eth1) interface.

#	Act	Proto	Source	Destination	Dest Port	Log ?	Description
1	Permit	TCP	10.167.105.16	10.42.251.31	80		HTTP Requests from Reverse Proxy to Web Server
2	Permit	TCP	10.167.105.16	10.42.251.31	443		HTTPS Requests from Reverse Proxy to Web Server
3	Permit	TCP	10.167.105.11	10.42.251.43	25		Incoming Mail from Mail Proxy to Mail Server
4	Permit	TCP	10.167.105.19	Router S0	80		Outgoing HTTP Requests from Web Proxy to Internet
5	Permit	TCP	10.167.105.19	Router S0	443		Outgoing HTTPS Requests from Web Proxy to Internet
6	Permit	TCP	10.167.105.11	Router S0	25		Outgoing Mail from Mail Proxy
7	Permit	UDP	10.167.105.13	Router S0	53		Allows Screened Subnet DNS Record Updates from Internet
8	Permit	UDP	10.167.105.0/24	10.42.251.62	514		Syslog Data from Screened Subnet Servers to Internal Syslog Server
9	Drop	Any	Any	Any	Any	Log	Clean-up Rule to Drop Everything Else & Log Those Attempts

Table 3.4 - Rules - Incoming Data to Firewall Interface Eth1

Internet users must go through either the reverse proxy to request GIAC Web pages or the mail proxy to send E-mail to GIAC. Rules 1 and 2 allow the incoming flow of HTTP and HTTPS requests to pass from the reverse proxy to the actual internal Web server. Rule 3 allows incoming mail to be forwarded from the mail proxy to the actual internal mail server.

Internal users are allowed to access Web pages from the Internet and send external E-mail by passing those requests through either the Web proxy or the mail proxy. Rules 4 and 5 allow their HTTP and HTTPS data to pass from the Web proxy to the Internet. Rule 6 allows their outgoing E-mail to pass from the mail proxy to the Internet.

Rule 7 allows the DNS server in the screened subnet to update its records from other DNS systems on the Internet.

Rule 8 passes the Syslog data from all servers in the screened subnet to the Syslog server on the internal network.

Rule 9, the clean-up rule, drops all traffic that has not been explicitly allowed. All packets that use this rule are logged, which helps monitor any interesting packets that do not fit the rest of the rules.

Table 3.5 controls data that is entering from the internal network (Eth2) interface.

#	Act	Proto	Source	Destination	Dest Port	Log ?	Description
1	Permit	TCP	10.42.251.0/24	10.167.105.19	80		Outgoing HTTP Requests from Internal Network to Web Proxy
2	Permit	TCP	10.42.251.0/24	10.167.105.19	443		Outgoing HTTPS Requests from Internal Network to Web Proxy
3	Permit	TCP	10.42.251.43	10.167.105.11	25		Outgoing Mail from Mail Server to Mail Proxy
4	Permit	UDP	10.42.251.55	Router S0	53		Allows Internal DNS Server to Receive Record Updates from Internet
5	Drop	Any	Any	Any	Any	Log	Clean-up Rule to Drop Everything Else & Log Those Attempts

Table 3.5 - Rules for Data Coming Into Firewall Interface Eth2

When systems on the internal network attempt to access the Internet or send outgoing E-mail, their request must pass from the internal network to the proxies in the screened subnet. From there, the request passes again through the firewall out to the Internet. Rules 1 and 2 allow the HTTP and HTTPS data to pass from the internal computers to the Web proxy. Rule 3 permits the mail to pass from the mail server to the mail proxy.

Rule 4 allows the internal DNS server to receive record updates from Internet. This server is not allowed to pass data outside the internal network.

Rule 5, the clean-up rule, drops all traffic that has not been explicitly allowed. All packets that use this rule are logged for security monitoring purposes.

For the actual implementation, the permit rules must be at the top of the list, and the drop rule must be at the bottom. For performance concerns, the permits with the highest hit rate will be moved to the top of the list. Temporarily allowing the firewall to log every packet and then sorting the log entries by protocol and direction of travel shows which data flows generate the most traffic. This performance tuning technique will be repeated regularly to keep the system operating efficiently.

Check Point Express provides stateful inspection and application intelligence, both of which add additional protection to the rules listed above. By keeping track of state data for both stateful and stateless protocols, many popular attack exploits can be blocked. The router and firewall are both configured to allow packets from the Internet to pass to the proxies on the screened subnet. Many exploits have been developed to allow an attacker to deliver malicious content inside packets that appear to be legitimate Web or E-mail requests. The application intelligence of Check Point recognizes and blocks the malicious packets even though the rule base would have allowed them.

