Global Information Assurance Certification Paper

SunScreen Protection For GIAC Enterprises

GCFW PRACTICAL ASSIGNMENT

Firewalls, Perimeter Protection and VPN's

Version: 1.8

Washington, D.C. October 2002

Prepared By
Fred Gross III

TABLE of CONTENTS

ABSTRACT

Information technology security for GIAC Enterprises has been designed on a defense in depth concept using multiple security zones to contain resources with similar risk levels. The primary perimeter protection is provided by a Cisco router with extended ACLs and Sun Microsystems's SunScreen firewall product configured in stealth mode. Interior security zones are created and protected by a combination of a Cisco layer three switch and a second SunScreen firewall. VPN services are achieved by a combination of secure shell and IPSec servers providing secured transmissions for external business and GIAC employees respectively. Desktop security issues are partially addressed by using a Linux solution based on diskless workstations and open software products.

Verification of the external firewall policy for Assignment III was completed by building a test rack with a fully configured SunScreen firewall and four target/test systems, one on each of the four defined firewall interfaces.

I used William Chan's GCFW Practical, November 30, 2002 for Assignment IV.

INTRODUCTION:

Company Background:

GIAC Enterprises is 34-year-old fortune cookie company based in the San Francisco Bay area and founded by Mr. Flowers in 1968. GIAC Enterprises grew into a fully integrated company, which included all business aspects from fortune writings, baking, sales, warehousing and shipping. Originally, GIAC Enterprises was known as Goodtime and Inspirationally Aware Cookies. In this period the company flourished and grew to have extensive US and foreign markets.

In the early 1990s Mr. Flowers was growing tired of running such a large complex business and saw the dawning of a new age with the rapid expansion of the .com industry. His new inspiration lead him to understand that the real essence of his company and the source of his wealth were the fortunes themselves and the cookies. In short, order he set about to restructure his company keeping the creative core, and much of his marketing and sales force. He turned the rest of his operation over to his long time friend Mr. Rose and thus became an immediate customer. In the restructuring process to become a privately held .com company, he followed suite with many other businesses and reduced the business name to the acronym GIAC Enterprises.

The new operation was a great success in the fortune cookie industry where getting new and creative sayings was always challenging for the cookie bakers. However, the good times hit a major obstacle recently when it was discovered that GIAC's most recent and best fortunes started showing up in foreign made cookies. Mr. Flowers contracted with a computer security company to inquire as if there was any proof of theft. Not only was there proof, but the crackers left a rather unsettling fortune on GIAC's web server, suggesting worse things to come.

While discussing this situation with his friend Mr. Rose at their favorite Chinese restaurant he opened a fortune cookie, which read: "Security will be your future". As the wheel of fortune turns, Mr. Flowers also became the sole winner of a multi-million Power Ball drawing.

Mr. Flowers' passion and love in life is bringing people brief moments of happiness with his fortunes. The future of GIAC Enterprises become clear to him, with renewed vigor and resources Mr. Flowers became committed to ensuring that GIAC Enterprises would remain the world's Fortune 1 Company in on-line fortunes. Thus, he contracted with 12^2 Computer Security to rebuild his entire computer and network architecture to help ensure that GIAC Enterprises would remain the world's Fortune 1 on-line fortune company.

Business Model:

After their recent restructuring GIAC Enterprises became a small well focused operation whose vision is to be the world's fortune 1 on-line fortune company. Their sole business objective is to sell fortunes to customers on the Internet. Contracted fortune writers or suppliers provide GIAC Enterprises with new fresh fortunes. Typical the writers operate from small office or home office environments (SOHO) where imagination greatly exceeds computer expertise. Major customers include businesses that print and/or bake fortune cookies and want a steady source of high quality fortunes. GIAC Enterprises also has established partnerships with a number of independent companies that purchase fortunes in bulk and translate them into a variety of foreign languages. Experience has shown that the computer expertise in both the customer and partnership companies varies considerably and is generally low. A very successful recent addition to the GIAC Enterprises product line is an on-line fortune subscription. Customers may purchase a daily or weekly subscription to receive a new fortune via e-mail as either a plain e-mail or as an executable attachment that the customer can crack open and unfold their fortune with proper sound effects.

Long-term success of GIAC Enterprises depends on their ability to readily provide wide variety of unique and high quality fortunes at a competitive price. A critical aspect of this business model is to maintain the confidentiality and integrity of their intellectual property, the fortunes.

A central tenant in Mr. Flower's business strategy is to ensure against the theft or alteration of any fortune while in transmission to or from or while in storage at GIAC Enterprises and impose a minimum constraint on business flow. Implementation of this tenant is the essence of the contractual agreement with 12^2 Computer Security.

Company Structure:

GIAC Enterprises is organized into the following discrete business units:

- Senior Management: Sets direction, policy and makes all major business decisions.
- Quality Control: Responsible for reviewing all fortunes submitted by the fortune writers and classifying them into proper categories, (humorous, politically correct, politically incorrect, seasonal, etc.). They also submit acceptance reports to both the writers and Accounts Payable.
- Sales: Responsible for customer contacts and expanding business opportunities. The sales team woks within both the main office and extensive on-the-road operations.
- Marketing: Responsible for monitoring impact of competition and proposing new business opportunities to management.
- Accounts Receivable: Responsible to monitor on-line sales and collect all payables.

- Accounts Payable: Responsible to monitor the fortunes acceptable by the Quality Control Unit from the contract fortune writers and ensure all other debts are paid.
- Information Technology: The IT unit is future organized into a database/applications group and an operations group, which includes IT security.

GIAC Enterprises employees approximately sixty employees at the central office plus eight traveling sales representatives.

Assumptions:

Changes made to the network and computing design at GIAC Enterprises cannot require significant expenditures on the be-half any of GIAC's suppliers, partners or customers.

Mr. Flowers has made it clear that we are to change or replace any equipment, software or technical processes deemed necessary to help accomplish his overall business vision. It is understood that the business application software maybe reworked to avoid un-necessary security problems.

A general principal of uniformity in service will be utilized. By this, we mean that one technical solution (IPSec, ssh/scp, ftp, etc.) will be used for all members in a single relationship class (suppliers, partners, etc.). We want to avoid security and design complexities induced from special solutions for individual members in s given relationship class.

Please note that through out this practical I have used the expression "we" to more accurately represent how a security proposal from 12^2 Computer Security might actually read. In no case does this suggest or imply anyone other than myself as the author of this work.

ASSIGNMENT I: NETWORK DESIGN:

Design Process:

Starting with the simple instructions from GIAC Enterprises discussed above we will move through a series of steps that will result in a computing and network infrastructure that will meet the stated objectives and be maintainable in the future. These steps will include:

- Business process and information flow.
- Risk analysis.
- Development of a high-level security policy.
- Network design options.
- Development of a logical network design.
- Development of the physical network model.
- Assignment of subnets and IP addresses.

Design Principals:

GIAC's requirements are to ensure against theft or alterations of their fortunes while in transmission to or from or in storage at GIAC's offices and minimum impediment to business flow. This is essentially the standard security requirement of availability, confidentiality and integrity. Defense-in-depth principals will be used, including physical security, network/system security and social issues. What we mean by defense-in-depth is that we will use multiple and complimentary technology and techniques to protect sensitive resources. All business activities include a measure of risk. Possibility of resource loss because of deficiencies in IT infrastructure or IT operations is part of the total risk factor. The objective is to reduce the risk to an acceptable level in accordance with the value of the resource and business requirements.

Business Process and Information Flow:

The drawings in Section 2.3 are simply business flow models and do not represent network diagrams. Access requirements and rationales are provided partially in the following sections are developed more fully in Section 2.9.3.

Foreign County Issues:

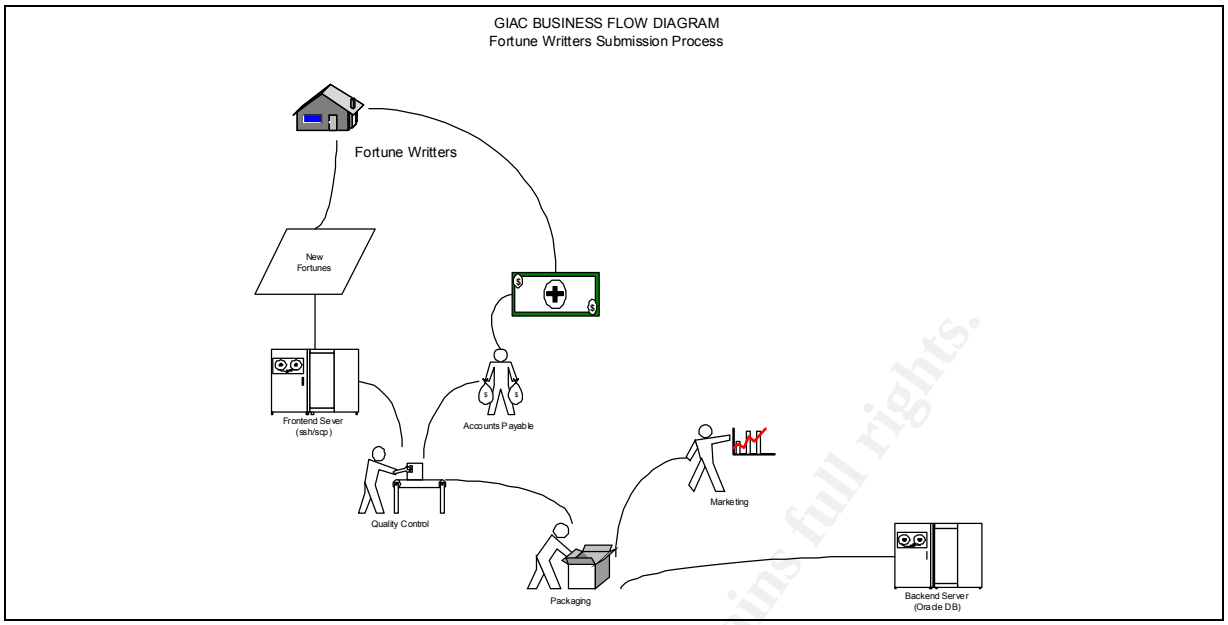
Some of GIAC's suppliers, customers and partners are located outside of the United States. To reduce potential problems with encryption restrictions, we have elected to use standard https, 128 bit, and secure shell, 192bit 3DES, to provide transmission security with GIAC's business associates.

Suppliers or Fortune Writers:

GIAC Enterprises obtains all of their fortunes from contract writers. These writers have a quota of fortunes they are to provide on a monthly basis. The fortunes maybe submitted to GIAC Enterprises for approval and acceptance in any quantity at any time. For the most part these writers operate from small offices or home offices, and are located in various regions of the U.S., but a few are abroad.

Once a writer submits a batch of fortunes, or in fortune parley, a batch of cookies, they are reviewed and classified into categories by a GIAC employee in the Quality Control unit. Subsequently, accounts payable is notified and they initiate payment for the number of accepted sayings. Marketing also receives notice of the new products. The final step is the transfer of the classified fortunes to the production database server.

Generally, the writers are much more creatively inspired than technically skilled. We do not want to use software that the writers may find particularly unusual or difficult to install and configure.



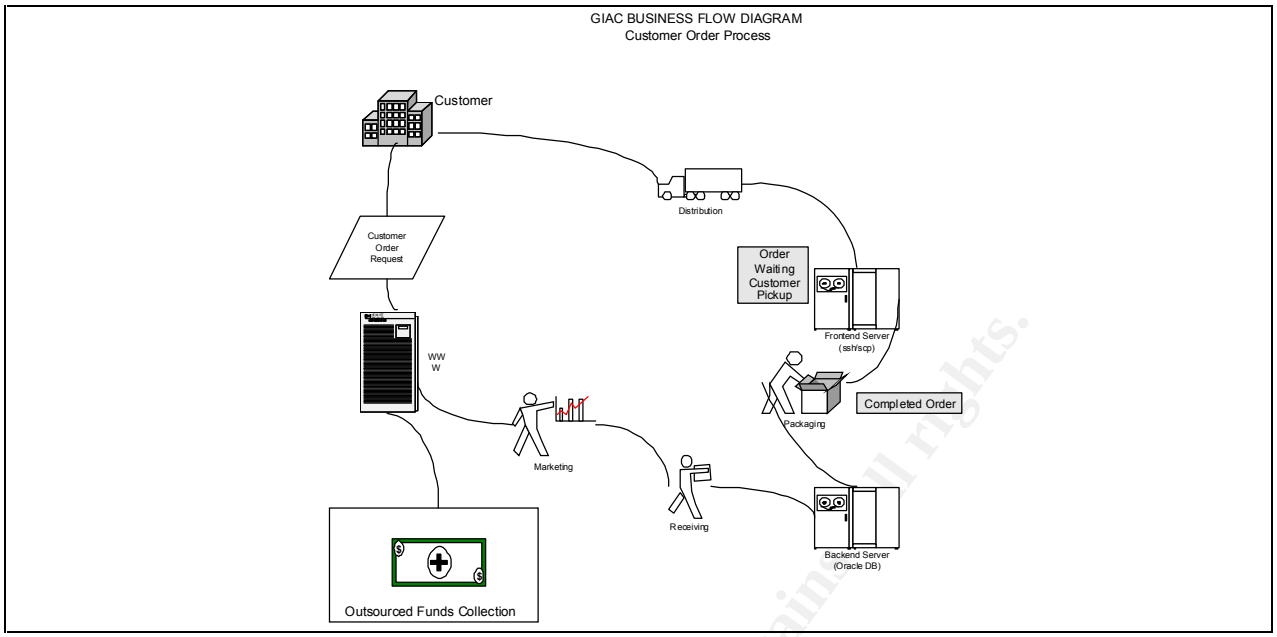
Suppliers will transfer their fortunes via secure copy to GIAC's VPN (secure shell) server. This can be accomplished with either freeware or commercial ssh/scp products. A cron job on the interior fortune database server will perform periodic sweeps for all new fortunes on the VPN server pull them into the GIAC network via secure copy.

Customers:

GIAC Enterprises reports strong sales from many global locations. While this is good for business, it does create a significant issue in terms of funds collections. To alleviate this problem GIAC Enterprises utilizes an external funds collection agent. Once the agent provides a confirmation number, the order is filled.

We do not want to use software that our business partners may find to be particularly unusual or difficult to install and configure.

Customers will place fortune orders via https to GIAC's public web server. A set of custom scripts of the web/application server will repackage the orders as a PKG encrypted mail attachment and forward them to the GIAC sales team for processing. Once their order is ready, they will be notified by e-mail to retrieve their fortunes on GIAC's VPN server with secure copy. Customer usernames and passwords are included as a PKG attachment to the e-mail notice.

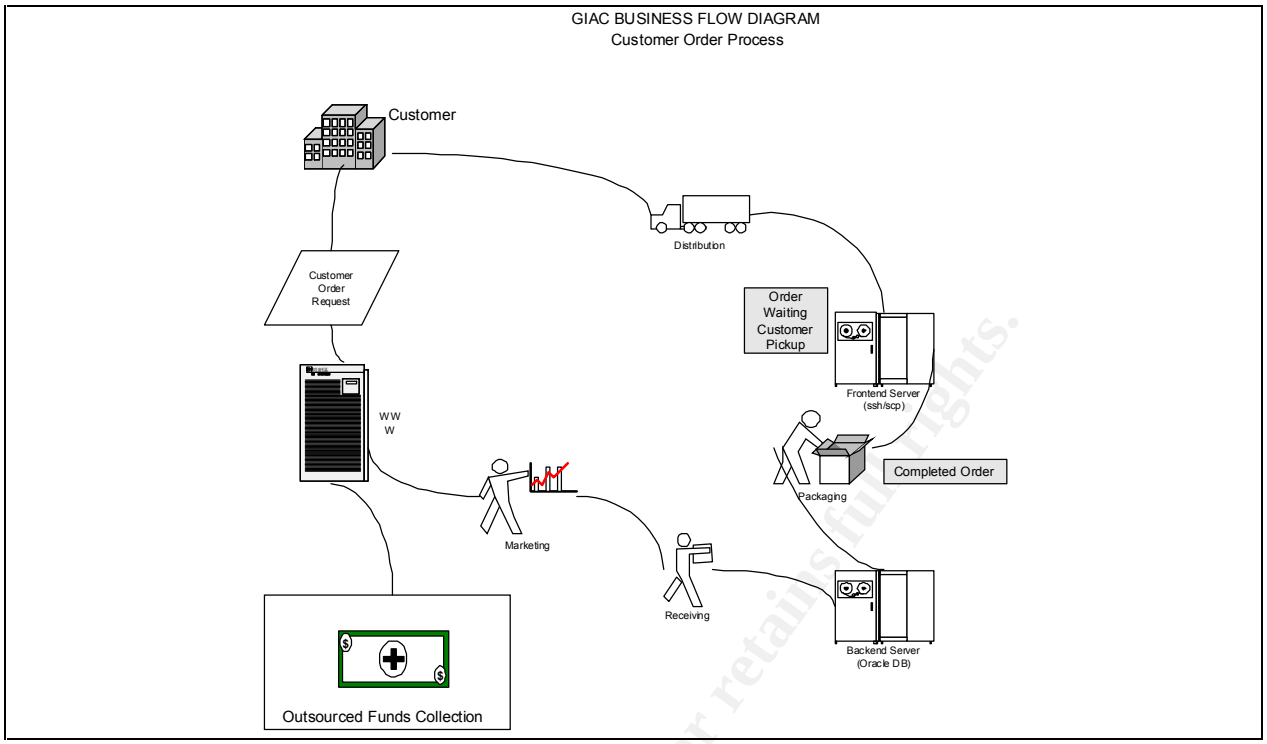


Business Partners:

GIAC Enterprises has agreements with a number of predominantly foreign firms who purchase fortunes in bulk from GIAC and translate them into other languages. Because of the global nature of these partners, they must be able to request and receive fortunes on a 24 x 7 basis. GIAC business policy does not treat this group of customers particularly different from other customers, except the order is processed prior to confirmation of the funds transfer process. This is deemed acceptable via contractual agreements.

Whereas, some of these partners have technically skilled IT staffs, others do not. We do not want to use software that our business partners may find to be particularly unusual or difficult to install and configure.

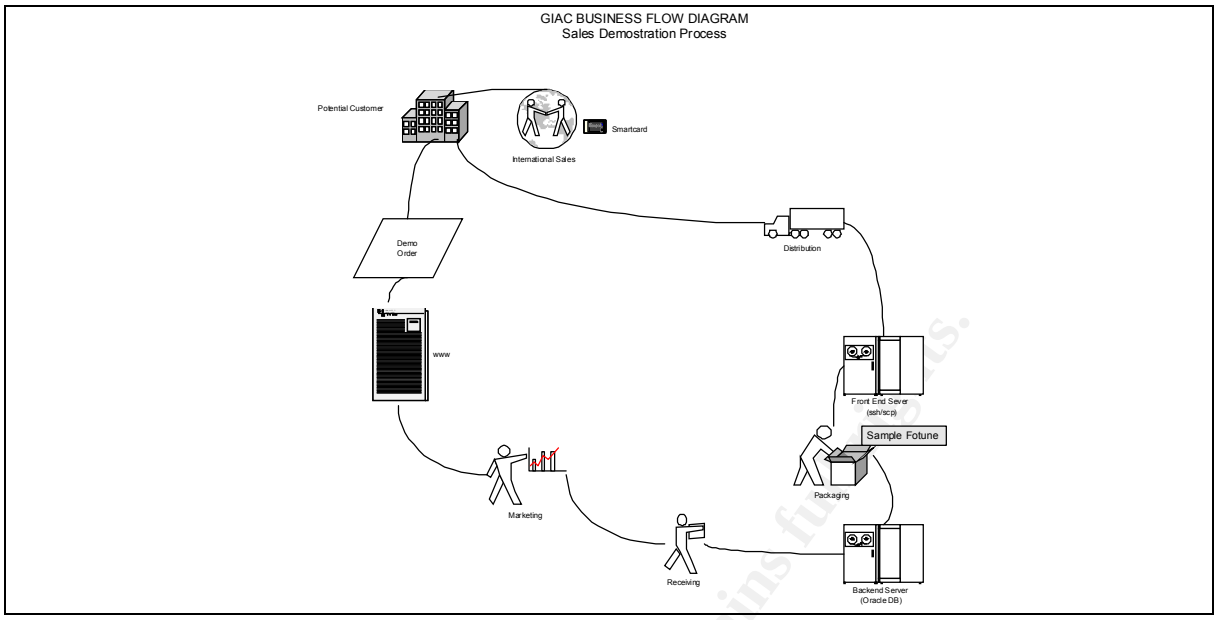
Business Partners will place fortune orders via https to GIAC's public web server. A set of custom scripts of the web/application server will repackage the orders as a PKG encrypted mail attachment and forward them to the GIAC sales team for processing. Once their order is ready, they will be notified by e-mail to retrieve their fortunes on GIAC's VPN server with secure copy. Partner's usernames and passwords are included as a PKG attachment to the e-mail notice.



