Global Information Assurance Certification Paper

SANS GIAC Certified Firewall Analyst

Practical Assignment

Version 1.7

Submitted by:
James L. O'Brien

Submitted on:
October 12, 2002

Table of Contents

Abstract

The purpose of this paper is to provide an overview of the information technology security architecture of GIAC Enterprises. The GIAC Enterprises security architecture is intended to focus on the basic security concepts of availability, integrity, and confidentiality through a Defense in Depth methodology of protection. The paper is divided into four parts. The first part covers the general security architecture of GIAC Enterprises' infrastructure. The second part focuses on the network security policies implemented on routers and firewalls. The third section covers an audit of the network security policies. Finally, the fourth section reviews an alternate design for GIAC Enterprises and potential vulnerabilities present.

Assignment 1 – Security Architecture

GIAC Enterprises Limited (GIAC) is a small startup corporation, founded in 2001 and based in the Midwest region of the United States. Four displaced business and technology professionals from local marketing and consulting companies in Milwaukee, Wisconsin formed GIAC. As such, the organization's corporate headquarters and technology facilities are located in Milwaukee.

GIAC's business model focuses on the creation, production, and sale of fortune cookie sayings. There are two distinct lines of business within this model. The first is the sale of fortunes to companies that produce fortune cookies for resale to restaurants and grocery stores. The second is the sale of fortune cookies with customized fortunes designed to market a company on a promotional basis. Within both lines of business, GIAC's true product is intellectual property, the actual fortune. Through strategic partnerships with a variety of other organizations, GIAC is able to remain small and to minimize the overhead normally associated with distribution and manufacturing facilities.

GIAC would like to branch out into other lines of business including the sale of motivational posters and greeting cards. The current economic environment and the failure of many Dot-Com businesses have made the owners of GIAC Enterprises cautious. They do not want to spread themselves to thin or risk the success of the company by pursuing additional lines of business before reaching profitability.

Business Profile and Processes

There are four key business operations that influenced the design of GIAC's information technology (IT) infrastructure. These operations include:

- Online sales to customers within both lines of business
- Interaction with and purchases from suppliers
- Strategic partnerships with other businesses
- Sales and internal business processes by employees

Understanding each of these operations is essential in grasping the business decisions that influenced the development of IT infrastructure.

Customers

As mentioned previously, GIAC maintains two lines of business. Both lines of business are conducted via the organization's e-commerce Web site. The resultant Web site has two key areas for business with customers including:

- Business-to-business Web interface to allow food producers to download the latest set in the series for their subscription.
- Business-to-business Web interface to allow a company to request a selection of fortune cookies and marketing focused fortunes for their organization.

Strategic Business Customers

For the first line of business, GIAC has produced four series of fortunes, consisting of 250 fortunes each, that are available via the Web site. Additional sets for each series are added quarterly and consist of 50 fortunes. For first time customers or customers interested in a new series, ten sample fortunes for each series are available for viewing on the Web site. Businesses interested in reviewing the samples must click-through an online agreement that prohibits the party from using the samples without a subscription to the fortune series.

A series is available on a subscription basis, by which a business customer agrees to a quarterly subscription service for the rights to license and produce fortune cookies using the fortunes. GIAC's subscription service requires a business customer to open an account and establish a line of credit.

When a new series is released to a customer, an automated batch process identifies the current series subscription for the customer, retrieves the latest set for a series from the GIAC fortune database, and bundles them into a text-formatted file. The file is encrypted and the batch process then transfers the file to a secure directory on the Web site accessible only by the customer and GIAC. The batch processor is also capable of creating or deleting files for other series in the event the customer adds or removes subscriptions. Customer authentication to the Web site is currently handled using a username and a one-time password (OTP) generated by a token.

Potential strategic business customers can also use the business-to-business Web interface to open an account. All accounts are subject to a credit check of the requesting organization, after which, the business customer is provided with an online account for accessing their selected fortune series. Subscription customers are invoiced on a quarterly basis.

Casual Business Customers

For the second line of business, GIAC provides a Web interface that allows a customer to create their own fortunes for their business using GIAC's random fortune generator or by typing in their own fortunes. A customer's order may

consist of up to five of any combination of random and personally written fortunes. The customer is allowed to reject specific random fortunes, in which case the random fortune generator selects a new fortune for each one rejected. This process is repeated until the customer is satisfied with their selection of fortunes.

The customer can then select from a variety of flavors of fortune cookies, and the quantity of each for each fortune. To complete the order, the customer must provide a mailing address, select a method of shipping, and a credit card. The credit card is then verified using GIAC's credit card verification system.

After the order is verified and approved, the customers order is written to the customer database, including the authorization code from the credit card verification. The customer's credit card is not written to the database; however, the customer's name, business, and mailing address are retained as part of the order entry recorded in the database. On a nightly basis, the batch processor exports all of the casual business customer orders for the day, and transfers the file to GIAC's business partner, Dave's Bakery.

Suppliers

While GIAC develops some of its fortunes in-house, free-lance authors around the world provide other fortunes. Authors under contract by GIAC to produce fortunes are provided access to the organization's collaboration Web site. This Web site allows suppliers to submit new fortunes, in a text format, on a regular basis. Authentication to the collaboration Web site is also handled using a username and a one-time password (OTP) generated by a token.

Once an author has submitted fortunes for review, GIAC staff review the submissions and determine which fortunes will be purchased from the author. Approved fortunes are incorporated into a set for the next release of a series and are loaded into the GIAC fortune database. A supplier can review the list of approved and denied fortunes online through the collaboration site. GIAC has the ability to provide comments and feedback for each denied fortune. Suppliers are paid a variable fee based on the number of accepted submissions; GIAC's finance department issues such payments as submissions are approved.

Partners

GIAC has two distinct types of business partners based on the two lines of business it maintains. International translations partners license GIAC fortunes, translate them into different languages, and then resell them through their own channels. Order fulfillment partners produce fortune cookies ordered for business promotional purposes as part of GIAC's casual business customer base.

International Translation Business Partners

GIAC's international partners currently translate fortunes into five different languages. These partners are allowed access to fortunes through the collaboration Web site. Through this site, GIAC's partners can download

bundled series of fortunes for translation. Like all other access to the collaboration Web site, authentication is handled using a username and a one-time password (OTP) generated by a token. GIAC's business partners then resell the translated fortunes via their own Web sites or through direct sales.

While GIAC receives a percentage of the revenue generated by its international partners, this business is secondary to their English based fortune business. In addition, several of GIAC's strategic business customers are international organizations. In these cases, GIAC does not wish to provide incentive to its strategic business customers to acquire GIAC fortunes through alternate channels, especially channels where GIAC will receive less revenue. For this reason, GIAC's business partners cannot download current fortune sets. Sets are made available to business partners two quarters after their release to strategic business customers.

When a bundled series is released to a partner, an automated batch process checks the release date for each set in a series and excludes the two most current releases (i.e. two quarters). This process then retrieves the sets for each series from the GIAC fortune database, and bundles them into a text-formatted file. The file is encrypted and the batch process then transfers the file to a secure directory on the collaboration Web site accessible only by the partner and GIAC.

Order Fulfillment Business Partners

GIAC also maintains a partnership with a local food producer to support its second line of business. Customers ordering fortune cookies with stock promotional fortunes for their business are actually serviced by Dave's Bakery. Dave's Bakery receives a list of customer orders each day via an automated process that securely transfers the files to a secure area of the bakery's e-commerce site.

As part of its relationship with GIAC, Dave's Bakery prints each customer's requested fortunes and inserts them into the cookies as part of the production process. The cookies are then packaged and shipped to the customer directly from Dave's Bakery. Depending on the size and complexity of a given customer's order, the order typically ships from Dave's Bakery within 72 hours. Dave's Bakery invoices GIAC for all cookies produced and shipped as part of its strategic partnership. Invoices are typically generated as orders are completed and are sent to GIAC's finance department through the mail.

GIAC Employees

There are approximately 25 people on GIAC's staff. A majority of these employees work out of the corporate headquarters. Employees perform a variety of duties based on job function. In general, all employees have access to systems for the following:

- Email, calendaring, and business collaboration
- File and print services

- Internet access
- Infrastructure services (DHCP, DNS, NTP, etc.)

Employees working within the finance department have access to GIAC's financial system, while the human resources department has access to the human resources management system (HRMS). Employees who author fortunes have access to the GIAC collaboration Web site, where they can submit new submissions for review in a fashion similar to free-lance authors. The editorial staff also has access to the collaboration Web site and performs regular reviews of new fortune submissions. An editor has the ability to approve or reject submissions, as well as to provide comments on each individual submission. Approved submissions are loaded into the GIAC fortune database through the collaboration Web site interface. GIAC's customer service representatives have access to the customer Web site for the purposes of supporting customer questions or problems.

The GIAC sales force is a predominantly mobile team that travels internationally attempting to identify and sell to new customers. The sales force also frequently meets with existing customers to discuss new business opportunities. The mobile sales force has access to the standard GIAC technology services as well as GIAC's customer relationship management (CRM) application. The sales staff also has access to the customer Web site for the purposes of reviewing customer subscriptions and assisting customers with pre-sale questions.

GIAC also maintains a five user IT staff. The five members of the staff support the applications and infrastructure that allow GIAC to function as an e-commerce company specializing in intellectual property. The IS staff consists of two application developers who provide ongoing development and support of the customer and collaboration Web sites. There are two individuals responsible for security, networking, and administration of the e-commerce servers. The remaining individual is responsible for support of GIAC's internal network and desktop systems.

Technical Architecture

Based on the above business profile and associated business processes, GIAC has developed a technical architecture for its applications and infrastructure. The GIAC technical architecture operates under five business objectives including:

- Facilitate business processes with Web services to enhance the customer experience.
- Develop technology solutions in-house to gain a competitive advantage in the marketplace.
- Leverage open source software to provide flexibility in design and to minimize costs.
- Implement a security posture that allows GIAC to pursue e-commerce, and to allow employees remote access to its infrastructure.

- Maintain a scalable and modular architecture that is capable of sustaining growth.

Defense in Depth

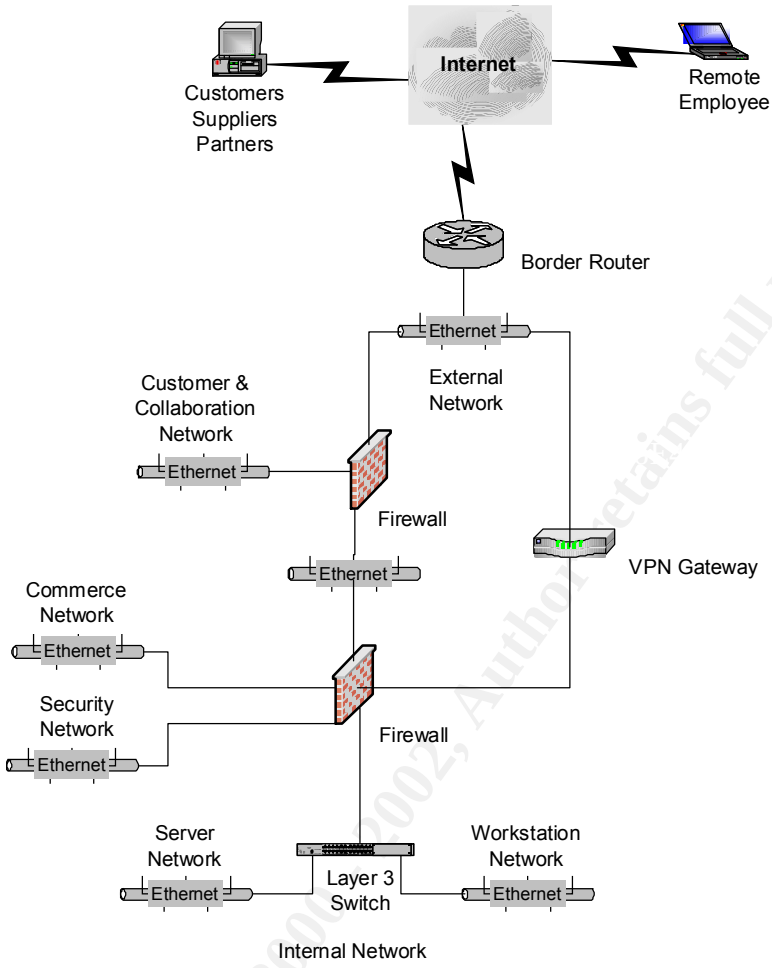
The architecture of GIAC 's network consists of a layered security approach. This allows GIAC to maintain separation of services for the public, its customers, suppliers, and business partners. GIAC has adopted the Defense in Depth approach to information security from the SANS Institute and US Navy. The significant complexity of technology systems, coupled with the numerous bugs and architectural limitations of solutions makes the reliance on any one system for security an ineffective method of protection. By using multiple layers of security, with complimentary technologies, it is possible to establish a robust security posture that is resistant to compromise (Galik).

GIAC had combined application access management, network security, and host security to accomplish the Defense in Depth approach to security. Access to non-public applications is controlled through the use of authentication and authorization controls in each application. The customer and collaboration Web sites use a revere proxy agent to provide authentication and authorization for access to specific resources within each Web application.

Publication of customer subscriptions and international translation partner fortunes undergo encryption before they are stored for access on the corresponding Web site. The data is encrypted to add an additional layer of protection to GIAC's intellectual property in the event a customer's credentials or a Web site is compromised. GNU Privacy Guard (GPG) and the OpenPGP standard are used to provide encryption of this data. According to RFC2440, "OpenPGP uses two encryption methods to provide confidentiality: symmetric-key encryption and public key encryption." The data is actually encrypted with a symmetric encryption algorithm and a unique one-time use session key. The session key is encrypted using asymmetric encryption, specifically the customer or partner's public key (Callas, Donnerhacke, Finney, and Thayer, p.5).

GIAC retains a copy of all of its customer and partner's public keys. Only the private key of the customer or partner that corresponds to the public key used to encrypt the session key can be used to decrypt the file.

The first layer of defense through network security for GIAC is the border router, which relies on access lists to enforce inbound and outbound policy at the edge of the network. GIAC uses a combination of firewalls to control access to key networks. These key networks include: External, Customer and Collaboration, Commerce, Security, Server, and the Workstation networks. This segregation is supplemented through the use of network based intrusion detection systems (IDS). Remote access to the network by employees is delivered through the implementation of a virtual private network (VPN) gateway. The following diagram illustrates the compartmentalization of GIAC's network:



Logical Network Diagram

All host systems accessible over the Internet directly or indirectly (i.e. accessed by another host that is directly accessible) are hardened to minimize the opportunity for compromise. At the minimum, host hardening includes the installation of the minimum operating system (OS) configuration, the removal of unnecessary services, and tuning of common host parameters (Lawson).

Access Requirements

There are several groups that require access to GIAC resources over the Internet and internally through GIAC's network. The general public requires access to readily available information about the company. GIAC has also attempted to standardize its access requirements for its customers, suppliers, and business partners. Access for employees varies depending on the employee's role within the organization.

Public Access

The general public has access to the company's public Web site. This Web site resides on the same systems as the customer Web site. GIAC's public Web site is accessible through the hypertext transfer protocol (HTTP) on port 80/tcp.

The public Web site is on a separate server from the customer and collaboration Web sites. As such, there are no authentication requirements for viewing Web pages on this system.

In addition, the general public has the ability to send email to GIAC. To support this access, the Internet has the ability to connect to GIAC's mail server via the simple mail transfer protocol (SMTP) on port 25/tcp. To support these activities, GIAC allows access to its external domain name service (DNS) servers for DNS on port 53/tcp and 53/udp.

As this access is available to the general public, this access also applies to customers, suppliers, and partners of GIAC.

Customer Access

Customers access the customer Web site over the Internet using HTTP and HTTPS. The casual business customer can make purchases of fortune cookies through GIAC's public Web site (the user actually conducts the transaction through the secure Web server). This access is conducted via HTTP over the secure socket layer (SSL) on port 443/tcp (also known as HTTPS).

Strategic business customers have access to a separate Web server known as the secure Web server. This server houses the strategic customer and collaboration Web sites. Access to these sites is conducted via HTTPS. Access to this secure Web server is actually handled by a reverse proxy agent on the system. This reverse proxy agent requires user authentication before it will provide access to any URL on the Web server. For this reason, customers over the Internet can connect to the Web login server via HTTPS to perform authentication using a username and token.

An authenticated customer belongs to a specific user role, the Strategic_Customer role. The revere proxy agent will only allow users with this role access to the strategic customer site. In addition, the reverse proxy passes the user's identity back to the Web application. Using this information, the Web application only serves data to the customer that belongs to the customer. In addition to restricting data based on the customer from the database, each customer has their own directory on the Web site that the application can serve data from.

Supplier Access

Like the strategic customers, suppliers have access to the secure Web server using a username and token. Suppliers also access the site over HTTPS on port 443/tcp. The reverse proxy agent handles authentication and authorization of access to the site. As such, authenticated suppliers belong to the Supplier role and are allowed access to the supplier area of the collaboration site. The reverse proxy also passes the user's identity to the collaboration Web application. The application can then use this information to restrict the data served back to the supplier and to track submissions by the supplier when they are recorded in the collaboration database.

Partner Access

GIAC's international translation partners have access to the secure Web server for the purposes of accessing the collaboration Web site. Again, access to the site is handled by the reverse proxy agent, which uses the Web login server to provide authentication via a username and token. Once authenticated, a partner is allowed access to the collaboration site as part of the Partner role. Access to the collaboration Web site is restricted to the area reserved for partners. The collaboration Web site relies on the user's identity to restrict the data the partner is served to the specific partner's directory on the Web site.

Order fulfillment business partners do not have any special access to the GIAC secure Web server. As such, partners like Dave's Bakery have the same level of access as public users. This may change in the future depending on the overall success of GIAC's promotional fortune cookie business.

Employee Access

Internal Employees

GIAC employs layers of protection for its IT applications and infrastructure. In addition to restrictions placed on network level access, employees are assigned unique accounts for access to the systems they need access to. Employees are only assigned access that is necessary to perform their job duties. For example, the human resources department does not have access to the collaboration Web site.

GIAC has standardized its internal network on Windows 2000 Professional workstations and Windows 2000 servers. GIAC maintains a domain controller for authentication, file, and print sharing. In addition, GIAC uses Microsoft Exchange as its email, calendaring, and business collaboration suite. Employees access Exchange via Microsoft Outlook. As internal employees are located on the internal network with the Windows 2000 servers there are no restrictions in place to restrict access based on IP addressing or ports.

Employees also access the companies limited enterprise resource planning installation via a Web interface for the purposes of financial, human resources, and customer relationship management. This access is performed via HTTPS on port 443/tcp. This access is restricted through authentication and authorization within the application. Employees are allowed access to only the sections of the application (i.e. finance) necessary for their job. This relationship is tracked within the application through the use of role-based security. An employee is assigned a specific role which determines whether an employee can access the financial services, human resources, or customer relationship management section of the application.

Those individuals in the organization that require access to the customer Web site or the collaboration Web site access these sites through the secure Web server using a username and token. This access takes place over HTTPS as well. The reverse proxy agent handles authentication and authorization to both sites.

Employees within customer service and sales belong to the Customer_Service role. This role allows them to access the strategic customer Web site. The application allows these individuals to review customer subscription information and customer accounts. Individuals within the editorial staff belong to the Editor role, which allows employees to access the collaboration Web site. The reverse proxy agent passes the user's identity back to the supplier Web application where the editor may review submissions by authors, provide comments on submissions, and approve submissions for loading into the fortune database.

Finally, GIAC's in-house authors belong to the Supplier role and are allowed access to the supplier area of the collaboration site. Once authenticated, the application restricts the data served back to the author and tracks submissions by the author when they are recorded in the collaboration database.

All GIAC employees are allowed access to the Internet. Access is allowed for HTTP (80/tcp) and HTTPS (443/tcp). Employees use the proxy and cache server for their access to the Internet.

Remote Employees

While employees may request remote access, access is only granted for one of the following business reasons:

- An employee travels as part of their job duties and must access the network to be able to perform their job duties
- An employee is responsible for the support of information technology applications and infrastructure
- An employee performs a minimum of 15% of their work after hours or from the comfort of their home

Employees using remote access can access all of the functions and services they would normally access from within the network. An employee must authenticate to the VPN gateway before they are allowed access. This authentication is provided by a username and token. Once the employee is authenticated, they are issued an IP address that allows them access to the server segment of the Internal network and the Customer and Collaboration network. Access to the Customer and Collaboration site and the Internet are restricted to the protocols allowed by employees on the Internal network.

IT Employees

GIAC's IT staff have access to the resources provided for all employees on the Internal network. In addition, GIAC's IT staff also have the ability to use secure shell (SSH) on port 22/tcp to access all servers on the protected network segments as well as the firewalls and VPN gateway. The firewalls and VPN gateway are also accessible using HTTPS on port 443/tcp. The IT staff is allowed to connect to the management server on the Security network. To support Firewall-1 access from the management client, connections for Firewall-1 require 257/tcp (FW1_Log), 18190/tcp (CPMI), 18191/tcp (CPD), and 18192/tcp (CPD_Amon) (Check Point). It is important to note that communication between

a GUI client and the management station requires a different subset of the protocols used to communicate between the management station and the enforcement point.

The IT staff requires access to the SafeWord authentication server on the Security network using SafeWords's proprietary management protocol on 5040/tcp. This access is restricted to the IP addresses of the IT staff's workstations on the Internal network as well as an IP address pool reserved for the IT staff on the VPN gateway.

Since there are security disadvantages with allowing either active (i.e. normal) or passive mode FTP, not even the IT staff are allowed to use the file transfer protocol (FTP) over the Internet. Active mode FTP requires that the FTP server be allowed to connect back to the client for the data channel. While this connection will most likely come from a source port of 20/tcp, the server will connect to a random high-level TCP port on the client. This increases the risk to the FTP client by allowing a connection to a high-level port from an Internet accessible host. Passive FTP attempts to resolve this issue be opening the data control channel to a random high-level TCP port on the FTP server. While this reduces the risk associated with an Internet host connecting to the FTP client, it still requires that an FTP client be able to open a connection to any high-level port on the FTP server (Zwicky, Cooper, and Chapman, p. 456-457).

Security Architecture

IP Addressing

GIAC's Internet service provider (ISP) has assigned a classless address range of 198.7.46.128/25 (i.e. 198.7.46.128 – 198.7.46.255) to GIAC for use. GIAC has opted to divide this range into two unique segments. The first segment, 198.7.46.128/26 (i.e. 198.7.46.128 – 198.7.46.191) is used for the network segment between the border router and the Internet firewall. The second segment, 198.7.26.192/26 (i.e. 198.7.46.192 – 198.7.46.255) is used for the Customer and Collaboration network. GIAC has opted to use public addressing for its publicly addressable resources to minimize the complexity of its DNS configuration, routing, and network address translation (NAT).

GIAC's remaining network segments use private address space as established by RFC 1918. GIAC has chosen to use private addresses from 172.16.0.0/12 block of reserved addresses (Rekhter, Moskowitz, Karrenberg, de Groot, and Lear, p. 3). GIAC uses a combination of segments from 172.16.0.0/16. As part of the planning for addressing, GIAC has left space between its service networks and the choke network as well as the server network and the workstation network. This was done to accommodate future growth.

GIAC uses hide NAT at the Internet firewall to masquerade outbound connections from the private address space to the Internet. Hide NAT is provided using 198.7.46.145 for all connections initiated outbound to the Internet. The only static NAT entry is 198.7.46.146 for the logging server on the Security network. This address exists for the sole purpose of allowing the border router to

send logging messages and retrieve time from the system. The following table provides an overview of the IP addressing used for each network segment:

ID	Network Segment	IP Network/Subnet Mask
A	WAN Network	66.84.218.0/30
B	External Network	198.7.46.128/26
C	Customer & Collaboration Network	198.7.46.192/26
D	Access Network	172.16.1.0/24
E	Commerce Network	172.16.2.0/24
F	Security Network	172.16.3.0/24
G	VPN Network	172.16.4.0/24
H	Choke Network	172.16.8/0/24
I	Server Network	172.16.9.0/24
J	Workstation Network	172.16.12.0/24

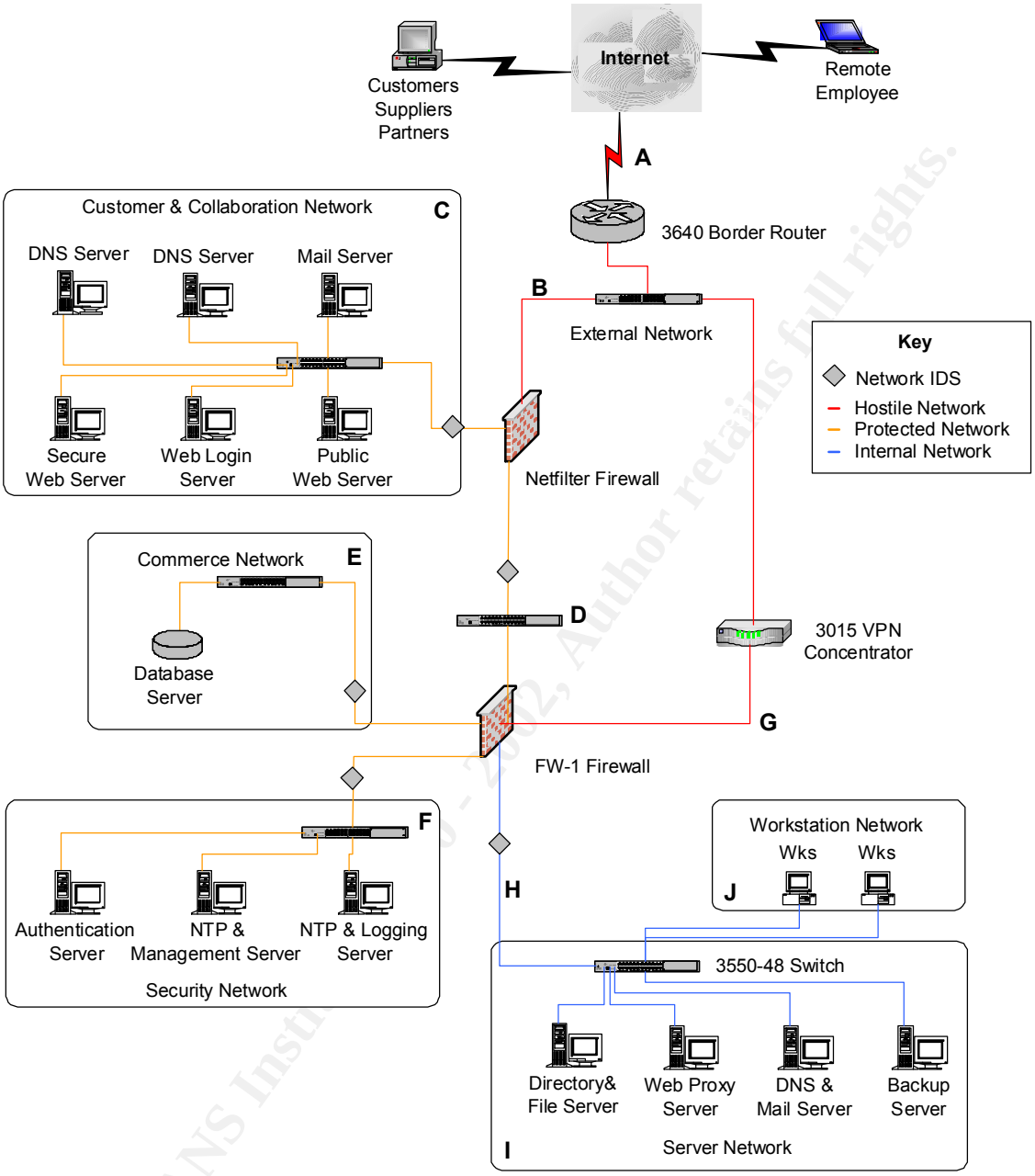
The following table includes a breakdown of the key devices used within GIAC's technical architecture:

Device	IP Address	IF Name	Network Segment
Core Network Devices			
3640 Border Router	66.84.218.2	S0	WAN
	198.7.46.129	Eth0	External
Internet Netfilter Firewall (Linux)	198.7.46.132	Eth0	External
	198.7.46.145	Eth0	External (NAT)
	198.7.46.146	Eth0	External (NAT) for Logging Server
	198.7.46.254	Eth1	Customer & Collaboration
	172.16.1.254	Eth2	Access
3015 VPN Concentrator	198.7.46.135	Eth2	External
	172.16.4.254	Eth1	VPN
	172.16.4.1 – 172.16.4.62	Eth1	VPN (Employee IP Pool)
	172.16.4.225 – 172.16.4.230	Eth1	VPN (IT IP Pool)

Corporate FW-1 Firewall (IPSO)	172.16.1.1	Eth1c0	Access
	172.16.2.1	Eth2c0	Commerce
	172.16.3.1	Eth3c0	Security
	172.16.8.1	Eth4c0	Choke
	172.16.4.1	Eth-s1p1c0	VPN
3550-48 Layer 3 Switch	172.16.8.254	Eth0	Choke
	172.16.9.254	Eth1	Server
	172.16.12.254	Eth2	Workstation
Customer & Collaboration Network			
Sendmail Mail Server (Linux)	198.7.46.200	Eth0	Customer & Collaboration
BIND DNS Server (Linux)	198.7.46.201	Eth0	Customer & Collaboration
BIND DNS Server (Linux)	198.7.46.202	Eth0	Customer & Collaboration
Apache Public Web Server (Linux)	198.7.46.203	Eth0	Customer & Collaboration
Apache Secure Web Server (Solaris)	198.7.46.204	Hme0	Customer & Collaboration
SC Web Login Server (Solaris)	198.7.46.205	Hme0	Customer & Collaboration
Commerce Network			
Oracle Database Server (Linux)	172.16.2.10	Eth0	Commerce
Security Network			
NTP and Syslog Server (Linux)	172.16.3.10	Eth0	Security
NTP and Management Server (Solaris)	172.16.3.11	Eth0	Security
SafeWord Server (Solaris)	172.16.3.12	Hme0	Security
Server Network			
Directory & File Server (Windows 2000)	172.16.9.10	Eth0	Server
DNS & Exchange Mail Server (Windows 2000)	172.16.9.11	Eth0	Server

Squid Web Proxy Server (Linux)	172.16.9.12	Eth0	Server
Networker Backup Server (Linux)	172.16.9.13	Eth0	Server
Workstation Network			
Employee Printers	172.16.12.10 – 172.16.12.19	Eth0	Workstation
Employee Workstations (Windows 2000)	172.16.12.20 – 172.16.12.239	Eth0	Workstation
IT Workstations (Windows 2000)	172.16.12.240 – 172.16.12.250	Eth0	Workstation
IDS Sensors			
Snort IDS Sensor (Linux)	172.16.3.100	Eth0	Security
	0.0.0.0	Eth1	Customer & Collaboration
Snort IDS Sensor (Linux)	172.16.3.101	Eth0	Security
	0.0.0.0	Eth1	Commerce
Snort IDS Sensor (Linux)	172.16.3.102	Eth0	Security
	0.0.0.0	Eth1	Security
Snort IDS Sensor (Linux)	172.16.3.103	Eth0	Security
	0.0.0.0	Eth1	Choke
Snort IDS Sensor (Linux)	172.16.3.104	Eth0	Security
	0.0.0.0	Eth1	Access

The resulting GIAC technical architecture is displayed below:



Infrastructure Diagram

External Network

Border Router

GIAC's border router is a Cisco 3640. This is a modular router that should provide acceptable performance when coupled with access control lists (ACL). The router has one T1 serial interface and a single 100 Mbps Fast Ethernet interface. The router runs version 12.2 of the Cisco internetworking operating system (IOS). Additional information regarding the router is available in the

Cisco 3600 Series Modular, High-Density Access Routers document at Cisco Systems Web site at http://www.cisco.com/univercd/cc/td/doc/pcat/3600.htm

The router has been hardened by stripping away unnecessary services and through the addition of banners. GIAC has implemented extended ACLs on the router to control inbound and outbound traffic. The router has been configured to ignore source-routed packets. As part of this filtering, the router uses ingress filtering to:

- Deny spoofed IP packets with a source of a private IP address
- Deny packets with GIAC IP addressing arriving inbound on the serial interface
- Deny packets that are not necessary to support critical business functions

The router also performs egress filtering to:

- Deny packets with a source address in a private IP range from leaving the GIAC network

Internet Firewall

GIAC has chosen to run Netfilter, version 1.2.7a, as its primary Internet firewall. Netfilter was chosen because of its comprehensive logging capabilities. Netfilter can be found at http://www.netfilter.org. GIAC uses a hardened version of Red Hat Linux 7.2 for the operating system. The bare minimum packages are installed as part of the initial setup and configuration (see Appendix B, OS hardening). Red Hat's up2date program is then used to update 7.2 to the latest version of all installed packages. GIAC runs up2date on a bi-monthly basis to maintain current revisions on its Linux systems. The Linux hardening application, Bastille version 1.3.0, is then run against the system using a customized configuration script (see Appendix B, OS hardening). Additional information regarding Bastille can be found at http://www.bastille-linux.org. GIAC spent considerable time and energy developing its hardened version of Red Hat Linux and as such has leveraged this process to build all of its Linux systems.

The Internet firewall is used to filter inbound and outbound access for the Customer and Collaboration network as well as outbound access to the Internet from the GIAC network. The firewall implements a security policy outlining rules for this access as well as perform NAT for outbound traffic to the Internet.