Summary

Together, the filtering abilities of the router and the firewall block a large percentage of potential attack vectors. By keeping all systems updated with the latest patches, anti-virus, and personal firewalls, further protection is realized. Monitoring log files on a daily basis keeps security personnel abreast of any break-in attempts. This knowledge helps them take a proactive stance against security breaches.

GIAC network users and systems are provided very limited access to resources that lie outside the network segment they are connected to. If there is a legitimate business need for allowing additional access, a formal request must be made stating the nature of the access and supporting information to justify the need. Both management personnel and the network security team will review the request, which may be granted if all parties agree that it is a legitimate business need and that it can be granted in a secure manner. No single person is allowed to make changes to the firewall or router rules without first having the approval of the security team and then documenting both the nature and the reason for the change.

References

Assignment One

[1] - Patch Installation Statistics - 4-11-05
http://www.softwarespectrum.com/business/TAAP_Library/Trend_docs/Redefining_Security_to_Combat_Malware_Threats.pdf

[2, 3] - SecureStack and StackGuard Buffer-Overflow Solutions, June 2001
http://infosecuritymag.techtarget.com/articles/june01/departments_products1.shtml

[4] - Trustix OS Information - 4-11-05
http://linux.trustix.com/whatsnew.html

[5, 6] - Randomized instruction set emulation to disrupt binary code injection attacks
University of New Mexico, Department of Computer Science, Gabriela Barrantes, David H. Ackley, Trek S. Palmer, Dino Dai Zovi, Stephanie Forrest, Darko Stefanovi´c, 4-8-05
http://www.cs.unm.edu/~moore/tr/03-02/rise.pdf

[7] - "What is a Rootkit?" - 4-10-05
http://www.tech-faq.com/rootkit.shtml

Assignments Two and Three

[8, 10, 11, 20] – Checkpoint Express Features Datasheet – 3-10-05
http://www.checkpoint.com/products/downloads/express_datasheet.pdf

[9] – Red Hat Enterprise Linux ES v.4 – 3-18-05
http://www.redhat.com/software/rhel/features/

[12] – VPN-1 Edge Datasheet – 3-10-05
http://www.checkpoint.com/products/downloads/vpn-1_edge_datasheet.pdf

[13] - Inside Network Perimeter Security, Northcutt, Zeltser, Winters, Frederick & Ritchey, New Riders Publishing, Indianapolis, 2003

[14] - Linksys Compact Wireless-G Router – 3-7-05
http://www.linksys.com/press/press.asp?prid=197

[15] - Internet Protocol V4 Address Space – 1-27-05
http://www.iana.org/assignments/ipv4-address-space

[16] - Cisco 3725 Multiservice Access Router Whitepaper – 2-27-05
http://www.cisco.com/en/US/products/hw/routers/ps282/ps283/index.html

[17] - Cisco IOS 12.3 Mainline Whitepaper – 2-27-05
http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/index.html

[18, 21] - Bugtraq - Cisco IOS 12.3 and Check Point – 3-11-05
http://www.securityfocus.com/bid/vendor/

[19] – Application Intelligence R55 Datasheet – 3-10-05
http://www.checkpoint.com/techsupport/ng_application_intelligence/r55_updates.html

[22] - NetOptics Gigabit Copper Taps Whitepaper – 3-3-05
http://www.netoptics.com/products/product_family_details.asp?cid=1&pid=54&Section=products&menuitem=0

[23] - Snort – Current Version – 2-20-05
http://www.snort.org/

[24] – ClamAV - How-to – 3-12-05
http://www.linuxjournal.com/article/7778

[25] - Squid Home Page – 3-13-05
http://www.squid-cache.org/

[26] - Host-redirect Hazards – 3-19-05
http://www.insecure.org/sploits/arp.games.html

Additional Works Consulted

SANS Network Security, Firewalls, Perimeter Protection and VPNs
Las Vegas, Fall 2004, Class Lecture, Notes, and Book Set

http://www.giac.org/certified_professionals/practicals/gcfw/0547.php
Jason Hall GCFW Practical – 3-12-05

http://www.giac.org/certified_professionals/practicals/gcfw/0528.php
Don Murdoch GCFW Practical – 2-21-05

http://www.giac.org/certified_professionals/practicals/gcfw/0525.php
Kim Guldberg GCFW Practical – 12-22-04

http://www.giac.org/certified_professionals/practicals/gcfw/0525.php
Gregory Lalla GCFW Practical – 12-22-04

http://www.giac.org/certified_professionals/practicals/gcfw/0521.php
John Swartzendruber GCFW Practical – 12-22-04

http://www.giac.org/certified_professionals/practicals/gcfw/0514.php
Diane Johnson GCFW Practical – 12-22-04