Sales Team

The GIAC sales force operates mainly as global traveling sales representatives, making occasional and short visits to the GIAC Corporative offices. Their primary tactic is to setup their laptop at a potential customer site, connect to the Internet via a local ISP or dial-up if required and demonstrate how simple and easy it is to make a purchase from GIAC Enterprises. In doing so they will present the potential customer with a small sample of fortunes they might expect from fortune categories of interest.

The sales representative may also provide necessary technical assistance in getting a potential customer's computer system setup for business with GIAC Enterprises. In support of this function and to complete other routine company business, these traveling sales representatives must have access to their e-mail system at the home office and access to download software and instructions as necessary to provide required assistance to potential customers. They also must have secured access to confidential customer contact information and other company documents.



GIAC IT Staff:

Select members of the GIAC IT staff are authorized to perform certain operations from home on an emergency basis in order to maintain a 24 x 7 operation. This includes members from both the database and operations groups. Company policy requires these individuals use a dedicated system provided by GIAC Enterprises to perform all such operations. These systems are Linux based with a minimum build and hardened according to specifications described elsewhere in this document and running Netfilter/IPTables for a firewall. This is to limit potential problems resulting from a remote workstation being compromised.

Ideally, they would have the same functionality from home as they do at the office. Due to several network design options the firewall and NIDS management networks are inaccessible from any external device. This will somewhat "blind" external security investigative capabilities. However, use of an event alarm system will provide limited notification of significant events.

External access permissions required by the GIAC IT group far exceeds that required for any other group of GIAC employees. Access requirements may include investigative, testing and/or corrective action on some network component or host device. This may require a wide variety of protocols and destination ports, with potentially sensitive packet content. GIAC security policy requires all of this traffic be encrypted and authenticated from the remote host to a GIAC gateway device.

VPN access for the GIAC IT staff will be made from a remote host to a gateway device via IPSec. We will use two forms of authentication to

access the VPN server, iPass, http://www.ipass.com/, and standard user ID and password pair. Once properly authenticated they will have normal access to the rest of the GIAC intranet, excluding the firewall and NIDS management networks.

Internal GIAC Staff:

By policy, GIAC employees Internet privileges are limited to web access and e-mail, which is partially controlled by proxies, and firewall rules. Since GIAC Enterprises uses a diskless Linux desktop solution, users have no reason to download software or most other executables. These restrictions are in place to assist in maintaining a known software and hardware install base.

The GIAC IT staff are permitted greater access, including downloads, ftp, secure shell, telnet and ICMP pings. The IT staff needs to be able to access external resources and download new software, updates, patches, etc to maintain GIAC's IT systems.

All GIAC employees may access GIAC's Internet web and ftp servers to conduct normal GIAC business operations.

Risk Analysis:

A network security policy needs to be aligned with known and potential risks. It is impossible to protect all resources against all possible threats. So we must identify and rank those threats we deem the most significant. In this section we will list and rank resources which we wish to protect. This list should be considered dynamic, in that inclusions and rankings may change according to changes in business strategies, policies and technical considerations, such as new software and newly found vulnerabilities. Our security policy may need to be adjusted accordingly because of changes.

RANKING	RESOURCE	PERCEIVED THREAT
1	Fortunes	Unauthorized access/loss
2	Internal business information	Unauthorized access/loss
3	Customer information	Unauthorized access/loss
4	Sales/delivery system	Degraded service
5	Supply system	Degraded service/modification or loss
6	E-mail system	Degraded service/unauthorized access
7	Intranet network and hosts	Degraded service/unauthorized use
8	External www access	Degraded service

We know from prior loss of fortunes that there are hostile parties that have directed GIAC Enterprises via network based attacks. In addition, GIAC Enterprises is subject the standard array of script-kiddie and other common Internet based attacks.

Development of a high-level security policy:

Objective:

In this section, we will present portions of a high-level security policy, which provides the basis to design the network and create the detailed device specific security policies. Please note only a portion of the items listed below will be addressed for this practical assignment. In reality most of these items would have to be fully developed with written polices and procedures including training schedules defined where appropriate.

High Level Security Policy:

- Use company standard hardware and operating system configurations.
- All system builds will be done from a build server to help ensure consistency across systems and emergency rebuilds.
- The build network and build server will be fully independent and isolated from any other network.
- All network traffic will be denied except for what is explicitly permitted.
- Device security policies will restrict permitted traffic from and to the minimum number of hosts.
- GIAC IT staff will have ssh/scp. ftp, http and ping access to the outside.
- GIAC IT staff will have external access to manage servers and network equipment. This traffic must be encrypted and authenticated.
- Non-IT GIAC staff will have http access the outside and ftp access to GIAC ftp drop box.
- Traveling GIAC sales force will have access to GIAC sales support systems.
- All GIAC employees have remote access to GIAC e-mail services.
- All e-mail including attachments is limited to 3 MB per message.
- The GIAC user segments will be able to only ping up to and not including the interior firewall.
- All inbound Internet, local and outgoing e-mail will be filtered for viruses, executable attachments and content.
- All in-bound e-mail will pass through a mail gateway.
- All outgoing e-mail must pass through the interior mail server.
- Suppliers, partners, customers and GIAC sales staff have access to the exterior fortune pickup service.
- All internet users have access to GIAC's public web server.
- Except as otherwise noted Internet users have no other access into the GIAC network.

- GIAC IT staff will have access to "all" systems from home based workstations.
- Except as otherwise noted GIAC employees have external access to the GIAC external networks from inside the exterior firewall..
- The GIAC ftp drop box is accessible by all GIAC employees from the inside and the public via registered guest accounts with passwords.
- No company proprietary or sensitive information will be placed on the ftp server.
- Only GIAC IT staff will have ssh/scp access to GIAC servers, except as otherwise noted.
- The only GIAC business unit, other than IT, to the fortunes database is the Quality Control unit.
- GIAC business units are assigned subnets.
- All database applications will be based on a three-tier model.
- Servers will be assigned to subnets based on risk factors.
- "All" servers will utilize a central syslog facility.
- NIDS equipment will be installed at significant network choke points.
- All NIDS devices will be centrally managed and monitored.
- HIDS facilities will be used on key servers.
- Firewalls will be managed from a central management console.
- All network and server equipment will be contained in a restricted area.
- All IT database and operations staff will work in a restricted area.
- All updates and changes will be tested on the development and test network prior to implementation on the production network.
- All system and network changes, updates and additions will be recorded in a timely fashion.
- All user desktop systems will be centrally configured and managed. No user (non-IT staff) will have authority to add, change or delete software on their desktop systems.
- All desktop data will be centrally backed up on a daily basis.
- Servers with volatile information will be backed up on a daily basis.
- All other servers will have a duplicated system image disk(s) created and stored as a backup.
- All backup media will be taken off-site Monday through Friday for storage.
- Abnormalities in daily backups will be properly recorded.

Development of a Logical Network Design:

Approach:

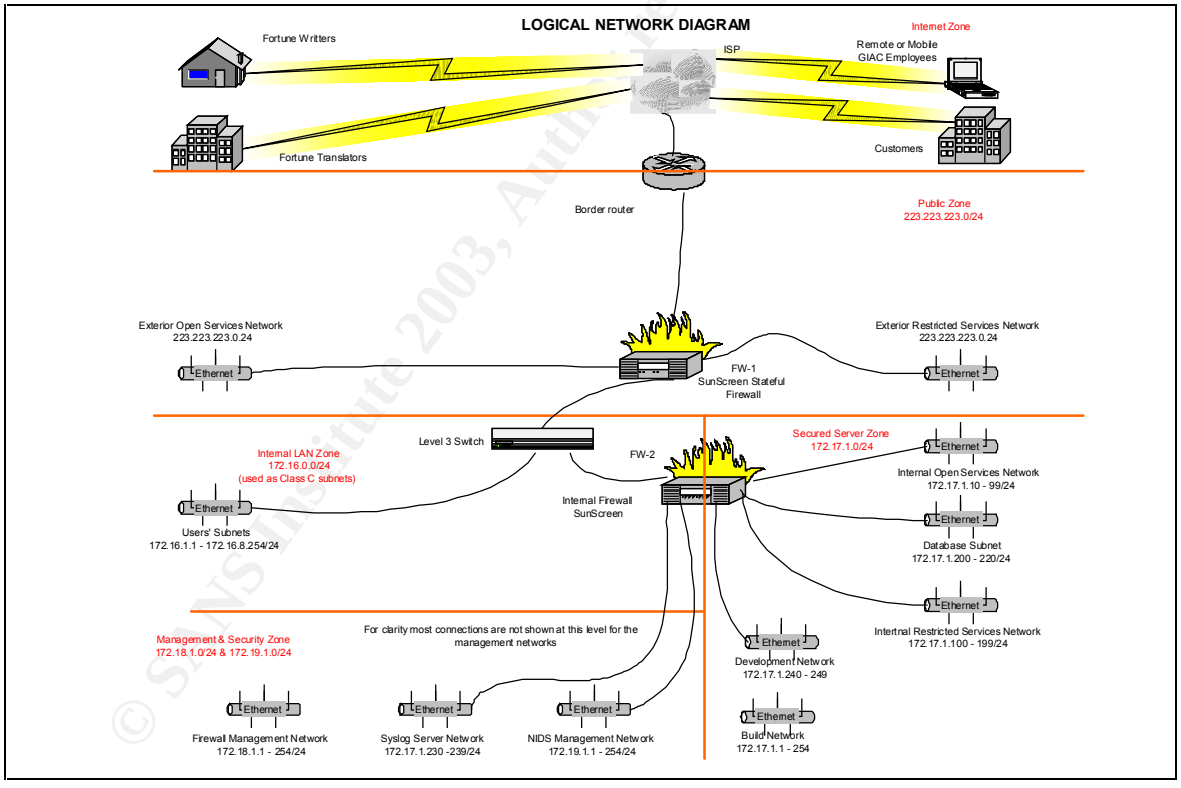
In order to develop a clear picture of the network details and device security polices, it best to start with a simple view of the network design.

To achieve defense-in-depth we will place multiple barriers between would be crackers and certain network resources. This approach will create zones and sub-nets in-which devices are placed with similar functions

and/or risk factors. The over-ridding idea is to protect the most important systems by placing these devices farther away from systems that are more vulnerable to unauthorized access or compromise. We will use a variety of different kinds of hardware and software barriers, so if exploitation of a particular vulnerability provides a foothold somewhere in the network, a cracker maybe stopped or at least slowed down while other measures can be taken to limit damage.

Security Zones:

We have created a network based on multiple zones. The Internet Zone is unprotected Internet address space. The Public Zone includes servers that may be accessed by Internet users. The Server Zone includes the rest of GIAC's production servers and are assigned to five different "subnets". The Internal LAN Zone is comprised of all the user subnets. Finally, the Security and Management Zone includes all the subnets and systems that are used to manage the various security facilities in the GIAC network.



The boundaries of the security zones have been created by the insertion of a firewall and filtering routers. Most security subnets are created by attachment to a port on a firewall. The multiple user subnets, which are collectively a security zone, have been created as multiple VLANs on a Cisco 6506 level 3 switch. We have chosen this switch because of its proven track record, expandability and feature rich software

http://www.cisco.com/en/US/products/hw/switches/ps708/index.html that includes VOIP, QoS, IDS and local firewall capabilities. The firewall central management and NIDS management networks are isolated from the rest of the production subnets and are considered to be in the network security and management zone. The grouping of devices in a particular security subnet is based on similarity of risks and protection requirements.

Subnet Naming Scheme:

We have adopted a subnet naming system that attempts to describe both the placement and function of the subnet. For example, the public web server is placed in the Exterior Open Services Network (EOSN), while the VPN server is in the Exterior Restricted Services Network (ERSN). Likewise, the intranet web server is in the Interior Open Services Network (IOSN).

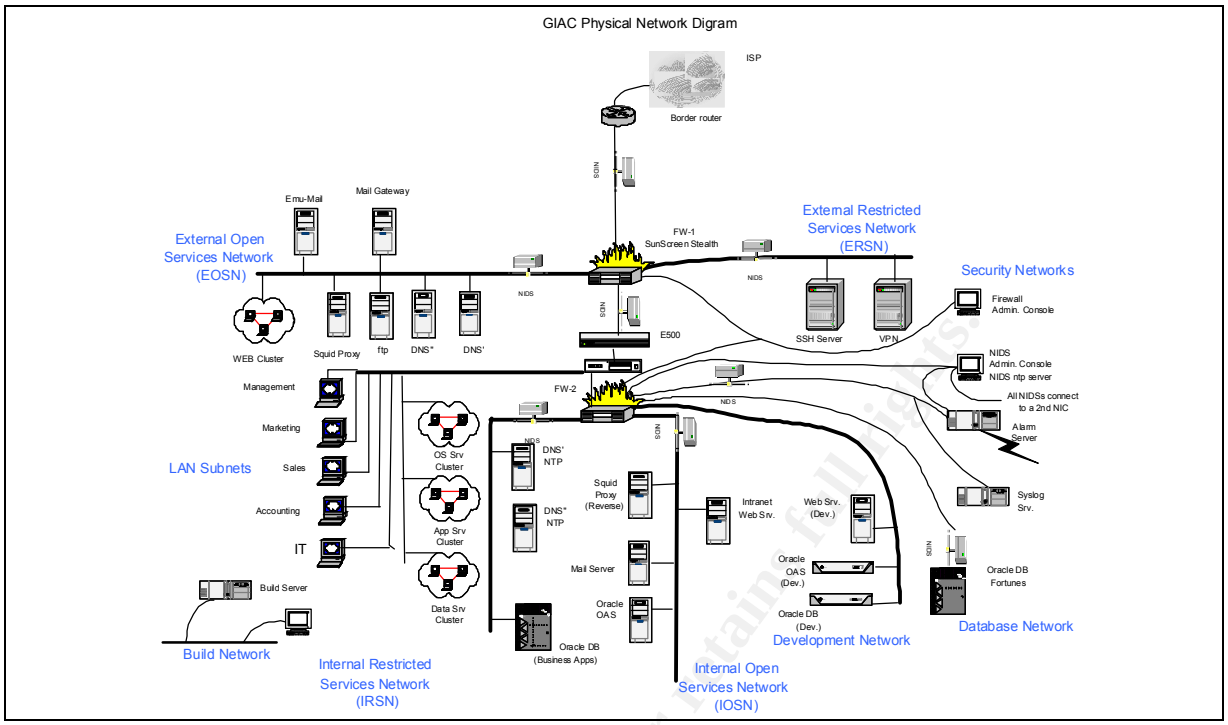
Security Objectives:

Frequently, it is not realistic to attempt to provide maximize protection for every device on the network, nor is it likely to be economically feasible. Following is a list of the top security requirements required to achieve the business objectives.

- Prevent unauthorized release or modification of the fortunes.
- Prevent unauthorized access or modification of the company's financial and other business related documents, plans, customer information, etc.
- Provide for a 24 x 7 operation.
- Prevent unauthorized use of company's IT resources.

Physical Network Design:

The following sections will go into depth relating to selection of hardware/software and device relationships in the network. Details of the device security policies will be covered in Assignment II of this document. The entire network design is shown in the following diagram. For clarity in this crowded drawing, we have omitted the security zone boundaries and IP addresses.



Hardware and Operating System Standards:

We will standardize on a few hardware/operating system configurations. This will permit the stocking of spare systems and components on-site for when hardware failures do occur. For some of the servers we will duplicate the system disk(s) for a server and store these in a protected cabinet. When a major hardware failure does occur we will be able to swap the hard drives into a standard spare system, reload data files from media if required and be back on-line with minimum downtime. Most importantly, this procedure can be accomplished by most GIAC IT staff and have confidence that all of the application specific and hardening details have been met. This will eliminate the tendency to hurry up a repair job cutting corners just to get a server back on-line, ultimately bypassing security controls.

Although we believe that Windows based servers can be reasonably secured, we will rely on Solaris and LINUX based systems wherever possible.

Even through Red Hat Linux is a supported Oracle OS platform, Oracle is certified, at the present time, only on older Red Hat releases and Red Hat Advanced Server. Since Oracle does stays in-sync with the Sun Sparc/Solaris product line we will standardize on this platform for primary database requirements.

Preference will be to use Solaris on Sparc platforms for database applications and a mix of Red Hat Linux V 8.0, kernel v 2.4.20 and Red

Hat Advanced Server 2.1, kernel 2.4.9 on Dell platforms for most other server and infrastructure requirements, routers and switches excluded. Advanced Server 2.1 has special enhancements for clustering and is considered very stable.

We have selected Dell models 1650 for the smallest servers, DNS, etc.; Dell 2650 for mid-sized servers, Web and Dell 8450 for high-end requirements. By standardizing on two vendors and product families, we will be able to easily reallocate spare parts and peripheral subsystems. Desktop standards will be discussed elsewhere.

System Hardening:

System hardening is one of our last defensive measures and is useful when outer defensive systems have failed or someone is attempting exploitation from the local network or the local host. All servers will be built using the minimum software base possible for the required services, and well hardened. Unless it is required for application software, such as Oracle, X-windows or other GUI software will not be installed on any server. This eliminates a whole series of vulnerabilities. In recent months, the use of back channels has gained popularity. These typically work by originating a connection from an inside server outwards to the hacker on a port frequently left open for outbound traffic on the firewall(s), such as 22, 23 or 80. We will deny all outbound server traffic on these ports.

There are several measures we can take which will prevent other exploits from working in the event our outer defensive layers fail. All systems should be carefully examined to remove SUID and SGID permissions from as many files as possible. This procedure remains a somewhat trail and error process and may vary according to use of each server. Part of the problem of executables with SUID bit set is how they handle creation of files in /tmp. This can provide an avenue to a symlink attack. Use of a more restrictive umask value, 025, can help somewhat. We can monitor this type of activity by installing L0pht Watch from http://www.L0pht.com/advisories/L0pht-watch.tar.gz.

Another technique is to disable stack execution. This will prevent some of the fixed-length buffer overflow exploits from working, but it will not work for heap-based overflows. For Solaris systems this is changed by setting noexec_user_stack=1 and noexec+user+stack_log=1 in the file /etc/system. For Linux systems it can be achieved by applying a kernel patch from http://www.openwall.com/linux/.

Core dump files may contain a wealth of system information to a hacker. Even through they can be essential for some debug purposes, they also pose a liability. For normal operations we will set core file size to 0 by ulimit –c 0.

The following table illustrates which ports, /etc/services, and associated services, /etc/inetd, maybe left open on servers depending upon their function. All others will be closed and when possible the associated software will not be installed.

Port	Service	Server Types
tcp/22	ssh & scp	all
tcp/25	smtp	mail server
tcp/80	http	web server
tcp/443	https	web server
tcp/514	syslog	all
udp/53	dns	DNS servers
tcp/20 & 21	ftp	ftp server
upd/500	isakmp	VPN server
tcp/123	ntp	all

Unnecessary services in /etc/rc2.d and /etc/rc3.d will be removed by renaming the link. These services may include, depended upon the server's function:

- sendmail
- autofs
- automount
- vold
- volmgt
- nfs.client
- nfs.server
- lockd
- lp
- utmp
- ncsd
- snmpd
- dmi
- keyserv
- statd

The Solaris systems will be hardened in part by using the YASSP hardening script for Solaris, Beta #15, http://www.yassp.org. Even through this script does a great deal of good work, a few issues still need to be completed by hand. The YASSP post installation steps explains these procedures and reasons very well, http://www.yassp.org/after.html. Of particular interest are the mount options for /etc/(v)fstab file; patching features; SUID limitations; logging and Tripwire. By creating dedicated file systems for /opt, /var and /usr/local we can usually mount these file

systems as read-only. This makes it a little harder to modify many executables.