VPN Gateway

GIAC considered using its FW-1 firewall to provide VPN services, but opted for a dedicated VPN solution for the following reasons:

- Combining the corporate firewall and VPN would add complexity to troubleshooting and configuration management
- VPN tasks such as encryption and decryption could impact the performance of the corporate firewall

- The VPN gateway could be placed outside a firewall and treated as an external device for the purposes of restricting future access for business partners or suppliers

The Cisco 3015 VPN Concentrator was selected based on its modular design. This design allows GIAC to use software-based cryptography and later upgrade to hardware-based cryptography as its needs grow. The GIAC VPN implementation runs version 3.6 of the software, is IPSec protocol based and uses the Encapsulating Security Payload (ESP) protocol. Internet Key Exchange (IKE) is used to provide authentication of peers and to negotiate policies for communications. Packet authentication is performed using an MD5-HMAC-128 hash with encryption performed via 3DES (Cisco Systems, Tunneling Protocols, p. 9).

The VPN concentrator uses RADIUS to validate the username and one-time password generated by the SafeWord token to authenticate the user. Employees are assigned an IP address from one of the two IP address pools based on the class attribute returned by the RADIUS server. When the class attribute is set to OU=employee, the user receives an address from the general employee IP address pool. When the class attribute is set to OU=itemployee, the user receives an address from the IP address pool reserved for IT staff (Cisco Systems, User Management, p.41).

The VPN concentrator is configured to enforce the operation of a firewall on the VPN client end. The VPN client uses a feature known as Are You There (AYT) that polls the personal firewall on the client every 30 seconds to verify it is running (Cisco Systems, Configuration Information for an Administrator, p. 6). GIAC allows its employees some flexibility in the policy present on the client system and provides Zone Alarm Pro for employees that receive VPN access.

Customer & Collaboration Network

External DNS Servers

GIAC maintains the domain name giacfortunes.com. The external DNS servers are part of a split DNS configuration; as a result, they only maintain information about publicly accessible systems in the giacfortunes.com DNS zone. As a precaution, GIAC has also registered giac-fortunes.com and simply creates aliases (i.e. CNAMEs) for each address (i.e. A) record in giacfortunes.com.

The DNS servers run the hardened version of Red Hat Linux 7.2 developed for the Internet firewall. GIAC has chosen to run BIND version 9.2.1 to provide DNS service. GIAC considered the benefits of using the version provided by Red Hat (9.2.1) versus an in-house maintained version of BIND (8.3.3) and opted to use the version provided by Red Hat. This was done to minimize the administrative overhead in maintaining BIND. The GIAC Internet firewall configuration allows DNS queries for 53/udp and 53/tcp from the Internet and the internal DNS server. The systems also have the ability to perform queries using 53/udp and 53/tcp to the Internet.

GIAC has configured BIND to prevent zone transfers from all hosts except for the secondary/slave external DNS server. The DNS server on the Server network is allowed to perform recursive queries against the external DNS servers, while Internet hosts are not allowed to do so. GIAC has also decided to perform extensive logging using syslog. A sample of the configuration within named.conf used to provide this functionality is provided below:

```
options {
        directory "/var/named";
        max-transfer-time-in 180;
        cleaning-interval 45;
        allow-transfer { 198.7.46.202; };
        allow-recursion { 172.16.9.11; 198.7.46.192/26 };
};

logging {
      channel giac_syslog {
            syslog local4;
            severity info;
            print-category yes;
            print-severity yes;
            print-time yes;
      };

      category default        { giac_syslog; };
      category panic          { giac_syslog; };
      category xfer-in        { giac_syslog; };
      category xfer-out       { giac_syslog; };
      category response-checks      { giac_syslog; };
      category security       { giac_syslog; };
};
```

External Mail Server

GIAC's external mail server, also known as the mail gateway, accepts mail from the Internet for delivery internally as well as mail from the Exchange server for delivery externally. This traffic takes place over SMTP on port 25/tcp. The server runs the hardened version of Red Hat Linux 7.2 discussed above. For the purposes of mail relaying, the system is configured to use 8.11.6 as provided by Red Hat. Sendmail is configured to use a mailertable to send all email for giacfortunes.com and giac-fortunes.com to the internal Exchange Server. It is also configured to only relay email to the Internet that has a sender address in giacfortunes.com. GIAC is currently considering whether it will restrict this further by identifying the known email addresses in giacfortunes.com on the Sendmail server.

Public Web Server

The public GIAC web site is served by Apache 1.3.22 provided by Red hat running on the GIAC hardened build of Red Hat 7.2. GIAC has taken steps to harden the configuration of Apache based on the SANS Institutes, Securing Linux Step-By-Step. GIAC has made the following changes to Apache:

- Run the Apache binary (httpd) as the user and group apache

- Set the default access to the root directory to more restrictive permissions:

```
<Directory />
Options None
AllowOverride None
order deny, allow
allow from all
</Directory>
```

- Set the ExecCGI option only on the directories that have CGI scripts
- Remove the default CGI scripts provided with Apache and audit all CGI scripts used for proper bounds checking
- Use SSL for the casual customer portion of the site with the SSLRequireSSL directive

Web Login and Secure Web Server

The Web Login and secure Web servers run a hardened version of Solaris 8. GIAC performs an installation using the core build. Once this is complete, GIAC adds a few essential packages for support of the system and other applications including Documentation Tools, On-Line Manual Pages, Bundled libC, GNU Zip Compression Utility, and the Sun Perl Packages. GIAC installs a customized version of OpenSSH and OpenSSL on the system for secure remote access. OpenSSH can be found at http://www.openssh.org and OpenSSL can be found at http://www.openssl.org

GIAC then applies the latest Sun patch cluster. GIAC performs the first level of hardening by removing unnecessary services and disabling or deleting unneeded accounts (see Appendix B, OS hardening). The Solaris hardening tool, Titan version 4.0 Beta 5, is then run against the system using a customized configuration script (see Appendix B, OS hardening). Additional information regarding Titan can be found at http://www.fish.com/titan/TITAN_documentation.html. This version of Solaris is used on all GIAC systems running Solaris.

GIAC has chosen to use Secure Computing's SafeWord Premiere Access to provide authentication and authorization of access to the secure Web server. The secure Web server runs a hardened version of Apache 1.3.22 similar in configuration to setup for the public Web server. The secure Web server also has a SafeWord Universal Web Agent (UWA) installed. All connections to the Web site are intercepted by the UWA, where the UWA redirects the user to the SafeWord Web login server (WLS). The WLS then connects to the SafeWord authentication server over 5031/tcp. If the user is trusted, the WLS sends a session cookie to the user and passes the confirmation for authentication back to the UWA. The UWA then allows access to the Web site and passes the user's identity back to the Web application. Both the secure Web server and the Web login server are accessible via HTTP and HTTPS on port 80/tcp and port 443/tcp.

Server and Workstation Networks

Corporate Firewall

GIAC has implemented a firewall appliance solution for its corporate firewall. The appliance solution lessens the administrative overhead of building and maintaining a secure operating system for the firewall application. In this scenario, Nokia's IP 530 running IPSO 3.6 FCS2 are used. Nokia's IPSO operating system is based on FreeBSD (Welch-Abernathy). The Nokia IP530 has four integrated 10/100 Ethernet ports, plus room for expansion to support Gigabit Ethernet (Nokia). GIAC's configuration includes an additional four port 10/100 Fast Ethernet card. Administration of the operating system is handled via SSH and HTTPS. Telnet and HTTP have been disabled through the Web interface, Voyager.

Check Point Firewall-1Next Generation with feature-pack 3 is installed on the Internet firewall. GIAC recently upgraded to the IP 530 from the older IP 330 for performance reasons. During this upgrade, GIAC seriously evaluated Firewall-1 Next Generation (NG) with feature-pack 3 and chose to migrate from version 4.1.

As a stateful inspection firewall, Firewall-1 leverages a distributed model to provide enforcement, management, and logging. Firewall-1 works by placing a "shim" between layers 2 and 3 of the TCP/IP stack. The operating system's stack still handles traffic, but only after it is processed by the Inspection engine in Firewall-1 (SANS Institute, Perimeter Protection: Firewall Technology, Firewall-1 and PIX, p. 158). GIAC chose a commercial firewall product at the core of its network to provide an alternate defense in the event a vulnerability in Netfilter leads to compromise of its Access or Customer and Collaboration networks.

The corporate firewall protects the Commerce, Security, Server and Workstation networks and is responsible for controlling access to each of the networks. It restricts connections from the Customer and Collaboration network back to the Commerce and Security networks. It is also responsible for regulating Internet access and access from the VPN gateway back into the network. The VPN gateway was placed behind the firewall to support future plans to allow business partners and suppliers access to GIAC internal systems. This allows GIAC's IT staff to control access by partners and suppliers. It should be noted that the firewall allows access by the dedicated Squid proxy to the Internet and the Customer & Collaboration networks for HTTP and HTTPS.

Squid Web Proxy

The Squid proxy and cache server, version 2.4, is used for employee Internet access. Employee's Web browsers are automatically configured using an auto-proxy configuration script (see Appendix B, code samples). GIAC has implemented Squid as a Web proxy to improve Web access by caching frequently accessed content. GIAC has not experienced significant problems with employee Internet abuse, but it may consider implementing content filtering for users in the future. Additional information on Squid can be found at http://www.squid-cache.org.

The proxy server runs a hardened version of Red Hat Linux 7.2. The same process and principles used to secure Linux systems on the Customer and Collaboration network were followed in securing the system. As part of supporting access to the Internet and Customer and Collaboration networks, the system is allowed to initiate connections using HTTP and HTTPS on ports 80/tcp and 443/tcp.

Windows 2000 Active Directory

The Windows 2000 Active Directory domain controller also functions as a file, print, and dynamic host configuration protocol (DHCP) server. NetShield antivirus software is installed on all Windows servers. It should be noted that all Windows 2000 workstations also run VirusScan antivirus software.

Exchange Mail and Internal DNS Server

Microsoft Exchange 2000 is GIAC's internal email, calendaring, and business collaboration solution. ScanMail for Exchange is installed on the system and provides virus defense for inbound and outbound email messages. The Exchange server is allowed to send and receive mail on port 25/tcp to and from the mail gateway in the Customer and Collaboration network.

The Exchange server also runs Windows Dynamic DNS (DDNS). Only devices on the Commerce, Security, Server, and Workstation networks use the system. It performs recursion for these clients. The server forwards lookups to the external DNS servers on the Customer and Collaboration network when it cannot provide an authoritative answer. The system connects to the external DNS servers on the Customer and Collaboration network over 53/tcp and 53/udp. This is the same access that all DNS clients are allowed for resolving queries through this system.

Backup Server

The backup server provides routine backups of the systems on the Commerce, Security, and Server networks. The systems on the Customer and Collaboration network have content that is dynamically created by other portions of the network. An image of each systems' original configuration was created in the event of a failure or security breach that requires a recovery of the system.

The server performs backups to a digital linear tape (DLT) autoloader using Legato Networker 6.2. GIAC performs full backups once per week, with incremental backups performed the other six days of the week. The backup server runs a hardened version of Red Hat Linux 7.2. The same process and principles used to secure Linux systems on the Customer and Collaboration network were followed in securing the system. Networker requires the ability to connect and receive connections for hosts on 7937/tcp and 7938/tcp. In addition, Networker has been configured to communicate over 10001/tcp through 10601/tcp as well. This is the optimal configuration for Networker based on the default port allocations allowed (Legato).

Commerce Network

Database Server

GIAC's database server, Oracle 9i, resides on a hardened Red Hat Linux 7.2 system on the Commerce network. The hardened operating system follows the same principles used to secure the Linux servers on the Customer and Collaboration network. In addition to the database application, the batch processor resides on this system. The batch processor is responsible for generating releases for strategic customers and partners. It is also responsible for transmitting the nightly order processing to Dave's Bakery for casual business customer orders. The batch processor uses secure copy (SCP) to transfer files to individual directories on GIAC's secure Web server. The batch processor authenticates using public key authentication.

The Internet and corporate firewalls are configured to allow access to the database server from the public and secure Web servers on 1521/tcp. It is important to note that only the public and secure Web servers have access to the database server for the purposes of communicating with the Oracle 9i database. This aids in isolating the database not only from the Internet, but internal employees as well.

Security Network

Authentication Server

The GIAC authentication server performs authentication for two business functions. The first is the authentication of strategic customers, suppliers, and partners to the secure Web server as outlined in Web access management section above. The second is the authentication and authorization of user access to the VPN concentrator.

The server runs a hardened version of Sun Solaris 8. The same process and principles used to secure Solaris systems on the Customer and Collaboration network were followed in securing the system. Secure Computing's SafeWord Premier Access is installed on the system. This includes the core servers such as the authentication, authorization, and accounting (AAA) server, administration server, and directory server, as well as the RADIUS server. The Web login server is allowed to connect to the server to validate authentication attempts using 5031/tcp. The VPN Concentrator also connects to the management server to perform authentication. This authentication is handled using RADIUS on 1645/udp and accounting is handled through 1646/udp.

Management Server

GIAC's management server provides several functions for the IT staff. The system runs the Firewall-1 management module, which allows for centralized management of the corporate firewall. The server accepts log messages from this firewall as well. It then exports the logs on a nightly basis to the logging server using secure copy and public key authentication.

The management server also functions as the primary NTP server and as a backup server for the configuration of the firewall enforcement points. Maintaining time synchronization across the infrastructure is essential for the purposes of incident handling. The NTP daemon synchronizes with two stratum 2 servers (128.105.39.11 and 216.27.190.202). These servers were taken from a list of NTP servers maintained at http://www.eecis.udel.edu/~mills/ntp/clock2.htm.

The Firewall-1 firewall enforcement point performs a nightly backup of its configuration to the system using the Configuration Creator script (see Appendix B, code samples). This script creates a tarball of critical configuration files, compresses the tarball, and then uploads the file to the management server using secure copy as the transport with public key authentication.

The server also runs a hardened version of Solaris 8. Each firewall is configured to allow network devices and server systems to perform NTP queries on port 113/udp to the system. The management server communicates with the corporate firewall (i.e. Firewall-1) for 257/tcp, 18191/tcp, 18192/tcp, 18210/tcp, and 18211/tcp (Checkpoint). Both firewalls also have the ability to send file transfers using secure copy over 22/tcp to the management server.

Logging Server

The logging server provides a syslog daemon to support remote logging from the host systems and IDS sensors on the GIAC network. The logging server rotates log files on a nightly basis and retains all logs for 180 days. The logging server runs Swatch version 3.0.4, which is configured to monitor log files for suspicious activity and send email alerts to the IT staff. Swatch can be downloaded from the Web at http://www.oit.ucsb.edu/~eta/swatch. The IT staff receives these alerts to their personal mailboxes on the Exchange server as well as to their pages through an email paging gateway provided by the vendor's paging service. GIAC chose to send these messages over the Internet to avoid having to connect its logging server to a modem.

The logging server also functions as the secondary NTP server in the event of a failure on the primary NTP server. The server runs a hardened version of Red Hat Linux 7.2. Each firewall is configured to allow network devices and server systems to perform NTP queries on port 113/udp to the system. All server systems are allowed to connect to the host for syslog logging using 514/udp. The management server connects to the server for the purposes of file transfers using secure copy over 22/tcp.

Intrusion Detection System

GIAC has installed network based intrusion detection on five network segments. These segments include the Customer and Collaboration, Security, Commerce, Choke, and Access networks. GIAC did not implement intrusion detection on the External network segment because doing so would generate alerts on activity that would be blocked by the Internet firewall. GIAC's philosophy in using network-based IDS is to serve as a monitoring and alert system for activity that

has successfully passed other network security defenses. Each sensor has two network interfaces, one for monitoring and one for administration and logging. GIAC had considered the use of network taps to monitor network connections but chose to leverage the Cisco switches on each network segment. The network interface designated for monitoring is set to run in promiscuous mode and listen promiscuously to traffic spanned to a switch port from the port firewall interface for the segment is connected to. The administration network interface is connected to the Security network switch and is assigned an IP address. This interface is used to send alerts to the logging system on the Security network.

The GIAC network IDS sensors run the hardened Red Hat Linux 7.2 build noted above. Each system runs Snort version 1.8.7 and the following Snort plug-ins:

- Frag2 which reassembles fragments before rule checks
- Stream4 which performs stateful inspection and can detect port scans

Additional information regarding Snort is available at http://www.snort.org.

Assignment 2 – Security Policy and Tutorial

The GIAC technical architecture implements multiple levels of defense. This includes several tiers of network security. GIAC's corporate security policy states, "Access to GIAC resources by external parties must be restricted to only those resources necessary for business services." To meet the expectations set forth by the corporate policy, GIAC has implemented access control at the following devices:

- Border router
- Internet firewall
- Corporate firewall
- VPN gateway

This section provides a detailed overview of the access control implemented within each device. A tutorial of the Internet firewall is provided as part of the implementation of the Internet firewall's policies.

Border Router Security

There are two steps in the configuration of GIAC's border router; hardening and access control lists. GIAC maintains a copy of the router's configuration on the management server on the Security network. This is performed through a cut-and-paste operation from a console session on the router to an SSH session to the management server. GIAC does not use the trivial file transfer protocol (TFTP) to back up its router configurations for 2 reasons (Sollins, p.1):

- TFTP is implemented on top of UDP, making a source address easy to spoof
- TFTP does not support authentication, giving a potential attacker an easy avenue to retrieve critical configuration information

Border Router Hardening

Following the basic principles outlined in the SANS Institutes', Perimeter Protection: Defense In-Depth course-work and Cisco Systems', Increasing Security on IP Networks, GIAC has hardened the border router.

The first step in the router's configuration is to enter privileged mode using the *enable* command. This mode provides access to enter the router configuration mode and can be distinguished by the # sign after the router name. At the prompt we enter the "config term" command to begin the router configuration:

```
router#>config term
Enter configuration commands, one per line.  End with CNTL/Z.
router(config)#
```

From the global configuration mode, we configure settings for the entire router (i.e. settings independent of an interface). The first configuration task is to assign

a password for console access. When creating a password, it is important to select a strong password that contains alphabetic, numeric, and non-alphanumeric (i.e. special) characters:

```
router(config)#line console 0
router(config-line)#login
router(config-line)#password !!ch31t3h
```

The second step is to set a password for privileged mode:

```
router(config)#enable-password $c0pl3x!
```

The third step is to set a timeout for logins on the console session. The *exec-timeout* command below sets the timeout for two minutes:

```
router(config)#line console 0
router(config-line)#exec-timeout 2
```

Once the passwords are set, it is important to encrypt the passwords. Passwords are kept in clear-text by default. The following command creates an MD5 hash of the passwords:

```
router(config)#service password-encryption
```

While not a critical host-hardening step, we can also set a hostname for the router which will make it easier to track which device we are connected to:

```
router(config)#hostname border-router
border-router(config)#
```

As a precautionary measure, GIAC allows telnet access to the router from the GIAC network. The following commands sets a password and timeout for the virtual terminal sessions:

```
border-router(config)#line vty 0 4
border-router(config-line)#login
border-router(config-line)#password !Wr@1th$
border-router(config-line)#exec-timeout 2
```

A standard access list is also implemented to prevent access to the virtual terminal ports. This access list is numbered as access list 10 using the *access-list* command:

```
border-router(config)#access-list 10 permit host 198.7.46.145
border-router(config)#line vty 0 4
border-router(config-line)#access-class 10 in
```

An access list is also applied to the auxiliary port to deny all access:

```
border-router(config)#access-list 20 deny 0.0.0.0 255.255.255.255
border-router(config)#line aux 0
border-router(config-line)#access-class 20 in
border-router(config-line)#exec-timeout 2
```

GIAC does not use the simple network management protocol (SNMP) to manage or monitor the border router. The command *no snmp* can be used to disable SNMP (SANS Institute, Perimeter Protection: Defense In-Depth, p. 54):

```
border-router(config)#no snmp
```

There are several other services available on a Cisco router that GIAC also disables. The basic services including echo, chargen, discard, and daytime are disabled:

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

The Web based configuration server is disabled:

```
border-router(config)#no ip http server
```

The bootp agent is disabled:

```
border-router(config)#no ip bootp server
```

The finger service is disabled:

```
border-router(config)#no ip finger
```

The Cisco discovery protocol (CDP) is disabled:

```
border-router(config)#no cdp enable
```

GIAC also disables support for packets with source routing using the following command:

```
border-router(config)#no ip source-route
```

The following command is applied to the Ethernet interface of the router to prevent attacks that would use the router as a Smurf amplifier:

```
border-router(config)#int FastEthernet0/0
border-router(config-if)#no ip directed-broadcast
```

The following commands are applied to the Ethernet interface of the router to prevent it from returning ICMP error messages:

```
border-router(config-if)#no ip unreachables
border-router(config-if)#no ip redirects
```

A section of GIAC's corporate security policy states, "All GIAC maintained devices must identify themselves as the property of GIAC and prohibit unauthorized access." To implement this corporate policy requirement, GIAC has applied a banner to its border router using the *banner* command:

```
border-router(config)#banner /
Enter TEXT message.  End with the character '/'.
This system is the property of GIAC Enterprises and is for the use of
authorized users only.
Individuals using this computer system without authority, or in excess
of their authority, are subject to having their activities monitored.
```

```
Unauthorized access is expressly prohibited.
/
```

The final step to hardening the router is to allow it to configure it to send log messages to the logging server on the Security network. To support logging, we configure logging and NTP on the router:

```
border-router(config)#logging 198.7.46.146
border-router(config)#logging trap debug
border-router(config)#logging console emergencies
border-router(config)#ntp server 198.7.46.146
```

Border Router Policy

There are several different kinds of access control lists available when using Cisco routers. Access control lists use a set of *permit* and *deny* commands to restrict access. The rules are applied to traffic in the order they are listed, top down (SANS Institute, Perimeter Protection: Defense In-Depth, p. 26). As a result, our rules start with more precise or specific restrictions and become more general. The following table provides a brief overview of each type of access control list:

ACL Type	Description
Standard	Filtering is accomplished based on source IP address
Extended	Filtering is accomplished based on source or destination IP address, protocol, UDP or TCP port, ICMP type, or TCP flag settings
Reflexive	Filtering is accomplished based on standard or extended types, also uses a state table to control replies
CBAC	Filtering is accomplished based on the conventions used in standard or extended types as well as by examining higher level protocol behavior

GIAC has chosen to implement extended ACLs on the border router to implement the router's security policy. The format or syntax for extended access lists is:

```
access-list <number of list> <action of permit|deny> <protocol>
<source> [wild-card] [source-port] <destination> [wild-card]
[destination-port] [options]
```

The border router has two extended access control lists enabled, one on the serial interface (ACL 110) and one on the Ethernet interface (ACL 120).

Ingress Filtering

The first step in implementing the policy is to deny IP packets with a source of a private IP address as defined by RFC 1918. This includes 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 networks:

```
border-router(config)#access-list 110 deny ip 10.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 172.16.0.0 0.15.255.255
any log
border-router(config)#access-list 110 deny ip 192.168.0.0 0.0.255.255
any log
```

Next we block reserved addresses (240.0.0.0/4) and multicast (224.0.0.0/4) traffic (IANA) that comprise the class D and E network numbers:

```
border-router(config)#access-list 110 deny ip 224.0.0.0 31.255.255.255
any log
```

We then block the other IANA reserved address spaces:

```
border-router(config)#access-list 110 deny ip 1.0.0.0 0.255.255.255 any
log
border-router(config)#access-list 110 deny ip 2.0.0.0 0.255.255.255 any
log
border-router(config)#access-list 110 deny ip 5.0.0.0 0.255.255.255 any
log
border-router(config)#access-list 110 deny ip 6.0.0.0 0.255.255.255 any
log
border-router(config)#access-list 110 deny ip 23.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 27.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 31.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 36.0.0.0 1.255.255.255
any log
border-router(config)#access-list 110 deny ip 39.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 41.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 42.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 49.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 50.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 58.0.0.0 1.255.255.255
any log
border-router(config)#access-list 110 deny ip 60.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 70.0.0.0 1.255.255.255
any log
border-router(config)#access-list 110 deny ip 72.0.0.0 7.255.255.255
any log
border-router(config)#access-list 110 deny ip 82.0.0.0 1.255.255.255
any log
border-router(config)#access-list 110 deny ip 84.0.0.0 3.255.255.255
any log
border-router(config)#access-list 110 deny ip 88.0.0.0 7.255.255.255
any log
border-router(config)#access-list 110 deny ip 96.0.0.0 31.255.255.255
any log
```

```
border-router(config)#access-list 110 deny ip 127.0.0.0 0.255.255.255
any log
border-router(config)#access-list 110 deny ip 197.0.0.0 0.255.255.255
any log
```

Next, we configure the router to block traffic from the auto-configuration client addresses and packets from the invalid host 0.0.0.0 (Winters, p.8):

```
border-router(config)#access-list 110 deny ip 169.254.0.0 0.0.255.255
any log
border-router(config)#access-list 110 deny ip host 0.0.0.0 any log
```

GIAC's public IP address space should not arrive inbound on the serial interface. The following rule will prevent spoofed packets with a source address of GIAC's public IP address space from passing onto the External network:

```
border-router(config)#access-list 110 deny ip 198.7.46.128 0.0.0.127
any log
```

The router is now ready for the ACLs that allow access to services provided by the GIAC network. The first rules allow inbound HTTP and HTTPS to the public Web, secure Web, and Web login servers. These rules are added near the top in an attempt to optimize the ACL for frequently matched traffic:

```
border-router(config)#access-list 110 permit tcp any host 198.7.46.203
eq 80 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.203
eq 443 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.204
eq 80 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.204
eq 443 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.205
eq 80 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.205
eq 443 log
```

These rules allow access, via DNS, to the DNS servers on the Customer and Commerce network. As UDP is stateless, it is necessary to add rules to allow responses to DNS queries from the external DNS servers:

```
border-router(config)#access-list 110 permit udp any host 198.7.46.201
eq 53 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.201
eq 53 log
border-router(config)#access-list 110 permit udp any eq 53 host
198.7.46.201 gt 1023 log
border-router(config)#access-list 110 permit udp any host 198.7.46.202
eq 53 log
border-router(config)#access-list 110 permit tcp any host 198.7.46.202
eq 53 log
border-router(config)#access-list 110 permit udp any eq 53 host
198.7.46.202 gt 1023 log
```

This rule allows access, via SMTP, to the mail server on the Customer and Commerce network:

```
border-router(config)#access-list 110 permit tcp any host 198.7.46.200
eq 25 log
```

These rules allow mobile clients to connect to the VPN gateway. As ESP and AH are implemented directly on top of IP, we will implement access rules that allow ESP (type 50) traffic (JNT Association):

```
border-router(config)#access-list 110 permit esp any host 198.7.46.135
log
border-router(config)#access-list 110 permit udp any eq 500 host
198.7.46.135 eq 500 log
border-router(config)#access-list 110 permit udp any eq 10000 host
198.7.46.135 eq 10000 log
```

Note: The Cisco VPN client requires 10,000/udp as part of its communication with the VPN concentrator. This traffic has the same source and destination port.

To support established sessions for devices on the GIAC network, it is necessary to allow responses back into the network.

```
border-router(config)#access-list 110 permit tcp any 198.7.46.128
0.0.0.127 established log
```

Finally, the router will filter ICMP echo-requests and drop all other traffic:

```
border-router(config)#access-list 110 deny icmp any 198.7.46.128
0.0.0.127 echo-request log
border-router(config)#access-list 110 deny ip any any log
```

The ingress ACL is then applied to the serial or external interface for the inbound direction:

```
border-router(config)#int serial0
border-router(config-if)#ip access-group 110 in
```

Egress Filtering

The GIAC border router also implements egress filtering to loosely restrict the traffic headed out to the Internet. The first rules apply to egress filtering block any traffic with a source IP address within private IP address space:

```
border-router(config)#access-list 120 deny ip 10.0.0.0 0.255.255.255
any log
border-router(config)#access-list 120 deny ip 172.16.0.0 0.15.255.255
any log
border-router(config)#access-list 120 deny ip 192.168.0.0 0.0.255.255
any log
```

The router is then configured to block outbound echo-replies and ICMP time-exceeded messages to minimize the success of denial of service attacks, which use ICMP:

```
border-router(config)#access-list 120 deny icmp any any echo-reply log
border-router(config)#access-list 120 deny icmp any any time-exceeded
log
```

Finally the router is configured to allow GIAC's public IP address space to initiate connections to the Internet and to drop all other IP traffic:

```
border-router(config)#access-list 120 permit ip 198.7.46.128 0.0.0.127
any log
border-router(config)#access-list 120 deny ip any any log
```

The egress ACL is then applied to the serial or external interface for the inbound direction:

```
border-router(config)#int FastEthernet0/0
border-router(config-if)#ip access-group 120 in
```

Firewall Security

GIAC's network security is further enforced through the placement of two tiers of firewalls. The first tier, the Internet firewall, is designed to regulate access from the Internet to GIAC's public services and e-commerce applications. The second tier, the corporate firewall, is designed to control access between GIAC's business networks and to protect the back-end processing within the e-commerce applications. GIAC uses two different technologies for firewalls. This reinforces the Defense in Depth philosophy. It is important not only to maintain multiple tiers or layers of defense, but also to use differing solutions. This enhances security because the organization has not placed the strength of its design in one vendor. In the event a vulnerability in one solution is discovered, the use of multiple solutions can assist in preventing compromise of all of the layers of protection implemented.

Internet Firewall Security and Tutorial

Like the border router, there are two steps in configuring GIAC's Internet firewall. The first, system hardening, is absolutely crucial to the successful implementation of the firewall. If the firewall's operating system is not adequately secured from intrusion, a compromise of the operating system could result in the compromise of the firewall application and the networks protected by the firewall. The second step is the configuration of firewall rules to enforce the security policy. Due to the importance of securing this system and implementing a solid and effective policy, a tutorial of the implementation of the firewall is provided as part of the discussion of the firewall.

Basic Internet Firewall Configuration and Hardening

The first step in building the Internet firewall is the installation of the base Red Hat 7.2 operating system from commercially produced CD-ROMs. During the installation of the OS, we choose a custom install and specify only the base packages necessary to run a firewall (see Appendix B, OS hardening). After the OS is installed, the first step to securing the system is to perform any necessary patching on the system. To patch the OS, we will use Red Hat's up2date application. This application allows us to download the latest patches from Red Hat, verify the validity of the patches via a GPG public key, and then automate the installation of the patches.

To run up2date, we must first configure the system preferences for the Red Hat Network. At the prompt we enter the "rhn_register" command to define these preferences:

```
[root@inetfw01 /]# rhn_register
```

This starts the text-based interface for configuration. After reading through the initial information screens, simply highlight the "Next" button (you can use the tab to move from field to field) and hit "enter" to continue. This brings us to step 2 in the process where we will create an account for the Red Hat Network:

```
---------- Step 2: Register a User Account ----------------------
|     Are you already registered with redhat.com?                |
|     Yes: Enter your current user name and password below.      |
|     No: Choose a new user and and password and enter below     |
|     User name: giacenterprises_____                           |
|     Password: ***********_________                            |
|     Again, for verification: ***********_________             |
|     E-mail address: hostmaster@giacfortunes.com_______        |
|          Next        Back        Cancel                       |
-----------------------------------------------------------------
```

We are then asked for optional information including name, company, and address. As this information is not necessary, do not fill any of the fields in. Simply highlight the "Next" button and hit "enter" to continue. We are then asked if you want to include the hardware and network configuration information in your profile. Uncheck the "Include the following…" box, highlight the "Next" button, and hit "enter" to continue. The next screen provides you with the option of including the RPM packages installed on the system in your profile. We do not want this information available to the Red Hat Network, so uncheck the "Include RPM packages…" box and then continue on to the next screen. We are then asked to send your profile to the Red Hat Network by highlighting the "Next" button. The system then creates a profile.

At this point we want to install the Red Hat Network GPG key so that we can verify the packages we download. At the command prompt we enter the following command to import the key into our key ring:

```
[root@inetfw01 root]# /usr/bin/gpg --import /usr/share/rhn/RPM-GPG-KEY
gpg: key DB42A60E: public key imported
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: Total number processed: 1
gpg:               imported: 1
```

We can now run up2date to download and install the latest packages available for the system. The –u flag tells up2date to update the system with all relevant packages, while the –v flag provides detailed output of the activity:

```
[root@inetfw01 root]# up2date -uv
Retrieving list of all available packages...
########################################
Removing installed packages from list of updates...
########################################
Removing packages marked to skip from list...
```

```
########################################
Getting headers for available packages...
########################################
Testing package set / solving RPM inter-dependencies...
########################################
Retrieving selected packages...
<list of packages omitted>
```

Once up2date has completed the patch process, the system should be rebooted to verify that the new configuration functions correctly. Once the system has rebooted, we can then begin the process of installing and running Bastille to harden the system. Bastille disables unnecessary services, corrects permissions on files, and performs a variety of other system enhancements that reduce the vulnerability of the operating system. To install Bastille and Perl-Curses we simply run the following commands in the directory we have copied the RPMs to:

```
[root@inetfw01 root]# rpm -ivh Bastille-2.x.y-z.a.i386.rpm
[root@inetfw01 root]# rpm -ivh perl-Curses-d.e-f.i386.rpm
```

We have two options for running Bastille, interactive or automated modes. Since we have a configuration file for Bastille, we run it in automated mode. We simply place the contents of our configuration into the file /etc/Bastille/config. We then run the command:

```
[root@inetfw01 /]# /usr/sbin/AutomatedBastille
Bastille is now locking down your system in accordance with your
answers in the "config" file.  Please be patient as some modules
may take a number of minutes, depending on the speed of your machine.
```

Once Bastille has completed the hardening process, the system should be rebooted to verify that the new configuration functions correctly. At this point we conduct a quick scan of the system using Nmap from the external (i.e. Internet facing) interface. The output of the Nmap scan shows that the system is only listening for SSH connections:

```
[root@buttercup /]# nmap -v -sS -O 172.16.1.254 –p '1-65535'

Starting nmap V. 2.54BETA22 (www.insecure.org/nmap/)
Host  (198.7.46.254) appears to be up ... good.
Initiating SYN Stealth Scan against  (172.16.1.254)
Adding TCP port 22 (state open).
The SYN Stealth Scan took 2 seconds to scan 65535 ports.
For OSScan assuming that port 22 is open and port 1 is closed and
neither are firewalled
Interesting ports on  (198.7.46.254):
(The 65534 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2017709 (Good luck!)
IPID Sequence Generation: Duplicated ipid (!)
Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds
```

Since root is not allowed to log in directly to the system from a remote connection, we then create an account for each system administrator. In this case we create a user named "ssajlo" with a unique userid. The user's primary group is the "users" group, with their secondary group set to the "wheel" group. We also set a password for the user:

```
[root@inetfw01 /]# useradd -u 1000 -g users -g wheel -d /home/ssajlo
-s /bin/bash -c "James OBrien" ssajlo
[root@inetfw01 /]# passwd ssajlo
```

To synchronize the system's time, we configure /etc/ntp.conf to point to the two NTP servers on the Security network:

```
server 172.16.3.10   # ntp01
server 172.16.3.11   # mgmt01
driftfile /etc/ntp/drift
```

Finally, we want to configure remote logging to the syslog server on the Security network. On the firewall, we edit /etc/syslog.conf and add the following line to the end of the file. We then need to restart syslog (note the process number used in the kill command comes from the id of the syslog process):

```
*.*                                     @172.16.3.10
[root@inetfw01 etc]# ps -ef |grep syslog
root       618     1  0 Sep06 ?        00:00:00 syslogd -m 0
[root@inetfw01 etc]# kill -HUP 618
```

We also need to start the syslogd process on the log server with the –r flag. We can make this change permanent by modifying /etc/init.d/syslog and changing the variable SYSLOGD_OPTIONS to include the –r flag.