We will also use the SANS Institute checklist Solaris Security Step by Step version 2.0. By disabling stack execution, we can prevent some of the buffer-overflow exploits. This can be done by setting A final measure is to run the custom hardening script shown in Appendix A and verify closure of unneeded ports and services.

For all Red Hat Linux 8 installations, we will use Linux Bastille 2.0.4-1.0, from http://www.bastille-linux.org/ to harden our systems, this is the latest version at this time. To make this version of Bastille run, you also have to install the perl-Tk-800 module and set the $DISPLAY variable. We will also run a custom script based in part on "Armoring Linux" by Lance Spitzner, http://www.enteract.com/~lspitz/papers.html that performs several additional functions. Please note some details of this script must be changed for certain server functions.

Recently, kernel exploits have gained respect and importance. Whereas, it may be difficult to fully protect against this class of attack, the use of LIDS may limit damages by "sealing" the Linux kernel. LIDS is installed as a kernel patch, requiring recompiling the kernel, http://www.lids.org/.

All servers, Linux and Solaris will include tcpwrappers to monitor internet services, such as ssh/scp and the commercial version of Tripwire v 5.1 as a host IDS. We will use Tripwire for Network Devices as a HIDS on the Cisco equipment and Tripwire for Servers on all Solaris and Linux servers. Tripwire Manager permits remote and near-real-time monitoring of Tripwire clients via ssl and ssh connections.

While tripwire is good at detecting files changes, it may miss compromised kernels. As a compliment to Tripwire we will run chkrootkit, http://www.chkrootkit.org on a regular basis with a cron job. Exceptions to the chkrootkit output will be summarized by a script and delivered by means of the central alerting facility. This layering should work effectively as a host based IDS.

Database management functions involving the Oracle applications are not explicitly part of this assignment, but a mention is still reasonable. Oracle is known to ship with a large number of default accounts and passwords. If these accounts and passwords are not required they should be deleted or the password changed to a strong value.

All servers will be built on the dedicated build network, which is not attached to any other network, including the Internet. All current operating system and application patch sets deemed necessary will be applied in

the build process. Complete configurations for each system will be stored on the build server. This provides us with a benchmark to compare changes against and a means to rebuild a server to its original specifications with minimum effort. This does require multiple test builds in order to get the configuration right in the first place. Only after a system is fully built and tested including penetration testing will it moved to a production network. IP Address range on the build network is the same as used on the production server network, 172.17.1.0/24. This is not a problem since the two networks are never connected.

GIAC will use a single T3 circuit to connect to their ISP.

Routers and Switches:

Our selection to use Cisco equipment is largely based on its strong reputation for quality, reliability and upgradeability. We will use Cisco 7606 router at the border gateway and a Cisco 6506 switch to create the VLANs for the user subnets and attachment of the interior firewall.

Being the most exterior device on the GIAC network the border router provides our first line of defense and partially protects the firewall. We will use extended ACLs on the border router to provide ingress and egress filters. In some cases, the ACLs will intentionally duplicate capabilities of the external firewall. This is intentional to provide a different mechanism to stop some traffic. The router will help to prevent spoof attacks from either assigned or unassigned address spaces and block specific ports as desired. If in the future, we need to add more security at the border router we may use the firewall features of this router to improve security.

Every desktop device cable will be home runned back to the computer room and terminated directly on the Cisco 6506. This will improve physical security by eliminating remote data closets and permit easy reconfiguration of individual offices into different vlans when management reassigns office space. Any unused switch port will be closed to somewhat reduce the problem of unauthorized LAN attachments. The business unit subnets will be created by using VLANS and layer 3 static routes. We can apply some ACL type filtering between the VLANs if needed.

The Cisco routers and switches will use Cisco IOS 12.2, which is the most current at this time. The Cisco 6506 permits loading new system images via a PCMCIA flash memory card. The same mechanism will be used to create backups of the router configurations after every change. Several generations of backups will be kept off site.

Since the router and switch configurations are expected to be reasonably static, remote administration is not considered important. We will prohibit

telnet, http(s) and ssh to any of the Cisco equipment. Instead, we will have a dedicated console with an A/B type switch to manage the Cisco units. This will eliminate the security problem of permitting remote access to a boundary router located outside of the external firewall. Recently a Cisco Security Advisory announced a buffer overflow in the ssh implementation of several Cisco products, http://archives.neohapsis.com/archives/cisco/2002-q2/0017.html.

Firewalls:

Selection of a firewall solution can be complex and is one of the most important decisions for a network design. We have elected to build a two-tier level of defense at the gateway. We believe many of today's firewalls are technically capable devices and the larger source of problems comes from miss-configuration of the device and not a failure of the device it self. Therefore, a key selection consideration is to use a product the staff is competent on and understands.

Of the three major types of firewalls, packet filtering, statefull and proxy, the statefull and proxy are serious candidates in this application. A packer filter firewall is better suited when applications requirements are simpler and cost is a prime consideration. We favor a statefull firewall for its speed and rule simplicity because of its state table. Proxy based firewalls tend to provide superior security when the content of the data payload is significant, such as for common Internet services, such as http, ftp, smtp and DNS. NetScreen 208 and Netfilter/IPTables were both serious options in our design. We believe the NetScreen to be very effective and have an extensive management and reporting capability. Netfilter/IPTables is also an excellent product with great logging features and an effective statefull design like the NetScreen. Another player in the firewall field is Sun's SunScreen product. Whereas, it has only a minor market share, it has proven to be very effective and fast as a statefull firewall. It has a configuration option to define some interfaces in a stealth mode providing statefull inspections, while other interfaces can be configured with IP addresses in proxy mode. It also has good central management facilities.

Our choice here is to use a SunScreen v 3.1 firewall running on a Sunblade 2000 with Solaris 8. The most current version of SunScreen is 3.2, which comes bundled with Solaris 9. We chose to go with the older version because of our familiarity with product and to keep all Sun systems at Solaris 8. The SunScreen product permits us to configure the same system as both a statefull and proxy based firewall at the same time. We will use the statefull service for all interfaces. There are no IP addresses or ports associated with any network interface within the statefull configuration, which makes it very stealthy. In fact, it sits in the

middle of a subnet with the same network address on all interfaces and no ports. This sometimes makes it interesting in certain configurations.

Sun Microsystems designed SunScreen to operate between layers 2 and 3 in the ISO model. When it is configured in stealth mode there no IP stack is associated with the interfaces. Packets are forwarded or dropped on a first match rule opposed to a best match rule as found in some other firewall products.

If we used both stealth and proxy interfaces on the same firewall, we would assign IP addresses and open ports for the proxy interfaces only. In addition, we would have to add a router on a loop between a stealth interface and an addressable interface. Upon careful review, we determined the option for proxy filtering on the SunScreen firewall did not gain enough to justify the added complexity.

The external firewall (FW-1) provides our second line of defense and the primary means of enforcing most of the network security policies relating to Internet access. It also splits the assigned GIAC public address space, 223.223.223.0, into three "subnets", EOSN, ERSN and the internal GIAC or inside network. If we had used proxy firewall interfaces, we would have subnetted this address space with a 27-bit network mask, creating six subnets of 30 usable addresses each. We did assign IP addresses with this in mind, in case we elected to use proxy interfaces in the future.

The external SunScreen firewall (FW-1) will provide both dynamic and possibly limited static NAT services. This will leave the external open and restricted service networks with the public IP addressing scheme for GIAC Enterprises.

Significant system abuse and misuse originates within an organization from curious and/or disgruntled employees. In many situations, it is difficult to stop this type malicious activity since the employee may be authorized access to criteria devices and information. However, we still need to limit this activity and potential secondary losses if an outside cracker is successful in penetrating the network and manages to compromise a system. We have specified an interior SunScreen firewall running on a Sun Sunblade 2000 with Solaris 8 in statefull mode. This firewall will create many of the required security "subnets".

All of the SunScreen firewalls will be attached to a separate central management network, which is not connected to any other network. The NICs used for this connection will have to have addressable IP addresses assigned. A Sun Sunblade 2000 workstation, running Solaris 8 will be used as the central management console. There is no known vulnerability to access this central management network via traffic passing though the

SunScreen. Consequently, this private network does not provide a backdoor around any other security measure.

VPN Server:

This is most difficult portion of the network to design and one of the most important. Business requirements demand integrity and loss prevention of the fortunes. We could use IPSec across the Internet with all of our suppliers, partners, technical staff and traveling sales team. Except for the GIAC employees, these groups have expressed strong reservations requiring use of particular hardware and expensive technical support to establish and maintain IPSec services with GIAC Enterprises.

Our VPN solution for suppliers, partners and customers will be done via scp. The business requirements include authentication, confidentiality and ease-of-use. scp version 3.5 will meet these requirements. Whereas, IPSec permits any protocol to be tunneled through a connection, ssh will not, unless required ports are open. We will lock this server down very well and deny port forwarding.

This secure shell server will be a clustered system based on Red Hat Advanced Server 2.1. This will help to ensure order completion even when the ssh server is down for planned work or otherwise. We will run ssh/scp with tcpwrappers to log transfer histories.

A supplier/customer may use an open source version of scp or a commercial version as desired, provided it is compatible with version 2 to avoid the known issues with version 1. Whenever a supplier is ready to upload new fortunes, they may simply use scp to transfer these to their assigned user account and directory on the ssh server in the External Restricted Services Network. A cron job initiated from the internal Oracle database server will periodically sweep all the suppliers' directories and use scp to transfer the new fortunes to the internal fortunes Oracle database server for review by the Quality Control unit. After confirmation of the sweep process the external ssh server will deleted any remaining new fortunes older than a given time stamp.

After customers or partners place an order with GIAC, they are notified by e-mail that their order is ready for pick-up at the GIAC ssh server. They will use scp to pull any files present in their assigned directory to fill their order. Again, a cron job will periodically delete any fortunes in a customers' directory after the tcpwrappers log shows a successful transfer.

This VPN solution and physical location removes any need for an Internet user to access any data or server within the GIAC private IP address space.

GIAC sales and technical staffs will be required to use a VPN service when remotely accessing the internal GIAC network. We considered using ssl as the preferred VPN method because of its simplicity. Due to some of the recent problems with ssl, we have chosen to use IPSec/ESP as our VPN solution for GIAC employees. Each GIAC employee remotely accessing the internal GIAC network is required to use a dedicated hardened computer provided by GIAC Enterprises. IPSec will run on the remote host and a VPN gateway at GIAC Enterprises. The question is where to place the GIAC VPN gateway.

We are not anticipating a heavy or even constant IPSec traffic from remote GIAC employees. Therefore, loading factors will not be as important as if all remote traffic were to use the VPN gateway.

Placement of the VPN gateway requires careful review of the devices used in the outer defensive layers, particularly since we will be using NAT on the external firewall, FW-1.

We are using a VPN gateway for GIAC employees, IT support and sales, to access many different internal devices and perform very sensitive work remotely. We need to be able to tunnel a wide variety of protocols through the VPN connection. AH is not even a consideration, since it only provides authentication and no encryption services and will not work with NAT, because the packet headers are rewritten. ESP in tunneling mode will provide both encryption of the data payload plus authentication of the header information. ESP will require IP ID 51 and isakmp on udp port 500.

To provide additional access security for all GIAC VPN users, we will require a two stage authentication process. We had considered using a standard SecurID card solution, but believe the efforts to keep the cards in sync with the server and the DoS attacks makes the SecurID card a poor choice in this application. As an alternative solution we selected iPass, http://www.net-roamer.com/index.htm. iPass is based on SSL has access points from over 150 countries world wide. To simplify management, we will take advantage of the outsourced authentication services. This eliminates the requirement for us to maintain a RSA/ACE server in the background. All of our road warriors will be able to establish a very secure channel with this solution. The second authentication process is a standard user ID and password pair for the VPN server

Another option is to place a VPN gateway in the External Restricted Services Network (ERSN) along with the ssh/scp server. This network is part of the public IP address space 223.223.223.0 assigned to GIAC Enterprises. On a SunScreen firewall this traffic will be directed to the

ERSN interface prior to any NAT. Therefore, all of the traditional IPSec problems with NAT are nullified. This approach places both of our VPN servers on the same "subnet".

We will use a dedicated server for each of the VPN functions because we are permitting GIAC IT employees secure shell access to any server on the GIAC network. The IPSec server will be built on Red Hat 7.3 using freeware S/WAN v 1.99, which is the most current at this time. At this time FreeS/WAN has not been ported to Red Hat Linux 8. This product is interoperable with many other common IPSec implementations, such as Windows 2000, http://www.treeswan.org/treeswan_trees/treeswan-f.99/doc/interop.html. We have chosen this product as the IPSec solution for the GIAC IT staff's remote systems as well.

SSH/SCP Service:

Remote system access is always a significant security problem. Yet to conduct business, you must permit limited information transfer with "trusted" systems.

GIAC Enterprises strictly prohibits the use of telnet and rlogin. As replacements, we will use Open ssh/scp version 3.5, http://www.openssh.com/, on all systems. This will avoid the vulnerabilities known in version 1 of ssh/scp. If a router or switch does not support ssh2, we will use direct connect consoles on those specific devices. ftp is not to be used anywhere on the production network, except to reach the external ftp drop box from desktop workstations. All other internal file transfers will be done via scp.

We will edit sshd.config to permit only GIAC IP addresses and specific users to use ssh/scp, except on the external fortune scp server. X-Windows forwarding and ssh tunneling will be blocked as well in the configuration files. For a little obscurity, we will change the ssh port from 22 to 65322. Port scans frequently look for tcp/22, but are not likely to reach into the high port numbers.

We will also utilize the additional security features of ssh, such as:

- Use of DSA instead of RSA keys.
- Strict mode enabled.
- Banners set to a meaningful not welcomed message.
- Authorized users set to the appropriate list of users for that particular system, where possible.
- Disabled root login.

Database Solution:

Oracle Database 9i version 9.2 will be used for implementation of all major database applications. We will utilize a classic three-tier model, workstations with a web browser, a web/application server and a database backend. The application server(s) will use Oracle Application Server 9iAS version 9.0.3. Oracle Listener will be located with the database engine, using port 1521. Since we are running Oracle in a single threaded configuration, we will not need any other port opened. All Oracle database products will run on Sun Sparc systems using Solaris 8, 02/02 edition that is the most current version of Solaris 8 as of this writing. Even through Solaris 9 has been released for some time now, we have selected the older version because of our experience and confidence in the product.

Web Servers:

Web servers remain a favorite target of crackers. By using Apache as the primary web interface we have eliminated many problems associated Microsoft IIS products. However, recent problems to be considered include the Apahce/mod_ssl worm, Apache Chunk Handling Exploit and CGI scripts in general. http://www.redhat.com/support/alerts/linux_slapper_worm.html.

There will be public or Internet web servers located on the External Open Services Network and intranet or private web server(s) on the Interior Open Services Network. Both Internet and intranet web servers will include static and dynamic pages acting as the front end of the custom Oracle applications.

We have selected Red Hat Advanced Server 2.1 as the operating system on a Dell 2650. Since we will be using Oracle Application Server 9iAS version 9.0.3 on this box, we will take advantage of the Apache product bundled into the Oracle Application Server. This Oracle product is certified to run on the above OS version. Consequently, we expect to have a very stable, integrated and manageable environment.

We did consider using Stronghold Enterprise Apache or even a standard edition of Apache. However, we believe if the proper security patches are installed, the advantages of a fully certified and integrated environment outweighs benefits of the other options. At this time, Red Hat has not published any security related patches for the Advanced Server product.

To help ensure 24 x 7 operations all web servers will be clustered systems. Red Hat Advanced Server is specifically designed for high availability enterprise situations where clustering is a requirement.

Using the most recent versions of Oracle Application Server and Apache, with all relevant security patches, we expect to resolve the problems with the Apahce/mod_ssl worm and the Apache Chunk Handling Exploit. We will also change the default Apache http response to provide miss-information by adding "ServerTokens Prod" to httpd.conf. Changes to the httpd.conf file will limit directory access by the web server. Apache will be run in a chroot environment with a user name of "nobody". All web files will be root owned and write access only by root. ACLs for executables will be set at 755 and web page content set to 744. All Server Side Includes will have "IncludesNOEXEC" set. This will reduce the amount of damage a cracker can do if an exploit is successful.

All default CGI scripts will be removed in the build process and strict CGI script standards enforced. We will install a VeriSign certificate on the public web server to provide a measure of authenticity to external users. At this time there is not sufficient reason to include a certificate server in the GIAC network, we believe it represents more security risks than benefits. No GIAC web server will be permitted to establish a web session to another web server.

As an extra measure of protection for the external web server, we will use Squid v 2.5 as a reverse proxy server, with the Jeanne plug-in module, in front of the public web server to improve performance and validate all incoming http and https packets. This will force all remote clients to establish an http(s) session with the proxy server, thus preventing any direct traffic between the client and the web server. Starting with version 2.5 squid has the ability to terminate ssl sessions and perform standard filtering operations, http://www.squid-cache.org/. It will then check and validate all URL requests via filtering and access controls. If the request contains anything except expected traffic, the proxy should drop the packet, thus providing another layer of defense for our web server. This process severely limits which files and directories that are accessible to a remote client. The Squid server will listen on port 80 and the squid.conf file includes:

```
http_port 80
```

Other standard best practices as listed in the current SANS/FBI Top 20 List includes:

- All default CGI scripts installed at built time will be removed.
- Ensure best practices are observed by all CGI programmers. A chef-
- Remove all compliers, interpreters and un-needed libraries.
- Use CGI alerting scripts.

- DO not allow Directory Indexing and Servers Side Includes, this provides un-necessary information.
- Do not allow the server to follow symbolic links.

As an extra caution, we will create a small cron job that will run at small random intervals, to create a variable and random load on the CPU. This may help avoid some techniques to break ssl encryption.

DNS:

DNS reconnaissance is a common first step in preparing for an attack. Common problems include old vulnerable versions and misconfigured servers. DNS problems have consistently made SANS Top 10 and 20 lists.

A split DNS system will be used to protect the interior network structure from hostile Internet users. Two DNS servers will be placed in the Exterior Open Services Network (EOSN). These servers will be configured to answer only non-recursive queries about specific servers in the exterior networks, such as the proxy, ftp, mail gateway and VPN servers. There will not be any entries relating to the interior network on these servers. A major drawback f this design, is that the DNS server reside on the same segment as the ftp, mail and web servers. We considered, but rejected, the idea of creating a dedicated DNS segment off the external firewall. It is not clear that the complexity justified the potential gains at this time.

By making both of these servers “masters” we can eliminate any need to allow zone transfers and therefore close tcp port 25. We will also gain a measure of redundancy for the GIAC public address space. This can avoid the loss of responses from a secondary DNS server when the primary has failed and the time-to-live has exceeded on the secondary, rendering the exterior network dysfunctional.

One of the two servers will still be considered a primary in terms of manual updates. This will require a forced database transfer and restart of the named process on the second master server following any database modification. We will accomplish by a script that will edit the required database files, scp the files to the second server and use ssh to force a restart of the named process. By requiring the operator to enter a password for ssh and scp commands we have not created any open trust issues. Particularly, on the exterior DNS servers, this arrangement should not pose any significant workload, as there will be very few changes made to this database once the network is operational. Configuration files for these servers will include lines to give out misleading version information. This will cause trouble for some automated reconnaissance tools.

```
options {
        directory "/var/named";
        query-source address 223.223.223.65 port 53;
        version "foobar";
        interface-interval 0;
        cleaning-interval 180;
        listen-on { 223.223.223.65; };
};
```

Two intranet DNS servers will be implemented on the Interior Restricted Services Network. To provide a measure of redundancy both DNS servers will be configured as masters for the GIAC private address space in much the same manner as done for the exterior DNS servers. Configuration files for these servers will include lines to use specific forwarders outside of the GIAC address space, to answer queries recursively and give out misleading version information. This will cause trouble for some automated reconnaissance tools.

```
Options {
        #recursion only for GIAC
        allow-recursion { 172.16.0.0; 172.17.0.0; };

        forwarders {1.2.3.4; 1.2.3.5; };
        # 1.2.3.4-5 is the DNS  IP Address of our ISP
}
```

This solution allows us to deny all tcp/53 traffic anywhere on the network, which effectively shuts down ability or need to perform zone transfers.

We will not permit DNS traffic between the exterior and interior DNS servers. The interior DNS servers will include records about all servers located in the exterior segments. All DNS servers will be built on an Dell platform with Red Hat 8 using BIND version 9.2.1 from http://www.isc.org, which is most the current at this time. The named process will run as user "named" in a chroot environment. To help reduce DNS related information flow between the exterior and interior, any IP address/hostname resolution required by servers in the exterior zone will be based exclusively on local /etc/hosts files. Again, since this portion of the network should remain static, the manual editing overhead should not be excessive.

Network Time Protocol:

It is imperative to keep all systems and network device clocks synchronized to perform useful log cross checks and validations. The two interior DNS servers will be used as local ntp servers. Both of these systems will synchronize with three different external stratum two clocks. First external clock (140.221.9.20) is common to both ntp servers and both

use a unique second external clock (140.221.9.6 and 128.4.40.12). We will provide the normal customary e-mail notification to the stratum 2 service providers.

The latest NTP software from www.ntp.org is version 4.1.1. We will run the ntp deamon in a chroot environment with user ntp.

These two local ntp servers will consider each other as peers and synchronize between themselves and the stratum 2 clocks. This will at least keep these local servers synchronized in the event connections are lost with the stratum two clocks.

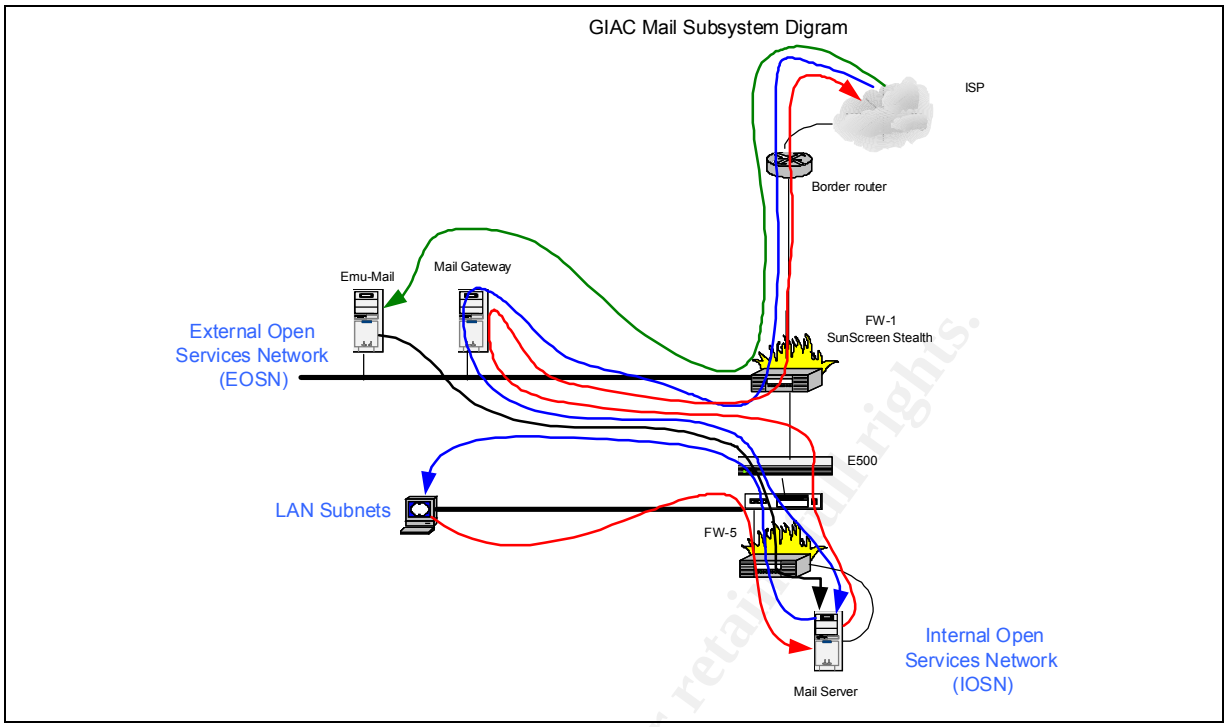
All other network devices will synchronize with these two servers, including the servers on the exterior networks.

NIDS probes will not have direct access to the two-ntp servers. Instead, the NIDS Management Console will synchronize with the two-ntp servers and will act as an ntp server for the NIDS network. This will reduce the presence of the NIDS devices to any sniffers, which maybe installed on the production network.

Mail Services:

Some of the earliest exploited Internet vulnerabilities included smtp, and it has remained a problem for many organizations. The most common problem is the use of older and/or un-patched versions of sendmail and similar products that permit buffer overflow conditions. Other common problems include un-intentional forwarding of mail, mail relaying, and SPAM to third parties as well denial of service conditions typically caused by out-of-spec attachments and viruses.

GIAC e-mail services are rather large and complex. The requirements included normal internal and Internet e-mail for all GIAC employees and that all employees should have secured access to office e-mail from the Internet. Also all e-mail is to be scanned for viruses and un-authorized attachment types, including .exe, .cmd, .bat, .pif, .lnk, .com, .vxd, .exe, .dll .vbs, and .vb, at both ingress and egress points. There are longer lists of potentially hostile attachment types available from NSA and Microsoft. These types will be allowed for internal GIAC e-mail.



Sendmail has traditionally been used for mail relay and mail servers. Even though sendmail is a very mature product and is reasonably secure when the latest patched versions are used, it is still old and was designed when the Internet was not particularly hostile. Consideration was given to qmail v 1.03 from http://www.qmail.org for both the mail relay and MTA. Whereas, qmail is a very fine product it does not include the level of virus and content filtering desired in a single product. We have opted to use Microtrend's InterScan Message Security Suite v 5.1, most current at this time, as the inbound e-mail relay in the External Open Services Network (EOSN). This product will provide a wide variety of ingress and egress security services at the perimeter. It may be configured to discard SPAM, prevent anti-relaying, check for viruses, including multiple levels attachments, prevent certain kinds of DoS attacks and content filtering. Finally, it will forward all GIAC e-mail to the internal e-mail systems. The firewall will forward all outbound e-mail only to this server prior to it being put on the Internet. This system will be built on an Intel server running Red Hat Linux 8. Resolution for external host names is required for this system. Since the DNS servers in the external networks will not resolve external addresses and we do not want out internet DNS servers communicating with our external service networks, we will set the /etc/resolv.conf file to use the DNS servers of our ISPs as per written agreements. The configuration files will be set so as not to give out any version information.

The next hop in the mail process is a McAfee E-500 transparent proxy server, which is a sealed turnkey system. This system will check all in-

bound and out-bound e-mail for viruses, including within attachments. The E-500 will also check all ftp and http traffic for viruses as well. It can do this by reassembly of the entire message prior to sending it onto the next hop. We considered using Trendmicro's InterScan AppletTrap in this application. This option was attractive to maintain consistency in a product line, but we felt using a second anti-virus product in series is worth the added management effort. The E-500 also provides McAfee's anti-virus scanning service for http and ftp traffic as well.

Finally, all e-mail is delivered to a Red Hat Linux 8 server running Microtrend's InterScan Message Security Suite v. 5.1 and Qpopper version 4.0 from http://eudora.com, which are the most current versions at this time. All users' desktop systems will be configured to use this server as both their in-coming and out-going e-mail server. This design will ensure all e-mail at GIAC Enterprises is checked and filtered the same way. The reason we are using the same InterScan product as the gateway and inside mail server is to allow different policies and to ensure all e-mail is examined according to potentially changing policies. Desktop systems are configured not to leave read mail on the server.

The finial requirement for the e-mail system is to permit all GIAC employees to get their e-mail while away from the office. Several options where considered, including use of IMAP4 and setting up secured tunnels from their remote systems through the firewall and into the e-mail server. This option was ruled out because of the ease of other protocols being tunneled equally well on potentially un-trusted systems.

The solution is to use a product called Emu-Mail version 5 that runs on an Apache web server under Red Hat Linux 8. Any GIAC employee, using https can connect to this web server only from the Internet, provide a user ID and password and gain access to any e-mail still residing on the e-mail server. The entire process is secured by a https session between the users' browser and the Emu-Mail server. Any mail read this way is left on the POP3 server, so it can be downloaded to the users' desktop upon return to the office. Some of the advantages of this system is that all new mail or responses from a GIAC employee will use the same filtering processes regular GIAC mail does.

FTP Services:

FTP has always been one the more troublesome Internet services. In part due to the use of separate command and data ports and the nature of the service itself, particularly, if anonymous ftp is permitted. There is a multitude of DoS style attacks that might be used against an ftp server.

From experience, it is known that some GIAC business e-mail requires transfer of large attachments that exceed the maximum 3 MB limitation

GIAC imposes. In order to accomplish this function, an ftp drop box system will be installed in the Exterior Open Services Network. Since there are a limited number of business partners requiring this service, they will be issued individual guest accounts on this ftp server. Anonymous ftp logons will not be permitted.

This system will use Red Hat Linux 8 and the latest version of wu-ftp, version 2.6.2 from http://www.wu-ftpd.org. /var/ftp will be a separate file system, sized sufficiently large to preclude an accidental DoS situation from normal use, about 30GB on a large disk. All directories within /var/ftp will be set for minimum permissions required, typically 550. A cron job runs nightly, deleting all files older than 7 days. From the /var/ftp/upload directories. We will also limit depth of directory creation, number and size of individual files, rate of connections, etc. TCP Wrappers will be used to future limit and log access to this service.

Another technique useful in protecting ftp uploads to a public directory is to create a "blind" directory with permissions set to rwxrws-wt (chmod 3773), and ownership at ftpadm:ftpadm. This results in a trivial error when using ls or ls –al commands. In order to get a file from such a blind directory, a user must know the full pathname of the target file. This name might be transmitted to the receiving party via e-mail or a phone call.

To avoid some of the most common current exploits we again follow the best practices as setout in the SANS/FBI Top 20 List:

- Add usernames all system account named and "ftp" and "anonymous" to the /etc/ftpusers file.
- Delete the "ftp" user name from the /etc/passwd file.

Desktop Systems:

Desktop systems can be the Achilles heel of a network. They are frequently setup and configured by the manufacture and/or the end user. It is common to find un-necessary and un-patched software, default passwords and out-of-date anti-virus and firewall products if any at all. Collectively, these weaknesses can provide a cracker with a base from which to launch an attack against other systems.

We have chosen a diskless Linux solution for most desktop requirements. A few standard MS Window 2000/XP systems maybe required for specialized applications, such as high-end graphics in the marketing unit. These Linux systems will load their operating system and applications from two dedicated OS and application server clusters, running Red Hat Advanced Server 2.1. All data files will be stored and served from a dedicated NFS server cluster based on Red Hat Advanced Server 2.1.

These server cluster clusters will be located in the User LAN Zone on a dedicated VLAN. We have selected this option over placing then behind the internal firewall for purely performance reasons. As with all GIAC servers these systems will be well hardened and can be afforded some additional protection from ACL capabilities on the Cisco 6506 switch. We have chosen clusters in this case to reduce impacts o the user desktops when one of these three servers fails or is being updated.

Clustered servers will permit us to install patches, updates, etc. on one server and verify its operation while the second server continues production work. After testing is completed, the "older" cluster unit can be synchronized.

This solution does mean the use of some questionable protocols and services, pxe, tftp and NFS. However, by design we have isolated these protocols to only the User LAN Zone, thereby providing some isolation from our main production servers. tftp and port tcp/69 & udp/69 will be activated only on the OS server and blocked at every firewall. The NFS server will be configured with a directory structure that closely matches GIAC's business units. Export controls in /etc/exports will limit directory access only to the appropriate business unit subnet IP addresses, for example:

/data/management 172.16.2.0/24(rw,no-root-quash,insecure)

We considered using a single cluster for all three of these servers. Testing suggests that a large file transfer with NFS could cause a noticeable performance hit for OS and application downloads. Therefore, we felt it best to use three dedicated server clusters. We believe that the improved control of the desktop environment is worth the potential risks of permitting these protocols within this single network zone.

Experience has shown that open software solutions for standard office requirements are functionally equivalent to Microsoft products and permits document interchange. Design strengths include consistency and verified software from the OS level up, greatly reduced problems from viruses, centralized storage and controlled sharing and backup of all data files. We have nearly eliminated desktop virus problems since 1) all mail must pass through one or more mail systems with anti-virus software; 2) all files are stored on central server with anti-virus software; 3) users lack anyway to install new software and 4) use of non-Microsoft office solutions (Outlook and Internet Explorer).

Many of the desktop design concepts presented here are part of a product called IDEAS from Integrity Networking Systems, Inc., http://www.integrityns.com/.

The few MS Windows 2000/XP systems will be protected with McAfee PC Security Suite (anti-virus and firewall), properly patched with the MS update services and hardened in accordance with Windows NT Security Step by Step and Microsoft's new Window 2000 security guidelines, "Microsoft Solution for Securing Windows 2000 Server" http://www.microsoft.com/technet/security/prodtech/Windows/SecWin2k/Default.asp. We will also treat all notebooks computers assigned to the sales team in the same manner.

Intrusion Detection Systems:

NIDS devices will be placed at choke points on most security subnets. The notable exception is the user subnet. Since there are NIDS on every other adjoining segment, user subnet traffic can not get anywhere without passing at least one NIDS.

Besides using NIDS to alert for potentially hostile traffic, we have found it very beneficial to identify miss-configured systems and network services.

All NIDS devices will utilize Snort 1.9.1 from http://www.snort.org, the most current at this writing, running on Red Hat Linux 8, the most current at this writing. These systems will have P4 1.6 GHz CPUs with 1GB memory and a 30 GB system disk and 72 GB logging disk. Each system will also have two NICs. eth0 will be connected to the man and an man the private NIDS management network. All NIDS devices will be attached to the network with 3Com Model 500 hubs (3C16610), these are inexpensive rack mount hubs. The second NIC, eth1 will not have an IP assigned and be placed in promiscuous mode. By not assigning an IP, the system cannot put datagrams on the network and thus becomes largely stealth. These systems can also be attached to the monitored segment via a stealth cable, with the transmit pairs cut and looped back. This cable design works well when attached to most hubs, http://snort.sourcefire.com/docs/faq.html.

Each Snort probe logs detects to an individual file system remotely mounted from the NIDS central management system. The local logging disk is used to store the first 100 bytes of binary data for each packet. These local log files are rotated on a daily basis and kept on-line for a maximum of seven days. Just after midnight every day the most recent binary log file is transferred to the NIDS Central Management System for inclusion in the daily backup. The following two commands are used to start or restart the two required Snort processes.

```
Snort –D –b –l eth1 –l /var/log/snort
Snort –c snort.conf –D –l eth1 –l /var/log/snortext
```

The NIDS Management Network connects to a dedicated central management console. This system is also built on Red Hat 8 with the same hardware configuration as the probes have, except it includes two 120 GB mirrored drives for logging and a DLT tape array. To reduce the Snort logs to a manageable format, we will run SnortSnarf v 1.9, from http://freshmeat.net/projects/snortsnarf/?topic_id=245%2C43. Apache will be used to provide the required web server functions of SnortSnarf. All SnortSnarf logs are kept on-line for 90 days prior to being deleted. They are archived daily on DLT, in case older records are required.

Use of a private NIDS network largely precludes the possibility of alerting a cracker to the existence of NIDS traffic and possible attempts to disable or cover their tracks. Since we have prevented any transmission from our NIDS probes on the production networks, we expect to be able to observe and/or capture most hostile traffic.

Normal NIDS operation will not translate IP address to host names, thus avoiding generating DNS traffic that might alert a cracker that he is being watched. The only time when a cracker might see telltale-monitoring traffic is and when a GIAC security person requests a IP address lookup from say http://www.geektools.org.

As stated earlier we will use the commercial version of Tripwire v 5.1 on all Solaris and Linux servers as a host IDS. Tripwire also markets a version of the product for common network devices. We will place Tripwire for Network Devices v 5.1 on all Cisco equipment. To manage all of these tripwire clients we will use Tripwire Manager on a system in the syslog network. This manager can be configured to send alerts to another alerting system. Copies of the file signatures for Tripwire will be generated while new systems are still on the Build Network and these will be burned to CD-ROM and transferred to the Tripwire Manager for scheduled comparisons. This subsystem will permit us to know whenever files have been modified, either intentionally or otherwise.

Syslog Server:

System logging for all servers and most other network devices will be recorder to a central clustered syslog server. We will user syslog-ng 1.5.23 from www.baliatbit.hu running on Linux 8.

A good network wide syslog solution is important to cross validate events and maintain an accurate record of activity. Frequently, crackers will attempt to discover what type of logging is being done and how they might cleanse their tracks.

In designing the syslog network, we have the option of using an in-band or out-of-band solution. The simplest option is to route all syslog traffic on

the regular production network. The disadvantage is that crackers only have to sniff the traffic to understand what is happening and start covering their tracks. The out-of-band solution removes the syslog traffic from the product network and places it on a private syslog network connected to each device by an additional interface. The design must also consider implications of this design on the various firewalls and their rule sets. Frequently, the first targets of cracker will be the exterior servers, www, ftp, mail & DNS. A hybrid solution might allow a private syslog network on the exterior subnets passing through the Internet firewall on a dedicated interface and merged with other syslog traffic once on the inside network.

Another option designed to throw crackers off-track is to use both systems. Thinking is that if they see syslog traffic on the production network, all they got to do to shutdown down that one stream. While the syslog traffic on the private network might continue to record their real actions. This is based on the assumption they can sniff the production network and have not root compromised a server to be able to see the syslog configuration, at which point the decoy strategies become obvious.

For this network, we have chosen the simpler solution to route all syslog traffic through the production network.

Regardless of the routing of the syslog traffic, the syslog server will be well hardened and isolated on a dedicated segment behind the interior firewall. We will run swatch v 3.0.4 from http://www.oit.ucsb.edu/~eta/swatch/swatch-3.0.4.tar.gz on this server to assist in alerting significant events. Such alerts are sent via e-mail to the alert server. A recordable CD drive will be added to this host to permit the monthly archiving of log files.

To make remote syslog work we must start syslogd on the syslog server with:

```
#Start central logging
syslogd –r
```

in the /etc/init.d/syslog file, otherwise the syslog server will not accept incoming log data.

On each non-syslog server we must start syslogd with:

```
#Start central logging
        syslogd –h
```

to permit forwarding of syslog data. Then in the /etc/syslog.conf file we must add:

. @172.17.1.231

to make each server transfer it's logs files to the central logging server. If a cracker gains access to the syslog.conf file, they will understand how GIAC is handling logging and may try to defeat it.

Build Server:

We will build our servers built using pre-tested installations that reside on a build server opposed to a fresh install from distribution media each time an operating system needs to be installed on a system. This represents a significant time investment up front in doing test builds and verifications, but we believe that this effort is justified by time savings and compliance with the security policy when systems have to be rebuilt quickly. The real value is the consistency between systems and reduced time spent on hardening each new or rebuilt system. A side advantage is that if file systems on some servers are setup right, we reduce the need to perform tape backs of the standard OS file systems, as we have a master copy on the build server.

Since we are using both Solaris and Red Hat, we actually need two build servers. Solaris Flash Archive, http://www.sun.com/solutions/blueprints/browsesubject.html#jumpstart successor to Jumpstart, will be used to build all Solaris systems and will run on a Sun V120 Netra, Solaris 8. Kickstart will be used on the Linux build server. Our build server will use Linux 8 on a small Dell system. These systems need ample disk space and an independent backup system.

It is important that systems being built are not attached to a production network, particularly the Internet, until the new server is fully configured and tested. Consequently, the build servers will reside on an independent network. These servers will also house a Tripwire database of all system configurations. By burning this database to CD-ROM disks, we can perform periodic tests on the production servers to determine integrity of our system files.

To test standard network functions, such as DNS, ntp and syslog, we will add these services to the build servers. By using multiple interfaces and the same IP addresses for these standard network functions as occurs on the interior production network, we will be able to test and verify most network interactions. In addition, we will add a Linux and Windows 2000 workstation to the build network permitting us to test functions such as web servers, DNS, ntp and syslog services when we are building these systems. To avoid confusion when testing such servers, that have a

conflicting service on the build server, we will temporarily disable that service(s) on the build server.