Netfilter Tutorial

Normally, Netfilter support must be compiled into the Linux kernel. The Red Hat 7.2 distribution has support enabled in the kernel. There are three basic tables available consisting of filter, nat, and mangle. If no table is specified, the filter table is considered the default. While some of the built-in chains can belong to more than one table, user created chains can consist of chains from only one table. The filter table contains rules designed to control traffic using the packet filter. The nat table handles network address translation and the mangle table modifies other information in the IP header (Bandle, p.3). Most of the work we will perform will focus on the filter table.

The filter table has three built-in chains in addition to user-defined chains. The following table provides a brief overview of each type of chain (SANS Institute, Perimeter Protection: Firewall Technology, Netfilter and Gauntlet, p.202):

Chain Type	Description
Input	Controls traffic coming into the firewall (i.e. sent to an IP address of the firewall or a broadcast to a local network for the firewall).

Output	Controls traffic leaving the firewall.
Forward	Controls traffic attempting to pass through the firewall.
User	Customized chains that can be called by one of the built-in chains. Intended to provide modularity in rule base management.

The format or syntax for iptables is fairly complex as shown by the iptables –h command. We will cover the relevant options for each of these flags as we progress through the policy configuration. The first flag for the iptables command is the –t flag and is used to specify the table. The next part of the command specifies the chain, by name. The rule-specification comes next and is the actual pattern to match against the IP or ICMP header. The –j flag then indicates the target, which can be followed by options for the target:

```
iptables –t <table> -[options] <chain-name> <rule-specification> –j
<target> [options]

Usage: iptables -[ADC] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
        -A append to chain
        -D delete from chain
        -C check this packet against chain
        -R replace rule X in chain
        -I insert rule at point X in chain
        -L list the rules in a chain or all chains
        -F delete all rules in a chain or all chains
        -Z reset counters on a chain
        -N create a user defined chain
        -X delete a user defined chain
        -E rename a chain
        -P change policy on chain to a new target
```

To list the current rules installed in a given chain, we can use the command "iptables –L <chain> -nv" to display a chain. If no chain is specified, all of the chains will be listed. The –n flag tells iptables to display numeric output, while the –v flag prints verbose ouput.

```
iptables –L
```

The following displays the output of the command:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere            LOG level
warning tcp-options
ACCEPT     tcp  --  172.16.0.0/16        172.16.1.254        state NEW
tcp spts:1024:65535 dpt:ssh
```

```
ACCEPT     all  --  anywhere             anywhere           state
RELATED,ESTABLISHED
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

To flush all of the rules from a given chain, we can use the command "iptables –F <chain>." We can then clear the chain using the command "iptables –X <chain>." If no chain is specified, all of the chains will be processed.

```
iptables -F
iptables -X
```

The following displays the firewall tables after the rules have been cleared:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

There are several options available to manage the rules in a chain. To append a rule in a chain, we use the –A flag. For example to append a rule to the INPUT chain in the filter table:

```
iptables –t filter –A INPUT remainder-of-rule
```

The rule above would create static filter. Netfilter does have the ability to maintain stateful awareness of connections. To maintain state, the rule is modified slightly. In conjunction to using the –m flag to tell Netfilter to maintain state, the "—state NEW" flag tells Netfilter to write a new state table entry when the packet is matched. The second rule then tells Netfilter to allow all established traffic related to previous matches that are now in the state table:

```
iptables –t filter –A INPUT –m state --state NEW remainder-of-rule
iptables –t filter –A INPUT –m state --state ESTABLISH,RELATED
remainder-of-rule
```

To insert a rule in a chain, we use the –I flag and specify the number of the start position in the chain. For example, to insert rule 4 in the OUTPUT chain:

```
iptables –t filter –I 4 INPUT remainder-of-rule
```

To delete a rule in a chain, we use the –D flag and specify the number of the rule to delete in the chain. For example, to delete rule 4 in the OUTPUT chain:

```
iptables –t filter –D 4 INPUT
```

We can also replace a rule in a chain using the –R flag and by specifying the number of the rule to replace in the chain. For example, to replace rule 2 in the INPUT chain:

```
iptables –t filter –R 2 INPUT remainder-of-rule
```

When adding rules to a chain, there are several options that can be specified as part of the rule's parameters.

Parameter	Description
-I	Interface the packet is received on (i.e. eth0, eth2)
-p	Protocol (i.e. tcp, udp)
--icmp-type	Optional ICMP type (number)
-s	Source IP address (address/bitmask)
--sport	Optional range for source port (start:finish)
-d	Destination IP address (address/bitmask)
--dport	Optional range for destination port (start:finish)
-j	Target action (ACCEPT, REJECT)
!	Exception or exclusion, negates arguments

There are several targets that we can set as part of a Netfilter rule in the filter table including:

Target	Description
Accept	Let the packet through
Drop	Drop the packet (i.e. do not let it through)
Queue	Pass the packet to userspace
Return	Stop traversing the chain and go back to the calling chain
Reject	Do not let the packet through and send an error back to the source
Log	Log packets that match this rule

The REJECT target supports sending different kinds of errors based on the type of traffic. This allows us to customize the error message sent back to the source address. These options are specified using the --reject-with switch and include:

- tcp-reset
- icmp-net-prohibited
- icmp-host-prohibited
- icmp-net-unreachable

- icmp-host-unreachable
- icmp-port-unreachable

The LOG target also supports several options. The LOG target does not control traffic so it can be used immediately preceding a rule that processes traffic. The log options are specified immediately after the LOG target and include:

- log-ip-options
- log-tcp-options
- log-tcp-sequence
- log-prefix "message"

The first three options can be used to regulate the level of logging from IP options up to TCP sequence numbers. The final option allows us to specify a value that can prefix each line in the log. This allows us to set a value that makes log processing and parsing based on keywords possible.

At this point we can begin to craft rules to accept, drop and reject traffic. If we wanted to allow any users to connect to a Web server at 192.168.1.1 on port 80, we could write a rule that allows connections from any high-level port on a client to port 80/tcp on 192.168.1.1 as seen below:

```
iptables –t filter –A FORWARD –m state --state NEW –p tcp –s 0/0 --
sport 1024:65535 –d 192.168.1.1 --dport 80 –j ACCEPT
iptables -t filter -A INPUT -m state --state ESTABLISH,RELATED –j
ACCEPT
```

The first rule allows connections to the server on 80/tcp and adds a new entry to the state table for this connection. The second rule tells Netfilter to check the state table for a match on additional packets to determine if the traffic is part of an existing connection.

We can also allow traffic for a specific ICMP type such as echo-request and echo reply. If we want to allow any user to ping 192.168.1.1 we can use the following rules:

```
iptables –t filter –A FORWARD –p icmp --icmp-type echo-request –s 0/0 –
d 192.168.1.1
iptables –t filter –A FORWARD –p icmp --icmp-type echo-reply –s
192.168.1.1 –d 0/0
```

A complete list of the ICMP types supported by Netfilter can be displayed using the following command:

```
iptables –p icmp –h
```

We can drop traffic using the drop target. For example, if we want to drop any traffic coming to our Web server at 192.168.1.1 on port 135-139 we could use the following rules:

```
iptables –t filter –A FORWARD –p tcp –s 0/0 –d 192.168.1.1 --dport
135:139 –j DROP
```

```
iptables –t filter –A FORWARD –p udp –s 0/0 –d 192.168.1.1 --dport
135:139 –j DROP
```

Notice that we have written one rule for TCP and one for UDP traffic to these ports. We could use the REJECT target instead, which would send a TCP reset for TCP connections and an ICMP port unreachable for UDP traffic:

```
iptables –t filter –A FORWARD –p tcp –s 0/0 –d 192.168.1.1 --dport
135:139 –j REJECT --reject-with tcp-reset
iptables –t filter –A FORWARD –p udp –s 0/0 –d 192.168.1.1 --dport
135:139 –j REJECT --reject-with icmp-port-unreachable
```

Finally, we can use the LOG target independent of the other target options to log specific types of traffic. If we wanted to log all traffic to the Web server at 192.168.1.1 we could use the following rule:

```
iptables –t filter –A FORWARD –p tcp –s 0/0 –d 192.168.1.1 –j LOG
--log-tcp-options
```

We could specify a prefix for each log message that will make it easier to identify traffic that matches this pattern:

```
iptables –t filter –A FORWARD –p tcp –s 0/0 –d 192.168.1.1 –j LOG
--log-prefix "web server"
```

The nat table has three built-in chains as well. The following table provides a brief overview of each type of chain (Bandle, p.2):

Chain Type	Description
PREROUTING	Translation is handled before routing of the packet.
POSTROUTING	Translation is handled after routing of the packet.
OUTPUT	Translation occurs as packet leaves interface.

Both masquerading and source NAT use postrouting while destination NAT uses prerouting. The first type of NAT used by Netfilter is commonly referred to as masquerading and is the equivalent of "hide NAT" in other products. This allows us to hide traffic from multiple internal systems behind one IP address. For example, if we want to masquerade all traffic leaving Ethernet interface 0 (i.e. eth0), we can use the following rule:

```
iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE
```

We can also user source NAT (SNAT) to change the source address of a packet passing through the firewall. This allows us to map one internal IP address to one external IP address as the source or to map multiple internal IP addresses to one external IP address as the source. The following rule would allow us to map all traffic to a source of 198.7.46.135:

```
iptables –t nat –A POSTROUTING –o eth0 –j SNAT --to 198.7.46.135
```

Finally, we can use destination NAT (DNAT) to change the destination address of a packet passing through the firewall. This allows us to map an internal server to a public IP address and make it available over the Internet. The following rule would allow us to map the internal Web server at 192.168.1.1 to 198.7.46.204:

```
iptables -t nat -A PREOUTING -i eth0 -d 198.7.46.204 -j DNAT --to
192.168.1.1
```

Internet Firewall Policy

Now that we understand the basics of developing firewall rules within Netfilter, we can begin writing the policy for the GIAC Internet firewall. For the purposes of automating the installation of the firewall policy, we create a startup script that will be run at the system's boot time called /etc/rc.d/rc.iptables. This script calls a rules file called /etc/rc.d/iptables.rules.

```
#!/bin/sh
#
## History:
#       James O'Brien Aug 20 08:03:54 CST 2002
#               Initial version.
#
# this script processes the firewall rules for iptables
#
# Set each variable
#
SCRIPTNAME="IPTables Firewall"
HOSTNAME=`hostname`
DIR=/etc/rc.d
RULES="${DIR}/iptables.rules"
IPTABLES=/sbin/iptables
#
# Load Environment Variables
. /etc/init.d/functions
#
case "$1" in
start|restart)
      echo "Flushing current rules: "
      $IPTABLES -F
      echo "Clearing current rules: "
      $IPTABLES -X
      if [ -f "$RULES" ]; then
            . $RULES
      else
           echo "No rules available."
      fi
      # turn on IP forwarding
      echo 1 > /proc/sys/net/ipv4/ip_forward
      ;;
stop)
      # turn off IP forwarding
      echo 0 > /proc/sys/net/ipv4/ip_forward
      echo "Flushing current rules: "
      $IPTABLES -F
      echo "Clearing current rules: "
      $IPTBLES -X
```

```
      ;;
status)
      iptables -L
      ;;
*)
      echo "Usage: $0 start|stop|restart|status }"
      exit 1
esac
exit 0
```

The script starts by flushing and clearing all of the chains in each table using the "iptables –F" and "iptables –X" commands.

Input Chain

We then begin the configuration of the firewall policy by configuring the INPUT chain in the filter table. To modify the rules in the filter table we use the –t flag with filter set as the option. We can then specify the –A flag to append a rule to a specific chain.

The first rule checks packets to determine if they are a response to a connection initiated out from the firewall. This rule checks to see if the packet is part of any established connection (i.e. ACK and FIN packets) and matches an existing entry in the state table. The third, fourth, and fifth rules actually control the incoming SSH connections from the internal network space (reserved for IT staff) to the firewall interface on the Access network. These rules actually tell Netfilter to add the connections to the state table:

```
iptables -t filter -A INPUT -m state --state ESTABLISH,RELATED -j  LOG
--log-ip-options
iptables -t filter -A INPUT -m state --state ESTABLISH,RELATED -j
ACCEPT
iptables -t filter -A INPUT -m state --state NEW -p tcp -s
172.16.3.11/32 -d 172.16.1.254 --dport 22 -j LOG --log-ip-options
iptables -t filter -A INPUT -m state --state NEW -p tcp -s
172.16.3.11/32 -d 172.16.1.254 --dport 22 -j ACCEPT
```

The final rule in the INPUT chain is used to drop any other traffic coming into the firewall:

```
iptables -t filter -A INPUT -j LOG --log-prefix "INPUT-DROP-"
iptables -t filter -A INPUT -j DROP
```

Output Chain

Once the base INPUT chain has been established, we can then move onto the OUTPUT chain. The firewall is told to check the state table for potential matches when processing traffic using the previously mentioned rule that checks for established connections:

```
iptables -t filter -A OUTPUT -m state --state ESTABLISH,RELATED -j LOG
--log-ip-options
iptables -t filter -A OUTPUT -m state --state ESTABLISH,RELATED -j
ACCEPT
```

The firewall needs to communicate with several servers using SSH, SMTP, DNS, and NTP. The next rule is designed to allow the firewall to initiate SSH connections out to the management server (172.16.3.11):

```
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
172.16.3.11/32 --dport 22 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
172.16.3.11/32 --dport 22 -j ACCEPT
```

The firewall occasionally needs to send email to members of the IT staff for the purposes of nightly processing results (i.e. cron) and alert messages. The firewall uses the mail server (198.7.46.200) on the Customer and Collaboration network as a relay:

```
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.200/32 --dport 25 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.200/32 --dport 25 -j ACCEPT
```

The firewall also needs to be able to resolve hostnames using DNS. Again, the firewall is only allowed to communicate with the servers (198.7.46.201 and 198.7.46.202) on the Customer and Collaboration network. In this case, the firewall is allowed to perform UDP and TCP DNS queries:

```
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.201/32 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.201/32 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.202/32 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.202/32 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.201/32 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.201/32 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.202/32 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.202/32 --dport 53 -j ACCEPT
```

To maintain synchronization of the system clock, the firewall is also allowed to communicate with the NTP servers (172.16.3.10 and 172.16.3.11) on the Security network:

```
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.10
--dport 123 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.10
--dport 123 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.11
--dport 123 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.11
--dport 123 -j ACCEPT
```

The final rule in the OUTPUT chain is used to drop any other traffic leaving the firewall:

```
iptables -t filter -A OUTPUT -j LOG --log-prefix "OUTPUT-DROP-"
iptables -t filter -A OUTPUT -j DROP
```

Forward Chain

The FORWARD chain is the most complex chain in the set. This chain handles all traffic that must pass through the firewall from one network to another. We start the configuration of the firewall rules with a set of rules designed to log and drop specific types of network attacks (Briddle, p.122-124). The first rules log and drop packets with both the SYN and FIN packets set, commonly known as a SYN-FIN port scan:

```
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j LOG --log-prefix
"SYNFINSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j DROP
```

These rules log and drop packets with the FIN set, commonly known as a FIN port scan:

```
iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j LOG --log-prefix
"FINSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j DROP
```

These rules log and drop TCP packets with none of the flags set, commonly known as a NULL port scan:

```
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j LOG --log-prefix
"NULLSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP
```

These rules log and drop packets with the FIN, PSH, and URG flags set, usually created by nmap's XMAS tree port scan option:

```
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j LOG --log-
prefix "NMAPXMAS-"
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP
```

Finally, these rules log and drop ICMP fragments:

```
iptables -A FORWARD -p icmp -f -j LOG --log-prefix "ICMPFRAG-"
iptables -A FORWARD -p icmp -f -j DROP
```

The next set of rules we add allow the public to access resources on the Customer and Collaboration network. The firewall is told to check the state table for potential matches when processing traffic using the previously mentioned rule that checks for established connections:

```
iptables -t filter -A FORWARD -m state --state ESTABLISH,RELATED -j LOG
--log-tcp-options
iptables -t filter -A FORWARD -m state --state ESTABLISH,RELATED -j
ACCEPT
```

We need to allow everyone access to the GIAC mail server at 198.7.46.200. The following rules allow anyone not on a GIAC internal network to connect to the mail server (we'll develop rules for the internal network later):

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
```

We then allow the mail server on the Customer and Collaboration network to send email to the Internet

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d ! 172.16.0.0/16 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d ! 172.16.0.0/16 --dport 25 -j ACCEPT
```

Next, we allow DNS queries to the DNS servers at 198.7.46.201 and 198.7.46.202:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j ACCEPT
```

We also add rules that allow the external DNS servers to communicate with Internet DNS servers:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
```

Finally, we need to allow HTTP and HTTP connections to the Web servers at 198.7.46.203-198.7.46.205:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 443 -j ACCEPT
```

Several of the servers on the Customer and Collaboration network must communicate with servers on the other GIAC networks as well. We add rules that allow the internal DNS server at 172.16.9.11 to communicate with the external DNS servers at 198.7.46.201 and 198.7.46.202:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j ACCEPT
```

Next we create rules that allow the internal Exchange server at 172.16.9.11 to send and receive email from the mail server on the Customer and Collaboration network at 198.7.46.200:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
```

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.200 --dport 25 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d 172.16.9.11 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d 172.16.9.11 --dport 25 -j ACCEPT
```

We also allow the Web Login Server at 198.7.46.205 to connect to the SafeWord authentication server on the Security network at 172.16.3.12 via 5031/tcp:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.205 -d 172.16.3.12 --dport 5031 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.205 -d 172.16.3.12 --dport 5031 -j ACCEPT
```

The Secure Web server needs to be able to connect to the Oracle database on the Commerce network. To allow this access, we add a set of rules that allows connections to the Oracle database:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.204 -d 172.16.2.10 --dport 1521 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.204 -d 172.16.2.10 --dport 1521 -j ACCEPT
```

The batch processor on the database server also needs to communicate with the Secure Web server via SSH:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.2.10 -d 198.7.46.204 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.2.10 -d 198.7.46.204 --dport 22 -j LOG --log-tcp-options
```

As each of the servers on the Customer and Collaboration network will need to send logging messages to the syslog server on the Security network at 172.16.3.10, we will allow access for syslog back to this system:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.10 --dport 514 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.10 --dport 514 -j ACCEPT
```

Finally, we allow the servers on the Customer and Collaboration network to synchronize their time with the NTP servers on the Security network:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.12 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.12 --dport 123 -j ACCEPT
```

The GIAC border router has been configured to log to the syslog server on the Security network. To support this function, GIAC has configured the Internet

firewall to allow syslog traffic from the border router to the public IP address of the syslog server (198.7.46.146):

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.129 -d 198.7.46.146 --dport 514 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.129 -d 198.7.46.146 --dport 514 -j ACCEPT
```

GIAC allows employees to access Web sites via HTTP and HTTPS. At this point in time, employees have not requested access to any Web site that is present on a non-standard port. GIAC plans to review such requests and allow access as appropriate, but it will not open this access until requested by employees. The following rules allow the Squid proxy server to access Web sites on the Internet and on the Customer and Collaboration Network. As the Squid proxy server logs access, GIAC has decided to reduce the impact of logging by not logging this traffic at the firewall:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.203 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.203 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.204 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.204 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.205 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.205 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d ! 198.7.46.192/26 --dport 80 -j
ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d ! 198.7.46.192/26 --dport 443 -j
ACCEPT
```

The NTP servers on the Security network must be able to communicate with the NTP servers on the Internet to synchronize time:

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 128.105.39.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 128.105.39.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 128.105.39.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 128.105.39.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 216.27.190.202 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 216.27.190.202 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 216.27.190.202 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 216.27.190.202 --dport 123 -j ACCEPT
```

We then allow the batch processor on the Commerce network to send data to Dave's Bakery using SCP:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.2.10 -d 201.210.1.32 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 201.210.1.32 --dport 22 -j ACCEPT
```

Finally, telnet and SSH are allowed to the border router and SSH is allowed to the Customer and Collaboration Network:

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 22 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 23 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 23 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.192/26 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.192/26 --dport 22 -j ACCEPT
```

The firewall is then configured to log and reject ident queries and log and drop any other traffic attempting to pass through the system:

```
iptables -t filter -A FORWARD -j LOG --log-prefix "FORWARD-DROP-"
iptables -t filter -A FORWARD -s 0/0 -d 0/0 --dport 113 -j reject --
reject-with tcp-reset
iptables -t filter -A FORWARD -j DROP
```

Network Address Translation

The firewall will also implement several NAT rules within the nat table. GIAC uses a combination of source and destination NAT. The first NAT rules handle the static mapping of the syslog server's private IP address to its public IP address for communication with the border router:

```
iptables -t nat -A POSTROUTING -s 172.16.3.10 -d 198.7.46.129 -j SNAT -
-to 198.7.46.146
iptables -t nat -A PREROUTING -s 198.7.46.129 -d 198.7.46.146 -j DNAT -
-to 172.16.3.10
```

The final NAT rules cover any connections from the GIAC private address space to the Internet. The first rule is designed to prevent traffic from the Internet firewall destined for the internal network from being translated to the external IP address:

```
iptables -t nat -A POSTROUTING -s 172.16.1.254 -d 172.16.0.0/16 -j SNAT
--to 172.16.1.254
iptables -t nat -A POSTROUTING -s 172.16.0.0/16 -d ! 198.7.46.192/26 -j
SNAT --to 198.7.46.145
```

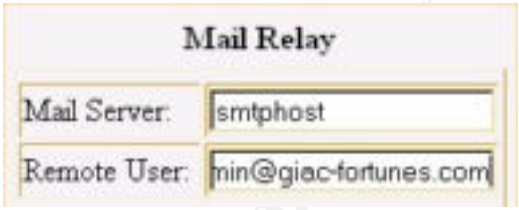
A completed copy of /etc/rc.d/iptables.rules can be found in Appendix B, Code Samples.

Corporate Firewall Security

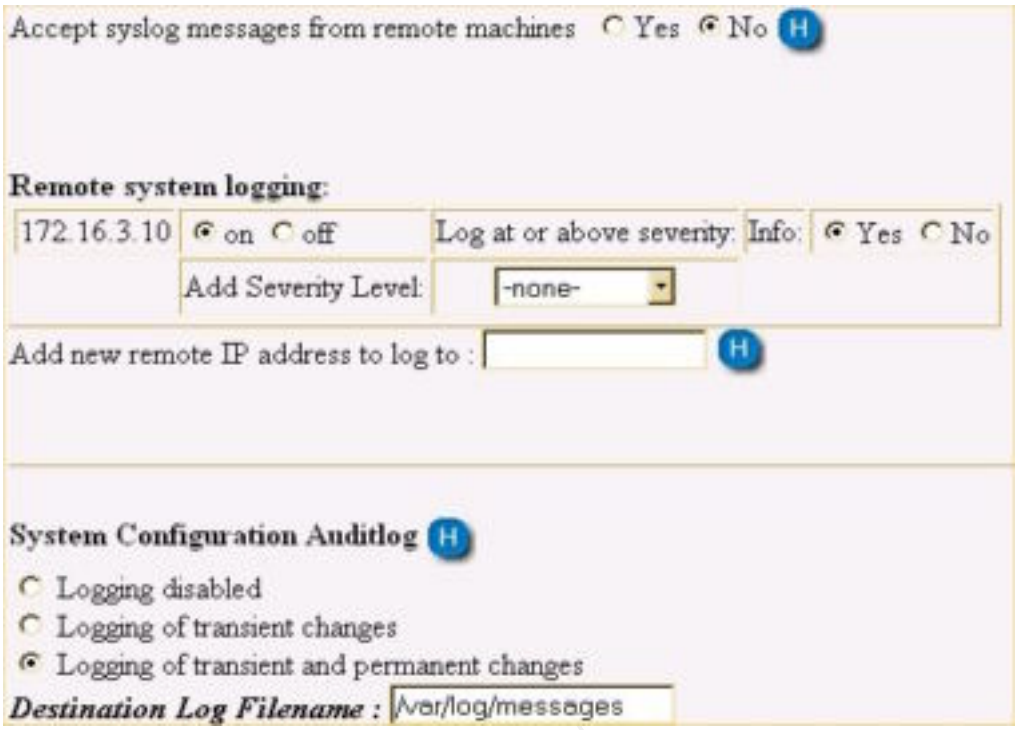
There are two steps in configuring GIAC's corporate firewall. The first is to harden the operating system. The use of the Nokia appliance greatly eases the hardening of the operating system. The second step is the configuration of firewall rules to enforce the security policy.

Basic Corporate Firewall Configuration and Hardening

The first step in setting up the corporate firewall is the modification of the default configuration of the Nokia IP530 appliance. We begin by configuring the system to send email alerts and system messages. This is done by setting the mail relay under Config → Mail Relay and by setting the alerts to a shared email mailbox under Config → System Failure Notification Configuration:

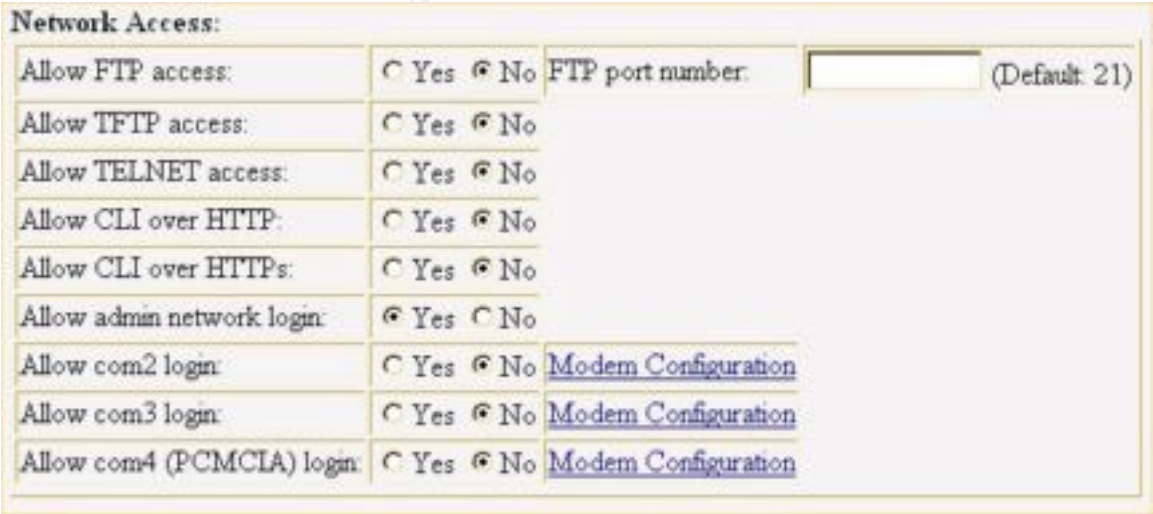


To configure the system to send logging events to the remote syslog server we configure the IP address and severity level under Config → System Logging Configuration. We also tell the system to audit any configuration changes, regardless of whether they are temporary or permanent:

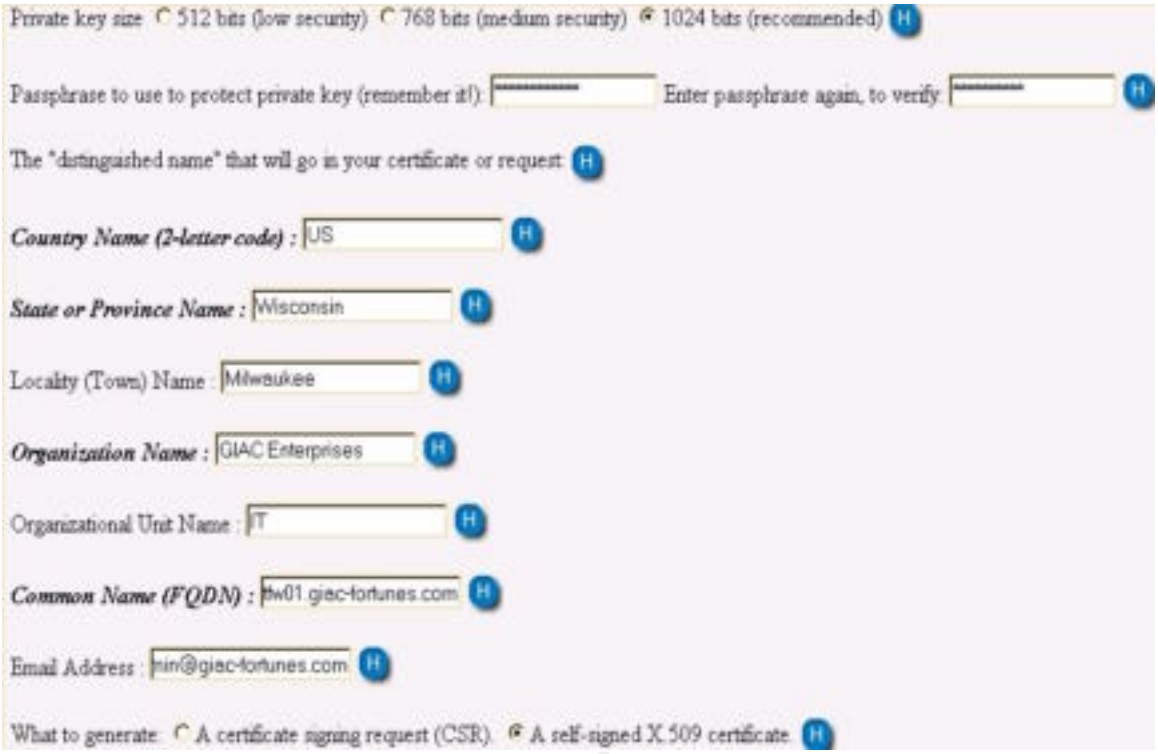


By default, the IP530 comes configured with HTTP, Telnet, and FTP running. We disable these services and use only SSH and HTTPS for administration of the operating system.