User Passwords

In this network design there are seven major categories of user accounts and passwords. Our password policy requires different passwords for the same user IDs on exterior and interior systems. So that system administrator's and root accounts for systems in the exterior network may not be the same as on any system on the interior network. This will provide a measure of protection against a cracker that might achieve a root compromise on an exterior server from easily rolling-over an interior system.

Passwords on security devices, firewalls, routers, switches, NIDS, syslog servers, etc. must be different from any password on a development or production class server.

With the exception of the ftp drop box and ssh server in the exterior network, users do not have system logon accounts outside of the firewall. These ftp user IDs are non-obvious derivatives of their user IDs used on interior systems to simplify memory issues. Otherwise, user IDs are restricted to their desktop systems, mail and file services.

The last category of user IDs are used for access within the Oracle applications. These user IDs are also different from any system logon ID.

All UNIX OS passwords must be at least eight characters in length, Window passwords are required to be exactly seven characters in length, and constructed from mixed alpha/numeric and special characters. Procedures encourage staff to substitute special characters for some letters. For example "bluedogs" might read "6|ue|)0g$". Testing with John the Ripper v 1.6 shows passwords constructed in this manner are very difficult to crack and are still easily remembered. This is important since all passwords are required to be changed every 90 days. All passwords are run though John the Ripper shortly after the mandatory change date to help identify staff who have not yet mastered the fine art of password crafting.

A system ID checklist is used to assist in making sure all devices on the network have their passwords changed at the same time. This helps to prevent the situation of forgetting the password to that one system stuck somewhere that was not changed for the last two cycles.

This password scheme helps to build a defense-in-depth, in that penetration at one level does not necessarily provide an easy path to other network resources.

Anti-Virus:

Anti-virus requirements cut across nearly all servers and desktop systems at all network levels. Experience has shown benefits in using more than one vendor's anti-virus solution, particularly in the early hours of a major virus outbreak. This provides significant defense in depth for virus problems. We have already noted the use of a McAfee E-500 transparent proxy inside of the firewall. In addition, we will use Trendmicro ServerProtect v 5.0 for Linux to protect all of the Linux servers. Historically, virus protection for Linux servers has frequently been ignored because of the few wild Linux viruses. We expect to see more Linux viruses as the popularity of Linux continues to increase.

All anti-virus systems will be configured to check for updated vendor files on a daily basis, even if most of the vendors normally provide weekly updates.

SNMP and RPC Services and R Commands:

We believe that proposed production server network design does not require the use of snmp or rpc services. Please note RPC and NFS are both required in the user LAN and NIDS networks. Therefore, these services and all associated ports will not be installed and/or active on any network device. For the server networks, this will eliminate the number one, four and six security vulnerabilities of the current SANS/FBI Top 20 List.

All of the network equipment will be installed in one of two secured facilities, the main production facility or the Disaster Recovery Facility. In our opinion what might be gained by use using snmp is probably overshadowed by the associated risks. RPC and related protocols is not needed.

All the r commands, rsh, rlogin, etc., will not be allowed on any system. The software should never be installed and all related ports closed, except as needed by other required applications, e.g. 514 for syslog.

Physical Security:

Network and Server Equipment:

All network devices and servers will be located in restricted areas protected by self-closing doors with combination locks. Only GIAC IT staff will have access to these areas. All air conditioning and electrical supply equipment will be located in an adjoining but separate secured space to reduce systems access by service personnel.

GIAC IT staff:

GIAC IT database and operations staff offices will be located in a restricted protected by self-closing doors with combination locks. This will minimize the opportunity for non-authorized personnel to observe or gain access to restricted workstations and other information.

Assignment of IP Addresses and Networks:

Subnet and Address Requirements:

For the purpose of this Practical Assignment, we will assume GIAC Enterprises has been given the IANA reserved IP address space of 223.223.223.0/24 and will use the private address space as defined in RFC 1918 of 172.16.1.0 – 172.20.1.255. This will provide a maximum of 254 routable addresses for our external subnets and sufficient addresses to create our internal subnets. We will subnet the 223.223.223.0 address space into six subnets with a 27-bit mask, 225.225.225.224. This will allow 30 hosts per subnet or 180 hosts total, which should be sufficient for our design requirements.

Since GIAC's partners, suppliers and customers are wholly independent business operations, we will not assume any particular IP addressing scheme for these external businesses. In addition, GIAC employees who have authority to access internal hosts from the public Internet will be assumed to be using IP addresses assigned by their respective ISPs.

Zone	Subnet Name	Network Address	Ending Address	Subnet Mask
Public	Router & ERSN	223.223.233.32	223.223.223.62	255.255.255.0
Public	EOSN	223.223.223.64	223.223.223.94	255.255.255.0
Public	NAT Block	223.223.233.50	223.223.223.59	255.255.255.0
User LAN	IT	172.16.1.1	172.16.1.254	255.255.255.0
User LAN	Management	172.16.2.1	172.16.2.254	255.255.255.0
User LAN	Marketing	172.16.3.1	172.16.3.254	255.255.255.0
User LAN	Quality Control	172.16.4.1	172.16.4.254	255.255.255.0
User LAN	Receivables	172.16.5.1	172.16.5.254	255.255.255.0
User LAN	Payables	172.16.6.1	172.16.6.254	255.255.255.0
User LAN	Sales	172.16.7.1	172.16.7.254	255.255.255.0
Secured Server	FW-2	172.17.1.1	172.17.1.9	255.255.255.0
Secured Server	IOSN	172.17.1.10	172.17.1.99	255.255.255.0
Secured Server	IRSN	172.17.1.100	172.17.1.199	255.255.255.0

Secured Server	Database	172.17.1.200	172.17.1.220	255.255.255.0
Mang. & Security	NIDS	172.17.1.221	172.17.1.225	255.255.255.0
Mang. & Security	Syslog	172.17.1.230	172.17.1.239	255.255.255.0
Secured Server	Developmen t	172.17.1.240	172.17.1.249	255.255.255.0
Build	Build	172.17.1.1	172.17.1.254	255.255.255.0
Mang. & Security	FWCM	172.18.1.1	172.18.1.254	255.255.255.0
Mang. & Security	NIDS	172.19.1.1	172.19.1.254	255.255.255.0
Gateway	Gateway	172.20.1.1	172.20.1.254	255.255.255.0

As explained previously stealth SunScreen firewalls sit in the middle of a subnet and do not have IP addresses assigned to the interfaces. The central management interface and any interfaces configured as a proxy are the only exceptions. In this model the group of devices off a single firewall interface does not constitute a true subnet, but is really just a range of addresses from the subnet that includes the firewall. We use the term subnet in this situation for our connivance.

The User LAN Zone is segmented by a level three switch, creating a series of VLANS, one for each of the subnets as noted in the above table. We have made each functional business group its own VLAN to permit creation of address ranges on the firewall. This scheme will also prevent noisy systems from degrading performance on other VLANS, and provide a slight security improvement by making sniffing across VLANS harder at the switch.

We use five interfaces on the intranet firewall, FW-2, to create five “subnets” within the Secured Server Zone; the Internal Open Services Network; the Internal Restricted Services Network; the Database network; the Development network and the Syslog Network. By reserving a range of IP addresses for each “subnet” we create address ranges in the firewall rules and it is easy to determine which subnet a server is assigned to by its IP address.

Our Management and Security Zone contains two true subnets. The first connects to the central management interface on each SunScreen firewall. This is an independent subnet without any external connection or route. There is a dedicated host, which is not a firewall to manage the SunScreen firewalls.

The NIDS subnet connects to the second interface (eth1) on each Snort probe and the dual homed central NIDS management workstation. There are no routes from the NIDS subnet to any other network. The NIDS central management console is connected to the intranet firewall, FW-2, to permit external IP address resolution and ntp synchronization with the two internal ntp servers.

An unusual aspect of this design is that the McAfee E-500 transparent proxy actually serves as the routing device between the Public Zone and internal GIAC subnets. The SunScreen firewall in stealth mode does not use IP addresses on the network interfaces. We need a unique subnet between the E-500 and the Cisco level 3 switch, 172.20.1.0/24. The latest version of the E-500 would require a router inside of the E-500 device.

The Build Network is independent of any other network or subnet. Most of the servers built on the Build Network will be assigned to the secured server zone with an IP address range of 172.17.1.1 to 172.17.1.254. Thus to minimize changes following a server's acceptance on the Build Network we will use the same range of IP addresses within the Build Network.

Server and Network IP Assignments:

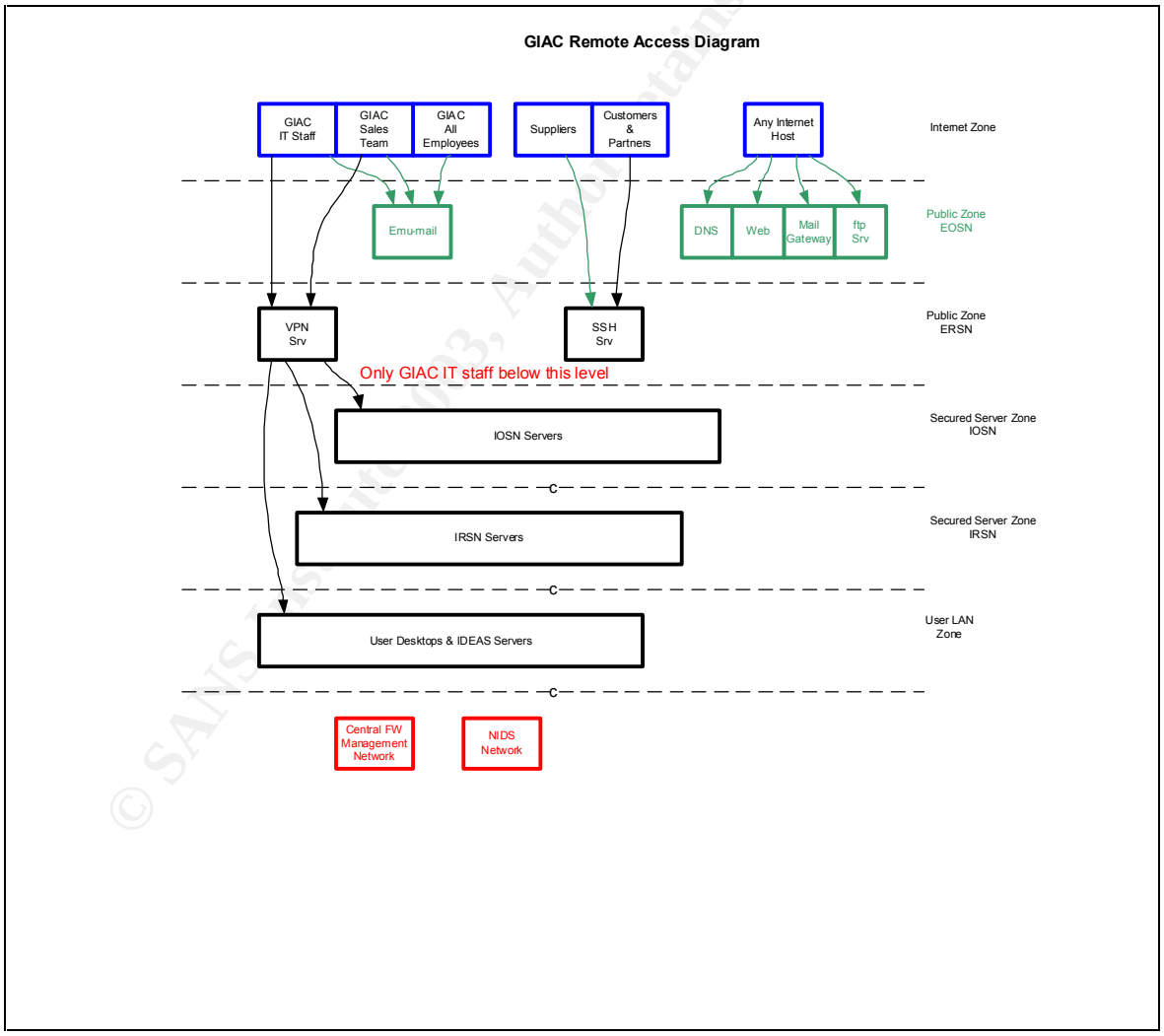
Host/Device	Network Zone	Interface	IP Address	Subnet Mask
Boundary Router	Public	eth1 (inside)	223.223.223.33	255.255.255.0
FW-1	Internet	qfe0	N/A	N/A
	ERSN (ssh)	qfe1	N/A	N/A
	(GIAC internal)	qfe2	N/A	N/A
	EOSN	qfe3	223.223.223.65	255.255.255.0
	FW Central Mang.	eri0	172.18.1.2	255.255.255.0
Static NAT	Public	N/A	223.223.223.50	255.255.255.0
ssh/scp Server	ERSN	eth0	223.223.223.40	255.255.255.0
VPN Server	ERSN	eth0	223.223.223.41	255.255.255.0
DNS '	EOSN	eth0	223.2	255.255.255.0
DNS "	EOSN	eth0	223.223.223.66	255.255.255.0
Squid Server	EOSN	eth0	223.223.223.68	255.255.255.0
Mail Gateway	EOSN	eth0	223.223.223.70	255.255.255.0
Emu Mail Server	EOSN	eth0	223.223.223.71	255.255.255.0
Web Server	EOSN	eth0	223.223.223.69	255.255.255.0
E-500	Public	eth0 (outside)	223.223.223.34	255.255.255.0
		eth1 (inside)	172.20.1.1	255.255.255.0

Switch	Internal LAN	VLAN-0	172.20.1.2	255.255.255.0
	(IT)	VLAN-1	172.16.1.1	255.255.255.0
	(Management)	VLAN-2	172.16.2.1	255.255.255.0
	(Marketing)	VLAN-3	172.16.3.1	255.255.255.0
	(Quality Control)	VLAN-4	172.16.4.1	255.255.255.0
	(Receivables)	VLAN-5	172.16.5.1	255.255.255.0
	(Payables)	VLAN-6	172.16.6.1	255.255.255.0
	(Sales)	VLAN-7	172.16.7.1	255.255.255.0
	(Linux servers)	VLAN-8	172.16.8.1	255.255.255.0
	(FW-2)	VLAN-9	172.17.1.1	255.255.255.0
FW-2	Secured Server	qfe0	N/A	N/A
	IOSN	qfe1	N/A	N/A
	IRSN	qfe2	N/A	N/A
	Database	qfe3	N/A	N/A
	NIDS	qfe4	N/A	N/A
	Syslog	qfe5	N/A	N/A
	Development Net	qfe6	N/A	N/A
	FW Central Mang.	eri0	172.18.1.3	255.255.255.0
FW Admin Console	FWMN	eri0	172.18.1.1	255.255.255.0
NIDS Console	NIDSMN (FW-2)	eth0	172.17.1.221	255.255.255.0
	(NIDS probes)	eth1	172.19.1.1	255.255.255.0
	Security Network	eth2	172.17.1.233	255.255.255.0
NIDS probes	NIDSMN	eth0	172.19.1.10- 17	255.255.255.0
		eth1	N/A	N/A
Syslog Srv.	Security Network	eth0	172.17.1.231	255.255.255.0
Alarm Srv.	Security Network	eth0	172.17.1.232	255.255.255.0
Oracle DB fortunes	Database Network		172.17.1.200	255.255.255.0
Web Srv. development	Development Net.	eth0	172.17.1.241	255.255.255.0
Oracle DB development	Development Net.	eri0	172.17.1.242	255.255.255.0
Oracle 9iAS development	Development Net.	eth0	172.17.1.243	255.255.255.0
Oracle 9iAS production	IOSN	eth0	172.17.1.11	255.255.255.0
Mail Server	IOSN	eth0	172.17.1.13	255.255.255.0

Squid Proxy	IOSN	eth0	172.17.1.14	255.255.255.0
Web Server	IOSN	eth0	172.17.1.15	255.255.255.0
Oracle DB business	IRSN	eri0	172.17.1.100	255.255.255.0
OS Server	Internal LAN	eth0	172.16.8.10	255.255.255.0
App. Server	Internal LAN	eth0	172.16.8.11	255.255.255.0
Data. Server	Internal LAN	eth0	172.16.8.12	255.255.255.0
DNS ‘	IRSN	eth0	172.17.1.103	255.255.255.0
DNS “	IRSN	eth0	172.17.1.104	255.255.255.0

Network Access Requirements:

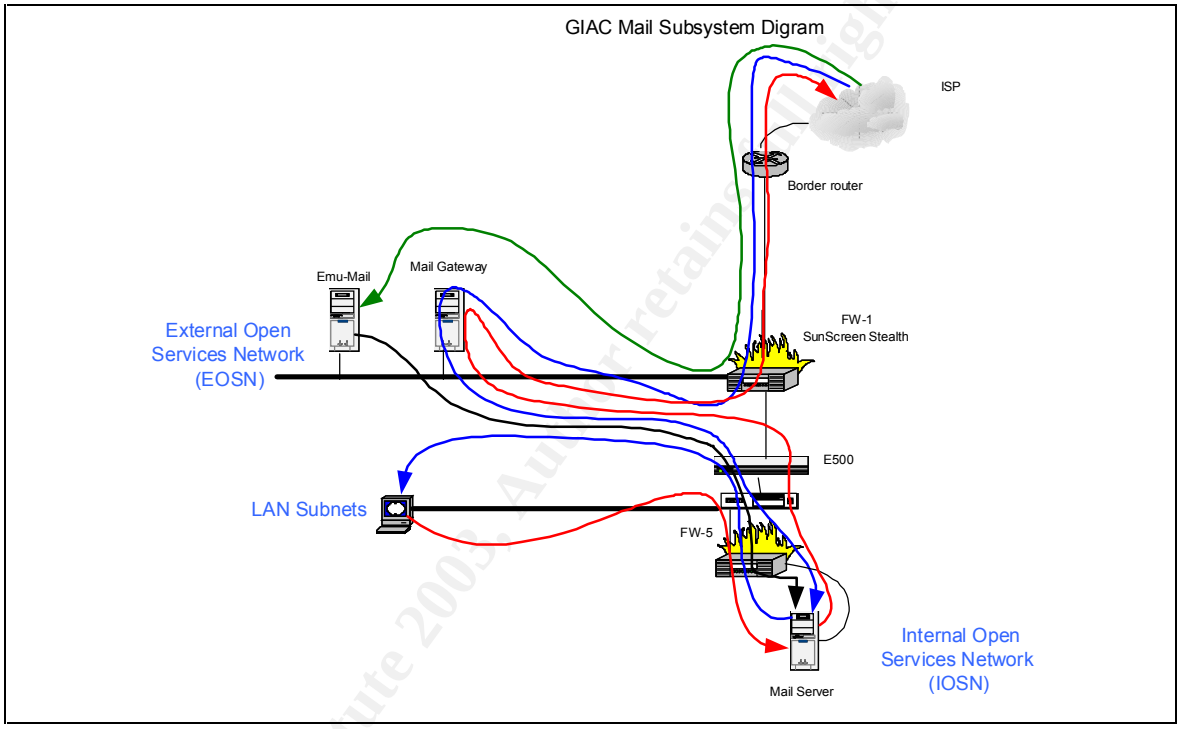
We have adapted a diagram found in Nick Reed’s GCFW Practical (January 2002) to clearly show the levels into GIAC’s network various external users are authorized to access.



Service level diagrams will provide a graphical view of information flow through a network. The accompanying tables show on a host basis

information flow and associated protocols, ports and interfaces. This entire section is presented with statefull firewalls in mind. In that, the direction of the flow arrows is from the originating host to the destination only. Firewall state tables will automatically handle the return path. We will have to build out the rule set more extensively for the proxy firewall in front of the External Open Services Network.

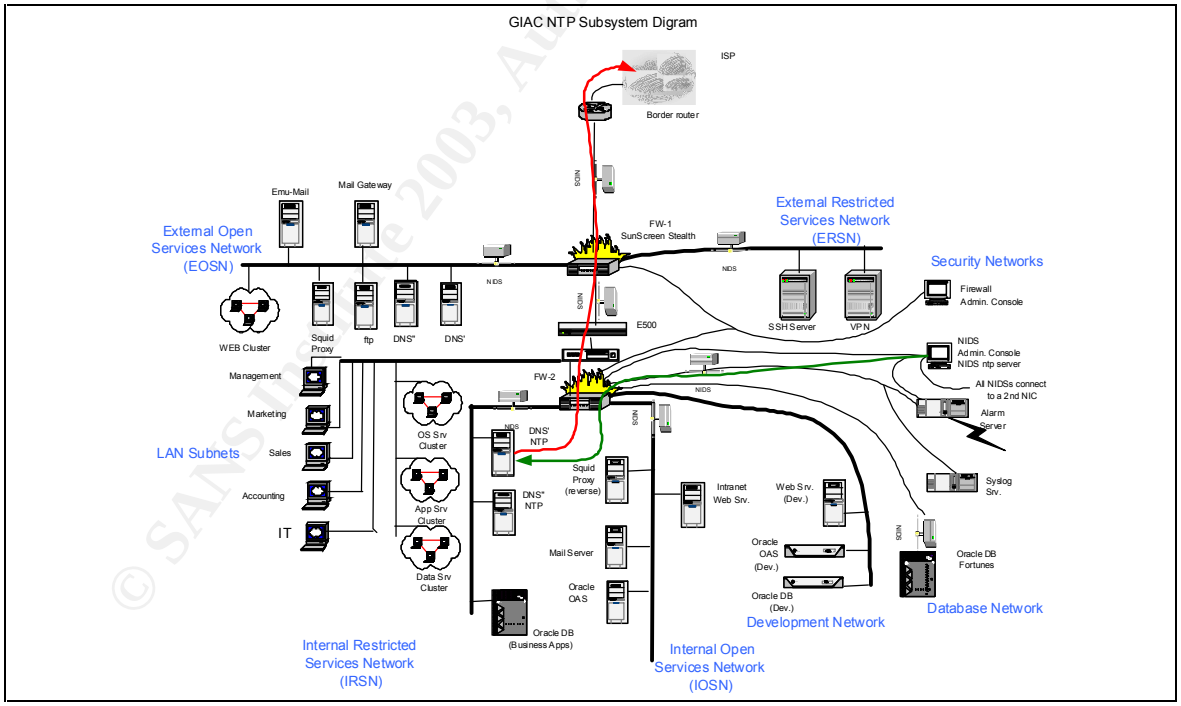
For clarity, purposes network components not associate directly with the target protocol have been deleted from each service level diagram.



The mail subsystem is one of the most complex in terms of information flow. There are two largely independent mail systems shown here. The regular mail system permits inbound Internet mail to the mail gateway > E-500 transparent proxy > mail server. All e-mail bound for the Internet must pass through the external mail gateway. This was setup to prevent unauthorized attachment types from being sent out. Basically, we are using this server to enforce the same set of rules for in-bound and out-bound e-mail opposed to building a more complex rule base on the internal mail server.

The Emu-mail sub-system permits authorized external users https access to the Emu-mail server, which then accesses the internal mail server and formulates a requested reply.

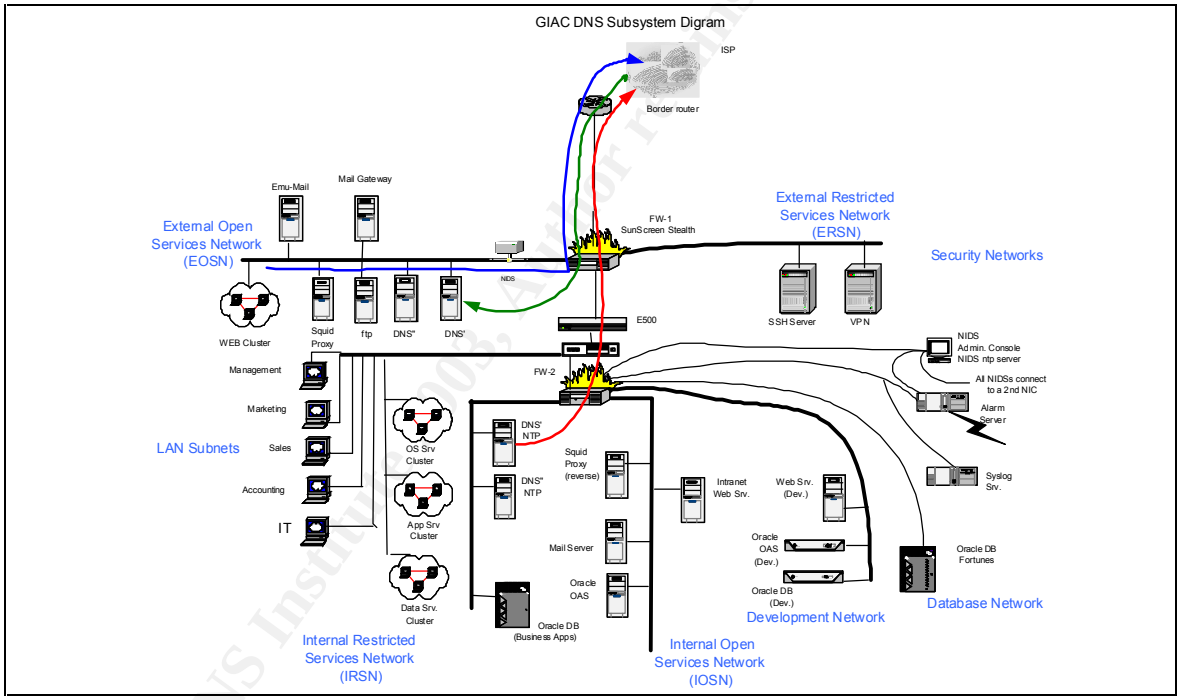
SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Outside	smtp	Outside	*	223.223.223.70	25
Mail Gateway	smtp	223.223.223.70	*	223.223.223.34	25
EOSN	smtp	223.223.223.64 - 94	*	223.223.223.34	25
ERSN	smtp	223.223.223.40 - 41	*	223.223.223.34	25
E-500	smtp	172.20.1.1	*	172.17.1.13	25
Desk Tops	smtp	172.16.1 - 8.0	*	172.17.1.13	25
Secured Server Networks	smtp	172.17.1.0	*	172.17.1.13	25
Mail Server	smtp	172.17.1.13	*	172.20.1.1	25
E-500	smtp	223.223.223.34	*	223.223.223.70	25
Mail Gateway	smtp	223.223.223.70	*	*	25
Outside	ssl	Outside	*	223.223.223.71	443
Emu Mail	pop3	223.223.223.71	*	172.17.1.13	110
E-500	pop3	172.20.1.1	*	172.17.1.13	110
Mail Server	pop3	172.17.1.13	*	223.223.223.71	110
Emu Mail	ssl	223.223.223.71	*	*	443



The ntp subsystem uses three ntp servers. The two primary ntp servers are co-hosted on the two internal DNS servers. Only these two ntp servers are permitted to request updates from four specific external stratum two clocks. The third ntp server is the NIDS administration console and it provides time synchronization for the NIDS probes. We do

not permit any routing across the dual-homed NIDS administration console. All other GIAC network devices are synchronized with the two primary ntp servers.

SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
DNS ‘ (internal)	ntp	172.17.1.103	*	Outside	123
DNS “ (internal)	ntp	172.17.1.104	*	Outside	123
EOSN	ntp	223.223.223.64 - 94	*	172.17.1.103 – 104	123
ERSN	ntp	223.223.223.33 - 62	*	172.17.1.103 - 104	123
IOSN	ntp	172.17.1.10 - 99	*	172.17.1.103 - 104	123
Development Net	ntp	172.17.1.240 - 249	*	172.17.1.103 - 104	123
Database Net	ntp	172.17.1.200 – 220	*	172.17.1.103 - 104	123
Syslog Net	ntp	172.17.1.230 - 239	*	172.17.1.103 - 104	123
NIDS Console	ntp	172.17.1.221	*	172.17.1.103 - 104	123

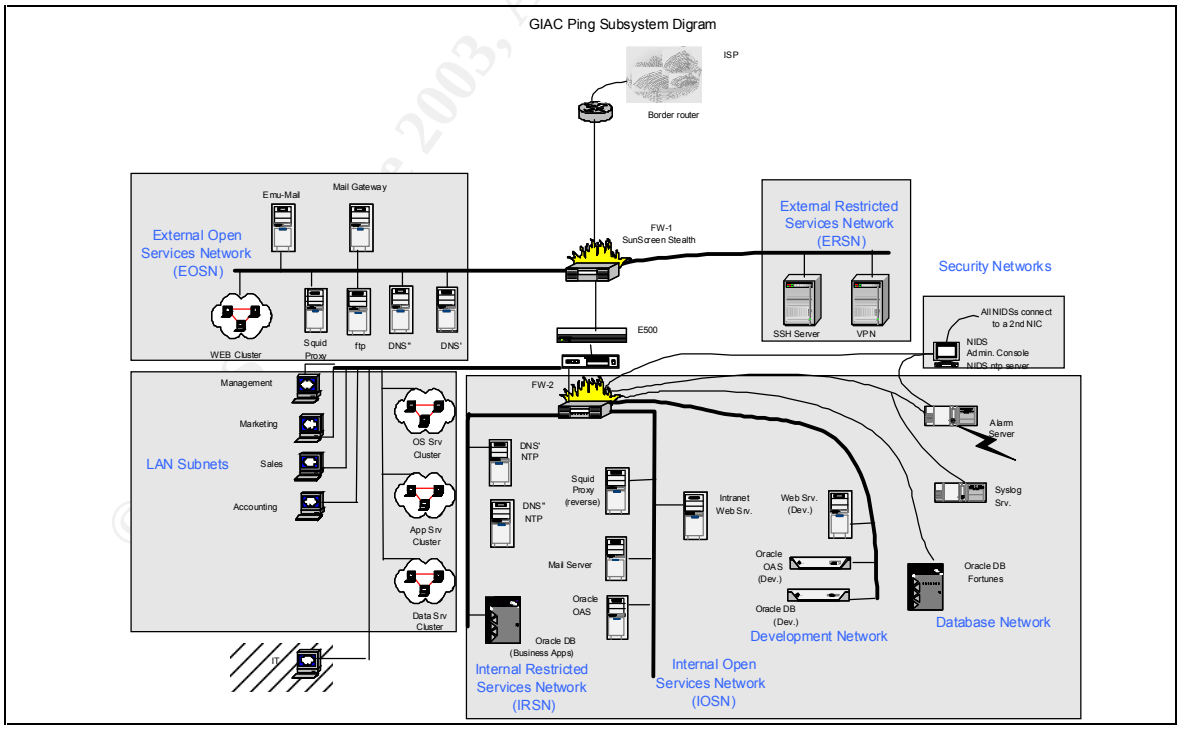


The GIAC DNS subsystem is build on a split DNS server scheme. The two external DNS servers will answer external queries only for devices located on the GIAC external subnets. Name resolution for most of the external GIAC hosts is handled through /etc/hosts table look-ups. A few hosts on the external networks do require name and reverse lookups in normal operations. This is accomplished by pointing these systems to the DNS of our ISPs. Consequently, the internal DNS servers never provide any information to any external device.

Most internal hosts use the two internal DNS servers for name resolution. The exceptions include the NIDS subsystem, which uses only /etc/hosts

files; the Build Network, which uses a local DNS server; and the Firewall administration subnet, which also uses local files only. Both internal DNS servers may forward queries to DNS servers external to GIAC Enterprises.

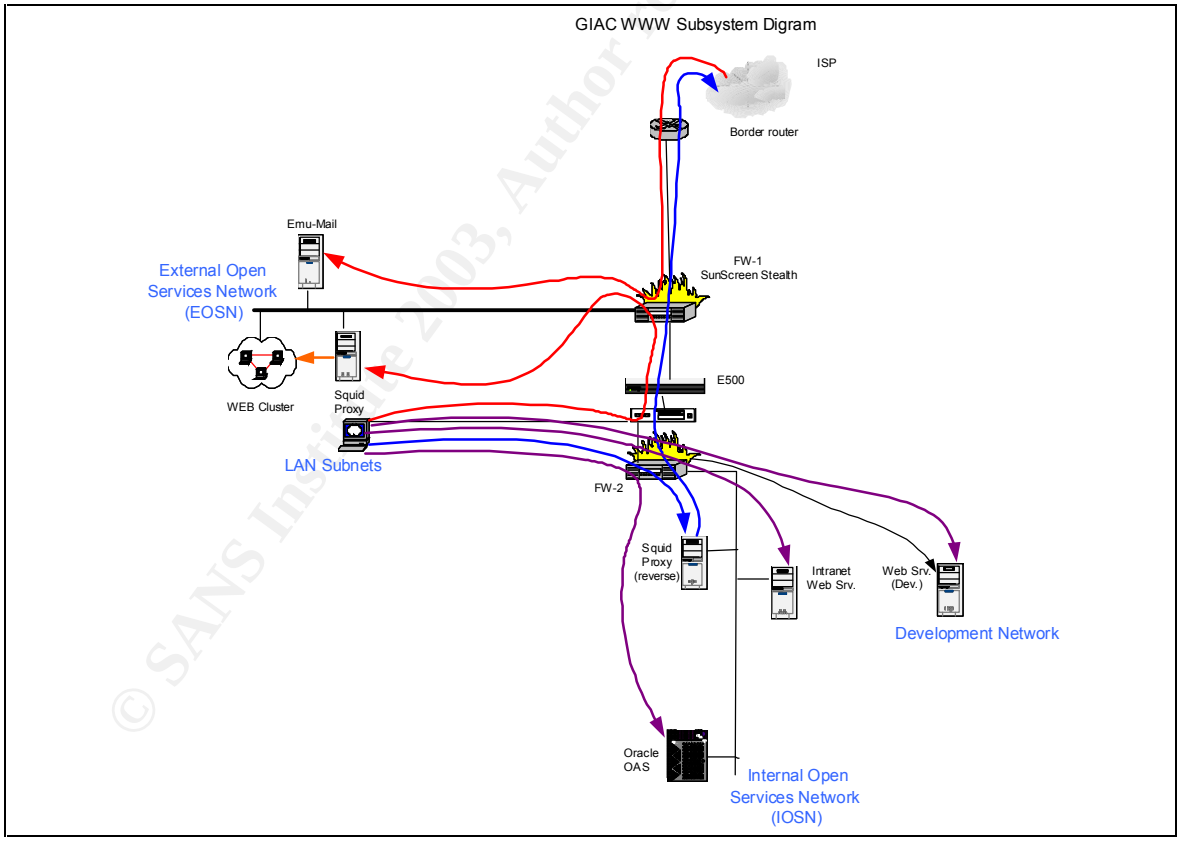
SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Outside	dns	Outside	*	223.223.223.65	udp 53
Outside	dns	Outside	*	223.223.223.66	udp 53
Mail Gateway	dns	223.223.223.70	*	Outside	53
Emu Mail	dns	223.223.223.71	*	Outside	53
ssh/scp Server	dns	223.223.223.40	*	Outside	53
User LAN	dns	172.16.1 - 8.0	*	172.17.1.103 - 104	udp 53
IOSN	dns	172.17.1.10 - 99	*	172.17.1.103 - 104	udp 53
Development Net	dns	172.17.1.240 - 249	*	172.17.1.103 - 104	udp 53
Database Net	dns	172.17.1.200 – 220	*	172.17.1.103 - 104	udp 53
NIDS Console	dns	172.17.1.221	*	172.17.1.103 - 104	udp 53



We have chosen to limit the use of the ping command to eliminate some DoS attack options and network probing. Hosts within each of the shaded

areas in the above drawing are permitted to ping anything within the same zone for local connectivity testing. Desktop systems can ping only other hosts within the LAN Subnets. The IT subnet is permitted to ping any host anywhere. Limited exceptions to these rules exist within the central firewall chain, the two firewalls, having stealth interfaces can not initiate or respond to a ping.

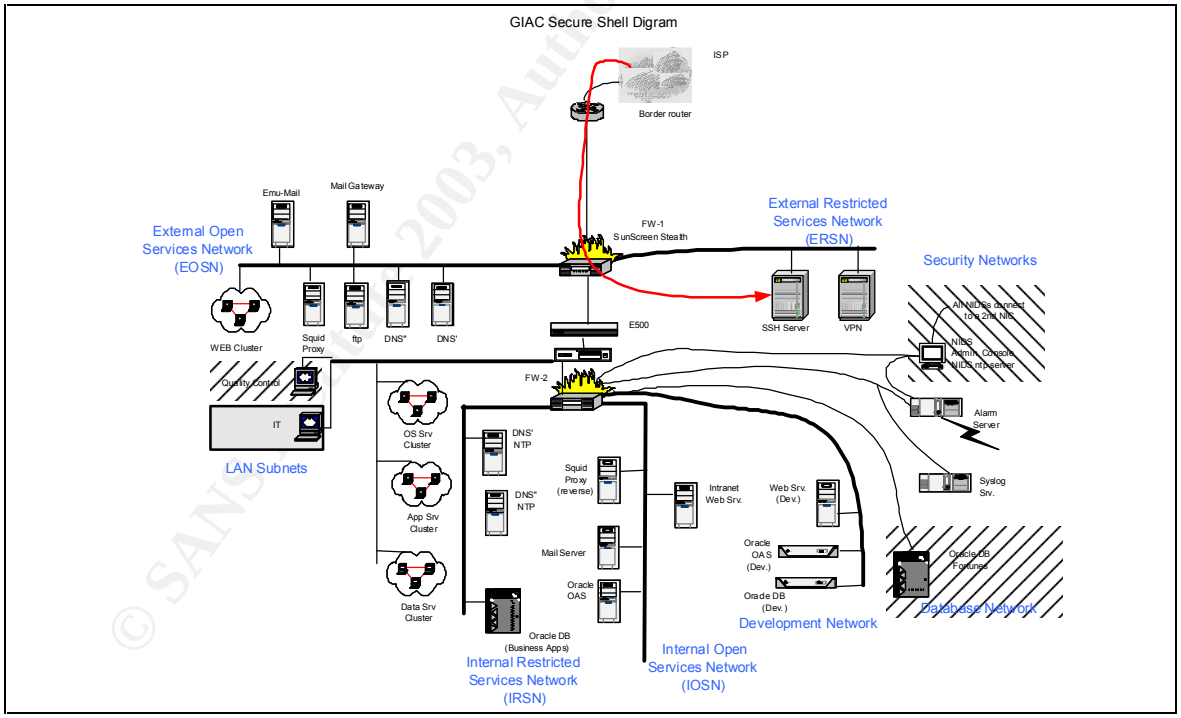
SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
IT	ping	172.16.1.1 - 254	*	*	*
IOSN	ping	172.17.1.10 - 99	*	172.17.1.1 - 254	*
IRSN	ping	172.17.1.100-199	*	172.17.1.1 - 254	*
Development Net	ping	172.17.1.240 - 249	*	172.17.1.1 - 254	*
Database Net	ping	172.17.1.200 – 220	*	172.17.1.1 - 254	*
NIDS Console	ping	172.17.1.221	*	172.17.1.1 - 254	*



There are three major web traffic clusters in the GIAC network. External users access the Squid proxy and external web server in the External Open Services Network. From there other protocols are used to transport users' request and return responses to and from the internal database

systems. Internal users may also access the external Squid/web server. GIAC Enterprises maintains an internal web server for company information requirements; all GIAC employees may access this system. Also all internal GIAC employees may access the Oracle 9iAS servers on the Internal Open Services and Developments Networks.