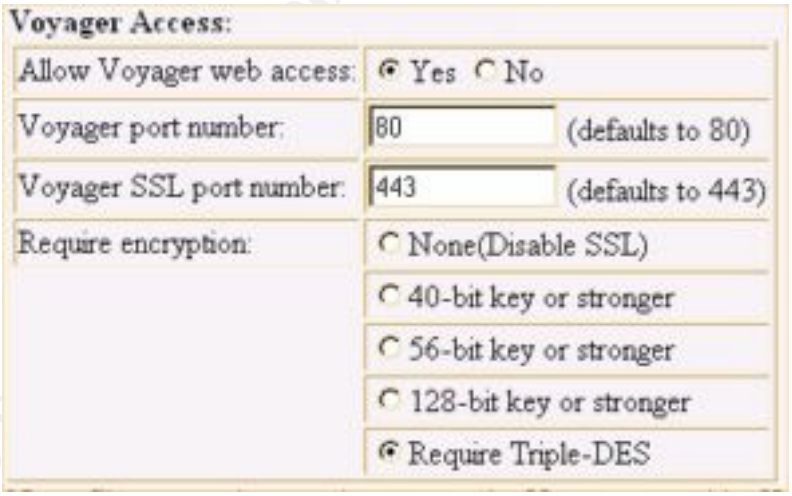
To disable Telnet and FTP, as well as other insecure services, we use the Voyager interface. These settings are access under Config → Network Access and Services (don't forget to click the apply and then the save buttons to make the changes permanent):



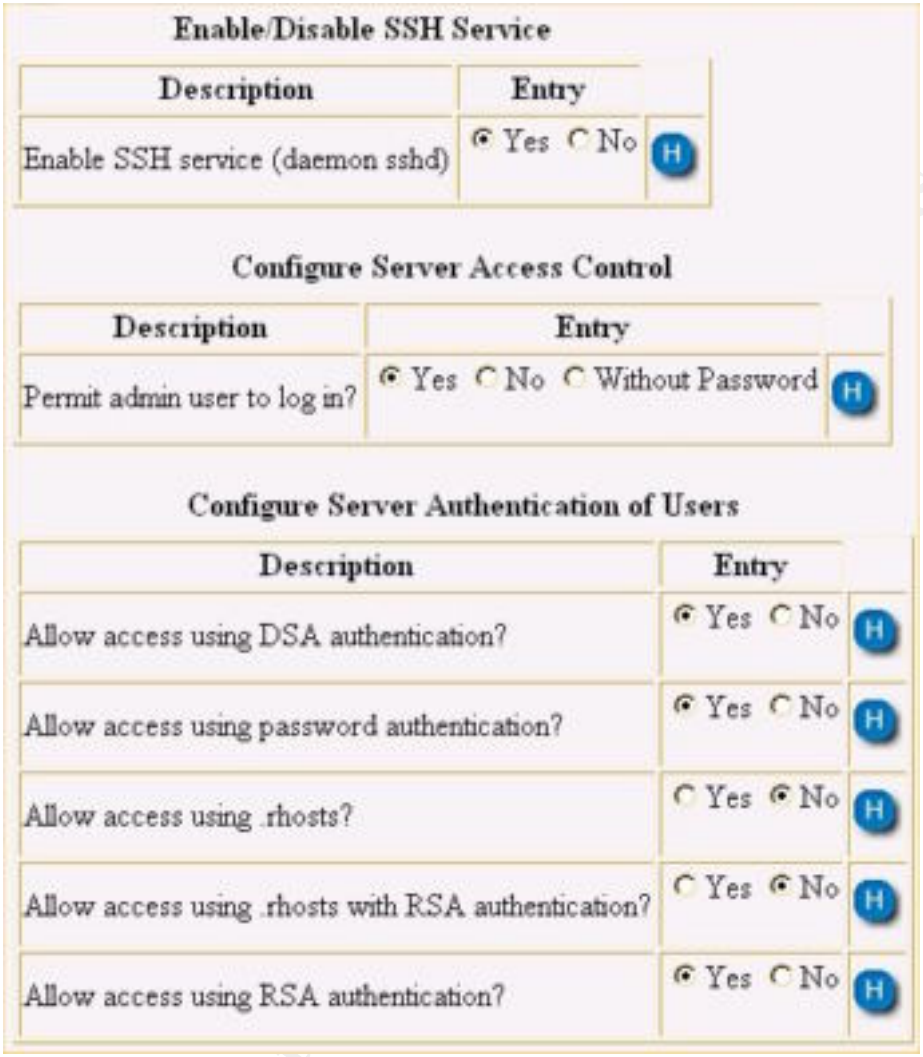
Next, we generate a self-signed X.509 certificate for use with the SSL Web server. Certificate generation is performed under Config → SSL Certificate Tool (Request):



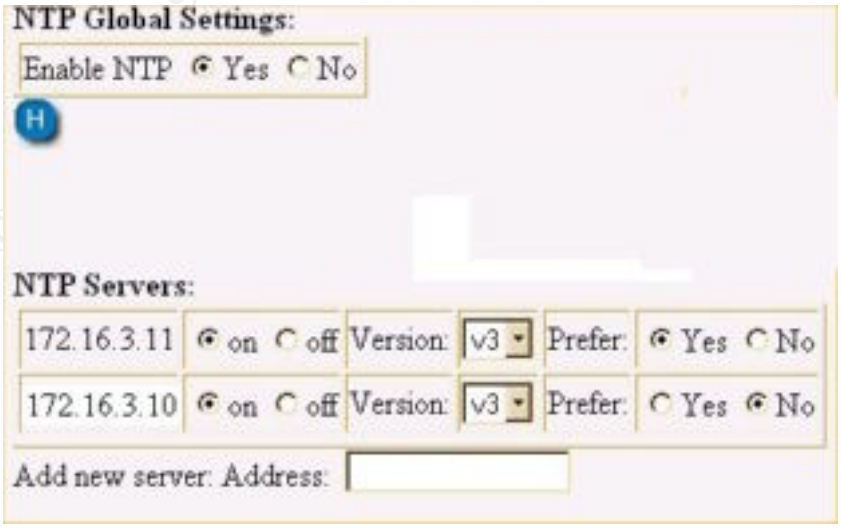
Once a certificate has been created, we can enable HTTPS. To support remote administration over an encrypted session, we turn on HTTPS and turn off HTTP. HTTPS access to the Web based interface is performed under Config → Voyager Web Access:



To support secure remote shell access, we turn on SSH. SSH access configuration is performed under Config → SSH Configuration:



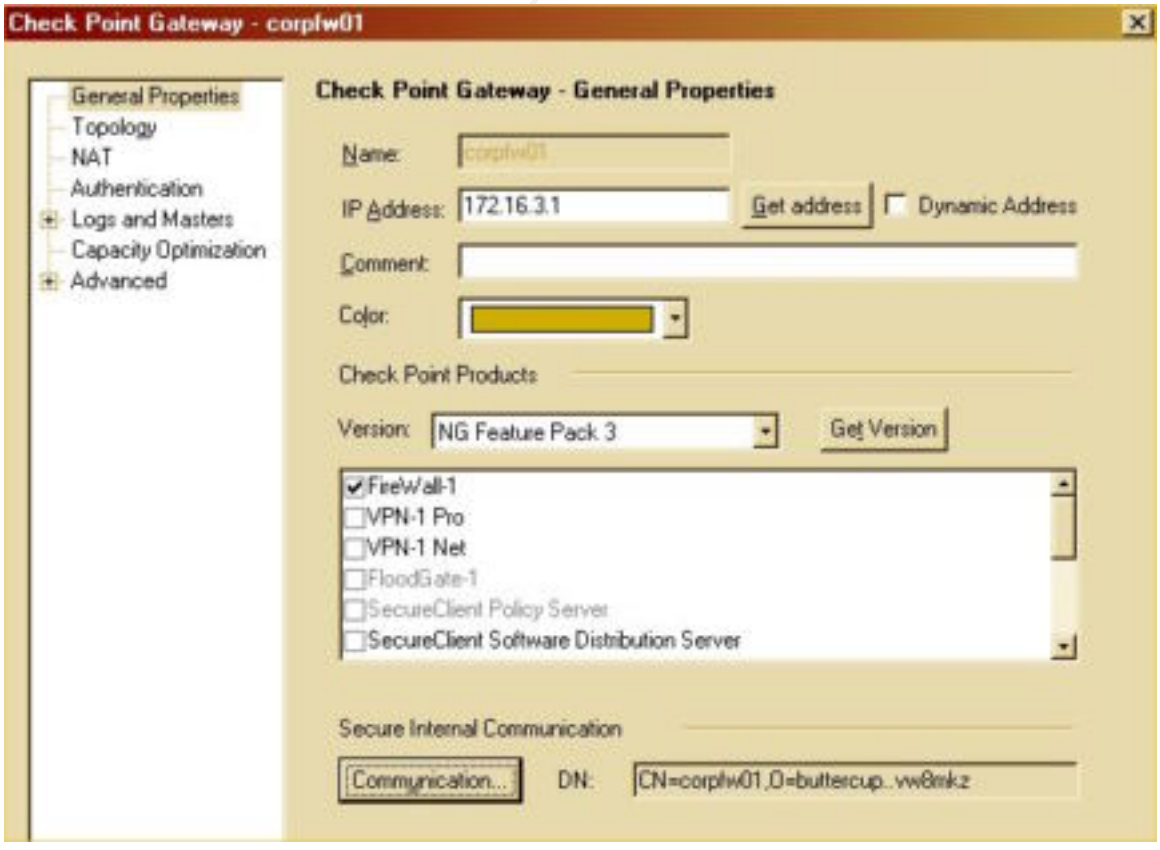
Finally, we configure the system to update its time from the two NTP servers on the Security network. This configuration is performed under Config → NTP:



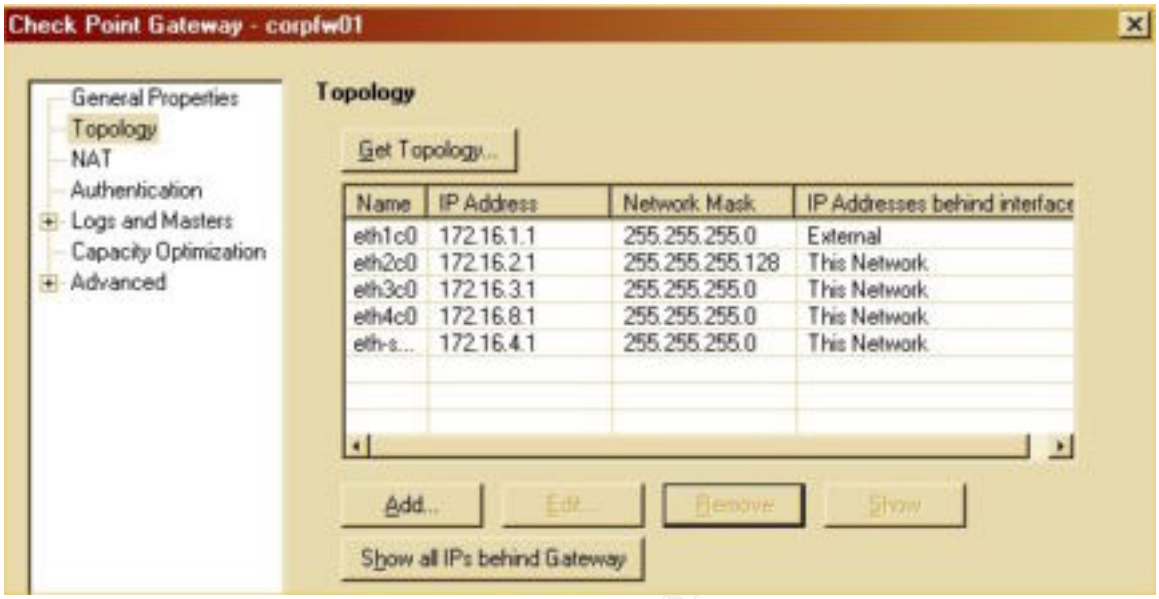
Corporate Firewall Policy

Once the firewall appliance's Operating System has been configured, Check Point Firewall-1 NG Feature Pack 3 is installed on the system. The application is configured to communicate with the management server on the security network at 172.16.3.11. We begin the creation of the firewall policy using the Check Point Smart Dashboard. This client replaces the Policy Editor as of Feature Pack 3. Firewall-1 uses objects to represent hosts, gateways, networks, services and other items within its rulesets. The use of objects is a careful balancing act. One must be careful in the use of objects in the ruleset. Without the appropriate precautions, an object used for one rule can very easily be used for another rule in a manner that is too permissive. For example, if we wanted to allow the IP pools for the VPN and the IP addresses for workstations access to the secure Web server via SSH we might create a group that contains these objects. Now if we reuse this group in a rule that allows access to the database server via SSH, we've probably allowed more individuals access to the server than is necessary. In general, objects should not be reused if by doing so; the security posture of the organization will be weakened.

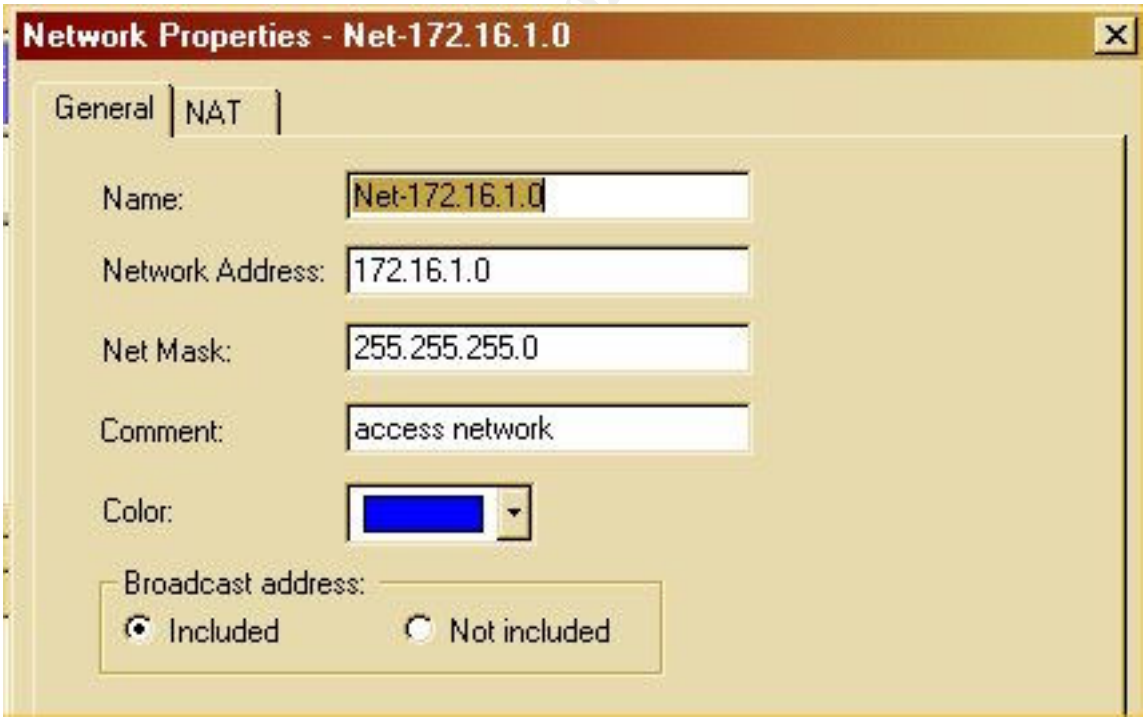
With this in mind, we start by adding an object for the firewall. To create the object we select Manage → Network Objects → New → Checkpoint → Gateway. We then fill in the information on the general properties page including the host name, IP address, and the communication activation key. The activation key replaces the older keys used by the 4.1 products. This key allows the management server to communicate with the firewall. :



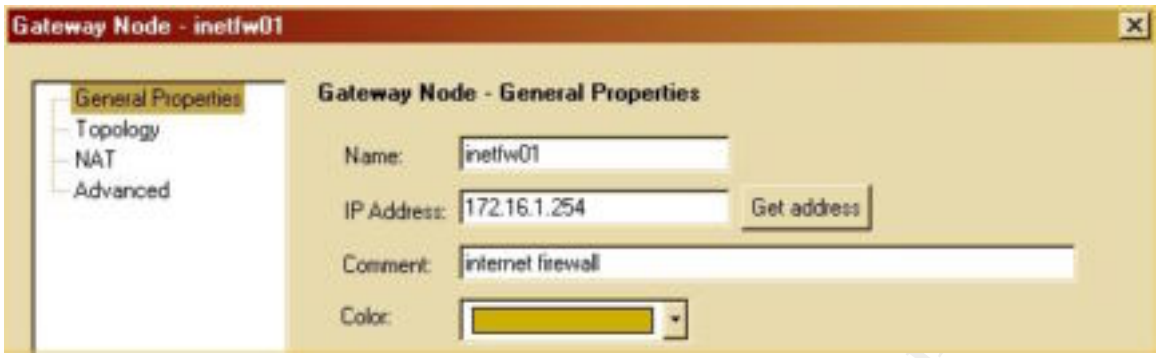
We also configure the object to recognize it's active network interfaces under the Topology option:



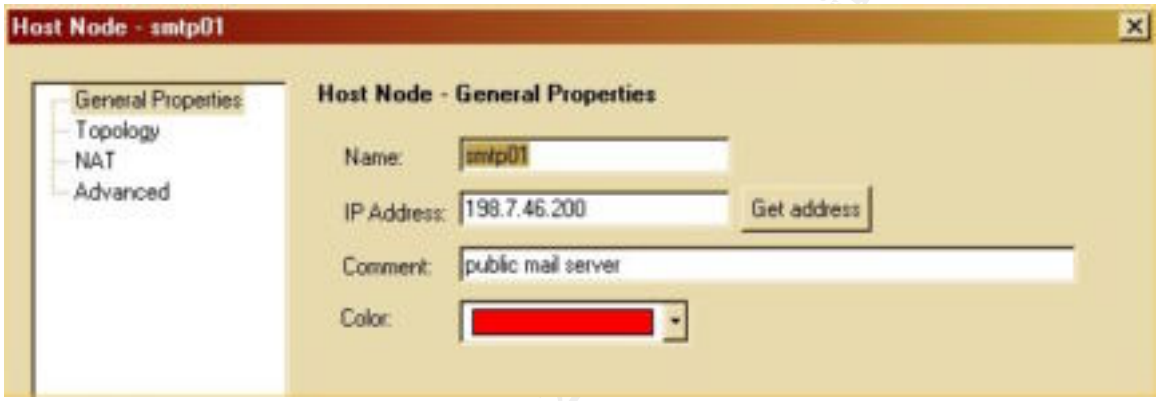
We also create a network object for each of our networks. To create the object we select Manage → Network Objects → New → Network:



Next, we create objects for each of the servers and networking devices on the GIAC network. To create objects for network gateways such as the border router, or Internet firewall we select Manage → Network Objects → New → Node → Gateway:



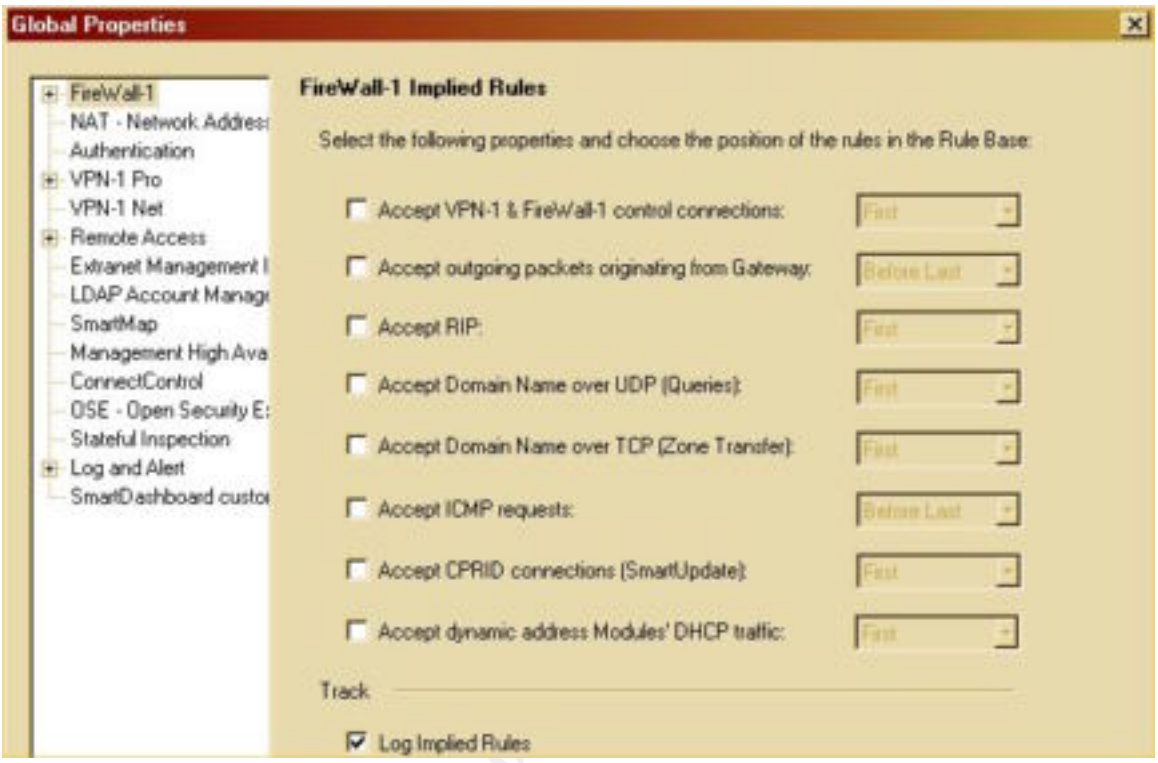
To create objects for hosts such as the NTP servers we select Manage → Network Objects → New → Node → Host:



To create objects for the VPN IP pools and the IT workstation address range we select Manage → Network Objects → New → Network Range:

With the basic network objects created, we can begin the development of the firewall policy. The first step is to adjust the default properties for all firewall enforcement points. The default implied rules are far too permissive. We uncheck the “Accept VPN-1 & FireWall-1 control connections”, “Accept CPRID

connections (SmartUpdate)", and "Accept dynamic address Modules' DHCP traffic" options. We also enable logging of implied rules. Finally, we disable the "Accept outgoing packets originating from gateway" option. This is done under Policy → Global Properties:

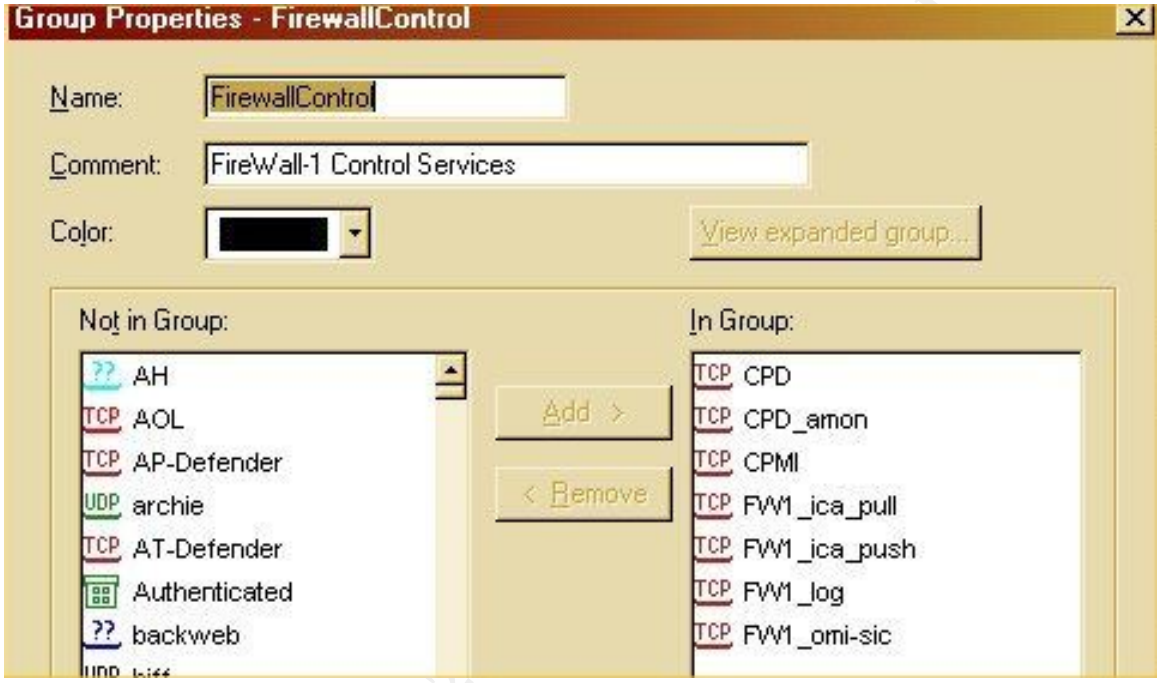


To allow connections to manage the firewall we manually recreate rules to allow the access based on the following table, drawn from Check Point's, *Common Ports Used by Check Point Next Generation*:

Port	Name	Description
257/tcp	FW1_Log	Used for logging between the enforcement point and the logging server as well as between the client and the logging server
18210/tcp	FW1_ica_pull	Used for creation and exchange of certificates from the enforcement point to the management station
18211/tcp	FW1_ica_push	Used for creation and exchange of certificates from the management station to the enforcement point
18186/tcp	FW1_omi-sic	Used for Secure Internal Communication (SIC) to allow communication between the management station and the enforcement point
18190/tcp	CPMI	Firewall management process for GUI control of the management station
18191/tcp	CDP	Used for communication between the

		management station and the enforcement point for policy install, certificate management, and status checks
18192/tcp	CPD_Amon	Used for application monitoring

We create a group that contains these services. To create a group of services we select Manage → Service → New → Group:



We also create a group that contains the IP addresses of the IT staff's workstations and VPN IP address pool for the purposes of connectivity to the management station. To create a group of network objects we select Manage → Network Objects → New → Group:

We then create rules that allow communication between the management station and the enforcement point as well as between the firewall administrators and the

management station. The SecurityServices group allows HTTP, HTTPS, SSH, and Telnet communications for the purposes of administration and system backups. To add a rule we select Rules → Add Rule → Top:

NO.	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON
1	mgmt01	corpfw01	Any	FirewallControl SecurityServices	accept	Log	Policy Targets
2	corpfw01	mgmt01	Any	FirewallControl SecurityServices	accept	Log	Policy Targets
3	FWAdmins	mgmt01	Any	FirewallControl SecurityServices	accept	Log	Policy Targets

We then create rules that allow the internal DNS server to forward DNS queries to the DNS servers on the Customer and Collaboration Network and to allow Exchange to relay email with the mail server on the same network:

4	w2kmail	ExternalDNSServers		dns	accept	Log	Policy Targets
5	w2kmail	smtp01		smtp	accept	Log	Policy Targets
6	smtp01	w2kmail		smtp	accept	Log	Policy Targets

We then create rules to allow GIAC's servers (Linux, Windows, and Solaris systems) and GIAC's network devices (Cisco devices, Internet and Corporate firewalls) access to the NTP servers for NTP and to the logging server (ntp01) for syslog. We also create a rule to let the NTP servers communicate to the NTP servers on the Internet for NTP:

7	GIACServers GIACNetworkDe	NTPServers		ntp	accept	Log	Policy Targets
8	NTPServers	io.berkeley.netdot.net caesar.cs.wisc.edu		ntp	accept	Log	Policy Targets
9	GIACServers GIACNetworkDe	ntp01		syslog	accept	Log	Policy Targets

We the create a rule that allows the proxy server to access the Customer and Collaboration Network and the Internet for HTTP and HTTPS and a rule that allows the IT staff to connect to server and network devices via the support protocols defined above:

10	proxy01	ProtectedNetworks InternalNetworks		WebServices	accept	Log	Policy Targets
11	FWAdmins	GIACNetworkDevices GIACServers		SecurityServices	accept	Log	Policy Targets

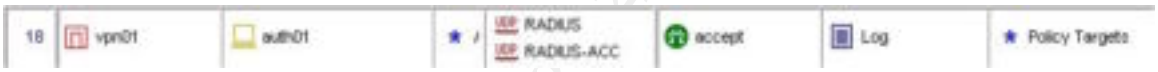
Next, we create a rule that allows the Web login server to send authentication attempts against the Safeword server via 5031/tcp. We also create a rule that allows the secure Web server to connect to the Oracle database server on the Commerce network via 1521/tcp and a rule that allows the patch processors on the database server to connect to the secure Web server and Dave's Bakery:

We then create a rule that allows the IT staff to administer the Safeword application on the Security network over 5040/tcp:

We then create a rule that allows the backup server to connect to the servers on the Commerce and Security networks for the purposes of backups using the restricted port ranges defined in section 1:

We also allow the IP address pool for employees and for IT staff to connect to any system on the Server network for any protocol:

The next rule for allowed traffic allows the VPN concentrator to validate user credentials against the RADIUS server:



The last rule for allowed traffic allows the Internet firewall to SSH to the management server:

Finally, the last rule drops any traffic that has not matched a previous rule:

A completed copy of the ruleset can be found in Appendix B, Code Samples.

VPN Security

Employees access the GIAC internal network through the Cisco 3015 VPN Concentrator. While the 3015 concentrator performs software cryptography, it is based on the same chassis as the 3030, 3060, and 3080. As a result, the 3015 can be field upgraded to support hardware-based encryption by adding Scalable Encryption Processing (SEP) modules.

The 3015 concentrator's primary functions are to provide:

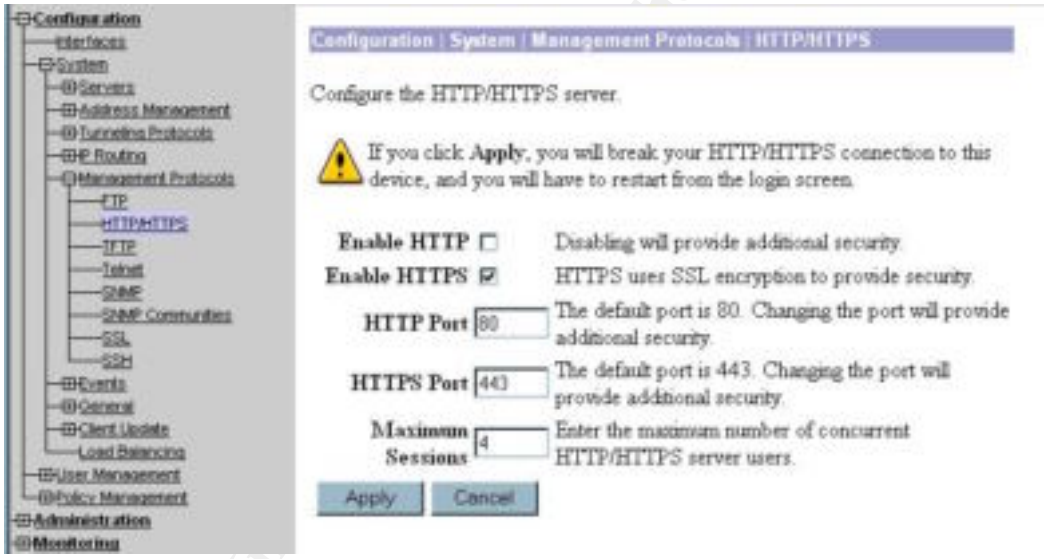
- Employees secure remote access to systems on the Server and Customer and Collaboration networks
- IT staff secure remote access to support GIAC's IT infrastructure, including the critical business systems on the Customer and Collaboration network

There are two active interfaces on the VPN 3015. Ethernet1 functions as the private address and is assigned an IP address of 172.16.4.254. Ethernet2 is the public (i.e. Internet accessible) interface and is assigned an IP address of 198.7.46.135. Traffic from a VPN client enters Ethernet2 encrypted, is decrypted by the concentrator, and then exits Ethernet1.

Base VPN Configuration and Hardening

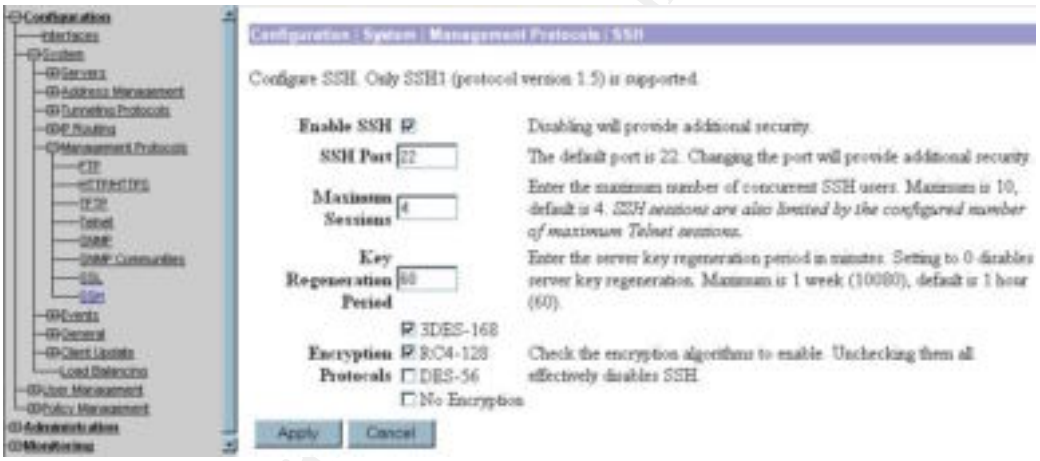
The VPN concentrator is administered via SSH and HTTPS from within the internal network. The 3000 series concentrators come equipped with a comprehensive Web based interface (https://corpvpn01.giacfortunes.com) that eases the setup and ongoing administration of the device. Administrative access via FTP, HTTP, TFTP, Telnet, and SNMP have been disabled. These protocols are considered insecure for a variety of reasons.