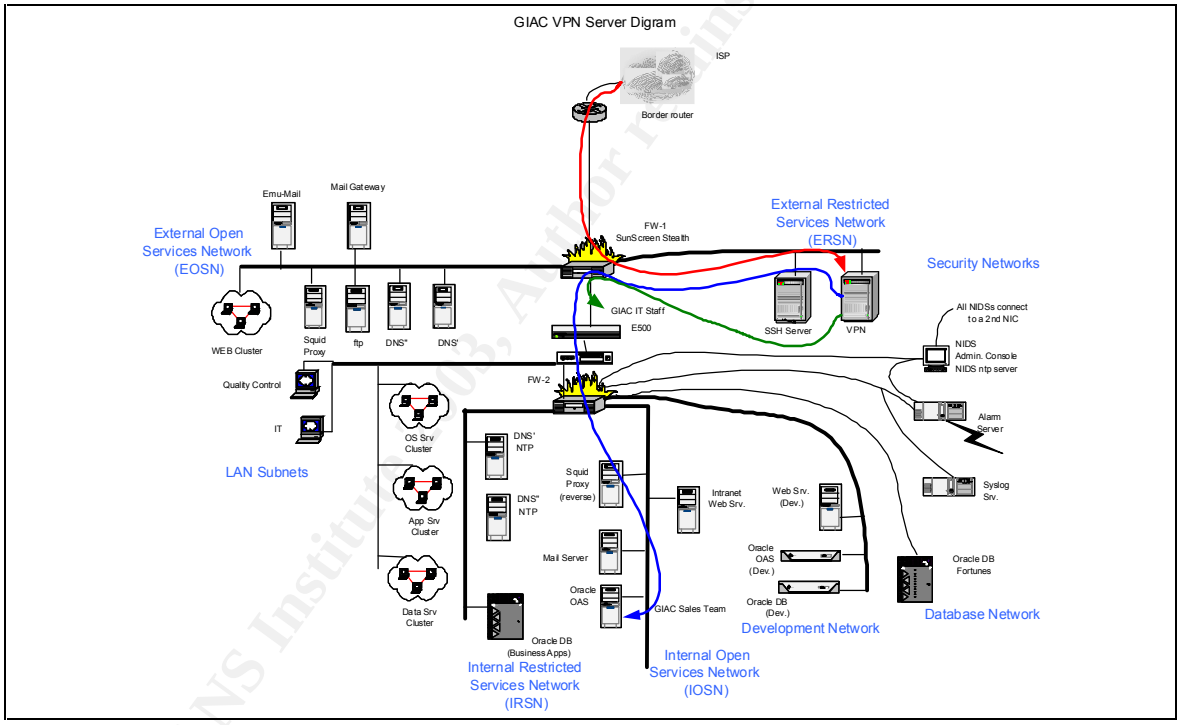
SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Squid (external)	https	Outside	*	223.223.223.68	443
Squid (external)	http	Outside	*	223.223.223.68	80
User LAN	https	172.16.1 - 8.0	*	223.223.223.68	443
User LAN	http	172.16.1 - 8.0	*	223.223.223.68	80
Emu-mail	https	Outside	*	223.223.223.71	443
User LAN	http	172.16.1 - 8.0	*	172.17.1.14	80
User LAN	https	172.16.1 - 8.0	*	172.17.1.14	443
Squid (internal)	http	172.17.1.14	*	Outside	80
Squid (internal)	https	172.17.1.14	*	Outside	443
User LAN	http	172.16.1 - 8.0	*	172.17.1.11	80
User LAN	http	172.16.1 - 8.0	*	172.17.1.241	80
NIDS Console	http	172.17.1.221	*	Outside	80



Secure shell and secure copy are the only authorized methods to logon to and move files between GIAC servers. In the above drawing, the left angled box shows ssh is permitted only within the NIDS subnet. No one may access any part of this subnet from another host. The Quality Control unit is the only non-IT GIAC group authorized to directly access the Oracle fortunes database server as shown by the right slating hashed boxes.

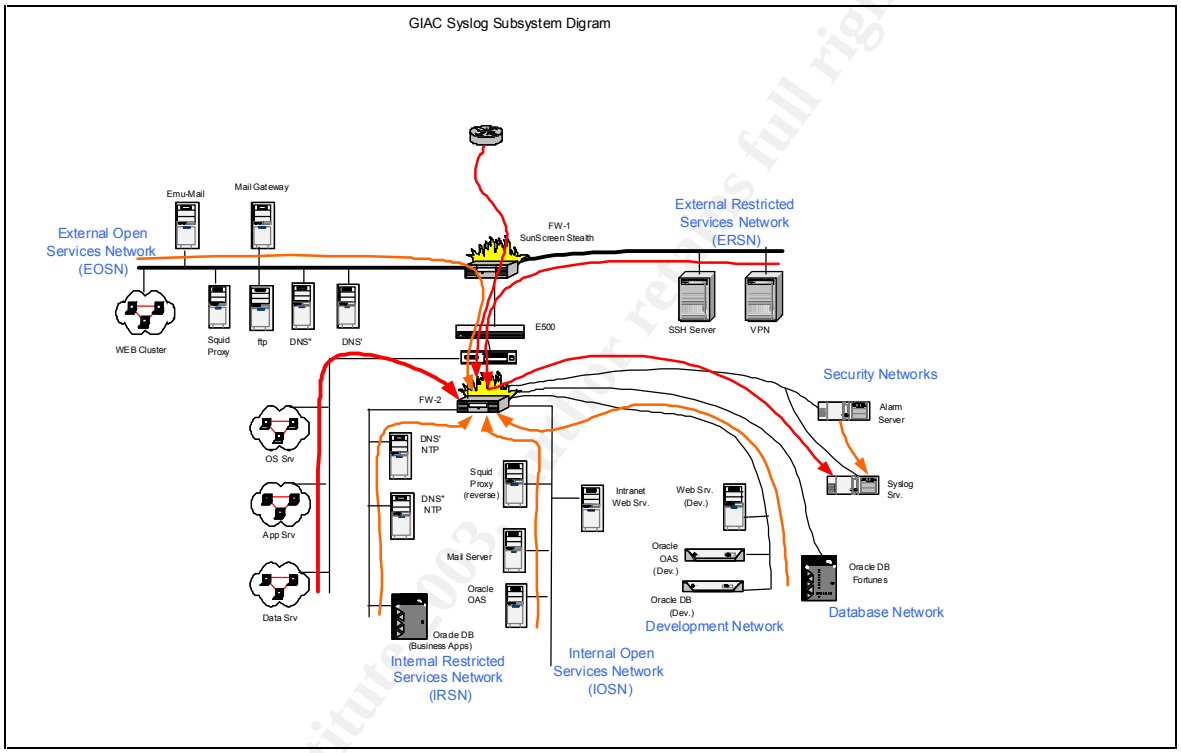
The IT group has unlimited ssh/scp access from any where in the GIAC network. Only exceptions are the firewalls and the border routers.

SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Outside	ssh	Outside	*	223.223.223.40	65322
IT	ssh	172.16.1.1 - 254	*	172.16.0.0 - 172.20.0.0	65322
IT	ssh	172.16.1.1 - 254	*	223.223.223.0	65322
Quality Control	ssh	172.16.4.1 - 254	*	172.17.1.200	65322
Quality Control	ssh	172.16.4.1 - 254	*	172.17.1.242	65322
Oracle DB (fortunes)	ssh	172.17.1.200	*	223.223.223.40	65322



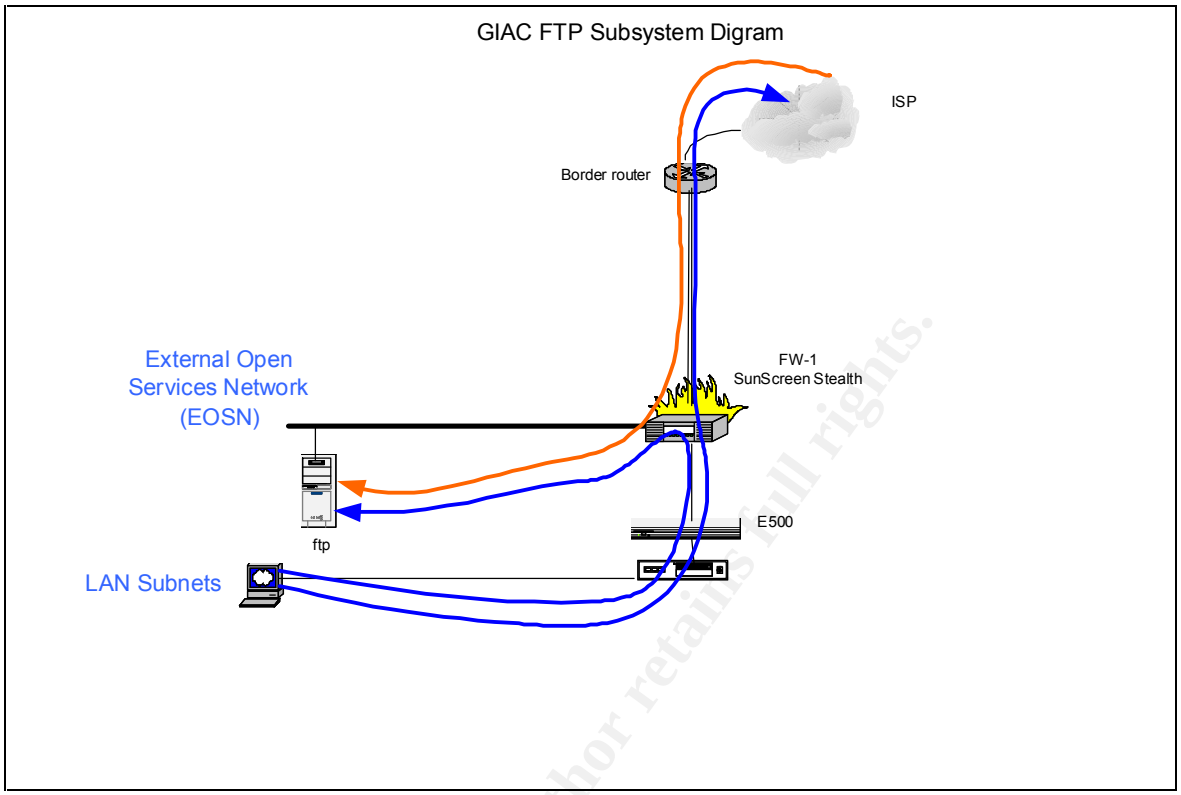
Only GIAC's external sales team and IT staff are authorized to access internal GIAC network resources via the VPN server. From the VPN server IT staff members may access any networked server, while the sales team may only access the internal Oracle application server in the IOSN. Each authorized VPN user runs in a chroot environment to limit access to the local VPN server. The sales team is further restricted with a startup shell that only permits access to a web browser.

SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Outside	isakmp	Outside	*	223.223.223.41	udp 500
Outside	ESP (IP ID 51)	Outside	*	223.223.223.41	
VPN Server	https	223.223.223.41	*	172.17.1.11	tcp/443
VPN Server	*	223.223.223.41	*	GIAC Net	*



Essentially all GIAC network servers use the syslog server to remotely log events. The only exceptions are the SunScreen firewalls, which log to the firewall central administration console.

SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
BR	syslog	223.223.223.33	*	172.17.1.231	514
IRSN	syslog	172.17.1.100 - 199	*	172.17.1.231	514
IOSN	syslog	172.17.1.10 - 99	*	172.17.1.231	514
Development Net.	syslog	172.17.1.240 - 250	*	172.17.1.231	514
Oracle DB (fortunes)	syslog	172.17.1.200	*	172.17.1.231	514
NIDS Console	syslog	172.17.1.221	*	172.17.1.231	514
EOSN	syslog	223.223.223.64 - 94	*	172.17.1.231	514
ERSN	syslog	223.223.223.33-62	*	172.17.1.231	514



At GIAC Enterprises ftp is used primarily as a means to transfer mail attachments that exceed the size limits imposed by the internal mail system. Consequently, both external and local users must access to the ftp server. Only IT employees are authorized to access external ftp sites.

SERVICE / HOST	PROTO	SIP	SPort	DIP	DPort
Outside	ftp	Outside	*	223.223.223.67	20 - 21
Desktops	ftp	172.16.1 - 8.0	*	223.223.223.67	20 - 21
IT	ftp	172.16.1.0	*	Outside	20 - 21

Subsystem Summary:

A review of the above service level diagrams clearly shows the internal firewall (FW-2) will be subject to heavy traffic and will require careful sizing.

SOCIAL ISSUES:

Invariably the weakest link in security for many organizations is the human and social factors. Physical, hardware and software security can be designed, engineered, built and tested, but the other factors require a human trust level. The compartmentization of business functions at GIAC Enterprises lends towards an environment where sensitive information or assets are accessible only to the minimum number of individuals, thus, reducing the measure of risk. Part of the

high-level security policy includes sections requiring background checks of all employees as well as initial and periodic security refresher training for everyone. This training in-part emphasis the need and value in correctly following establish operating procedures at all levels in the organization. Policy requires regular review of the procedures to ensure they are current and reflect any changes made to the business model, information flow or hardware and software changes. Periodic penetration testing, including social engineering, will be utilized to help identify remaining or new problems and failures in following established procedures.

The recent problems created by the SQL Slammer Worm in January 2003 exemplify the need to keep systems patched and be aware of all software and configuration files/accounts installed on your systems.

Through out this network design process we have mentioned various "best practices" to prevent or reduce common human errors. GIAC operating procedures will address issues raised in the SANS Institute's white paper titled "Mistakes People Make that Lead to Security Breaches" update October 23, 2001.

SUMMARY:

The nearly exclusive use of a single operating system on a single hardware platform might be seen as a weakness. Perhaps a cracker could be kept off guard somewhat by using a greater variety of hosts. We strongly believe, there is far more to be gained by selecting a narrow solution base and becoming very good at securing that base environment. Our experience shows most problems result from poor configuration and/or poor updating procedures than in product weakness themselves, particularly if new vulnerabilities are patched or corrected in a timely fashion.

The network design presented here for GIAC Enterprises is a fairly large and complex design for a small company. However, the sole owner, Mr. Flowers is financially independent and at this point considers GIAC's operation to be more of a passion and hobby than a profitable business.

ASSIGNMENT II: SECURITY POLICY:

Overview of Security Policies:

In Assignment II I will explain the rational and the individual steps and rules used to create the security policies for the key perimeter defensive systems. The network design presented in Assignment I was a layered defense, where defensive measures in each layer are intended to protect the assets in the underlying layer. Thus, each piece of equipment will be used for what it does best.

Border Router:

Policy:

Our border router, a Cisco 7606 with IOS 12.2 Release Software, is the first layer of defense for GIAC Enterprises. While we could use Cisco's firewall capabilities, we will use the router for primarily routing functions with named ACLs. Extended and named ACLs provide additional configuration options not available with standard ACLs. Reflexive access lists will not be used at this time because of the added processing and memory overhead. If the local security policy allows or requires responses to certain potentially hostile traffic, reflexive ACLs permit automatic generation of response packets. Release status of Cisco IOS can be checked at http://www.cisco.com/Products/Cisco-ios-software/key-release-dates

We do not anticipate frequent access requires to the border router and thus will require all router access to done locally via a console connection. No remote access will be permitted, which includes Auxiliary and all five vtys (0-4). There is no need to use external routing protocols since static routing will meet our requirements. According to our network design we do not have a need for services on the router, such as snmp, http, telnet, ssh, etc., these will be disabled at the router. Time synchronization will be accomplished with the two internal GIAC ntp servers. All router logs will be sent to the internal GIAC syslog server. TACACS and AAA are useful features without a problem to solve in this network design using a single router.

Ingress filtering for Private addresses (RFC 1918), non-assigned and reserved IP addresses (http://www.iana.org/) will be configured. Egress filtering will permit only GIAC's assigned public IP addresses to exit the GIAC network. We will add ACLs to reduce the potential for successful DoS style attacks and block traffic to other unnecessary services and ports. All router configuration and management will be done via a directly connected console.

Router Configuration:

In this section, we will show commands used to configure the border router.

In order to use the required command set, we must enable privileged mode, which is denoted by the # prompt.

```
router>enable
password  XXXXXXXXXX
router#
```

By default Cisco routers have three privilege levels enabled. This should be sufficient for GIAC's operations with out the complexity of implementing all 16 possible levels.

Next, we want to enter configure global configuration mode by typing config term, this mode is denoted by including "config" in the prompt. Then name our router "GIACBR1", with the idea we might add another border router to a second ISP in the future. If we do so, we will need to use border gateway protocol (BGP) to handle the complex routing issues.

```
router# config term
router(config)# hostname GIACBR1
GIACBR1(config)# interface ethernet 0
GIACBR1(config if)# ip address i.s.p.12 255.255.255.0
GIACBR1(config if)# no shutdown
GIACBR1(config if)# no ip directed-broadcast
GIACBR1(config if)# no ip proxy-arp
GIACBR1(config if)# no ip mask-reply
GIACBR1(config if)# no ip redirects
GIACBR1(config)#  ntp disable
GIACBR1(config if)# exit
```

We have set the outside interface, eth0, to i.s.p.12 with the assigned Class C subnet mask and block some potentially harmful traffic from reaching out exterior interface. There is usually no need to reply to requests for our network mask, so we explicitly disable it. Proxy arp can be used for network mapping by requesting the MAC address for host, we will disable this service. Directed broadcasts have been stopped to prevent ping floods to our network addresses. Many hosts would respond to such a request and create a DoS problem. Note that we have permitted ICMP unreachable. A cracker may be able to learn about our internal network because of this rule. Finally, we do not want the router to act as an ntp server.

A recent Cisco feature is unicast reverse packet forwarding (uRPF). This provides a sanity check to ensure traffic is sent only from the proper interface. uRPF can make ingress and egress rules in a complex network topology much simpler, because it will automatically adjust to network changes. However, uRPF does not log at the same level as ACLs do. Without a demonstrated need, we will not enable this feature.

Next, we will set the internal interface, eth1, to 223.223.223.33 and block potentially harmful traffic from leaving our network. There is no need to permit ICMP type 13, timestamp request or ICMP type 17, address mask request from going out, as this information can aid in network mapping.

```
GIACBR1(config)# interface ethernet 1
GIACBR1(config if)# ip address 223.223.223.33 255.255.255.0
GIACBR1(config if)# no shutdown
GIACBR1(config if)# no ip directed-broadcast
GIACBR1(config if)# no ip proxy-arp
GIACBR1(config if)# no ip mask-reply
GIACBR1(config if)# no ip redirects
GIACBR1(config)#  ntp disable
GIACBR1(config if)# exit
```

Directed broadcasts have been stopped to prevent our hosts from being used for ping floods of external network addresses.

We start hardening the router by setting the privileged level password, encrypting it and setting several access and console features.

```
GIACBR1(config)# line console 0
GIACBR1(config-line)# password “BRpassword”
GIACBR1(config-line)# login
GIACBR1(config-line)# enable secret “BRSecpassword”
GIACBR1(config)# service password encryption
GIACBR1(config)# line console 0
GIACBR1(config-line)# exec-timeout 5 00
GIACBR1(config-line)# exit
```

The “enable secret” command encrypted the password in a MD5 hash, which is now unreadable, but as with any MD5 hash if the password is stolen, it can still be learned by using a password cracking tool. This is far better than the older “enable password” command, which is not as secure. Console timeout was set to five minutes, to help cover busy administrators.

Next, we will set the login banner that is show for every login attempt.

```
GIACBR1(config)# banner /
***WARNING***
Only authorized access is permitted.  Any non-authorized use is
prohibited and violators may be prosecuted.  All activity o this
device and network is monitored and logged.
/
```

The exec banner is shown after a login and an shell session has been started. We do want both banners.

```
GIACBR1(config)# banner exec /
###Reminder###
Only authorized access is permitted.  Any non-authorized use is
prohibited and violators may be prosecuted.  All activity o this
device and network is monitored and logged.
/
```

Policy states that the border router may not be accessed via any network connection. So we will shut down both aux and vty (0-4) access. These measures should prevent any connections from telnet, rlogin, ssh and http. Typically, when the aux port is used, a modem is attached for backup routing or remote management. The aux port is of little significance in this design since we will not attach any device to it.

```
GIACBR1(config)# line aux 0
GIACBR1(config-line)# no exec
GIACBR1(config-line)# login
GIACBR1(config-line)# no password
GIACBR1(config-line)# exec-timeout 0 1
GIACBR1(config-line)# transport input none

GIACBR1(config-line)# line vty 0 4
GIACBR1(config-line)# no exec
GIACBR1(config-line)#  login
GIACBR1(config-line)# no password
GIACBR1(config-line)#exec-timeout 0 1
GIACBR1(config-line)# transport input none
GIACBR1(config-line)# exit
```

All of the measures taken above ensure that the aux and vty lines are unusable. Just using “login \ no password” will still permit a telnet connect with a message like “Password required, but none set”. The proposed solution will give a “connection refused” message.