To support remote administration over an encrypted session, we turn on HTTPS and turn off HTTP. HTTPS access to the Web based interface is performed under Configuration → System → Management Protocols → HTTP/HTTPS:



We then select the strength of the cryptographic algorithm used to communicate via HTTPS. In this case, we avoid algorithms that are easy to crack. Selecting an encryption algorithm and SSL version for HTTPS connections strength version can be found under Configuration → System → Management Protocols → SSL:

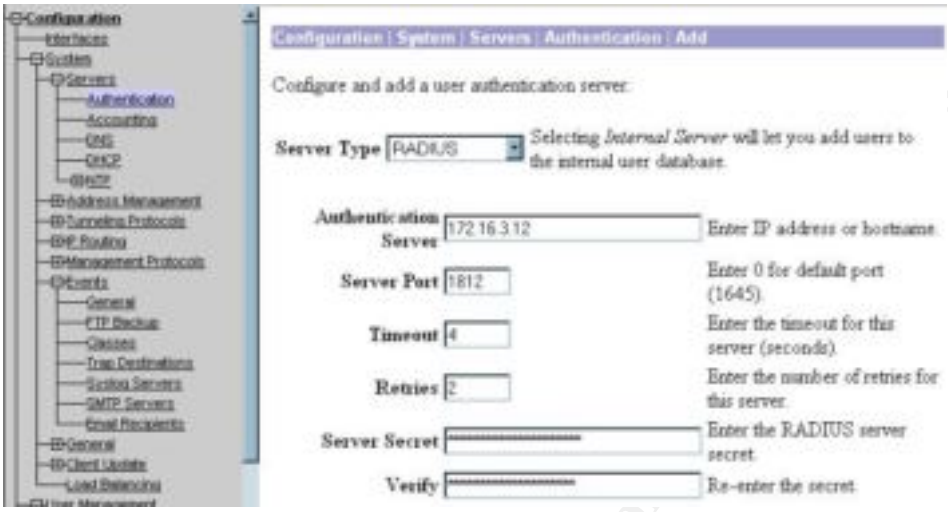
We then enable SSH for remote shell access over an encrypted session. Like HTTPS, this assures that authentication and configuration of the device is not done in clear-text. SSH is enabled under Configuration → System → Management Protocols → SSH:



The VPN concentrator is then configured to send system events to the logging server under Configuration → Events → Syslog Servers:

As the SafeWord server on the Security network will handle authentication and accounting, the concentrator is configured to validate authentications using

RADIUS under Configuration → System → Servers → Authentication (or Accounting):



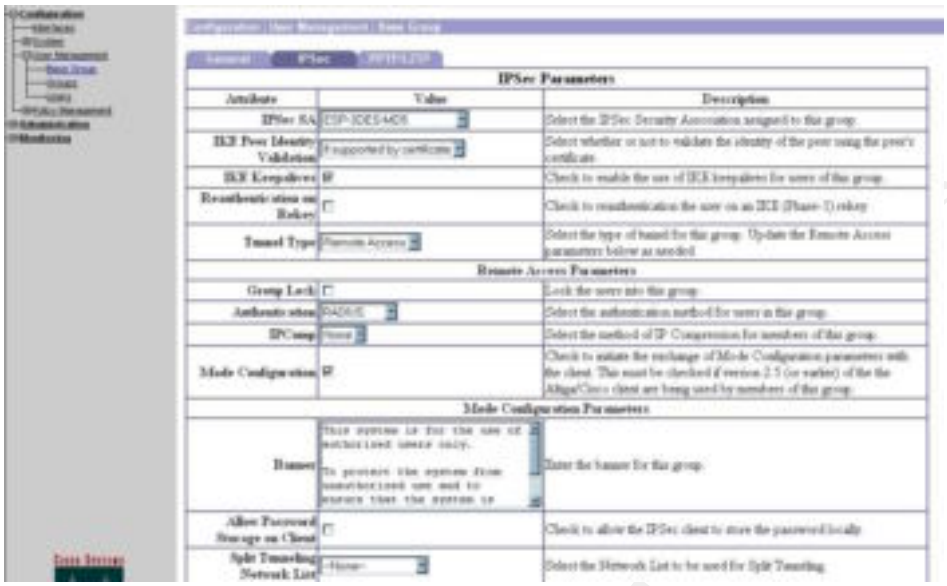
VPN Policy

The corporate firewall performs access control as traffic comes to and from the VPN concentrator on the private interface. Filtering at the corporate firewall is accomplished using the source IP address of the connection. As a result, there are two distinct IP address pools:

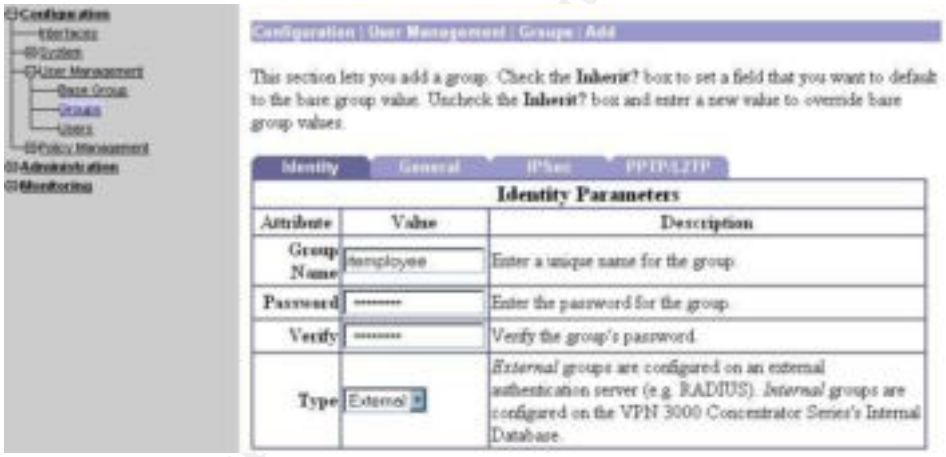
- 172.16.4.1 – 172.16.4.62 for employees
- 172.16.4.225 – 172.16.4.230 for the IT staff

The address space between the employee and IT pools is reserved for future use.

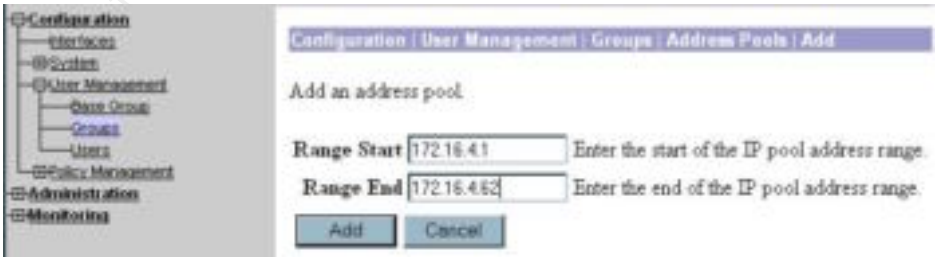
The GIAC VPN is IPSec protocol based, and uses the Encapsulating Security Payload (ESP) protocol. As previously mentioned, packet authentication is performed using an MD5-HMAC-128 hash with encryption performed via 3DES. GIAC also enables a warning banner for users connecting to the VPN concentrator. The base configuration for all users is performed under Configuration → User Management → Base Group:



The employee and itemployee groups are then created under Configuration → User Management → Groups:



Finally, by selecting the group under Configuration → User Management → Groups and then clicking the Modify Address Pools button the IP address pool for the group is assigned:



Assignment 3 – Verify the Firewall Policy

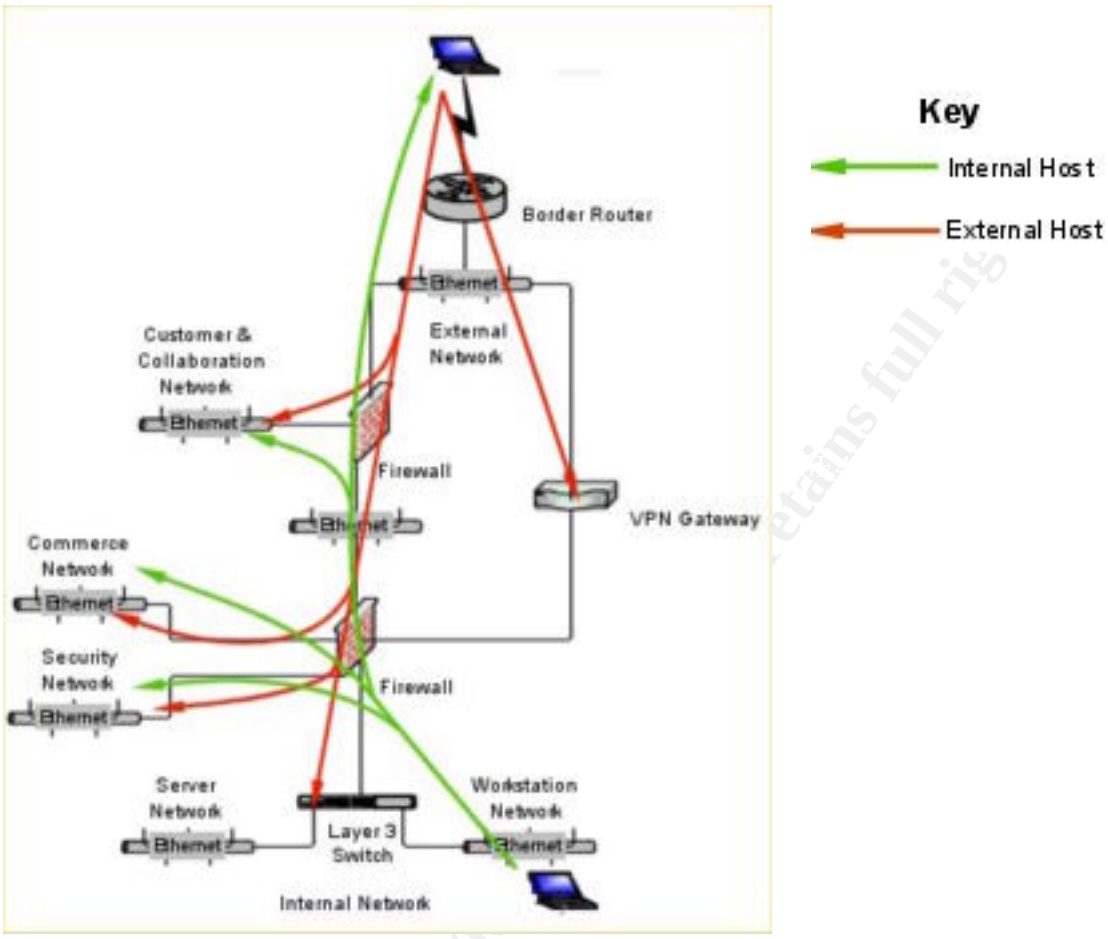
The purpose of the audit is to identify vulnerabilities in an organization's information-technology assets. Security is a dynamic issue, information gained from the audit helps establish a baseline for security of the IT infrastructure. This baseline can be used to quantify the existing risk to the GIAC IT infrastructure as well as to measure the baseline for improvement with subsequent audits.

Audit Approach

The firewall assessment process can be divided into five phases: network discovery, vulnerability identification, analysis, recommendations, and resolution. The phases are defined as:

- **Network discovery—**including the identification, profiling, and enumeration of network segments.
- **Vulnerability identification—**focusing on the verification of security vulnerabilities that can be exploited to affect the availability, confidentiality, or integrity of information.
- **Analysis—**including the review of results, discard of false positives/negatives, and documentation of potential vulnerabilities and associated risks.
- **Recommendations—**documentation of security best practices and specific steps to mitigate risks.
- **Resolution—**documentation of action items and planned steps by infrastructure teams to resolve issues.

The firewall assessment will evaluate the GIAC IT infrastructure from two viewpoints. The first viewpoint is that of an external entity on the Internet trying to access GIAC's internal and Internet facing resources. The second viewpoint is from an internal employee's perspective and the ability to gain access to systems protected by network security components as well as unprotected systems. The following diagram illustrates the two perspectives within the assessment.



Audit Tools

Several tools are used as part of the base audit. Multiple tools are used to leverage the strengths of each tool and to assist in controlling for false positives. By using multiple tools and reviewing the results of said tools, it is possible to validate identified vulnerabilities. The following tools are used during the audit:

Tool	Purpose
Nmap	Tool used to perform port scans and OS fingerprinting
Tcpdump	Protocol packet capture and dump utility that will allow us to view network traffic
Nessus	Security auditing tool that allows for testing of a vast array of vulnerabilities
Sara	Security auditing tool that allows for testing of a vast array of vulnerabilities

Audit Resource Requirements

The GIAC security audit is conducted by an independent consulting organization. As the IT staff was responsible for the implementation of the architecture, the staff believes an independent review may assist in uncovering vulnerabilities that the IT staff has not considered. The IT staff is involved in the audit both for reference and to assure that no harm comes to the infrastructure during the assessment.

The audit is conducted over the course of a weekend. As GIAC's primary customer base is the business customer, it is believed that a weekend audit has the lowest likelihood of impacting services for customers. The consulting firm GIAC has contracted with for the audit charges an hourly rate of $150.00. GIAC's IT staff is also present during the audit. The GIAC staff is factored at a rate of $65.00 per hour. The following table outlines the expected cost of the audit:

Resource	Cost per Hour	Hours	Extended Cost
On-Site Audit			
Security Analysts (2 staff)	$300.00	20	$6,000.00
GIAC Staff (2 staff)	$130.00	20	$2,600.00
Report Generation			
Security Analyst	$150.00	40	$6,000.00
Total Audit Cost			**$14,600.00**

Audit Risks

There are a few risks in conducting a security audit of the GIAC infrastructure. First and foremost, an audit that conducts a broad array of vulnerability testing may impact the availability of a critical system during a test for denial of service vulnerabilities. Second, if an attack were to occur during the audit, it may be difficult to identify the attack and to separate it from the activity conducted as part of the audit. Third, information gained during the audit may include confidential data regarding the GIAC network architecture, customer information, or intellectual property.

To address these issues, the GIAC IT staff has implemented several precautions. First, the IT staff has discussed the purpose of conducting the audit with the GIAC management staff. As part of this discussion the IT staff disclosed the potential impact to the availability of systems during the audit and the possible exposure of confidential data to a third party (the consulting company). The GIAC IT staff has received written approval from the GIAC management staff to conduct the audit. Second, GIAC has notified its ISP of the security audit as a courtesy in the event abnormal events are detected outside the GIAC network. Third, GIAC has received a signed non-disclosure from the consulting company and has conducted an independent check of the company's previous record of

assessments and conduct. Next, in preparing for potential negative impacts to systems, all systems have been fully backed up to assure easy recovery. Recovery tests are conducted to an isolated environment to verify that current backup processes will result in a successful recovery. Finally, the IT staff has established a "cut-out" policy with the consulting company. This policy provides the IT staff with a quick and effective mechanism for calling off the audit if an actual attack occurs or if the audit dramatically affects GIAC's infrastructure.

Audit Execution

Network Discovery

The technical architecture assessment begins with the mapping of the network infrastructure. The assessment team identified several network segments comprising the GIAC infrastructure used to probe during the assessment. The discovery designated addresses in the 172.16.0.0/16 private space as well as addresses in 198.7.46.128/25. The following table identifies the direction by which the segments where identified. The inbound scan focused on the evaluation of address space from outside the GIAC network, while the outbound scan focused on the evaluation of address space from inside the GIAC network.

Network Segments	Inbound	Outbound
198.7.46.128/26	√	√
198.7.46.192/26	√	√
172.16.1.0/24		√
172.16.2.0/24		√
172.16.3.0/24		√
172.16.4.0/24		√
172.16.8.0/24		√
172.16.9.0/24		√
172.16.12.0/24		√

Firewall Vulnerabilities

As part of the assessment of network wide vulnerabilities, the audit conducted a review of the effectiveness of the network intrusion detection system and the enforcement of firewall rulesets by firewall systems. The network intrusion detection system provides adequate detection of security events and notification of such events to the IT staff. GIAC does not have a formalized incident response to handle security events. As such, GIAC may not be fully prepared to handle a security event in real-time.

To validate the enforcement of firewall policy, the audit team focused on the manual verification of each firewall rule. A combination of tools was used to audit the firewall rulesets including tcpdump and nmap. Tcpdump was used on both

sides of a firewall to determine if the firewall is passing, dropping, or rejecting traffic. The command line mode for nmap was used to perform TCP and UDP port scans to find open ports. For the purposes of this discussion, an open port is one in which traffic is allowed to pass the firewall to the target system.

Nmap Connect Scan

The first nmap scan is an inbound scan to determine the ports open on the primary firewall. Since we expect the router and firewall to block ICMP echo requests, the –P0 option is used. This tells nmap not to ping the target before conducting the port scan. The –O flag is used to attempt to fingerprint the operating system of each host. As a result, Nmap was able to determine that the Internet firewall is running a version of RedHat Linux on an x86 platform. This information could be used to further exploit vulnerabilities on the system. The following output is from the initial nmap scan:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating Connect() Scan against inetfw01.giacfortunes.com
(198.7.46.132)
The Connect() Scan took 100,234 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
All 65535 scanned ports on inetfw01.giacfortunes.com (198.7.46.132)
are: filtered
Too many fingerprints match this host for me to give an accurate OS
guess
TCP/IP fingerprint:
SInfo(V=2.54BETA22%P=i386-redhat-linux-gnu%D=9/29%Time=3D978E76%O=-
1%C=-1)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)
```

We also conducted this scan from the internal network. In this case, the firewall accepts connections for SSH from the audit host on the internal network. Again Nmap was able to determine that the Internet firewall is running a version of RedHat Linux on an x86 platform:

```
[root@scanhost conf]# nmap -v -sT -P0 -O -p 1-65535 172.16.1.254

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (172.16.1.254) appears to be up ... good.
Initiating Connect() Scan against intefw01 (172.16.1.254)
Adding TCP port 22 (state open).
The Connect() Scan took 13,800 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
For OSScan assuming that port 22 is open and port 43969 is closed and
neither are firewalled
For OSScan assuming that port 22 is open and port 34486 is closed and
neither are firewalled
For OSScan assuming that port 22 is open and port 34892 is closed and
neither are firewalled
```

```
Interesting ports on inetfw01 (172.16.1.254):
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
No OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=2.54BETA22%P=i386-redhat-linux-
gnu%D=9/29%Time=3D9779A2%O=22%C=-1)
TSeq(Class=RI%gcd=1%SI=1A2EE2%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=1A2941%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=1A2940%IPID=C%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=1714496 (Good luck!)
IPID Sequence Generation: Duplicated ipid (!)
```

We also conduct this scan against the corporate firewall to determine the open ports on the system. As FireWall-1 accepts incoming packets with a SYN bit set, the connect scan is able to determine which ports are listening on the system. It should be noted that Nmap was unable to determine the operating system (this scan was conducted from a trusted host that is allowed to connect to the firewall):

```
[root@buttercup /]# nmap -v -sT -P0 -O -p 1-65535 172.16.8.1

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host corpfw01 (172.16.8.1) appears to be up ... good.
Initiating Connect() Scan against corpfw01 (172.16.8.1)
Adding TCP port 22 (state open).
Adding TCP port 443 (state open).
Adding TCP port 257 (state open).
The Connect() Scan took 13590 seconds to scan 65535 ports.
For OSScan assuming that port 22 is open and port 23 is closed and
neither are firewalled
For OSScan assuming that port 22 is open and port 23 is closed and
neither are firewalled
For OSScan assuming that port 22 is open and port 23 is closed and
neither are firewalled
Interesting ports on corpfw01 (172.16.8.1):
(The 65531 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
23/tcp     closed      telnet
80/tcp     closed      http
443/tcp    open        https
No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=2.54BETA22%P=i386-redhat-linux-
gnu%D=9/29%Time=3D97863A%O=22%C=23)
```

```
TSeq(Class=RI%gcd=1%SI=270E%IPID=I)
TSeq(Class=RI%gcd=1%SI=1A40%IPID=I)
TSeq(Class=RI%gcd=1%SI=2CB6%IPID=I)
T1(Resp=Y%DF=N%W=4000%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=11446 (Worthy challenge)
IPID Sequence Generation: Incremental
```

Nmap ACK Scan

The second nmap scan is an ACK scan. This scan is expected to evade the packet filtering router as it is expected to filter only on SYN packets. The results for this scan were the same from both networks. It should be noted that in the case of the internal scan, the traffic was actually blocked by the FireWall-1 on the corporate firewall. The scan is conducted on both firewalls, with the results virtually identical. To conduct this scan we use the –sA flag. The output of the scan is shown below:

```
[root@scanhost conf]# nmap -v -sA -P0 -p 1-65535 198.7.46.132

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating ACK Scan against  (198.7.46.132)
The ACK Scan took 113,821 seconds to scan 65535 ports.
(no tcp responses received -- assuming all ports filtered)
All 65535 scanned ports on (198.7.46.132) are filtered
```

Nmap FIN Scan

The third nmap scan is a FIN scan. This scan was completely ineffective in detecting open ports on the primary firewall. Again, the results of this scan were identical from the external and internal locations. The scan is conducted on both firewalls, with the results virtually identical. With the internal scan, FireWall-1 dropped the traffic and did not allow it to pass through to the Internet firewall. To conduct this scan we use the –sF flag. The output of the scan is shown below:

```
[root@scanhost conf]# nmap -v -sF -P0 -p 1-65535 198.7.46.132

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating FIN Scan against  (198.7.46.132)
The FIN Scan took 13,800 seconds to scan 65535 ports.
(no tcp responses received -- assuming all ports filtered)
All 65535 scanned ports on (198.7.46.132) are filtered
```

Nmap XMAS Scan

The nmap scan is known as an XMAS scan and sets the FIN, URG, and PUSH bits in the TCP packet. When conducting this scan against the system, all traffic

was detected, logged, and dropped. In the case of the inbound scan, the traffic was dropped by the Internet firewall. In the case of the outbound scan, traffic was dropped by the corporate firewall. The scan is conducted on both firewalls, with the results virtually identical. To conduct this scan we use the –sX flag. The output of the scan is shown below:

```
[root@scanhost conf]# nmap -v -sX -P0 -p 1-65535 198.7.46.132

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating FIN Scan against  (198.7.46.132)
The XMAS Scan took 78,600 seconds to scan 65535 ports.
(no tcp responses received -- assuming all ports filtered)
All 65535 scanned ports on (198.7.46.132) are filtered
```

Nmap UDP Scan

The final nmap scan conducted is an UDP port scan. The results for this scan were the same from both networks. It should be noted that in the case of the internal scan, the traffic was actually blocked by FireWall-1 on the corporate firewall. The scan is conducted on both firewalls, with the results virtually identical. To conduct this scan we use the –sU flag. The output of the scan is shown below:

```
[root@buttercup root]# nmap -v -sU -P0 -p 1-65535 198.7.46.132

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating UDP Scan against inetfw01 (198.7.46.132)
The UDP Scan took 12,042 seconds to scan 65535 ports.
(no udp responses received -- assuming all ports filtered)
All 65535 scanned ports on inetfw01-sec (198.7.46.132) are: filtered
```

Vulnerability Scans

Once these scans are completed, a brief check for vulnerabilities is conducted using both Nessus and Sara. The following information provides details, in port order, regarding the security status of the Internet and corporate firewalls. The assessment team identified potential opportunities for exploit based on the port scans conducted above.

Secure Shell (TCP Port 22)

The team determined that both firewalls are running the Secure Shell (SSH) server at the time of the assessment. The corporate firewall is running a version older than 3.0.1. Versions older than 3.0.1 are vulnerable to a flaw, in which an attacker may acquire unauthenticated access, provided Kerberos V support has been enabled. Kerberos V is not enabled on either system as part of the SSH server configuration. As such, this vulnerability is not directly present in the infrastructure at this point in time. In addition, the corporate firewall supports version 1.5 of the SSH protocol. This version is considered to not be cryptographically safe. The firewalls running a Secure Shell server include:

Hosts	Old SSH Version	Old SSH Protocol Version
198.7.46.132		√
172.16.8.1	√	√

Recommendation
The use of Secure Shell provides an alternative to several insecure protocols. Continue using Secure Shell wherever possible. Disable support for version 1.5 of the SSH protocol.

Hypertext Transfer Protocol (TCP Port 80 & 443)

The team also determined that the corporate firewall is running a Web server over HTTPS. The corporate firewall is running a version of OpenSSL which is older than 0.9.6g. Versions prior to 0.9.6g are susceptible to a buffer overflow vulnerability that may allow someone to obtain a shell session on the system. It is also running a version of mod_ssl older than 2.8.10 that is vulnerable to a buffer overflow, which will allow an attacker to obtain a shell on the host.

Hosts	Old OpenSSL Version	Old mod_ssl Version
172.16.8.1	√	√

Recommendation
Verify that the hardware vendor has patched the affected version of OpenSSL and mod_ssl. As the OpenSSL vulnerability is relatively recent, check with the vendor to determine if a patch is available for the vulnerability.

Ruleset Audit

To audit traffic handled by the Internet firewall, the audit team used a combination of nmap, telnet, and tcpdump to determine which traffic is able to pass the system. Other standard Unix utilities such as netstat, ntpdate, nslookup, dig, ping, ssh, and traceroute were also used to conduct the audit.

As there are two layers of firewalls in place, the firewall audit required the review of both firewall clusters. For traffic that must pass both firewalls, the audit team focused on the audit of such traffic when auditing the Internet firewall ruleset. The remaining rules in the corporate firewall were audited separately.

Internet Firewall Ruleset

While the Internet firewall ruleset audit focused on verifying the resources available to external users, it also validates rules that allow communication to hosts on the Internal network as well. The audit begins by verifying the INPUT filter. As we only allows SSH connections from the Internal network, any other traffic to the firewall should be dropped:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (172.16.1.254) appears to be up ... good.
Initiating Connect() Scan against intefw01 (172.16.1.254)
Adding TCP port 22 (state open).
The Connect() Scan took 13,800 seconds to scan 65535 ports.
Interesting ports on inetfw01 (172.16.1.254):
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
```

An attempt to connect to the firewall from outside of the GIAC network for any protocol is unsuccessful:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host inetfw01 (198.7.46.132) appears to be up ... good.
Initiating Connect() Scan against inetfw01.giacfortunes.com
(198.7.46.132)
The Connect() Scan took 100,234 seconds to scan 65535 ports.
All 65535 scanned ports on inetfw01.giacfortunes.com (198.7.46.132)
are: filtered
```

We then verify the OUTPUT filter by testing the traffic that is allowed out of the firewall. We first verify that we can SSH to the management system on the Security network. This also verifies rule 19 in the corporate firewall:

```
[ssajlo@inetfw01 /]# ssh 172.16.3.11
The authenticity of host '172.16.3.11 (172.16.3.11)' can't be
established.
RSA key fingerprint is 41:0d:86:83:61:05:50:04:e3:b7:f3:68:1a:bb:2c:0c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.3.11' (RSA) to the list of known
hosts.
ssajlo@172.16.3.11's password:
Last login: Sat Oct  5 13:36:03 2002 from 10.10.175.23
```

We also attempt to SSH to another host that is not allowed:

```
[ssajlo@inetfw01 /]# ssh 198.7.46.205
Secure connection to 198.7.46.205 refused.
```

We then verify the firewall has the ability to send mail to the mail server on the Customer and Collaboration network via 25/tcp using a telnet connection to port 25:

```
[ssajlo@inetfw01 /]# telnet 198.7.46.200 25
Trying 198.7.46.200...
Connected to 198.7.46.200.
Escape character is '^]'.
220 smpt01.giacfortunes.com ESMTP Some Mail Server; Sat, 5 Oct 2002
13:40:25 -0500
```

We then try to connect to the internal mail server to verify that the firewall is not allowed to do so:

```
[ssajlo@buttercup /]# telnet 172.16.9.11 25
Trying 172.16.9.11...
telnet: connect to address 172.16.9.11: Connection timed out
```

We then verify that we can perform DNS queries against the DNS servers on the Customer and Collaboration network using nslookup:

```
[root@inetfw01 root]# nslookup www
Server:         198.7.46.201
Address:        198.7.46.201#53
Name:   www.giacfortunes.com
Address: 198.7.46.203
```

We attempt to perform a DNS query against the internal DNS server with no success:

```
[ssajlo@inetfw01 /]# nslookup
> server 172.16.9.11
Default server: 172.16.9.11
Address: 172.16.9.11#53
> www.sans.org
;; connection timed out; no servers could be reached
```

Finally, we validate the rules allowing NTP to the internal NTP servers by running ntpdate to observe the traffic between the firewall and the NTP servers. This also validates rule 7 on the corporate firewall:

```
[root@inetfw01 /]# ntpdate ntp01.giac-fortunes.com
5 Oct 15:19:09 ntpdate[11157]: step time server 172.16.3.11 offset
4597.542075 sec
```

We then try to run ntpdate against one of the NTP servers on the Internet:

```
[root@inetfw01 /]# ntpdate 128.105.39.11
5 Oct 15:25:59 ntpdate[11163]: no server suitable for synchronization
found
```

With the INPUT and OUTPUT filters reviewed, the audit team then focuses on the FORWARD filter. There are several rules at the beginning of the FORWARD filter used to detect scans similar to those conducted above using nmap. These are logging rules and are reviewed later. The first rule to allow traffic is intended to allow Internet mail servers to connect to the external mail server for SMTP mail delivery. We test this rule using a telnet connection to the mail server from an external host on port 25/tcp:

```
telnet 198.7.46.200 25
Trying 198.7.46.200...
Connected to 198.7.46.200.
Escape character is '^]'.
220 smpt01.giacfortunes.com ESMTP Some Mail Server; Sat, 5 Oct 2002
13:40:25 -0500
```

We then verify that the external mail server is allowed to send mail to the Internet by using telnet to connect to port 25/tcp on a mail server on the Internet from the external mail server:

```
telnet mail1.giac.org 25
Trying 65.173.218.103...
Connected to smtphost.wpl.com (65.173.218.103).
Escape character is '^]'.
220 mail1.giac.org ESMTP
```

We then verify that the external mail server cannot send mail to an internal system other than the Exchange server:

```
[root@buttercup conf]# telnet 172.16.3.10 25
Trying 172.16.3.10...
telnet: connect to address 172.16.3.10: Connection timed out
```

The next rules allow systems on the Internet to perform queries against the external DNS servers. We use nslookup to perform a query against the servers:

```
[root@dns01 etc]# nslookup
> server 198.7.46.201
Default server: 198.7.46.201
Address: 198.7.46.201#53
> www.giacfortunes.com
Server:         198.7.46.201
Address:        198.7.46.201#53

Name:   www.giacfortunes.com
Address: 198.7.46.203
```

We then verify that systems on the Internet are allowed to connect to the public and secure Web servers as well as to the Web Login server for both HTTP and HTTPS. We use tcpdump to observe connection attempts to these systems:

```
Public Web Server
12:46:35.910944 I 198.7.47.200.44149 > 198.7.46.203.80: S
370052528:370052528(0) win 5840 <mss 1460,sackOK,timestamp
68543688[|tcp]> (DF) (ttl 64, id 27000)
12:46:35.913782 O 198.7.46.203.80 > 198.7.47.200.44149: S
899793864:899793864(0) ack 370052529 win 5792 <mss
1460,sackOK,timestamp 249922366[|tcp]> (DF) (ttl 62, id 0)
12:46:35.913895 I 198.7.47.200.44149 > 198.7.46.203.80: . ack 1 win
5840 <nop,nop,timestamp 68543688 249922366> (DF) (ttl 64, id 27001)
12:46:36.110944 I 198.7.47.200.44149 > 198.7.46.203.443: S
370052929:370052929(0) win 5840 <mss 1460,sackOK,timestamp
68543688[|tcp]> (DF) (ttl 64, id 27010)
12:46:36.113782 O 198.7.46.203.443 > 198.7.47.200.44149: S
899793964:899793964(0) ack 370052529 win 5792 <mss
1460,sackOK,timestamp 249922366[|tcp]> (DF) (ttl 62, id 10)
12:46:36.113895 I 198.7.47.200.44149 > 198.7.46.203.443: . ack 1 win
5840 <nop,nop,timestamp 68543688 249922366> (DF) (ttl 64, id 27010)
Secure Web Server
12:55:26.075637 198.7.47.200.44173 > 198.7.46.204.80: S [tcp sum ok]
3776331871:3776331871(0) win 5840 <mss 1460,sackOK,timestamp 69217613
0,nop,wscale 0> (DF) (ttl 63, id 27922, len 60)
```

```
12:55:26.075637 198.7.46.204.80 > 198.7.47.200.44173: S [tcp sum ok]
3718801292:3718801292(0) ack 3776331872 win 5792 <mss
1460,sackOK,timestamp 250596339 69217613,nop,wscale 0> (DF) (ttl 63, id
0, len 60)
12:55:26.075637 198.7.47.200.44173 > 198.7.46.204.80: . [tcp sum ok]
1:1(0) ack 1 win 5840 <nop,nop,timestamp 69217613 250596339> (DF) (ttl
63, id 27923, len 52)
15:57:14.435637 198.7.47.200.44174 > 198.7.46.204.443: S [tcp sum ok]
3884274323:3884274323(0) win 5840 <mss 1460,sackOK,timestamp 69228447
0,nop,wscale 0> (DF) (ttl 63, id 1292, len 60)
15:57:14.435637 198.7.46.204.443 > 198.7.47.200.44174: S [tcp sum ok]
3828585031:3828585031(0) ack 3884274324 win 5792 <mss
1460,sackOK,timestamp 250607174 69228447,nop,wscale 0> (DF) (ttl 63, id
0, len 60)
15:57:14.435637 198.7.47.200.44174 > 198.7.46.204.443: . [tcp sum ok]
1:1(0) ack 1 win 5840 <nop,nop,timestamp 69228448 250607174> (DF) (ttl
63, id 1293, len 52)
Web Login Server
12:58:38.355637 198.7.47.200.44175 > 198.7.46.205.80: S [tcp sum ok]
3964929593:3964929593(0) win 5840 <mss 1460,sackOK,timestamp 69236839
0,nop,wscale 0> (DF) (ttl 63, id 32350, len 60)
12:58:38.355637 198.7.46.205.80 > 198.7.47.200.44175: S [tcp sum ok]
3907573892:3907573892(0) ack 3964929594 win 5792 <mss
1460,sackOK,timestamp 250615567 69236839,nop,wscale 0> (DF) (ttl 63, id
0, len 60)
12:58:38.355637 198.7.47.200.44175 > 198.7.46.205.80: . [tcp sum ok]
1:1(0) ack 1 win 5840 <nop,nop,timestamp 69236839 250615567> (DF) (ttl
63, id 32351, len 52)
15:59:39.085637 198.7.47.200.44180> 198.7.46.205.443: S [tcp sum ok]
4020599599:4020599599(0) win 5840 <mss 1460,sackOK,timestamp 69242912
0,nop,wscale 0> (DF) (ttl 63, id 60175, len 60)
15:59:39.085637 198.7.46.205.443 > 198.7.47.200.44180: S [tcp sum ok]
3979929049:3979929049(0) ack 4020599600 win 5792 <mss
1460,sackOK,timestamp 250621640 69242912,nop,wscale 0> (DF) (ttl 63, id
0, len 60)
15:59:39.085637 198.7.47.200.44180 > 198.7.46.205.443: . [tcp sum ok]
1:1(0) ack 1 win 5840 <nop,nop,timestamp 69242912 250621640> (DF) (ttl
63, id 60176, len 52)
```

We then verify the rules that allow the internal DNS server to forward DNS queries to the DNS servers on the Customer and Commerce network. This also verifies rule 4 on the corporate firewall. We perform a DNS query against an Internet host (www.sans.org). If the firewalls are configured correctly, they should allow the internal DNS server to forward the query to the external DNS servers:

```
13:29:32.062700 I 172.16.9.19.51848 > 198.7.46.201.53:  40938 (44) (DF)
(ttl 254, id 39095)
13:29:37.638549 O 198.7.46.201.53 > 172.16.9.19.51848:  2665 q:
www.sans.org 1/3/2  (147) (DF) (ttl 254, id 16657)

Server:         172.16.9.19
Address:        172.16.9.19#53
Non-authoritative answer:
Name:   www.sans.org
Address: 63.100.47.46
```

```
Name:   www.sans.org
Address: 65.173.218.106
```

To confirm that the firewall does not allow direct queries from the internal network, we then point nslookup directly to the external DNS server and attempt to perform a query with no success:

```
[root@buttercup conf]# nslookup
> server 198.7.46.201
Default server: 198.7.46.201
Address: 198.7.46.201#53
> www.sans.org
;; connection timed out; no servers could be reached
```

We then verify the rules that allow the internal mail server to communicate with the external mail server on port 25/tcp and in the opposite direction. This verifies rules 5 and 6 on the corporate firewall as well. To conduct this test we perform a telnet connection from the internal mail server to the external mail server, connecting to port 25 and from the opposite direction back in:

```
telnet 198.7.46.200 25
Trying 198.7.46.200...
Connected to 198.7.46.200.
Escape character is '^]'.
220 smpt01.giacfortunes.com ESMTP Some Mail Server; Sat, 5 Oct 2002
13:40:25 -0500

[ssajlo@smtp01 conf]# telnet 172.16.9.11 25
Trying 172.16.9.11...
Connected to 172.16.9.11.
Escape character is '^]'.
220 exchg.giacfortunes.com ESMTP MS-Exchange 2000; Sat, 5 Oct 2002
13:40:25 -0500
```

We also try to connect to the servers from a host that should not be allowed to connect:

```
[root@buttercup conf]# telnet 198.7.46.200 25
Trying 198.7.46.200...
telnet: connect to address 198.7.46.200: Connection timed out

[root@buttercup conf]# telnet 172.16.9.11 25
Trying 172.16.9.11...
telnet: connect to address 172.16.9.11: Connection timed out
```

Next we check access from the Web login server to the SafeWord server for 5031/tcp. As part of this connection we also verify rule 12 on the corporate firewall. We observe this traffic using tcpdump:

```
12:46:35.910944 I 198.7.46.205.44150 > 172.16.3.12.5031: S
370052528:370052528(0) win 5840 <mss 1460,sackOK,timestamp
68543688[|tcp]> (DF) (ttl 64, id 27000)
12:46:35.913782 O 172.16.3.12.5031 > 198.7.46.205.44150: S
899793864:899793864(0) ack 370052529 win 5792 <mss
1460,sackOK,timestamp 249922366[|tcp]> (DF) (ttl 62, id 0)
```

```
12:46:35.913895 I 198.7.46.205.44150 > 172.16.3.12.5031:. ack 1 win
5840 <nop,nop,timestamp 68543688 249922366> (DF) (ttl 64, id 27001)
```

We then try to open a connection from a non-trusted host to verify the firewall does not allow the traffic through:

```
12:48:43.112591 I 198.7.46.200.44000 > 172.16.3.12.5031: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556408[|tcp]> (DF) [tos 0x10] (ttl 64, id 41328)
12:48:46.105807 I 198.7.46.200.44000 > 172.16.3.12.5031: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556708[|tcp]> (DF) [tos 0x10] (ttl 64, id 41329)
12:48:52.105812 I 198.7.46.200.44000 > 172.16.3.12.5031: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68557308[|tcp]> (DF) [tos 0x10] (ttl 64, id 41330)
```

To verify connectivity from the secure Web server to the Oracle database server on the Commerce network for 1521/tcp, we use telnet to attempt to connect to the Oracle server on 1521/tcp. The connection attempt also validates rule 13 on the corporate firewall:

```
[ssajlo@secureweb conf]# telnet db01 1521
Trying 172.16.2.10...
Connected to 172.16.2.10.
Escape character is '^]'.
```

We then attempt to connect to the Oracle server from the public Web server to determine if access is blocked:

```
[ssajlo@www conf]# telnet db01 1521
Trying 172.16.2.10...
telnet: connect to address 172.16.2.10: Connection refused
```

We then verify the batch processor on the Oracle server is allowed to connect to the secure Web server and to the application server at Dave's Bakery. A simple SSH attempt to each server confirms access, thus validating the rules on the Internet firewall and rule 14 on the corporate firewall:

```
[root@db01 /]# ssh -l filesync 198.7.46.204
filesync@198.7.46.204's password:
[root@db01 /]# ssh -l filesync 201.210.1.32
filesync@201.210.1.32's password:
```

We then try to perform an SSH attempt from the Oracle server to another host on the Internet that has SSH running to verify it is blocked:

```
[root@db01 /]# ssh 201.210.1.33
(the connection never prompts for a password)
```

We then check communications between the servers on the Customer and Collaboration network and the syslog server on the Security network for syslog (i.e. 514/udp). This also verifies rule 9 on the corporate firewall. We generate a log message using the logger command on the dns01 server and use tcpdump on the logging server to verify the traffic reaches the server:

```
[ssajlo@dns01 /]# logger -p kern.info "TEST"
15:16:59.158471 I 198.7.46.201.514 > 172.16.3.10.514:  udp 15 (DF) (ttl
64, id 0)
```

We verify that the border router is also allowed to send syslog messages back to the system using tcpdump to watch for traffic from the router:

```
14:26:59.128271 I 198.7.46.129.52290 > 172.16.3.10.514:  udp 15 (DF)
(ttl 64, id 1)
```

We also validate the rules allowing NTP to the internal NTP servers from the servers on the Customer and Collaboration network by running ntpdate to observe the traffic between dns01 and the NTP servers. This also validates rule 7 on the corporate firewall:

```
[root@dns01 /]# ntpdate ntp01.giac-fortunes.com
5 Oct 16:29:09 ntpdate[11157]: step time server 172.16.3.11 offset
4597.542075 sec
```

We then try to run ntpdate against one of the NTP servers on the Internet:

```
[root@dns01 /]# ntpdate 128.105.39.11
5 Oct 16:30:19 ntpdate[11163]: no server suitable for synchronization
found
```

We then verify the rules on the Internet firewall and rule 10 on the corporate firewall that allow HTTP and HTTPS to Internet sites from the proxy server. We use tcpdump to observe traffic to an Internet site (www.verisign.com):

```
16:48:54.758693 O 198.7.47.145.13899 > 65.205.249.56.80: S
2323434192:2323434192(0) win 6144 <mss 1460> (DF) (ttl 125, id 3127)
16:48:54.840429 I 65.205.249.56.80 > 198.7.46.145.13899: S
4112784079:4112784079(0) ack 2323434193 win 24820 <mss 1460> (DF) (ttl
46, id 62738)
16:48:54.841135 O 198.7.46.145.13899 > 65.205.249.56.80: P 1:380(379)
ack 1 win 6144 (DF) (ttl 125, id 3138)
16:49:28.330215 O 198.7.46.145.13936 > 65.205.249.56.443: S
2383304793:2383304793(0) win 6144 <mss 1460> (DF) (ttl 125, id 3805)
16:49:28.412371 I 65.205.249.56.443 > 198.7.46.145.13936: S
4175996821:4175996821(0) ack 2383304794 win 24820 <mss 1460> (DF) (ttl
46, id 62746)
16:49:28.412926 O 198.7.46.145.13936 > 65.205.249.56.443: . ack 1 win
6144 (DF) (ttl 125, id 3813)
```

We then try to connect to the site directly to verify the traffic is blocked:

```
16:49:28.330215 O 172.16.12.100.1200 > 65.205.249.56.443: S
2383304793:2383304793(0) win 6144 <mss 1460> (DF) (ttl 125, id 3900)
16:49:28.330215 O 172.16.12.100.1200 > 65.205.249.56.443: S
2383304793:2383304793(0) win 6144 <mss 1460> (DF) (ttl 125, id 3900)
16:49:28.330215 O 172.16.12.100.1200 > 65.205.249.56.443: S
2383304793:2383304793(0) win 6144 <mss 1460> (DF) (ttl 125, id 3900)
```

We then confirm that the rules allowing the NTP servers on the Security network to connect to the NTP servers on the Internet using ntpdate. This test also verifies that rule 8 on the corporate firewall functions as expected:

```
[root@ntp01 /]# ntpdate 128.105.39.11
5 Oct 16:29:19 ntpdate[11157]: step time server 128.105.39.11 offset
4597.542075 sec
[root@ntp01 /]# ntpdate 216.27.190.202
5 Oct 16:31:43 ntpdate[11157]: step time server 216.27.190.202 offset
4597.542075 sec
```

We then try to run ntpdate against a different NTP server on the Internet:

```
[root@ntp01 /]# ntpdate 128.105.39.12
5 Oct 16:30:19 ntpdate[11163]: no server suitable for synchronization
found
```

Finally, we test access to the border router from the Internal network to verify the rules on the Internet firewall allowing such access and rule 11 on the corporate firewall. These rules are verified by attempting to connect to the border router using telnet from the management server on the Security network:

```
[root@buttercup conf]# telnet 198.7.46.129
Trying 198.7.46.129...
Connected to 198.7.46.129.
Escape character is '^]'.

This system is the property of GIAC Enterprises and is for the use of
authorized users only.
Individuals using this computer system without authority, or in excess
of their authority, are subject
to having their activities monitored.
Unauthorized access is expressly prohibited.

User Access Verification
Password:
```

We also test to see if we can telnet to the border router from an untrusted host on the Internal network:

```
[root@proxy01 /]# telnet 198.7.46.129
Trying 198.7.46.129...
telnet: connect to address 198.7.46.129: Connection refused
```

Corporate Firewall Ruleset

The corporate firewall ruleset audit focused on validating each rule to verify that the desired traffic is allowed while other traffic is prohibited. Starting the FireWall-1 GUI and pushing policy to the firewall enforcement point from the management station checks the first rule. In addition, we conduct a port scan of the firewall from the management station to verify the ports allowed include the FireWall-1 management protocols and the necessary management protocols for the appliance:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host corpfw01-sec (172.16.3.1) appears to be up ... good.
Initiating Connect() Scan against corpfw01-sec (172.16.3.1)
Adding TCP port 22 (state open).
Adding TCP port 257 (state open).
Adding TCP port 443 (state open).
```

```
Adding TCP port 18264 (state open).
Adding TCP port 18191 (state open).
Adding TCP port 18192 (state open).
The Connect() Scan took 10068 seconds to scan 65535 ports.
Interesting ports on corpfw01-sec (172.16.3.1):
(The 65523 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
23/tcp     closed      telnet
80/tcp     closed      http
257/tcp    open        set
443/tcp    open        https
18186/tcp  closed      unknown
18190/tcp  closed      unknown
18191/tcp  open        unknown
18192/tcp  open        unknown
18210/tcp  closed      unknown
18211/tcp  closed      unknown
18264/tcp  open        unknown
```

We also then conduct a scan from a separate host to verify that it does not have access to the above services on the firewall:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host corpfw01 (172.16.8.1) appears to be up ... good.
Initiating Connect() Scan against corpfw01 (172.16.8.1)
The Connect() Scan took 100,234 seconds to scan 65535 ports.
All 65535 scanned ports on corpfw01 (172.16.8.1) are: filtered
```

We then verify rule 2 in which the firewall can connect to the management station by reviewing the SmartView Tracker for logging events from the firewall enforcement point. We also use SSH to connect back to the management station and by running netstat on the firewall to determine its currently open connections to the management station:

```
tcp        0      0  172.16.3.1.1811    172.16.3.11.22     ESTABLISHED
```

We then verify rule 3 by conducting an nmap scan from a trusted host on the Internal network to the firewall enforcement point

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host corpfw01 (172.16.8.1) appears to be up ... good.
Initiating Connect() Scan against corpfw01 (172.16.8.1)
Adding TCP port 22 (state open).
Adding TCP port 257 (state open).
Adding TCP port 443 (state open).
Adding TCP port 18264 (state open).
Adding TCP port 18191 (state open).
Adding TCP port 18192 (state open).
The Connect() Scan took 10068 seconds to scan 65535 ports.
Interesting ports on corpfw01 (172.16.8.1):
(The 65523 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
23/tcp     closed      telnet
80/tcp     closed      http
```

```
257/tcp    open        set
443/tcp    open        https
18186/tcp  closed      unknown
18190/tcp  closed      unknown
18191/tcp  open        unknown
18192/tcp  open        unknown
18210/tcp  closed      unknown
18211/tcp  closed      unknown
18264/tcp  open        unknown
```

We also conduct a scan from a separate host to verify that it does not have access to the above services on the firewall:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host corpfw01 (172.16.8.1) appears to be up ... good.
Initiating Connect() Scan against corpfw01 (172.16.8.1)
The Connect() Scan took 100,234 seconds to scan 65535 ports.
All 65535 scanned ports on corpfw01 (172.16.8.1) are: filtered
```

Rules 4 through 14 and 19 were audited above as part of the Internet firewall ruleset audit. Rule 15 however is specific to the corporate firewall. To verify rule 15 we open a connection from the one of the trusted hosts to the SafeWord server using the client. An analysis of the tcpdump output shows that the connection is allowed by the firewall:

```
12:46:35.910944 I 172.16.12.242.44149 > 172.16.3.12.5040: S
370052528:370052528(0) win 5840 <mss 1460,sackOK,timestamp
68543688[|tcp]> (DF) (ttl 64, id 27000)
12:46:35.913782 O 172.16.3.12.5040 > 172.16.9.242.44149: S
899793864:899793864(0) ack 370052529 win 5792 <mss
1460,sackOK,timestamp 249922366[|tcp]> (DF) (ttl 62, id 0)
12:46:35.913895 I 172.16.12.242.44149 > 172.16.3.12.5040: . ack 1 win
5840 <nop,nop,timestamp 68543688 249922366> (DF) (ttl 64, id 27001)
```

We then try to open a connection from a non-trusted host to verify the firewall does not allow the traffic through:

```
12:48:43.112591 I 172.16.12.100.44150 > 172.16.3.12.5040: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556408[|tcp]> (DF) [tos 0x10] (ttl 64, id 41328)
12:48:46.105807 I 172.16.12.100.44150 > 172.16.3.12.5040: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556708[|tcp]> (DF) [tos 0x10] (ttl 64, id 41329)
12:48:52.105812 I 172.16.12.100.44150 > 172.16.3.12.5040: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68557308[|tcp]> (DF) [tos 0x10] (ttl 64, id 41330)
```

To verify rule 16 we run a backup job from the backup server on the internal network and monitor traffic using tcpdump:

```
12:46:35.910944 I 172.16.9.13.4000 > 172.16.3.10.7937: S
370052528:370052528(0) win 5840 <mss 1460,sackOK,timestamp
68543688[|tcp]> (DF) (ttl 64, id 27000)
12:46:35.913782 O 172.16.3.10.7937 > 172.16.9.13.4000: S
899793864:899793864(0) ack 370052529 win 5792 <mss
1460,sackOK,timestamp 249922366[|tcp]> (DF) (ttl 62, id 0)
```

```
12:46:35.913895 I 172.16.9.13.4000 > 172.16.3.10.7937: . ack 1 win 5840
<nop,nop,timestamp 68543688 249922366> (DF) (ttl 64, id 27001)
```

We also perform a telnet attempt from a host that should not be allowed to the Networker service on one of the backup targets:

```
12:48:43.112591 I 172.16.12.100.44150 > 172.16.3.10.7937: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556408[|tcp]> (DF) [tos 0x10] (ttl 64, id 41328)
12:48:46.105807 I 172.16.12.100.44150 > 172.16.3.10.7937: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68556708[|tcp]> (DF) [tos 0x10] (ttl 64, id 41329)
12:48:52.105812 I 172.16.12.100.44150 > 172.16.3.10.7937: S
509654792:509654792(0) win 5840 <mss 1460,sackOK,timestamp
68557308[|tcp]> (DF) [tos 0x10] (ttl 64, id 41330)
```

Next, to verify rule 17, we use the VPN client to connect to the VPN concentrator, authenticate, and then attempt to telnet to mail server's SMTP port (i.e. 25/tcp):

```
telnet 172.16.9.19 25
Trying 172.16.9.19...
Connected to 172.16.9.19.
Escape character is '^]'.
220 exchg.giacfortunes.com ESMTP MS-Exchange 2000; Sat, 5 Oct 2002
13:40:25 -0500
```

During the verification of rule 17, we use tcpdump to watch traffic from the VPN concentrator to the SafeWord server for RADIUS authentication. This allows us to verify rule 18 allows the appropriate connections:

```
14:15:50.837546 O 172.16.4.254.1500 > 172.16.3.12.1645:  udp 58 (ttl
64, id 29377)
14:15:54.178199 I 172.16.3.12.1645 > 172.16.4.254.1500:  udp 58 (DF)
(ttl 254, id 5309)
```

An attempt to perform RADIUS authentication from a device that is not allowed to connect to the SafeWord server for 1645/udp shows the connection is not successful:

```
14:15:50.837546 O 172.16.8.254.1500 > 172.16.3.12.1645:  udp 58 (ttl
64, id 29377)
14:15:50.837546 O 172.16.8.254.1500 > 172.16.3.12.1645:  udp 58 (ttl
64, id 29378)
14:15:50.837546 O 172.16.8.254.1500 > 172.16.3.12.1645:  udp 58 (ttl
64, id 29379)
```

The final rule is the rule to drop any other traffic. This rule has been validated through the process of checking each rule for traffic that does not match an explicit pattern in a rule above. Any traffic that did not match a previous rule has actually been dropped by rule 20 as seen below:

Firewall Logging

A review of the GIAC firewall logs indicates that the Internet firewall provides substantial logging of activity. The following log excerpt indicates a sample of dropped traffic destined for the firewall:

```
Sep 30 00:11:52 inetfw01 kernel: INPUT-DROP-IN=eth0 OUT=
MAC=08:00:09:dc:fb:70:00:a0:c9:e9:54:1b:08:00 SRC=198.7.46.160
DST=198.7.46.132 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=39271 SEQ=33536
```

The Internet firewall also provides special logging intended to detect specific types of port scans as show in the following excerpt:

```
Sep 30 00:14:19 inetfw01 kernel: FINSCAN-IN=eth0 OUT=eth1
SRC=198.7.46.160 DST=198.7.46.200 LEN=40 TOS=0x00 PREC=0x00 TTL=56
ID=47878 PROTO=TCP SPT=57439 DPT=405 WINDOW=2048 RES=0x00 FIN URGP=0
```

The corporate firewall also provides adequate logging. Unfortunately, FireWall-1 only logs the first packet of a connection. In the case of the port scans, FireWall-1 provided different error messages based on the type of scan. For example, the ACK scan resulted in the alert message "TCP packet out of state." This is due to the fact that the ACK packet was sent out of order, where FireWall-1 was expecting a SYN packet. In the case of the FIN and XMAS scans, FireWall-1 reported the alert message "TCP flags do not make sense." It is believed that this is a result of invalid flag combinations set in each packet.

Both products provided sufficient logging of dropped and rejected traffic that attempted to pass through the firewalls as well. Again, Netfilter provided advanced logging information that surpassed FireWall-1's logging.

Recommendations and Results

To bring the GIAC IT infrastructure to fully acceptable levels of risk, the organization must be able to apply security to all levels of its infrastructure, practicing a defense in-depth approach to security. As such GIAC should conduct an assessment of its entire IT infrastructure. The Internet and corporate firewalls provide an acceptable level of protection against intrusion from the Internet as well as from internal sources. The audit of both firewalls' rulesets indicates that they are effective in enforcing network security policy. Both systems perform substantial logging of accepted and blocked traffic. The audit team did not uncover any rule configuration issues that allow unintended traffic. The audit did discover a few minor flaws with the Nokia firewall appliance used for the corporate firewall.

The following changes will improve the security profile of the GIAC infrastructure:

- Disable use of version 1.5 of the SSH protocol. Use of this version of the SSH protocol may needlessly expose communications to compromise.
- Apply the latest patch set to the Nokia operating system, IPSO, to eliminate the vulnerabilities in OpenSSL and mod_ssl. If Nokia is not

adequately responsive in addressing security vulnerabilities, consider an alternate platform for FireWall-1.

- Implement additional infrastructure to increase the availability of mission critical systems. The current IT infrastructure does not maintain redundant firewall infrastructure components or Internet connectivity. A failure in one device could cripple the organization's business.
- Implement an Incident Response Team (IRT) to respond to security events in real-time.
- Conduct a full IT security assessment that evaluates the security profile of the GIAC IT infrastructure and business applications. The organization must be able to apply security to all levels of its infrastructure, practicing a Defense in Depth approach to security.

Resolution of Issues

GIAC has taken several steps to address the issues raised in the audit. Where possible, the IT staff has made corrections to the firewall systems to resolve potential security issues.

SSH Protocol Version

GIAC has eliminated the use of SSH protocol version 1.5 on the Internet firewall. Unfortunately, the version of SSH shipped with the Nokia appliance does not provide an effective mechanism for permanently disabling this version of the protocol. In addition, GIAC's current backup process for the Nokia appliance relies on the use of SCP to migrate the nightly backup to another system. GIAC currently uses SCP due to the ability to script file transfers and the lack of SFTP on the Nokia platform.

Nokia Operating System

GIAC has applied the IPSO 3.6 FCS4, which addresses the OpenSSL and mod_ssl vulnerabilities. A quick check of the OpenSSL version at the command line indicates that the version is a sufficiently new enough release:

```
Corpfw01[admin]# openssl version
OpenSSL 0.9.6g 21 Sep 2002
```

The IT staff also conducts an internal test of the corporate firewall using Nessus to verify that the latest version of IPSO does not contain the vulnerability and that no new vulnerabilities have been introduced.

Continuous Availability of the Network Infrastructure

The GIAC IT staff has reviewed the need to meet growing availability expectations by increasing the redundancy of the network infrastructure. To address this need, the GIAC IT staff has implemented an additional T1 connection to the Internet. This connection is provided by the same ISP, but is provisioned through a separate Point of Presence (POP). This is done to minimize the likelihood of an outage at the ISP impacting GIAC's availability. In

addition, a separate telecommunications carrier is used to provision the actual circuit between the GIAC datacenter and the ISP.

GIAC has chosen to terminate this connection on a separate router and implement Cisco's Hot Standby Routing Protocol (HSRP). The use of HSRP allows GIAC to use several routers to appear as a single virtual router on the GIAC network. In the event of a failure with one router, all traffic will fail over to the second router.

GIAC then implements a second Nokia IP 530 as part of a high-availability cluster for FireWall-1. For this purpose, the Nokia IP 530's use the Virtual Router Redundancy Protocol (VRRP) to provide a functional equivalent of HSRP for the firewall appliance. FireWall-1 on both systems is configured to share state information. This allows a seamless fail-over of connections, including active connections, in the event a failure on the primary system.

GIAC also implements a basic high-availability cluster (also using VRRP) for the Linux Netfilter firewall. A second firewall is configured to function as a hot standby in the event of a system failure. Unfortunately, Netfilter does not currently support state table sharing between multiple systems. In the event of a system failure, the secondary will take over, but existing connections will be lost. If this becomes a serious issue, GIAC may examine alternate commercial products to provide a higher-level of availability.

Implement an Incident Response Team

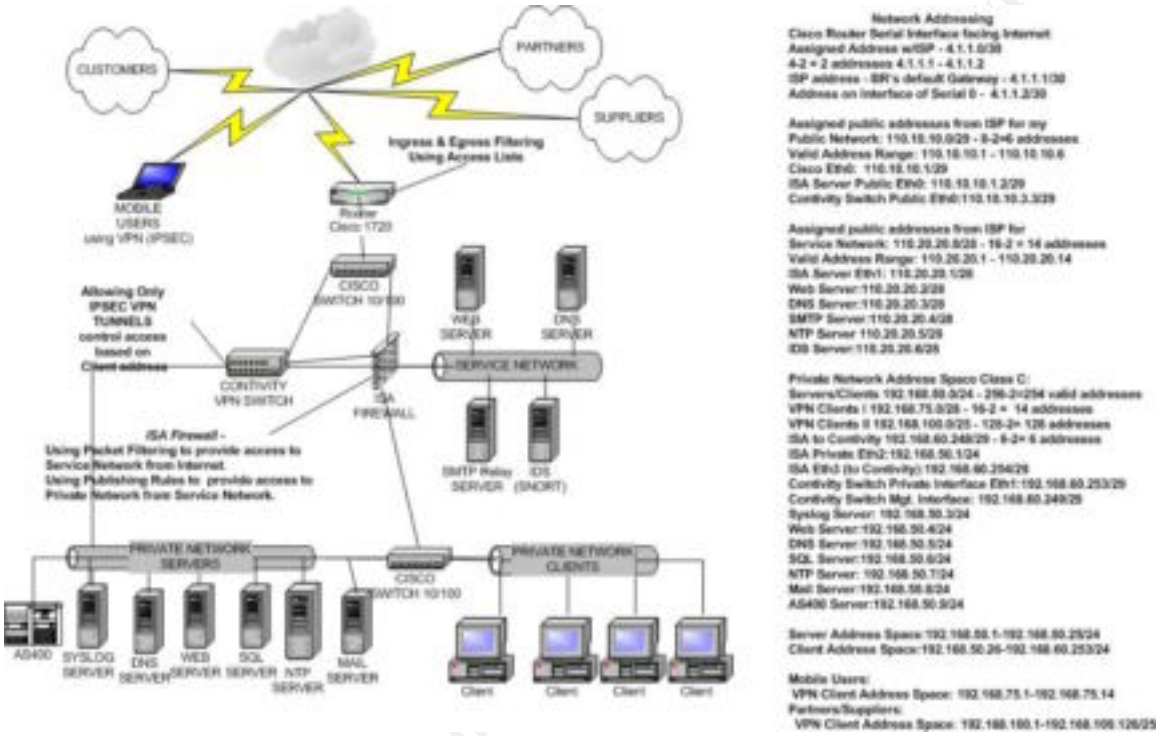
GIAC has implemented an IRT to handle security events. The IRT has developed a set of practices and a formalized procedure to respond to events. These procedures have been reviewed and approved by the GIAC management staff. On a monthly basis the IRT conducts a practice drill of its practices and procedures to test the effectiveness of its response.

Enterprise Wide IT Security Assessment

With the growing complexity in IT solutions, firewalls can no longer be relied upon as the sole mechanism for securing an IT infrastructure. GIAC must be able to apply security to all levels of its infrastructure, practicing a Defense in Depth approach to security. By doing so, security risks that cannot be protected against by a firewall or are the result of circumvention of a firewall, can still be defended against. GIAC has rigorously applied this philosophy to its entire IT infrastructure. To determine whether this approach has effectively mitigated risk, GIAC has contracted with the consulting company to return to perform an assessment of the entire IT infrastructure. This assessment will take place after the IT staff has had the opportunity to address the issues outlined above.

Assignment 4 – Design Under Fire

For the design under fire, we have chosen to conduct our exploits against Lloyd Ardoin's GCFW practical located at http://www.giac.org/practical/Lloyd_Ardoin_GCFW.zip. The architecture under fire is outlined below:



Network Discovery

To conduct a set of exploits against the architecture it is first necessary to perform a set of reconnaissance probes against the architecture. The first step in gathering information is to perform an nslookup on the IP address of the public Web server. We can also identify the DNS servers for the domain by setting the query type to name server records within nslookup (i.e. set type=ns). Finally, we can identify the mail servers by setting the query type to mail exchange records (i.e. set type=mx). With this information we can identify potential targets for attack. We also attempt zone transfers from the ISP as they are listed as maintaining a secondary DNS server in the whois information for the domain. It should be noted that the architecture appears to restrict zone transfers to only the ISP. If we are lucky, the ISP will not be as thorough.

We also use the American Registry for Internet Numbers (ARIN) to determine the range of addresses belonging to GIAC. We can anticipate one of two results; the address of one of the systems above is in an address block delegated to GIAC by their ISP. If this is the case, we can then use ARIN to determine the address range used by GIAC. The other possible outcome is that the address will belong to a block registered to the ISP. In this case we may still be able to derive some

information regarding GIAC's address space from ARIN. The ARIN whois database can be accessed at http://ws.arin.net/cgi-bin/whois.pl.

Once we've established a basic map of GIAC's address space, we run an nmap scan against each of these addresses with the –O flag to fingerprint the OS of each of these systems as well as the firewall. We also use the –sS flag to conduct a TCP SYN stealth scan to minimize the likelihood of detection by IDS. A second scan is run against the addresses using –sU flag to search for UDP ports that may be open. We then confirm the OS of the Web server by using telnet to connect to port 80 and provoke it to return an error page. This provides us with the type of Web server used; from which we may be able to abstract the operating system.

Initial Findings

The exterior firewall is a Windows 2000 Server, Service Pack 2, running Internet Security and Acceleration (ISA) server. We are able to run exploits against ISA and the operating system (especially vulnerabilities post Service Pack 2). The router does not adequately filter traffic. As can be noted by the final rule in the Ingress and Egress filtering, any traffic that has not matched a previous drop rule is allowed. This allows us to slowly probe the GIAC IP address space to identify nodes, fingerprint the OS of each node, and then conduct attacks against the node.

The external Web server is also a Windows 2000 Server and while not specified, we will assume that it is also running Service Pack 2. There is no mention of host hardening through the entire document and this is confirmed through the probing for the availability of common ISAPI filters within Internet Information Server. For the remainder of this discussion, the Web server will be treated as a non-hardened host protected by the firewall.

Attacking the Firewall

To research the possible attacks available against Internet Security and Accelerations server on Windows 2000, we review the vulnerabilities listed on several Web sites. We begin by searching the online database of Common Vulnerabilities and Exploits (CVE) for ISA vulnerabilities. We also run checks against the ISS X-Force database and the Security Focus database. Other sites are available including CERT; however, vulnerabilities typically are not released by CERT until the vendor has provided a patch. As such, CERT is not as effective when looking for potential exploits. The following URLs provide searchable indexes of vulnerabilities and bugs that may affect Microsoft products:

- CVE: http://www.cve.mitre.org/cve
- X-Force: http://www.iss.net/security_center/search.php
- Security Focus: http://online.securityfocus.com/search
- NTBugtraq: http://www.ntbugtraq.com

- Microsoft: http://support.microsoft.com/search

In preparing our attack profile, we identify potential attacks against ISA and continue to attack the system until an attack is successful.

Vulnerability A – User Enumeration through Null Sessions

The first vulnerability we would typically try to exploit is the anonymous connection to a Windows Null Session. The X-Force database has more information on the vulnerability at http://www.iss.net/security_center/static/171.php.

An anonymous null session allows a remote attacker to connect to the system without authentication and then perform a variety of functions including the enumeration of users, shares, or modification of the registry. We begin to conduct the attack through the enumeration of shares. We then attempt to brute force the password for the administrator account by remotely mounting a share such as C$. Once we have successfully identified the password for the administrator account we can modify the contents of any file on the system by remotely mounting the hidden shares for each drive. We can also modify the registry. Our goal is to ultimately gain remote control of the system through the installation of Back Orifice 2000. Back Orifice 2000 allows us to remotely shutdown the system, create network services, and capture keystrokes. Back Orifice 2000 can be found at http://sourceforge.net/projects/bo2k/.

An initial analysis of the router ruleset indicates that access to port 139/tcp, 139/udp, 445/tcp, and 445/udp are blocked for incoming packets. This effectively prevents our remote system from establishing an anonymous null session to the host. In addition, it prevents attempts to remotely mount a share. For these reasons, this attack is not expected to traverse the router and will not be successful.

Vulnerability B – UDP Flood Denial of Service

The second identified vulnerability for ISA server is the fragmented UDP packet flood denial of service attack listed as isa-udp-flood-dos (7446) within the X-Force database at http://www.iss.net/security_center/static/7446.php. Microsoft ISA server is vulnerable to an attack in which a high volume of fragmented UDP packets sent through the server can consume all of the CPU resources. By maintaining a sustained barrage of UDP fragments it would be possible to prevent the server from processing other packets.

This vulnerability has two distinct advantages. First, the server itself does not have to allow direct connections. By allowing the Internet to query the DNS server in the DMZ, the ISA server is in a position to be exploited by this vulnerability. The DNS server located in the DMZ is the target destination address for our attack. We can evade router filter rules by sending fragmented UDP packets to port 53 on this server. The router interprets these as legitimate DNS queries and allows them to pass. In addition, because the traffic used to create the attack is UDP, it is relatively easy to generate a high volume of traffic

with spoofed source addresses. This allows us to masquerade our activity as legitimate DNS traffic.

To implement this exploit, we require a tool that allows us to create fragmented UDP packets and to specify the source address, destination address, and destination port. To conduct this attack, we've chosen to modify the opentear exploit code. The original code was developed by RootShell and can be found at http://www.tamersahin.net/downloads/opentear.c. We've modified the exploit code to target only port 53/udp and to continuously randomize its source IP address. We then run the attack using the following command:

```
[root@attackhost DoS]# opentear2 110.20.20.3
```

A search of Microsoft's site does not uncover a patch for the vulnerability. It should also be noted that Ardoin's process for patching the system focuses on the hardening of the system prior to the installation of Internet Security and Acceleration Server. There is no mention of applying hot-fixes or Service Pack 1 for ISA after the installation. As such, there is a high-probability that the system is vulnerable to other attacks that have been corrected by Microsoft.

It should be noted that packet filtering of IP fragments is enabled within ISA, however, it is believed that the attack will still be successful. While the filtering of fragments presents them from actually being passed to the service network, the attack relies on forcing the ISA server into excess CPU utilization by forcing it to examine UDP fragments. As access to the target server is allowed through the router, it is possible to send a flood of fragmented UDP packets that appear to be DNS queries (53/udp) to the DNS server. Furthermore, by randomizing the source IP address of the traffic, it will be difficult to discern at the firewall what traffic is malicious in nature. As the forged UDP fragments are sent to the target, the ISA server will begin to consume more and more CPU resources until it ceases to process additional packets.

Attack Detection

The use of Snort on the service network substantially increases the probability that the event will be detected. As the opentear attack is based on the teardrop attack in which UDP fragments are sent to the target, it is possible that Snort will detect the event if the stock teardrop signature is used. This will occur only if the packet is passed by ISA to the service network. The Snort signature database indicates that the Snort teardrop signature checks for udp packets from an external network with the fragment flag set (Snort, Snort Signature Database – SID 270, p.1).

Unfortunately, the Snort sensor only has one network interface as is evident by the firewall rule specifically configured to allow the Snort sensor to send traffic to the syslog server (Ardoin, p. 75). While the Snort sensor may detect the traffic, it is unable to send alerts to the syslog server because the attack has impaired the firewall's ability to process packets.

Mitigation of Firewall Attack

There are three steps to take in mitigating the effectiveness of this attack. First and foremost, Internet Security and Acceleration Server Service Pack 1 and any available security hot-fixes from Microsoft should be applied to the system. To increase the effectiveness of intrusion detection, the Snort sensor should be configured with two network interfaces. The interface on the service network can be set to listen to traffic in stealth mode, without communicating on the network segment. The second interface would be attached to the private network and configured to use IP for communications to the syslog server. Finally, a traffic management tool could be implemented in front of the firewall on the Internet exposed network to monitor and regulate traffic.

Distributed Denial of Service Attack

To implement a distributed denial of service (DDoS) attack on the GIAC network from our collection of 50 compromised broadband systems, we will again begin by researching vulnerabilities in the various systems exposed to the Internet. Intended to have similar effects as denial of service attacks, distributed denial of service attacks function by flooding the target system or network with substantial volumes of traffic. Such an attack leads to network congestion and unresponsive access, ultimately leading to a denial of service for users. The primary point of distinction between distributed and non-distributed attacks is that distributed attacks are coordinated by multiple systems (Jeon, p.1). These systems are often compromised hosts, typically on broadband networks. Each of these systems is controlled by a central host system that initiates the attack by communicating specific commands to each compromised host or agent.

After careful review of the DDoS listed in Jeon's article, we have chosen to use Shaft to create a DDoS attack. While Trinity v3 supports a unique communication model between the handler and agents that makes detection and location of the handler more difficult, the tool is more commonly known. As a result, tools for identifying and isolating Trinity v3 have been developed (NIPC, p.1.) Shaft targets a computer by flooding it with UDP packets, SYN packets, or ICMP packets. The application consists of a client, handler, and agents. The client communicates with the handler system using telnet. The handler then directs agents to perform a variety of attacks against specified target IP addresses (Dietrich, p.2). Additional information on Shaft is available at http://security.royans.net/info/posts/bugtraq_ddos3.shtml.

To maximize the disruption of availability, we have decided to target the Web server. By targeting the availability of the Web server we are able to disrupt a majority of GIAC's business operations. To perform this attack, we use Shaft to send a SYN flood attack to port 80/tcp. In preparing for the attack we have compromised a collection of Solaris, Linux, and other Unix hosts around the Internet. These systems function as our agent systems. A modified Shaft agent that targets port 80/tcp has been set up on each system as part of a root kit and has been configured to communicate with our compromised Linux host which functions as the handler. When we are ready to begin the attack, we telnet to

port 20432/tcp on the handler, authenticate using our established credentials and then run the following command:

```
[root@attackhost DoS]# mdos 110.20.20.2 time 3600 type tcp
```

This command instructs the agents to attack 110.20.20.2 for 1 hour using a TCP SYN flood attack. The router and firewall pass this traffic because it appears to be legitimate Web traffic to the Web server. This attack should effectively cripple communications with the GIAC Web server. In this scenario CPU utilization is expected to increase until it reaches 100% at which point the IIS services, WWW Publishing Service and IIS Admin Service, will crash. As a result, once the DDoS attack has completed, the GIAC Web site will still be unavailable to GIAC's customers. It is possible that GIAC may be able to then restart the services to restore access, but is highly unlikely that this will be unsuccessful without a system reboot.

Attack Detection

Again, the use of Snort on the service network substantially increases the probability that the event will be detected. There are multiple Snort signatures for detecting Shaft based activity. These include communication between the client and the handler as well as between the handler and agents. These signatures are not designed for the detection of the incoming SYN flood. In fact, the only current Snort signature to detect Shaft SYN floods is for outbound traffic. In other words, for networks which have been compromised and now have a Shaft agent on a local system.

Mitigation of Denial of Service Attack

The prevention of distributed denial of service attacks is difficult; especially attacks that implement a SYN flood to impact service or system availability. Effectively counteracting SYN flood attacks is a balancing act. On one hand, if an organization aggressively configures its infrastructure to block SYN flood attacks; it risks blocking connections by legitimate users. On the other hand, if the organization is not aggressive enough, the result is a continued exposure to the attack. There are several steps GIAC can take to minimize the effectiveness of SYN flood attacks and to provide quicker detection of such incidents including:

- Additional intrusion detection at ISA firewall
- Modify the TCP/IP stack on the Web server
- Implement load balancing for Web services

Intrusion Detection at ISA Firewall

In addition to Snort, there are a couple of IDS systems that can be installed directly on the ISA firewall. Both ISS and GFI provide a plug-in for ISA that provides intrusion detection capabilities. The ISS RealSecure Server Sensor for ISA Server is designed to protect the ISA Server, and the networks behind the firewall. GIAC could install and configure RealSecure as a supplementary IDS system to detect and block SYN flood attacks. Additional information on

RealSecure Server Sensor for ISA Server can be found at http://www.iss.net/isaserver.

Modify the TCP/IP Stack on the Web Server

According to Microsoft's TechNet article Q315669 one can harden the TCP/IP stack against denial of service attacks by modifying values in the registry. The registry entries can shorten the time the host will wait on receiving a response to a SYN/ACK before timing out a connection attempt. The registry changes are found under: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services. The following values are taken from the TechNet article and are available in setting the value of the registry entry:

Item	Description
Value Name	SynAttackProtect
Key	Tcpip\Parameters
Value Type	REG_DWORD
Valid Range	0-2
Default Value	0

Each of the values implements increasing granularity and functionality in screening against SYN flood attacks. Additional information on the specific options and instructions on making the registry changes can be found on Microsoft's TechNet at http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q315669.

Implement Load Balancing for Web Services

There are several possible benefits to implementing load balancing for the Web server. First and foremost, by implementing load balancing it would be possible to add additional Web servers to support additional traffic to the system. This may assist in offsetting the impact of a SYN flood attack by distributing the load to multiple systems. In addition, one can achieve an additional layer of defense by using a Cisco CSS 11000 Web content switch to filter malicious traffic. The CSS 11000 series switches can defend against SYN flood attacks by blocking the flow of connections for any host that sends more than eight initial SYN packets with the same initial sequence number. In addition, any TCP connection attempt must return an ACK for the three-way handshake within 16 seconds or it will be purged from the connections table (Cisco, Web-Site Security and Denial-of-Service Protection, p. 4).

Host Attack

The Windows 2000 Web server at 110.20.20.2 was selected as the target for the host compromise. This system was selected for several reasons. First and foremost it is running Internet Information Server (IIS) version 5. This popular

Web server is subject to several different vulnerabilities. Second, this Web server provides customer access to GIAC's fortunes. It probably has connectivity to the back-end database, which may house intellectual property or customer data. Finally, as a Web server, the system receives substantial volumes of traffic that may make it possible for us to evade IDS by blending in with legitimate network traffic.

To research the possible attacks available against Windows 2000, we begin by searching the CVE and X-Force databases. We also check the Security Focus database and the NTBugtraq archives for possible vulnerabilities and exploit code.

Vulnerability – IIS ASP Chunked Encoding Buffer Overflow

The vulnerability we attempt to exploit is CVE candidate CAN-2002-0079. The X-Force database also refers to this exploit as the iis-asp-chunked-encoding-bo (8795). Additional information on the vulnerability can be found at http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0079 and http://www.iss.net/security_center/static/8795.php.

Internet Information Server 4.0 and 5.0 are both vulnerable to a buffer overflow within the Internet Services Application Programming Interface (ISAPI). The vulnerability lies within the chunked encoding data transfer mechanism. This transfer mechanism allows a client system to generate data, also known as a "chunk", notify the Web server of the size of the data prior to its transfer, and then transfer the data (ISS X-Force, ISS X-Force Database: iis-asp-chunked-encoding-bo, p.1). Since the client tells the server the size of the buffer, by telling it the size of the data, it is possible for a malicious client to lie about the size of the data before it is transferred. As a result, the Web server may allocate too small a buffer for the receipt of the data, which can lead to a buffer overflow when the data is sent. By overflowing the buffer, a malicious user may be able to execute code on the system, thus running commands or carrying out sets of functions designed to further compromise the system.

Buffer overflow attacks are significantly complex to carry out successfully. First and foremost, one must have knowledge of the size of the buffer to be able to successfully exceed the size of the buffer, while formatting the data at the point where the remaining code passed to the system is actually a valid command set. Second, the results of a successful overflow attack take time to manifest. In many cases, one set of commands may not be enough to accomplish the desired end result. In these cases, the attacker must incrementally accomplish tasks on the system without any knowledge of the success of the previous attacks. In the case of this attack, we have two advantages that make this buffer overflow exploit easier to carry out. First, the user defines the size of the buffer. This gives us distinct knowledge and control over the size of the buffer and the necessary code sizes to carry out the exploit. Second, the system provides Web services to the Internet. This allows us to view the results of an attack via the Web site.

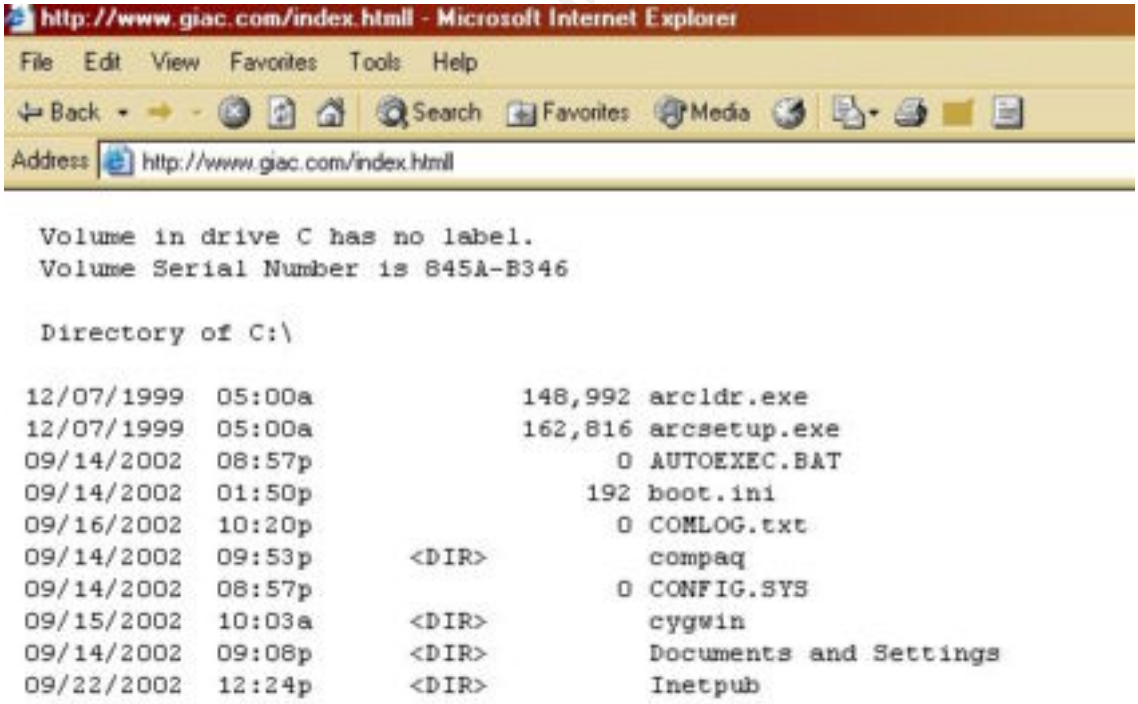
To begin this attack we craft a test of the exploit that allows us to validate the possible success of this attack against this system, allows us to determine the root directory of the Web site, and slowly indexes the file system layout on the server. Our test exploit attempts to perform a directory listing of C:\ and save the contents of this output as a Web page in the root directory of the public Web site. We are then able to view the directory listing using a Web browser. By default, IIS creates a directory on the primary drive that it stores the default Web site under. In most cases this is located at C:\InetPub\wwwroot. The test exploit attempts to execute the following command on the system:

```
dir c:\ /A /N /S  > C:\InetPub\wwwroot\index.htmll
```

The /A flag displays file attributes, the /N flag displays the listing using long file names, and the /S flag descends the directory tree to display all subdirectories. In the event the root directory of the Web site is not located on the primary partition, we increment the drive letter progressively to determine if we accomplish any alternate results. If this does not work, we can attempt to craft code that retrieves the location of the directory from the registry by pulling the value of PathWWWRoot and setting this value to a variable named WROOT:

```
set WROOT=<value from registry>
dir c:\ /A /N /S  > %WROOT%\index.htmll
```

The following screenshot shows the results of our test exploit:



Once the test exploit is successful, we then attempt to download a root kit from another compromised system on the Internet (i.e. a broadband user running a Web site). This makes it more difficult to track the source of the toolkit. As the

Web server is allowed to make connections to port 80/tcp to systems on the Internet, it should be easy to download the root kit and install each tool. The root kit tools were initially acquired through the Web site, http://neworder.box.sk/codebox.links.php?&key=hackfav, and include:

Tool	Purpose
ActiveState Perl	Win32 Perl programming language to assist in the automation of complex attacks by running advanced exploit scripts locally on the system.
L0phtcrack	Password cracking tool for cracking passwords on the local system as well as other systems. This will allow us to offload the cracking of Windows passwords from other hosts to this powerful system.
GNU Privacy Guard	Encryption tool that will allow us to encrypt the contents of data we acquire from this system and others.
PWDUMP2	Dumps password hashes from the SAM database, to be used in conjunction with L0phtcrack to grind passwords.
SQLAT	Toolset intended for penetration testing of MS SQL Server. Useful for attacking the internal MS SQL server the Web server has access to.
SQLPing	Tool used to perform reconnaissance of MS SQL Servers. Useful for profiling the internal SQL server.
FakeGINA	Intercepts communications between Winlogon and the normal GINA, allowing us to capture all successful login attempts and write them to a file.
Netcat	Tool that allows us to read and write data over a network socket via TCP or UDP.
NTDaddy	Active Server Page that supports remote file execution, file modifications, and other features that will allow us to modify the system.
File Protector	Tool to hide files and protect them from detection.
VNC	VNC allows a user to remotely view the remote desktop of the system via a VNC Viewer client or a Web browser. For our purposes, it will be configured to listen on port 443 when we start it manually.

Using these tools we attempt to compromise the local password database, audit the internal SQL server, and conduct attacks against other systems on the service network as well as the internal Microsoft SQL server. The NTDaddy ASP

code allows us to execute SQL attacks against the internal SQL server and immediately see the results of these attacks on the Web page. We also use the processing power of the system to crack the passwords of other compromises systems.

An initial analysis of the router and firewall rulesets indicates that access to the Web server to conduct the initial attack will be successful. In addition, because the server is allowed to initiate connections out to the Internet, it is possible to download the root kit. The ISA firewall also allows the system to connect to the MS SQL server on port 1433/tcp. This allows the compromised Web server to attack the SQL server, eventually leading to the compromise of the database, and the acquisition of any data retained within the database. Through the connection between the Web server and the SQL server it may also be possible to create SQL stored procedures that could be used to further compromise data outside the database system.

Attack Detection

The use of Snort on the service network substantially increases the probability that the event will be detected. There are no signatures designed to detect the initial buffer overflow attack. As such, this attack is expected to go undetected. With the vast array of Snort signatures for MS-SQL server, it is likely that the attacks against the SQL Server will be detected, assuming that we do not compromise the Snort sensor before conducting our attacks on the MS SQL Server.

Mitigation of Host Attack

There are many reasons this attack is dangerous. First and foremost, there is no adequate segmentation of the network. The compromise of the Web server allows for attacks to be carried out against other systems on the same local network. In addition, the Snort sensor is accessible from the Web server. The Snort sensor should be configured with two network interfaces. The interface on the service network can be set to listen to traffic in stealth mode, without communicating on the network segment. The second interface would be attached to the private network and configured to use IP for communications to the syslog server. The use of the Web server to compromise the SQL server could then use the SQL server to compromise other internal systems. The MS SQL server should be isolated from the internal network.

Second, this specific buffer overflow could be prevented through the installation of the Microsoft hot-fix for this vulnerability. This would only prevent this specific overflow attack. When a new attack is discovered, the system could be compromised again. It may be advantageous to switch to an alternate Web server that has fewer vulnerabilities and releases patches in a timelier manner. GIAC then must maintain the current patch level on the system to reduce the overall time of exposure.

Third, by allowing connections from the Web server out to the Internet on port 80/tcp and 443/tcp, it is possible to send and retrieve virtually any data from the

Internet. By implementing either a stateful packet filter or application proxy firewall, it would be possible to allow inbound connections from the Internet to port 80/tcp and 443/tcp without allowing connections to be initiated back out to the Internet for any services. This would hamper the download of the root kit and the compromise of other systems.

References

Books & White Papers

SANS Institute. Perimeter Protection: Defense In-Depth. Bethesda, MD: SANS Press, 2001.

SANS Institute. Perimeter Protection: Firewall Technology, Firewall-1 and PIX. Bethesda, MD: SANS Press, 2001.

SANS Institute. Perimeter Protection: Firewall Technology, Netfilter and Gauntlet. Bethesda, MD: SANS Press, 2001.

SANS Institute. Securing Linux Step-By-Step, Version 1.0. Bethesda, MD: SANS Press, 2000.

Zwicky, Elizabeth D., Cooper, Simon, and Chapman, Brent D. Building Internet Firewalls, 2nd Edition. Cambridge, MA: O'Reilly & Associates, Inc., June 2000.

Requests For Comments

Callas, J., Donnerhacke, L., Finney, H., and Thayer, R. "OpenPGP Message Format." RFC 2440. November 1998. URL: http://www.ietf.org/rfc/rfc2440.txt (22 August 2002).

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and Lear, E. "Address Allocation for Private Internets." RFC 1918. February 1996. URL: http://www.ietf.org/rfc/rfc1918.txt (22 August 2002).

Sollins, K. "The TFTP Protocol (Revision 2)." RFC 1350. July 1992. URL: http://www.ietf.org/rfc/rfc1350.txt (4 September 2002).

Web Sites

American Registry for Internet Numbers. "Output from Whois." URL: http://ws.arin.net/cgi-bin/whois.pl (19 September 2002).

Ardoin, Lloyd. "GCFW Practical v.1.7." 30 June 2006. URL: http://www.giac.org/practical/Lloyd_Ardoin_GCFW.zip (10 September 2002).

Atkins, Todd. "SWATCH: The Simple WATCHer." 8 November 2001. URL: http://www.oit.ucsb.edu/~eta/swatch (22 August 2002).

Bastille Linux. "Bastille Linux." URL: http://www.bastille-linux.org (22 August 2002).

Bandle, David A. "Taming the Wild Netfilter." 1 September 2001. URL: http://www.linuxjournal.com/article.php?sid=4815 (4 September 2002).

Briddle, Matt. "GCFW Practical v.1.6a." 29 April 2002. URL: http://www.giac.org/practical/Matt_Briddell_GCFW.zip (15 September 2002).

Check Point Software Technologies. "Common Ports Used by Check Point Next Generation (NG)." 30 January 2002. URL: http://support.checkpoint.com/public/idsearch.jsp?id=sk9408&QueryText=(%22C

ommon+ports+used+by+Check+Point+Next+Generation+%22)& (22 August 2002).

Cisco Systems. "Cisco 3600 Series Modular, High-Density Access Routers." 22 August 2002. URL: http://www.cisco.com/univercd/cc/td/doc/pcat/3600.htm (22 August 2002).

Cisco Systems. "Configuration Information for an Administrator." VPN Client Administrator Guide, Release 3.6. 8 August 2002. URL: http://www.cisco.com/univercd/cc/td/doc/product/vpn/client/3_6/admin_gd/vcach1.pdf (22 August 2002).

Cisco Systems. "Increasing Security on IP Networks." 26 April 2002. URL: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs003.htm (4 September 2002).

Cisco Systems. "Tunneling Protocols." VPN 3000 Series Concentrator Reference Volume I: Configuration. 8 August 2002. URL: http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_6/config/tunnel.pdf (22 August 2002).

Cisco Systems. "User Management." VPN 3000 Series Concentrator Reference Volume I: Configuration. 8 August 2002. URL: http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_6/config/usermgt.pdf (22 August 2002).

Cisco Systems. "Web-site Security and Denial-of-Service Protection." 29 November 2000. URL: http://www.cisco.com/warp/public/cc/pd/si/11000/prodlit/cswsc_wi.htm (19 September 2002).

CVE. "CVE-2001-0546." 9 March 2002. URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0546. (20 September 2002).

CVE. "CVE-2002-0079." 2 May 2002. URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0079 (20 September 2002).

Dietrich, Sven. "Shaft DDoS Tool Analysis." 16 March 2000. URL: http://security.royans.net/info/posts/bugtraq_ddos3.shtml (19 September 2002).

Fish.com "Titan 4.0 Beta5." 1 August 2002. URL: http://www.fish.com/titan/TITAN_documentation.html (22 August 2002).

Galik, Captain Dan. "Defense in Depth: Security for Network-Centric Warfare." April 1998. URL: http://www.chips.navy.mil/archives/98_apr/Galik.htm (22 August 2002).

IANA. "Internet Protocol V4 Address Space." 6 August 2002. URL: http://www.iana.org/assignments/ipv4-address-space (4 September 2002).

ISS. "RealSecure Server Sensor 6.5 for ISA Server." URL: http://www.iss.net/isaserver (19 September 2002).

ISS X-Force. "ISS X-Force Database: isa-udp-flood-dos." 2 November 2001. URL: http://www.iss.net/security_center/static/171.php (20 September 2002).

ISS X-Force. "ISS X-Force Database: iis-asp-chunked-endcoding-bo." 10 April 2002. URL: http://www.iss.net/security_center/static/8795.php (20 September 2002).

ISS X-Force. "ISS X-Force Database: nt-usernull." URL: http://www.iss.net/security_center/static/171.php (20 September 2002).

Jeon, DeokJo. "Understanding DDOS Attack, Tools, and Free Anti-tools with Recommendation." 7 April 2001. URL: http://rr.sans.org/threats/understanding_ddos.php (19 September 2002).

JNT Association. "Cisco ACL Example." 11 January 1999. URL: http://www.ja.net/CERT/JANET-CERT/prevention/cisco/cisco_acls.html (4 September 2002).

Lawson, Craig. "Hardening in the Enterprise: Always an After Thought?" 9 April 2001. URL: http://rr.sans.org/sysadmin/hard_enterprise.php (22 August 2002).

Legato. "354: The Use of Firewalls with Networker Server Release 5.5 and Later." Legato Technical Bulletin. 22 December 1999. URL: http://www.legato.com/resources/bulletins/354.html (22 August 2002).

National Institute Protection Center. "Trinity v3/Stracheldraht 1.666 Distributed Denial of Service Tool." 13 October 2000. URL: http://www.nipc.gov/warnings/advisories/2000/00-055.htm (19 September 2002).

Netfilter. "Netfilter/Iptables." 26 August 2002. URL: http://www.netfilter.org (4 September 2002).

New Order. "New Order – Computer Security and Networking Portal. " URL: http://neworder.box.sk/codebox.links.php?&key=hackfav (20 September 2002).

Nokia. "Nokia IP530." 22 August 2002. URL: http://www.nokia.com/securitysolutions/platforms/530.html (22 August 2002).

OpenBSD. "OpenSSH." 1 September 2002. URL: http://www.openssh.org (3 September 2002).

Engelschall, Ralf S. "OpenSSL." 1999. URL: http://www.openssl.org (3 September 2002).

"Public NTP Servers (stratum 2) Time Servers." 12 August 2002. URL: http://www.eecis.udel.edu/~mills/ntp/clock2.htm (22 August 2002).

RootShell. "opentear.c." URL: http://www.tamersahin.net/downloads/opentear.c (19 September 2002).

SANS Institute. "GIAC: Global Information Assurance Certification - GIAC Certified Firewall Analyst (GCFW) Practical Assignment." Version: 1.7. 8 April 2002. URL: http://www.giac.org/GCFW_assignment.php (22 August 2002).

Snort. "Snort: The Open Source Network Intrusion Detection System." 29 August 2002. URL: http://www.snort.org (30 August 2002).

Snort. "Snort Signature Database – SID 270." 20 September 2002. URL: http://www.snort.org/snort-db/sid.html?sid=270 (20 September 2002).

29 August 2002. URL: http://www.snort.org (30 August 2002).

Squid-cache.org. "Squid Web Proxy Cache." 15 August 2002. URL: http://www.squid-cache.org (22 August 2002).

Welch-Abernathy, Dameon D. "Firewall-1 and FreeBSD." Firewall-1 FAQ. 18 January 2002. URL: http://www.phoneboy.com/faq/0341.html (22 August 2002).

Winters, Scott. "Securing the Perimeter with Cisco IOS 12 Routers." 15 August 2000. URL: http://rr.sans.org/firewall/blocking_cisco.php (4 September 2002).

Appendix A – Project Requirements

The following information in this appendix is taken from the GIAC Certified Firewall Analyst (GCFW) Practical Assignment, version 1.7 and is intended to provide a context for understanding the requirements of the paper. The requirements can also be viewed online at the SANS Institutes GIAC Web site:

http://www.giac.org/GCFW_assignment.php

Basic Requirements

This assignment consists of four related parts. Please check your spelling and read through your wording! This is how the world will see you; you will not be allowed to "clean up" your paper once it has been submitted. You will be graded primarily on the accuracy and educational value of your submission, but appearance also counts.

Your completed submission (all four assignments) should be a minimum of 20 pages long, and should also meet all of the other requirements listed in the Administrivia under "Formatting and Minimum Length". Your work should include diagrams, screen shots, code examples, references, and/or appendices as appropriate.

Note that many students get bogged down with putting a lot of extraneous information into the assignment. In their efforts to write a comprehensive paper, they leave out the specific items that are asked for in the assignments below. Focus on the required items listed in the assignment, and then work on any additional information you want to include.

Assignment 1 – Security Architecture (15 points)

Define a network security architecture for GIAC Enterprises, an e-business which deals in the online sale of fortune cookie sayings.

Your architecture must consider access requirements (and restrictions) for:

- Customers (Companies or individuals that purchase bulk online fortunes)
- Suppliers (Companies that supply GIAC Enterprises with their fortune cookie sayings)
- Partners (International companies that translate and resell fortunes)
- GIAC Enterprises employees located on GIAC Enterprise's internal network
- GIAC Enterprises mobile sales force and teleworkers

You must explicitly define how the business operations of GIAC Enterprises will take place. How will each of the groups listed above connect to or communicate with GIAC Enterprises? How will GIAC Enterprises employees access the outside world? What services, protocols, or applications will be used?

Defining access requirements and the reasoning for those requirements is critical to this assignment. If you have not thought through how this access will take place, you will not be able to adequately define your security policy and ACLs/rulesets later in the paper.

In designing your architecture, you must include the following components:

- Filtering Router(s)
- Firewall(s)
- VPN(s)

Your architecture may also include the following optional components if they are appropriate to your design:

- Internal firewalls (Are internal firewalls appropriate for additional layered protection; to segment internal networks…?)
- Additional secure remote access (Is additional remote access – other than the VPN – required by administrators, salespeople, telecommuters…?).
- Intrusion detection systems

You must include a diagram or set of diagrams that shows the layout of GIAC Enterprise's network and the location of each component listed above. You must provide the specific brand and version of each perimeter defense component used in your design. Finally, include an explanation that describes the purpose of each component, the security function or role it carries out, and how the placement of each component on the network allows it to fulfill this role.

The network can be as complex or as simple as you like as long as it meets the functional requirements that you define according to the guidelines given above. The important thing is not how elaborate your network is, but that your design actually works.

You must justify the appropriateness of your design. Is it both technically reasonable and financially feasible? Are you building a $1000 fence to contain a $100 horse? You may provide a cost or bill of materials if you wish.

Assignment 2 – Security Policy and Tutorial (35 points)

Based on the security architecture that you defined in Assignment 1, provide a security policy for the following three components:

- Border Router(s)
- Primary Firewall(s)
- VPN(s)

You may optionally include policy for other devices (i.e., - internal firewalls).

By "policy" we mean the specific set of ACLs, ruleset, or IPSec policy for that device – **not** corporate or organizational policy (though note that organizational policy may dictate the specific ACLs or ruleset in effect).

For each component, be sure to consider the access requirements for customers, suppliers, partners, remote users, and internal users that you defined in Assignment 1. The policies you define must accurately reflect those business needs as well as appropriate security considerations.

You must include the complete policy (meaning explicit ACLs, Ruleset, IPSec policy, etc.) in your paper. It is not enough to simply state, "I would include ingress and egress filtering…" The policies may be included in an Appendix if doing so will help the "flow" of the paper (clearly state if this is the case).

For each rule in all policies, you must include the general purpose of the rule and why it is important.

You must also include a discussion of the order of the rules, and why order is (or is not) important.

For **one** of the three security policies defined above, you must incorporate a tutorial on how to implement the policy. Clearly separate and label your tutorial. Use screen shots, network traffic traces, firewall log information, and/or URLs to find further information to clarify your instructions. Be certain to include a general explanation of the syntax or format of the ACL, filter, or rule for your device, as well as a general explanation of how to apply a given ACL, filter, or rule.

Be certain to point out any tips, tricks, or potential problems.

Assignment 3 – Verify the Firewall Policy (25 points)

You have been asked to conduct a technical audit of GIAC's primary firewall in order to verify that the policies are correctly enforced as described in Assignments 1 and 2. To conduct the audit, you will need to:

- Plan the audit.
    - Describe the technical approach you will use to assess the firewall.
    - Be certain to include considerations such as what shift or day you would do the assessment.
    - Estimate costs and level of effort.
    - Identify risks and considerations and how they are addressed.
- Using the approach you described conduct the audit.
    - Demonstrate how you validated that the primary firewall is actually implementing GIAC Enterprise's security policy.
    - Be certain to include the tools and commands used. Include screen shots in your report if possible.
- Evaluate the audit. Based on your assessment (and referring to data from your assessment):
    - Provide an analysis of the audit results.

- Make recommendations for improvements or alternate architectures.
- **Supportive diagrams are strongly recommended for this part of the assignment.**

Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool you choose, but you must annotate/explain the output.

Assignment 4 – Design Under Fire (25 points)

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any GCFW practical posted in the previous **6 months** and paste the graphic into your submission. Be certain to list the URL of the practical you are using.

Research and design the following three types of attacks against the architecture:

1. An attack against the firewall itself.
   - Research and describe a vulnerability that has been found for the type of firewall chosen for the design.
   - Design an attack based on the vulnerability.
   - Explain the results of running that attack against the firewall.
2. A denial of service attack.
   - Subject the design to an attack from 50 compromised cable modem/DSL systems.
   - Describe the countermeasures that can be put into place to mitigate the attack that you chose.
3. An attack plan to compromise an internal system through the perimeter system.
   - Select a target and explain your reasons for choosing that target.
   - Describe the process to compromise the target.

Your attack information should be detailed – include the specifics of how the attack would be carried out. Do not simply say, "I would exploit the vulnerability described in Vendor Security Bulletin XXX". What commands would you use to carry out the attack? Are exploit tools or scripts available on the Internet? What additional steps would you need to take prior to conducting the attack (reconnaissance, determining internal network layout, determining valid account name…)? Would any of your methods be noticed (log files, IDS…)? What

"stealth" techniques could you employ to avoid detection? What countermeasures would help prevent your attack from succeeding?

If it is possible to carry out the attack on a test system, include screen shots; log files, etc. as appropriate to illustrate your methods.

In designing your attacks, keep the following in mind:

- The attack should be **realistic.** The purpose of this exercise is for the student to clearly demonstrate that they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks.
- The attack should be **reasonable.** The firewall does not necessarily have to be impenetrable (perfectly configured with all of the up-to-the-minute patches installed). However, you should **not** assume that it is an unpatched, out-of-the-box firewall installed on an unpatched out-of-the-box OS. (Remember, you designed GIAC Enterprises' firewall; would you install a system like that?)
- You must supply documentation (e.g., a URL to the security bulletin, bugtraq archive, or exploit code used) for any vulnerability you use in your attack.

The attack does not necessarily have to succeed. If, given the perimeter and network configuration you have described above, the attack would fail; you can describe this result as well.

Appendix B – Supplements

OS Hardening

Red Hat 7.2 Package Selection

The following list of packages are the base packages installed as part of the initial operating system build before. Please note that the versions indicated are the versions prior to patching (i.e. via up2date):

```
indexhtml-7.2-1
redhat-logos-1.1.3-1
filesystem-2.1.6-2
glibc-2.2.4-13
bzip2-libs-1.0.1-4
cracklib-2.7-12
db2-2.4.14-7
dosfstools-2.7-1
eject-2.0.9-2
gdbm-1.8.0-10
hdparm-4.1-2
ksymoops-2.4.1-1
mailx-8.1.1-22
mktemp-1.5-11
parted-1.4.16-8
perl-5.6.0-17
pwdb-0.61.1-3
setserial-2.17-4
slang-1.4.4-4
netconfig-0.8.11-7
setuptool-1.8-2
termcap-11.0.1-10
bash-2.05-8
crontabs-1.10-1
iproute-2.2.4-14
groff-1.17.2-3
ncurses-5.2-12
cpio-2.4.2-23
ed-0.2-21
at-3.1.8-20
gawk-3.1.0-3
ash-0.3.7-2
grub-0.90-11
less-358-21
openssl-0.9.6b-8
procps-2.0.7-11
raidtools-0.90-23
redhat-release-7.2-1
sed-3.02-10
kbdconfig-1.9.14-1
sysklogd-1.4.1-4
tcsh-6.10-6
dev-3.2-5
mkinitrd-3.2.6-1
time-1.7-14
setup-2.5.7-1
basesystem-7.0-2
bdflush-1.5-17
chkconfig-1.2.24-1
db1-1.85-7
db3-3.2.9-4
e2fsprogs-1.23-2
file-3.35-2
glib-1.2.10-5
iputils-20001110-6
losetup-2.11g-5
mingetty-0.9.4-18
net-tools-1.60-3
pcre-3.4-2
popt-1.6.3-1.03
reiserfs-utils-3.x.0j-2
shadow-utils-20000902-4
newt-0.50.33-1
ntsysv-1.2.24-1
syslinux-1.52-2
libtermcap-2.0.8-28
bzip2-1.0.1-4
hotplug-2001_04_24-11
libstdc++-2.96-98
logrotate-3.5.9-1
modutils-2.4.6-4
info-4.0b-3
diffutils-2.7.2-2
fileutils-4.1-4
findutils-4.1.7-1
grep-2.4.2-7
gzip-1.3-15
man-1.5i2-6
procmail-3.21-1
psmisc-20.1-2
readline-4.2-2
rootfiles-7.2-1
console-tools-19990829-36
slocate-2.6-1
tar-1.13.19-6
textutils-2.0.14-2
mount-2.11g-5
lilo-21.4.4-14
mouseconfig-4.23-1
tmpwatch-2.8-2
```

```
utempter-0.5.2-6                    vim-common-5.8-7
vim-minimal-5.8-7                   which-2.12-3
words-2-17                          cracklib-dicts-2.7-12
pam-0.75-14                         authconfig-4.1.19-1
cyrus-sasl-1.5.24-20                cyrus-sasl-md5-1.5.24-20
cyrus-sasl-plain-1.5.24-20          gpm-1.19.3-20
kudzu-0.99.23-1                     passwd-0.64.1-7
sh-utils-2.0.11-5                   krb5-libs-1.2.2-13
openldap-2.0.11-13                  sendmail-8.11.6-3
SysVinit-2.78-19                    zlib-1.1.3-24
rpm-4.0.3-1.03                      util-linux-2.11f-9
initscripts-6.40-1                  apmd-3.0final-34
ipchains-1.3.10-10                  iptables-1.2.3-1
kernel-2.4.7-10                     lokkit-0.50-6
pciutils-2.1.8-23                   quota-3.01pre9-3
timeconfig-3.2.2-1                  vixie-cron-3.0.1-63
anacron-2.3-17                      gmp-3.1.1-4
m4-1.4.1-5                          python-1.5.2-35
libtool-libs-1.4-8                  pspell-0.12.2-3
aspell-0.33.7-1                     make-3.79.1-8
xinetd-2.3.3-1                      gnupg-1.0.6-3
autofs-3.1.7-21                     logwatch-2.1.1-3
krbafs-1.0.9-2                      nmap-2.54BETA22-3
micq-0.4.6.p1-2                     nss_ldap-172-2
openldap-clients-2.0.11-13          openssh-2.9p2-7
openssh-clients-2.9p2-7             pam_krb5-1.46-1
pidentd-3.0.14-1                    stunnel-3.19-1
rmt-0.4b22-6                        tcp_wrappers-7.6-19
rhn_register-2.7.2-7.x.2            traceroute-1.4a12-1
sendmail-cf-8.11.6-3                wget-1.7-3
up2date-2.7.2-7.x.6                 fetchmail-5.9.0-1
whois-1.0.9-1                       mutt-1.2.5i-17
links-0.96-2                        pine-4.33-15
nmh-1.0.4-9                         metamail-2.7-28
sharutils-4.2.1-8                   openssh-server-2.9p2-7
glibc-common-2.2.4-13               sysstat-4.0.1-2
mailcap-2.1.6-1
```

Bastille Configuration File

The following is a sample configuration file used with Bastille on Linux systems:

```
# Q: Would you like to set more restrictive permissions on the
administration utilities? [N]
FilePermissions.generalperms_1_1="Y"
# Q: Would you like to disable SUID status for mount/umount?
FilePermissions.suidmount="Y"
# Q: Would you like to disable SUID status for ping? [Y]
FilePermissions.suidping="Y"
# Q: Would you like to disable SUID status for at? [Y]
FilePermissions.suidat="Y"
# Q: Would you like to disable the r-tools? [Y]
FilePermissions.suidrtool="Y"
# Q: Would you like to disable SUID status for usernetctl? [Y]
FilePermissions.suidusernetctl="Y"
# Q: Would you like to disable SUID status for traceroute? [Y]
FilePermissions.suidtrace="Y"
```

```
# Q: Would you like to disable SUID status for Xwrapper? [N]
FilePermissions.suidXwrapper="Y"
# Q: Should Bastille disable clear-text r-protocols that use IP-based
authentication? [Y]
AccountSecurity.protectrhost="Y"
# Q: Would you like to enforce password aging? [Y]
AccountSecurity.passwdage="Y"
# Q: Would you like to restrict the use of cron to administrative
accounts? [Y]
AccountSecurity.cronuser="Y"
# Q: Do you want to set a default umask? [Y]
AccountSecurity.umaskyn="Y"
# Q: What umask would you like to set for users on the system? [077]
AccountSecurity.umask="077"
# Q: Should we disallow root login on tty's 1-6? [N]
AccountSecurity.rootttylogins="N"
# Q: Would you like to password-protect the GRUB prompt? [N]
BootSecurity.protectgrub="N"
# Q: Would you like to disable CTRL-ALT-DELETE rebooting? [N]
BootSecurity.secureinittab="Y"
# Q: Would you like to password protect single-user mode? [Y]
BootSecurity.passsum="Y"
# Q: Would you like to set a default-deny on TCP Wrappers and xinetd?
[N]
SecureInetd.tcpd_default_deny="Y"
# Q: Should Bastille ensure the telnet service does not run on this
system? [y]
SecureInetd.deactivate_telnet="Y"
# Q: Should Bastille ensure the FTP service does not run on this
system? [y]
SecureInetd.deactivate_ftp="Y"
# Q: Would you like to display "Authorized Use" messages at log-in
time? [Y]
SecureInetd.banners="Y"
# Q: Who is responsible for granting authorization to use this machine?
SecureInetd.owner="its owner"
# Q: Would you like to disable the gcc compiler? [N]
DisableUserTools.compiler="Y"
# Q: Would you like to put limits on system resource usage? [N]
ConfigureMiscPAM.limitsconf="N"
# Q: Should we restrict console access to a small group of user
accounts? [N]
ConfigureMiscPAM.consolelogin="Y"
# Q: Which accounts should be able to login at console? [root]
ConfigureMiscPAM.consolelogin_accounts="root a09960"
# Q: Would you like to add additional logging? [Y]
Logging.morelogging="Y"
# Q: Do you have a remote logging host? [N]
Logging.remotelog="Y"
# Q: What is the IP address of the machine you want to log to?
[127.0.0.1]
Logging.remotelog_host="172.16.3.10"
# Q: Would you like to disable apmd? [Y]
MiscellaneousDaemons.apmd="Y"
# Q: Would you like to disable GPM? [Y]
MiscellaneousDaemons.gpm="Y"
# Q: Would you like to deactivate NIS server programs? [Y]
```

```
MiscellaneousDaemons.nis_server="Y"
# Q: Do you want to stop sendmail from running in daemon mode? [Y]
Sendmail.sendmaildaemon="Y"
# Q: Would you like to run sendmail via cron to process the queue? [N]
Sendmail.sendmailcron="Y"
# Q: Would you like to disable the VRFY and EXPN sendmail commands? [Y]
Sendmail.vrfyexpn="Y"
# Q: Would you like to disable printing? [N]
Printing.printing="Y"
# Q: Would you like to install TMPDIR/TMP scripts? [N]
TMPDIR.tmpdir="Y"
# Q: Would you like to run the packet filtering script? [N]
Firewall.ip_intro="N"
```

Solaris Hardening Script

The following script is a sample script for hardening Solaris systems:

```
#!/bin/sh
# modifications to /etc
cd /etc/rc2.d
for x in S71rpc S76nscd S70uucp S73nfs.client S74autofs S80PRESERVE
K60nfs.server *cachefs* S72inetsvc S69inet S88sendmail
do
      rm -f $x > /dev/null 2>&1
done
for x in defaultdomain auto* rc3.d/*
do
      rm $x > /dev/null 2>&1
done
echo 'umask 022' > /etc/init.d/umask.sh
chmod 744 /etc/init.d/umask.sh
for dir in /etc/rc?.d
do
      ln -s ../init.d/umask $dir/S00umask.sh
done
touch /etc/notrouter
chmod 444 /etc/notrouter; chown root:sys /etc/notrouter
# modifications to /var
mkdir /usr/guest
ln -s /usr/guest /var/guest
chmod 777 /usr/guest; chown root:root /usr/guest
rm /var/spool/cron/crontabs/adm >/dev/null 2>&1
rm /var/spool/cron/crontabs/lp >/dev/null 2>&1
for user in uucp nuucp adm lp listen
do
      /usr/sbin/passmgmt -d $user
done
for user in daemon bin nobody noaccess nobody4
do
      /usr/sbin/passmgmt -m -s /dev/null $user
done
for x in `find / -nouser`
do
      chown root $x
done
exit 0
```

Titan Configuration File

The following file is a sample configuration script used for Titan on Solaris systems:

```
# version 4.0.6  May 2 09:47:39 PDT 2001
#
add-umask.sh -f
adjust-arp-timers2.8.sh -f
# note aset should be run in medium on a server
aset.sh -f
# If automount is used; comment these out
automount.sh -f
automount2.sh -f
# BSM allows for greater logging, but may be a performance problem. By
default
# Titan doesn't set up a lot of logging, read Sun blueprints on
auditing for
# more
bsm.sh -f
# Might want to just do a pkgrm of all the CDE and windows modules on a
Server
# unless the windows needs to be run locally (doubtful on a server)
cde.sh -f
create-issue.sh -f
cronset.sh -f
decode.sh -f
defloginparams.sh -f
defpwparams.sh -f
disable-L1-A.sh -f
disable-NFS-2.6.sh -f
disable-accounts.sh -f
disable-core-sol8.sh -f
disable-ping-echo.sh -f
disable-services.sh -f
disable_ip_holes.sh -f
# DMI is aprt of snmp; some servers use this
dmi-2.6.sh -f
eeprom.sh -f
file-own.sh -f
fix-cronpath.sh -f
fix-modes.sh -f
fix-stack.sol2.6.sh -f
ftp-2.6_secure.sh -f
ftpusers.sh -f
hosts.equiv.sh -f
# inetd.sh -f
inetd2.sh -f
# inetsvc might need to be modified for DNS or DHCP Servers
# inetsvc.sh -f
keyserv2.8.sh -f
# Probabl login_failed_retries.sh -f
loginlog.sh -f
# if this is a lp server comment out the next line
lpsched.sh -f
nddconfig2.8.sh -f
nfs-portmon.sh -f
```

```
nsswitch.sh -f
nuke-dtlogin.sh -f
nuke-nfs-client.sh -f
# if this is a nfs server comment out the next line
nuke-nfs-serv.sh -f
# the NSCD can be poisoned but assuming the server is firewalled off
# we don't need to disable it
nuke-nscd.sh -f
nuke-powerd.sh -f
# You may need RPC services such as calendar or on SSP's if so comment
out
# the next line
nuke-rpc.sh -f
# On sendmail servers leave sendmail enabled by commenting out the next
line
#nuke-sendmail.sh -f
pam-rhosts-2.6.sh -f
passwd.sh -f
powerd2.8.sh -f
psfix.sh -f
rf_create-motd.sh -f
rhosts.sh -f
rmmount.sh -f
rootchk.sh -f
routed.sh -f
sendmail-forward.sh -f
sendmail.sh -f
smtpbanner-8.8.sh -f
snmpdx-2.6.sh -f
sulog.sh -f
syslog.sh -f
syslog_failed_logins.sh -f
tcp-sequence.sh -f
telnet-banner.sh -f
# Note this may need to be increased to 1048 for hugh oracle databases
tmpfs-fixsize.sh -f
useraddset.sh -f
userumask.sh -f
utmp2.7.sh -f
vold.sh -f
ziplock.sh -fy don't need to log all tcp connections on a server unless
# you see an intrusion
# log-tcp.sh -f
```

Code Samples

Configuration Creator Script

The following script is a sample script that can be used to backup critical system files on a Nokia firewall appliance. Please note, this script assumes that private/public key authentication is used:

```
#!/bin/sh
#
## History:
#       James O'Brien Thur Jan 20 08:03:54 CST 2000
```

```
#               Initial version.

# This script is used for backup and restoration of configuration files
# from a firewall backup.  The script uses tar to backup/restore files
# based on a configuration file.
#
# USAGE:cfgcreator.sh [-b <buildfile>] [-r <recoverfile>] [-h -help]
#
# Set each variable
#
SCRIPTNAME="Configuration Creator"
HOSTNAME=`hostname`
SYSLOGFAC="local2.notice"
DIR=/opt/scripts/cfgcreator
BINDIR=${DIR}/bin
DATADIR=${DIR}/data
BACKUPDIR=${DIR}/backups
RESTOREDIR=${DIR}/restore
ROOTDIR="/var/tmp"
DATE=`date '+%Y%m%d'`
TIME=`date | awk ' {print $4} '`
TARCF="/usr/bin/tar -cPf"
TARXF="/usr/bin/tar -xPf"
GZIP="/usr/bin/gzip -9f"
UNGZIP="/usr/bin/gzip -d"
TRANSFER="/usr/bin/scp"
COPY="/bin/cp"
FILE_FLAG="false"
BUILD_FLAG="false"
RESTORE_FLAG="false"
ERROR_FLAG="false"
TARFILE="cfg-${HOSTNAME}.${DATE}.tar"
GZTARFILE="${TARFILE}.gz"
BUILDFILE=""
BUILDLOC=""
RESTOREFILE=""
UNGZRESTOREFILE=""
OWRESPONSE=""
PM_PROFILE="/var/etc/pm_profile"
#
# Account information
#
# You will need to customize these options based on your setup
USER="username"                                 # remote account
REMOTEDIR="/var/fwbackups"
SERVER="hostname"                               # default backup server
#
# Load Environment Variables
if [ -f "$PM_PROFILE" ]; then
        . "$PM_PROFILE"
else
        FWDIR=/etc/fw
fi
#
# Define functions for later execution in the script
#
depend_check() {
```

```
        if [ ! -d "$DATADIR" ]; then
                mkdir "$DATADIR"
                chmod 750 "$DATADIR"
        fi
        if [ ! -d "$BACKUPDIR" ]; then
                mkdir "$BACKUPDIR"
                chmod 750 "$BACKUPDIR"
        fi
        if [ ! -d "$RESTOREDIR" ]; then
                mkdir "$RESTOREDIR"
                chmod 750 "$RESTOREDIR"
        fi
}
usage() {
        echo "cfgcreator.sh [-b <buildfile>] [-r <recoverfile>] [-h --
help]"
}
act_backup() {
        echo "Creating tar file $TARFILE, please wait..."
        cd /
        SED_VAR="s!\$FWDIR!$FWDIR!g"
        $TARCF "$BACKUPDIR"/"$TARFILE" `cat "$BUILDLOC" | grep -v ^# |
grep -v /$ | sed -e "$SED_VAR" `
        tarstatus=$?
        if [ "$tarstatus" -ne "0" ]; then
                echo "Error: Failed to create tar file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to create
tar file"
                ERROR_FLAG="true"
                return 1
        else
                echo "Done"
        fi
        echo "Compressing file, please wait..."
        $GZIP "$BACKUPDIR"/"$TARFILE"
        gzstatus=$?
        if [ "$gzstatus" -ne "0" ]; then
                echo "Error: Failed to compress tar file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to compress
tar file"
                ERROR_FLAG="true"
                return 1
        else
                echo "Done"
        fi
}
act_put() {
        echo "Uploading $GZTARFILE, please wait..."
        $TRANSFER "$BACKUPDIR"/"$GZTARFILE"
"$USER"@"$SERVER":"$REMOTEDIR"/"$GZTARFILE"

        trstatus=$?
        if [ "$trstatus" -ne "0" ]; then
                echo "Error: Failed to upload backup file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to upload
backup file"
                ERROR_FLAG="true"
```

```
                return 1
        else
                echo "Done"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Upload of backup
file successful"
        fi
}
act_get() {
        echo "Retrieving restore file $RESTOREFILE, please wait..."
        $TRANSFER "$USER"@"$SERVER":"$REMOTEDIR"/"$RESTOREFILE"
"$RESTOREDIR"/"$RESTOREFILE"
        trstatus=$?
        if [ "$trstatus" -ne "0" ]; then
                echo "Error: Failed to retrieve restore file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to retrieve
restore file"
                ERROR_FLAG="true"
                return 1
        else
                echo "Done"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Download of
restore file successful"
        fi
}
act_restore() {
        echo "Decompressing file, please wait..."
        $COPY "$RESTOREDIR"/"$RESTOREFILE" "$ROOTDIR"/"$RESTOREFILE"
        $UNGZIP "$ROOTDIR"/"$RESTOREFILE"
        ungzstatus=$?
        if [ "$ungzstatus" -ne "0" ]; then
                echo "Error: Failed to decompress tar file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to
decompress tar file"
                ERROR_FLAG="true"
                return 1
        else
                UNGZRESTOREFILE=`echo "$RESTOREFILE" | sed 's/.gz//g'`
                echo "Done"
        fi
        echo "Extracting files, please wait..."
        $TARXF "$ROOTDIR"/"$UNGZRESTOREFILE"
        untarstatus=$?
        if [ "$untarstatus" -ne "0" ]; then
                echo "Error: Failed to extract tar file"
                logger -p "$SYSLOGFAC" "$SCRIPTNAME: Failed to extract
tar file"
                ERROR_FLAG="true"
                return 1
        else
                echo "Done"
        fi
}
act_start() {
        echo "-----------------------------------------------------"
        echo "          Configuration Creator "
        echo " "
        echo "Starting..."
```

```
        logger -p "$SYSLOGFAC" "$SCRIPTNAME: Starting"
        if [ "$BUILD_FLAG" = "true" ]; then
                act_backup
                if [ "$ERROR_FLAG" = "true" ]; then
                        exit 1
                else
                        act_put
                fi
        elif [ "$RESTORE_FLAG" = "true" ]; then
                act_get
                if [ "$ERROR_FLAG" = "true" ]; then
                        exit 1
                else
                        echo "Do you want to overwrite the existing
configuration (y/n)? "
                        read OWRESPONSE
                        if [ "$OWRESPONSE" = "Y" ] || [ "$OWRESPONSE" =
"y" ]; then
                                act_restore
                        else
                                echo "Files will not be extracted"
                        fi
                fi
        fi
        logger -p "$SYSLOGFAC" "$SCRIPTNAME: Finished"
        echo "------------------------------------------------------"
        exit 0
}
#
# MAIN
#
depend_check
while getopts r:b:h opt
do
        case "$opt"
        in
        b|B)    BUILD_FLAG="true";
                VALUE=$OPTARG;;
        r|R)    RESTORE_FLAG="true";
                VALUE=$OPTARG;;
        h)      usage;
                exit 0;;
        esac
done
if [ "$BUILD_FLAG" = "true" ]; then
        BUILDFILE="$VALUE"
        if [ -f "$DATADIR"/"$BUILDFILE" ]; then
                echo "Build file verified"
                BUILDLOC="$DATADIR"/"$BUILDFILE"
        elif [ -f "$BUILDFILE" ]; then
                echo "Build file verified"
                BUILDLOC="$BUILDFILE"
        else
                echo "Error: Could not verify build file"
                exit 1
        fi
elif [ "$RESTORE_FLAG" = "true" ]; then
```

```
        RESTOREFILE=$VALUE
        echo "Using restore file $RESTOREFILE"
else
        echo "Error: Could not interpret command"
        exit 1
fi
act_start
```

The above script uses a configuration file to determine the files necessary to backup. The following is a sample configuration file:

```
# fwbackup.cfg
#
# build file for cfgcreator that will back up configurations on a
# nokia firewall
#
/config/db
/var/admin/ipsobackup
/var/admin/ipsobackuplist
/var/cron/tabs/root
/var/etc/rc.local
$FWDIR/conf/fw.license
$FWDIR/conf/objects.C
$FWDIR/conf/*.W
$FWDIR/conf/rulebases.fws
$FWDIR/conf/fwauth.keys
$FWDIR/conf/fwauthd.conf
$FWDIR/conf/masters
$FWDIR/conf/serverkeys.db
$FWDIR/conf/sync.conf
$FWDIR/conf/fwopsec.conf
$FWDIR/conf/omi.conf
$FWDIR/conf/slapd.conf
$FWDIR/conf/fwauth.NDB
$FWDIR/conf/fwmusers
$FWDIR/conf/gui-clients
$FWDIR/conf/smtp.conf
$FWDIR/conf/product.conf
$FWDIR/lib/fwui_head.def
$FWDIR/lib/init.def
$FWDIR/lib/setup.C
$FWDIR/database
$FWDIR/state
$FWDIR/log
```

Auto-Proxy Configuration Script

The following JavaScript code is used to automatically configure an employee's Web browser:

```
function FindProxyForURL(url, host)
{
  if  (
      (isPlainHostName(host) ||
      LocalHostOrDomainIs("inetfw01", "inetfw01.giacfortunes.com") ||
      LocalHostOrDomainIs("corpfw01", "corpfw01.giacfortunes.com") ||
```

```
      LocalHostOrDomainIs("vpn01", "vpn01.giacfortunes.com") ||
      dnsDomainIs(host, "127.0.0.1"))
    )
  return "DIRECT";
  else return "PROXY proxy.giacfortunes.com:80";
}
```

Netfilter Firewall Ruleset

The following script is a copy of the completed /etc/rc.d/iptables.rules used on the GIAC Internet firewall:

```
# add necessary routes
route add –net 172.16.0.0 netmask 255.255.0.0 gw 172.16.1.1
route add default gw 198.7.46.129

# input filter
echo -n "Loading INPUT filter..."
iptables -t filter -A INPUT -m state --state ESTABLISH,RELATED -j LOG
--log-ip-options
iptables -t filter -A INPUT -m state --state ESTABLISH,RELATED -j
ACCEPT
iptables -t filter -A INPUT -m state --state NEW -p tcp -s
172.16.0.0/16 --sport 1024:65535 -d 172.16.1.254 --dport 22 -j LOG
--log-ip-options
iptables -t filter -A INPUT -m state --state NEW -p tcp -s
172.16.0.0/16 --sport 1024:65535 -d 172.16.1.254 --dport 22 -j ACCEPT
iptables -t filter -A INPUT -j LOG --log-prefix "INPUT-DROP-"
iptables -t filter -A INPUT -j DROP
echo "DONE"

# output filter
echo -n "Loading OUTPUT filter..."
iptables -t filter -A OUTPUT -m state --state ESTABLISH,RELATED -j LOG
--log-ip-options
iptables -t filter -A OUTPUT -m state --state ESTABLISH,RELATED -j
ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d 172.16.3.11
--dport 22 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d 172.16.3.11
--dport 22 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.200 --dport 25 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.200 --dport 25 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.202 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p tcp -d
198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.201 --dport 53 -j ACCEPT
```

```
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.202 --dport 53 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d
198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.10
--dport 123 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.10
--dport 123 -j ACCEPT
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.11
--dport 123 -j LOG --log-ip-options
iptables -t filter -A OUTPUT -m state --state NEW -p udp -d 172.16.3.11
--dport 123 -j ACCEPT
iptables -t filter -A OUTPUT -j LOG --log-prefix "OUTPUT-DROP-"
iptables -t filter -A OUTPUT -j DROP
echo "DONE"

# forward filter
echo -n "Loading FORWARD filter..."
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j LOG --log-prefix
"SYNFINSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j DROP
iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j LOG --log-prefix
"FINSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j DROP
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j LOG --log-prefix
"NULLSCAN-"
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j LOG --log-
prefix "NMAPXMAS-"
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP
iptables -A FORWARD -p icmp -f -j LOG --log-prefix "ICMPFRAG-"
iptables -A FORWARD -p icmp -f -j DROP
iptables -t filter -A FORWARD -m state --state ESTABLISH,RELATED -j LOG
--log-tcp-options
iptables -t filter -A FORWARD -m state --state ESTABLISH,RELATED -j
ACCEPT
## allow Internet systems to send mail to the CCN mail server
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
## allow the CCN mail server to send mail to the Internet
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 ! -d 172.16.0.0/16 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 ! -d 172.16.0.0/16 --dport 25 -j ACCEPT
## allow Internet systems to perform DNS queries to the CCN DNS servers
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j LOG --log-ip-options
```

```
iptables -t filter -A FORWARD -m state --state NEW -p udp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.202 --dport 53 -j ACCEPT
## allow the CCN DNS servers to perform DNS queries on the Internet
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.201 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.202 -d ! 172.16.0.0/16 --dport 53 -j ACCEPT
## allow Internet systems to connect to the CCN Web servers
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.203 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.204 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 80 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 443 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s !
172.16.0.0/16 -d 198.7.46.205 --dport 443 -j ACCEPT
## allow internal DNS server to forward DNS queries to the CCN DNS
servers
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j LOG --log-tcp-options
```

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.201 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.202 --dport 53 -j ACCEPT
## allow the Exchange server to send and receive email from the CCN
mail server
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.200 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.11 -d 198.7.46.200 --dport 25 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d 172.16.9.11 --dport 25 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.200 -d 172.16.9.11 --dport 25 -j ACCEPT
## allow the Web login server to communicate with SafeWord
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.205 -d 172.16.3.12 --dport 5031 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.205 -d 172.16.3.12 --dport 5031 -j ACCEPT
## allow the secure Web server to connect to the Oracle database
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.204 -d 172.16.2.10 --dport 1521 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
198.7.46.204 -d 172.16.2.10 --dport 1521 -j ACCEPT
## allow the batch processor to communicate with the secure Web server
via SSH
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.2.10 -d 198.7.46.204 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.2.10 -d 198.7.46.204 --dport 22 -j LOG --log-tcp-options
## allow CCN servers to send syslog messages to the log server
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.10 --dport 514 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.10 --dport 514 -j ACCEPT
## allow CCN servers to sync time against the NTP servers
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.12 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.192/26 -d 172.16.3.12 --dport 123 -j ACCEPT
## allow the border router to send syslog messages to the log server
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.129 -d 198.7.46.146 --dport 514 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
198.7.46.129 -d 198.7.46.146 --dport 514 -j ACCEPT
## allow employees access to Web sites on the Internet and the CCN
```

```
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.203 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.203 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.204 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.204 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535  -d 198.7.46.205 --dport 80 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d 198.7.46.205 --dport 443 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d ! 198.7.46.192/26 --dport 80 -j
ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.9.12 --sport 1024:65535 -d ! 198.7.46.192/26 --dport 443 -j
ACCEPT
## allow NTP servers on SN to connect to external NTP servers
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 128.105.39.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 128.105.39.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 128.105.39.11 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 128.105.39.11 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 216.27.190.202 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.10 -d 216.27.190.202 --dport 123 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 216.27.190.202 --dport 123 -j LOG --log-ip-options
iptables -t filter -A FORWARD -m state --state NEW -p udp -s
172.16.3.11 -d 216.27.190.202 --dport 123 -j ACCEPT
## allow telnet and SSH to the border router and CCN
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 22 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 23 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.129 --dport 23 -j ACCEPT
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.192/26 --dport 22 -j LOG --log-tcp-options
iptables -t filter -A FORWARD -m state --state NEW -p tcp -s
172.16.0.0/16 -d 198.7.46.192/26 --dport 22 -j ACCEPT
## drop all other traffic
iptables -t filter -A FORWARD -j LOG --log-prefix "FORWARD-DROP-"
iptables -t filter -A FORWARD -s 0/0 -d 0/0 --dport 113 -j reject --
reject-with tcp-resetiptables -t filter -A FORWARD -j DROP
echo "DONE"

# source nat
echo -n "Loading Source NAT..."
```

```
iptables -t nat -A POSTROUTING -s 172.16.3.10 -d 198.7.46.129 -j SNAT -
-to 198.7.46.146
iptables -t nat -A POSTROUTING -s 172.16.1.254 -d 172.16.0.0/16 -j SNAT
--to 172.16.1.254
iptables -t nat -A POSTROUTING -s 172.16.0.0/16 -d ! 198.7.46.192/26 -j
SNAT --to 198.7.46.145
echo "DONE"

# destination nat
echo -n "Loading Destination NAT..."
iptables -t nat -A PREROUTING -s 198.7.46.129 -d 198.7.46.146 -j DNAT -
-to 172.16.3.10
echo "DONE"
```

FireWall-1 Firewall Ruleset

NO	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON	TIME
1	mgmt01	corpfw01	* / *	FirewallControl SecurityServices	accept	Log	Policy Targets	Any
2	corpfw01	mgmt01	* / *	FirewallControl SecurityServices	accept	Log	Policy Targets	Any
3	FWAdmins	mgmt01	* / *	FirewallControl SecurityServices	accept	Log	Policy Targets	Any
4	w2kmail	ExternalDNSServers	* / *	dns	accept	Log	Policy Targets	Any
5	w2kmail	smtp01	* / *	smtp	accept	Log	Policy Targets	Any
6	smtp01	w2kmail	* / *	smtp	accept	Log	Policy Targets	Any
7	GIACServers GIACNetworkDe	NTPServers	* / *	ntp	accept	Log	Policy Targets	Any
8	NTPServers	io.berkeley.netdot.net caesar.cs.wisc.edu	* / *	ntp	accept	Log	Policy Targets	Any
9	GIACServers GIACNetworkDe	ntp01	* / *	syslog	accept	Log	Policy Targets	Any
10	proxy01	ProtectedNetworks InternalNetworks	* / *	WebServices	accept	Log	Policy Targets	Any
11	FWAdmins mgmt01	GIACNetworkDevices GIACServers	* / *	SecurityServices	accept	Log	Policy Targets	Any
12	wts	auth01	* / *	tcp-5031	accept	Log	Policy Targets	Any
13	securewww	db01	* / *	tcp/telnet	accept	Log	Policy Targets	Any
14	db01	securewww app.devro-bakery.com	* / *	ssh	accept	Log	Policy Targets	Any
15	FWAdmins	auth01	* / *	tcp-5040	accept	Log	Policy Targets	Any
16	backup01	BackupClients	* / *	NetworkerServic	accept	Log	Policy Targets	Any
17	VPNClients	Net-172.16.9.0	* / *	* Any	accept	Log	Policy Targets	Any
18	vpn01	auth01	* / *	RADIUS RADIUS-ACC	accept	Log	Policy Targets	Any
19	inetfw01	mgmt01	* / *	ssh	accept	Log	Policy Targets	Any
20	* Any	* Any	* / *	* Any	drop	Log	Policy Targets	Any

Appendix C – Acronyms

Acronyms

The following acronyms were used in this paper and are provided for reference purposes:

ACL	Access Control List
ARIN	American Registry for Internet Numbers
CDP	Cisco Discovery Protocol
CRM	Customer Relationship Management
DHCP	Dynamic Host Configuration Protocol
DLT	Digital Linear Tape
DNS	Domain Name Service
DoS	Denial of Service
DDoS	Distributed Denial of Service
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
GCFW	GIAC Certified Firewall Analyst
GPG	GNU Privacy Guard
HRMS	Human Resources Management System
HSRP	Hot Standby Routing Protocol
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol - Secure
IDS	Intrusion Detection System
IKE	Internet Key Exchange
IOS	Internetworking operating system
IPSec	Internet Protocol Security
IRT	Incident Response Team
ISA	Internet Security and Acceleration
ISAPI	Internet Services Application Programming Interface
ISP	Internet Service Provider
IT	Information Technology
LAN	Local Area Network
NAT	Network Address Translation

NG	Next Generation
NTP	Network Time Protocol
OS	operating system
OTP	One-Time Password
PoP	Point of Presence
RFC	Request for Comment
RPM	Red Hat Package Manager
SCP	Secure Copy
SEP	Scalable Encryption Processing
SIC	Secure Internal Communications
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Socket Layer
TFTP	Trivial File Transfer Protocol
UWA	Universal Web Agent
VAR	Value Added Reseller
VPN	Virtual Private Network
WAN	Wide Area Network
WLS	Web Login Server