Since we are not using many of the software features found on the Cisco 6406, we can shutdown many unnecessary services. Source routing is seldom required or desirable, so we shut it down. Cisco enables DNS by default, so we shut it down. We do not need any bootp type process across router.

```
GIACBR1(config) no ip source-route
GIACBR1(config) no ip name-server
GIACBR1(config) no service.config
GIACBR1(config) no boot network
GIACBR1(config) no service pad
GIACBR1(config) no ip classless

GIACBR1(config)# no service tcp-small-servers
GIACBR1(config)# no service udp-small-servers
GIACBR1(config)# no snmp server
GIACBR1(config)# no service finger
GIACBR1(config)# no ip http server
GIACBR1(config)# no ip bootp server
GIACBR1(config)# no cdp run
```

This has stopped access to the unnecessary services on the low or small number tcp and udp ports. Eliminated snmp service, which by GIAC policy will not be used, and ensure the router does not run web, bootp or finger services. Cisco's web management feature was disabled by the "no ip http server" command. Since we do not have any other Cisco routers in the network we can disable Cisco Discovery Protocol (cdp). All of the remote configuration options, except ssh, sends the passwords and configuration information across the network in clear text. Since we have only a single router that should be fairly static in configuration, our console management plan should work very well and be secure.

We will disable reverse lookups for efficiency purposes.

```
GIACBR1(config)# no ip domain-lookup
```

The router's clocks will be synchronized with the two internal ntp servers with a typical ntp client/server model. A broadcast/multicast model will not work because of different subnets. Even if the internal ntp servers fails to synchronize with the external stratum two clocks, all of GIAC's hosts will be still be synchronized together. We need to set the source to be the inside interface and state we prefer ntp server 172.17.1.103. Presently Cisco does not support ntp version four.

```
GIACBR1(config)#  ntp server 172.17.1.103 version 3 source
172.17.1.33 prefer
```

```
GIACBR1(config)#  ntp server 172.17.1.104 version 3 source
172.17.1.33
```

To verify which ntp servers the router is really synchronizing with after configuration is complete, we can issue the command:

```
GIACBR1# show ntp associations
```

All GIAC servers are to log to the central syslog server located in the internal server network. Turn on logging; set it to log in local time with date and time in milliseconds.

```
GIACBR1(config)#  logging on
GIACBR1(config)#  service timestamps log datetime msec localtime
```

Set a rate limit for logging of 10 records per second to reduce the effects of a DoS type attack by overflowing our buffers and logs. We will disable console logging since we do not normally operate with an attached console and to improve router performance. All trapped messages will be sent to our central log server at 172.17.1.231 using logging facility local 7. Adding the "service sequence-numbers" forces logging to add sequence numbers to help identify missing or otherwise incomplete log records.

```
GIACBR1(config)#  service sequence-numbers
GIACBR1(config)#  logging rate-limit all 10 except error
GIACBR1(config)#  logging buffer 32000
GIACBR1(config)#  logging buffer notification
GIACBR1(config)#  no logging console
GIACBR1(config)#  service sequence-numbers
GIACBR1(config)#  logging facility local 7
GIACBR1(config)#  logging trap alerts
GIACBR1(config)#  logging trap emergencies
GIACBR1(config)#  logging 172.17.1.231
```

On the central logging server we will add a line to syslog.conf file to direct router logs to a specific log file.

```
local7.info                    /var/log/router
```

Router ACLs:

In this design the primary purpose of ACLs is to protect the router itself, while the secondary purpose is to provide an added layer of policies also covered by the firewall. Most of the requirements for our border router could be meet with standard ACLs, which check only the source IP address. Named ACLS make reading and editing ACLs much easier than standard numbered lists in the range of 1 to 99.

Since we are going to use extended named lists, we will add some ACLs over our base requirement to provide an extra layer of defense at the border router.

Ingress Rules:

At this defensive layer, we are not interested in logging all normal or permitted traffic, only those packets that are not permitted. Generally, we do not include a log statement on "permit" rules. Note that since we do not have a closing "deny all" type rule, the permit ACLs are not really required, but have been included for clarity and are defined if we chose to be more restrictive in the future. It would also make it easier to start logging on selected permit rules if the need arose. Standard ACLs only permit the "log" option that records type, date and time. We will use the "log-input" option, available with extended ACLs, which adds interface and source MAC address to the log entries.

We should never see any private IP address (RFC 1918) on the outside interface, but the following commands will drop any such packets. This will stop some spoofed packet issues.

```
GIACBR1 (config)# ip access-list extended ingress_filter
GIACBR1(config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 10.10.0.0 0.0.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log-input
```

Block any traffic with our assigned address space.

```
GIACBR1(config-ext-nacl)# deny ip 223.223.223.0 0.0.0.255 any log-input
GIACBR1(config-ext-nacl)# permit ip any any          # not required
```

Next block traffic without an IP address.

```
GIACBR1(config-ext-nacl)# deny ip host 0.0.0.0 any log-input
```

Block all traffic from localhost.

```
GIACBR1(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log-input
```

Block broadcast and multicast traffic and windows self assigned.

```
GIACBR1(config-ext-nacl)# deny ip 255.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 224.0.0.0 7.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 240.0.0.0 7.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 169.254.0.0 0.0.255.255 any log-input
```

Allow DNS traffic only to the two external DNS servers.

```
GIACBR1(config-ext-nacl)# permit udp any host 223.223.223.65 eq 53
GIACBR1(config-ext-nacl)# permit udp any host 223.223.223.66 eq 53
GIACBR1(config-ext-nacl)# deny udp any any eq 53 log-input
```

Allow ssh traffic only to the secure shell server in the External Restricted Services Network.

```
GIACBR1(config-ext-nacl)# permit tcp any host 223.223.223.40 eq 22
GIACBR1(config-ext-nacl)# deny tcp any any eq 22 log-input
```

smtp is permitted only to the mail gateway, we will deny all other mail related traffic. The Emu-mail server uses https for traffic with Internet users, so pop3 is not an issue here.

```
GIACBR1(config-ext-nacl)# permit tcp any host 223.223.223.70 eq 25
GIACBR1(config-ext-nacl)# permit tcp any host 223.223.223.71 eq 443
GIACBR1(config-ext-nacl)# deny tcp any any eq 25 log-input
GIACBR1(config-ext-nacl)# deny tcp any any eq 143 log-input
GIACBR1(config-ext-nacl)# deny tcp any any range 109 110 log-input
```

Allow passive ftp traffic only to the ftp server on ephemeral ports greater than 1023. The SunScreen firewall will do essentially the same thing, except it will not block destination ports in the well-known port range <1023.

```
GIACBR1(config-ext-nacl)# permit tcp any gt 1023 host 223.223.223.67 \
                          gt 1023 est
```

```
GIACBR1(config-ext-nacl)# deny tcp any any eq range 20 21 log-input
```

GIAC policy prohibits news services on nntp.

```
GIACBR1(config-ext-nacl)# deny tcp any any eq 119 log-input
```

There is no need for syslog traffic across the border router.

```
GIACBR1(config-ext-nacl)# deny udp any any eq 514 log-input
```

The only ICMP traffic we want coming into our network is return traffic for packets too big, ICMP type 3, code 4, so we can use a smaller MTU size to improve Internet transfer efficiency, described in RFP 1191. Then we can deny all other ICMP traffic.

GIACBR1(config-ext-nacl)# permit icmp any any packet-too-big

or alternatively

GIACBR1(config-ext-nacl)# permit icmp any any 3 4

Do not permit ICMP time exceeded, time stamp or ICMP information request messages out to reduce network mapping possibilities.

GIACBR1(config-ext-nacl)# deny icmp any any time-exceeded log-input
GIACBR1(config-ext-nacl)# deny icmp any any timestamp-request log-input
GIACBR1(config-ext-nacl)# deny icmp any any information-request log-input

We included the above three rules just to have specific logs.

GIACBR1(config-ext-nacl)# deny icmp any any log-input

Block all NetBios traffic.

GIACBR1(config-ext-nacl)# deny tcp any any range 135 139 log-input
GIACBR1(config-ext-nacl)# deny udp any any range 135 139 log-input

Block all telnet traffic.

GIACBR1(config-ext-nacl)# deny tcp any any eq 23 log-input

SunScreen firewalls are known to pass packets withy certain TCP Flag combinations. It blocks most illegal combinations, but a couple combinations are still improperly passed to the inside. To counter this problem we will add the following line to block all TCP traffic, which is not part of an established session.

GIACBR1(config-ext-nacl)# permit any any est

This by itself will still permit TCP SYN and ACK scans, but the firewall is known to stop these scans. The complimentary nature of these devices is starting to illustrate some of the value of defense-in-depth concepts.

Block all traffic from reserved and unassigned by IANA (Internet Assigned Numbers Authority), except GIAC's "assigned" IP range of 223.223.223.0/24. A current list can also be found at http://www.liguifried.com/docs/security/reservednets.html.

GIACBR1(config-ext-nacl)# deny ip 1.0.0.0 0.255.255.255 any log-input

```
GIACBR1(config-ext-nacl)# deny ip 2.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 5.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 7.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 14.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 23.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 27.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 31.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 36.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 37.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 39.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 41.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 42.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 49.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 50.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 58.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 59.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 60.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 69.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 70.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 72.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 82.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 840.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 88.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 96.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 197.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 201.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 221.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 222.0.0.0 0.255.255.255 any log-input
! Omitted GIAC's address block 223.223.223.0
GIACBR1(config-ext-nacl)# deny ip 224.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 240.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 248.0.0.0 0.255.255.255 any log-input
GIACBR1(config-ext-nacl)# deny ip 255.255.255.255 .0.0.0.0 any log-
input
```

Cisco offers a rich set of software features on this router and it could be configured as a firewall itself. However, our plans called for the border router to do primarily routing functions and we have added a few ACLs to provide extra defense for services and ports we do permit through the external firewall.

If needed to we could block most fragmentation attacks by adding the following line.

```
GIACBR1(config-ext-nacl)# deny ip any any fragments log-input
```

At this time, we do not believe the rule is needed on the router, because the firewall is expected to provide adequate fragmentation protection and we will not block legitimate fragmented traffic.

If the router were used more as a firewall, we would need to add ACLs to permit established sessions from within the GIAC network and then add an ACL to deny all other traffic, such as:

GIACBR1(config-ext-nacl)# deny ip any any log-input

We will now apply this ACL for inbound traffic on the external interface.

```
GIACBR1(config)# interface serial0
GIACBR1(config -if)# ip access-group ingress_filter in
```

Egress Rules:

Next we will develop our egress ACLs and apply these to outbound traffic on the outside interface, (eth0).

```
GIACBR1(config)# ip access-list extended egress_filter
```

Anti-spoofing egress ACLs can be done simply, since we do not want any IP address not assigned to GIAC Enterprises leaving the router.

```
GIACBR1(config-ext-nacl)# permit 223.223.223.0 0.0.0.255 any
GIACBR1(config-ext-nacl)# deny ip any any log-input
```

Now we add ACLs to block certain ports, even they do originate from our network. GIAC policy prohibits news services on nntp.

```
GIACBR1(config-ext-nacl)# deny tcp any any eq 119 log-input
```

There is no need for syslog traffic across the border router.

```
GIACBR1(config-ext-nacl)# deny udp any any eq 514 log-input
```

Block all NetBios traffic.

```
GIACBR1(config-ext-nacl)# deny tcp any any range 135 139 log-input
GIACBR1(config-ext-nacl)# deny udp any any range 135 139 log-input
```

We will now apply this ACL for inbound traffic on the external interface.

```
GIACBR1(config-ext-nacl)# exit
GIACBR1(config)# interface serial0
GIACBR1 (config-if)# ip access-group egress_filter out
GIACBR1 (config-if)# exit
GIACBR1(config)# Ctrl Z
```

As the final step we must save the running configuration to non-volatile memory (NVRAM), or else it will e lost at the next reboot.

```
copy running-config startup-config
```

If we were using the router with a "deny all" rule, we would need to add additional ACLs to permit internal traffic outbound to the Internet. In this configuration, it is not necessary. The above router hardening procedures and ACLs are expected to provide the border router with reasonable protection and block the only known problems with the firewall. In addition we have provided significant overlap protection to the firewall rule set, with many of the ACLs. Thus enhancing our defense in-depth network design.

External Firewall (FW-1):

Policy:

Our external firewall (FW-1) is a SunScreen v 3.1system and will be configured as a statefull device. SunScreen firewall management allows us to create host addresses, address groups and address range objects as well as service objects. The firewall rules will make use of these common objects opposed to specific IP addresses. This will permit easy setup and fewer errors when policy rules change. Also, all SunScreen firewalls can use the same set of objects, much the same as NetFilter does.

Our object names have been carefully selected to indicate if the object is a server (XxxSrv), a group of addresses (XxxGrp) or an entire network (XxxNet). A group is comprised of used IP addresses assigned to live hosts. Whereas, Net indicates the full possible compliment of IPs assigned to that network segment. Group objects allow inclusion of other groups. Thus by careful planning in object construction, when an IP address changes or we add a new host, we should have to add the new address to only a single group and all other objects are automatically updated. This approach reduces the potential for errors in complex firewall rule sets, particularly after a series of changes have been made over a long period.

By assigning net objects to the stealth interfaces, we automatically create ingress and egress filters that will meet many of our filtering requirements. We will duplicate these filters on the border router, just to remove such traffic prior to it hitting the outside firewall interface for efficiency purposes.

On a SunScreen stealth interface, one without an IP address, there are no ports to be probed and really no ports to be closed. The state table prevents unauthorized traffic that did not originated from within the GIAC network or otherwise permitted from the outside interface. To date there no known vulnerabilities against the SunScreen firewall v 3.1 running in stealth mode, other than DoS style attacks.

We will present all rules for the external and internal firewalls in this section. The details for the following rules were developed in Sections 2.9.x of Assignment I. At this point we will present details similar as they would appear on a screen from a SunScreen central management console.

The base firewall rule is that all traffic is denied except what is explicitly permitted. Packets are forwarded or dropped on a first match rule. Thus, on SunScreen firewalls, we want to include specific rules before general rules covering the same service and source and/or destination addresses. This is to ensure that exceptions or special cases are either allowed or denied as intended. Experience has shown that otherwise exact order does not make a significant difference on light to moderately loaded SunScreen firewalls. We will still place the most commonly accessed rules above less accessed rules, but we find that grouping rules by service enhances understandability and thus effectiveness of our access policies. For example, all of the http and ssh rules should precede rules for ntp and ftp. Please see the SunScreen Firewall Tutorial for an exact placement of the proposed rules.

For a little brevity, we will not detail how all of the individual IP addresses and groups roll into group objects and definition of the XxxNet objects. We believe our naming convention is reasonably clear and detailed IP addresses were identified in section 2.9 of Assignment I. Please see section 3.8 SunScreen Tutorial.

Operating System Install:

The first step is to install the Solaris core distribution with several add-on packages and patch sets.

SunScreen EFS Install:

First we install the SunScreen Administration product on the Administration console. This step will install the self-generated

administration certificate and SKIP to provide secure communication between the firewalls and the central management console.

Next we install the SunScreen software and screen's certificate on each firewall device. After the system is built we will run the hardening script, /usr/lib/sunscreen/lib/harden_os, to cleanup the OS. Finally, SKIP is enabled on the central management console creating a secure communication channel with the firewalls.

Next we install the SunScreen software on each firewall device and each screen's certificate. While installing we must check "routing mode" even through the firewall is running in stealth mode. This will allow it run in a mixed mode environment to support both statefull inspection and proxy rules. Finally, SKIP is enabled on the central management console creating a secure communication channel with the firewalls.

Interface Configuration:

First step is to create and enter all of the required objects, such as host names, group names and network ranges. This will permit a more efficient configuration process. Next we setup the interfaces as follows.

Firewall	Interface	Type	Address Group	Router Address
FW-1	qfe0	stealth	Outside	223.223.223.33
FW-1	qfe1	routing	Eosn	223.223.223.65
FW-1	qfe2	stealth	Ersn	N/A
FW-1	qfe3	stealth	Giac	223.223.223.34
FW-1	eri0	admin	Fw-1-Admin	N/A
FW-2	qfe0	stealth	NonSrv	N/A
FW-2	qfe1	stealth	Iosn	N/A
FW-2	qfe2	stealth	Irsn	N/A
FW-2	qfe3	stealth	Database	N/A
FW-2	qfe4	stealth	Nids	N/A
FW-2	qfe5	stealth	Syslog	N/A
FW-2	qfe6	stealth	Dev	N/A
FW-2	eri0	admin	Fw-2-Admin	N/A

Administration Configuration:

We need to add the screen configuration values that permit the firewalls to act as bridges in a subnet.

FW-1
Stealth Net Address 223.223.223.32 255.255.255.0
Allow Routing Traffic Yes
Name Service None
Certificate Discovery Yes

Administration IP Address FW-1-admin
Administration Certificate FW-1-cert

FW-2
Stealth Net Address 172.17.1.0 255.255.255.0
Allow Routing Traffic Yes
Name Service None
Certificate Discovery Yes
Administration IP Address FW-2-admin
Administration Certificate FW-2-cert

NAT Rules:

Since GIAC Enterprises employs about 50 people, a single dynamic NAT address (GiacHosts) is more than sufficient to handle all outbound traffic.

Firewall	Type	Src. Address	Dst. Address	Trans. Src.	Trans. Dst.
FW-1	dynamic	Inside	Outside	NatGiac	Outside
FW-1	dynamic	Outside	NatGiac	Outside	Inside

Inside is all possible IP addresses used on the internal GIAC network, excluding the NIDS and firewall central management networks.

NatGiac is the single IP address used for NAT at GIAC, 223.223.223.50.

Outside is all legal external routable IP addresses, excluding GIAC's assigned public address block.

Central Management Rules:

To permit central management on SunScreen firewall we must run SKIP on the firewalls and the central management system. This will permit all firewalls on the private central management network to communicate.

Service	Source	Destination	Access
Certificate discovery	*	*	allow
skip	*	*	allow

Mail System Rules:

GIAC's mail system is really comprised of two separate systems. First is for standard in and out bound e-mail. The second system is the Emu-mail system that is used by GIAC employees to securely obtain their GIAC e-mail from the Internet. Only the mail gateway is permitted to send smtp to the outside, deny smtp to all other servers.

Firewall	Service	Source	Destination	Access
FW-1	smtp	Outside	MailGate	allow
FW-1	smtp	MailGate	E500	allow

FW-1	smtp	ExtGrp	E500	allow
FW-2	smtp	E500	MailSrv	allow
FW-2	smtp	UserGrp	MailSrv	allow
FW-2	pop3	UserGrp	MailSrv	allow
FW-2	smtp	IntGrp	MailSrv	allow
FW-2	smtp	MailSrv	E500	allow
FW-1	smtp	E500	MailGate	allow
FW-1	smtp	MailGate	Outside	allow
FW-1	ssl	Outside	EmuMail	allow
FW-1	pop3	EmuMail	E500	allow
FW-2	pop3	E500	MailSrv	allow
FW-2	pop3	MailSrv	E500	allow
FW-2	pop3	E500	EmuMail	allow
FW-1	smtp	SrvGrp	Outside	deny
FW-1	pop3	SrvGrp	Outside	deny

ntp Rules:

The two primary internal GIAC ntp servers synchronize clocks with 5 different stratum two ntp clocks and all other GIAC servers synchronize with these two servers. The special case is the NIDS console serves the ntp server for the NIDS network that lacks a route to any other network. Only the internal ntp servers require access to the outside on port 123.

Firewall	Service	Source	Destination	Access
FW-1	ntp	NtpSrvGrp	NtpClocks	allow
FW-2	ntp	NtpSrvGrp	NtpClocks	allow
FW-1	ntp	EsnGrp	NtpSrvGrp	allow
FW-2	ntp	EsnGrp	NtpSrvGrp	allow
FW-2	ntp	UserGrp	NtpSrvGrp	allow
FW-2	ntp	Inside	NtpSrvGrp	allow
FW-2	ntp	NidsCon	NtpSrvGrp	allow
FW-1	ntp	BR	NtpSrvGrp	allow
FW-2	ntp	BR	NtpSrvGrp	allow
FW-1	ntp	SrvGrp	Outside	deny

DNS Rules:

GIAC will run a split DNS system, with two DNS servers on both the external and internal networks. Since we have built the system with peer DNS servers, as opposed to primary and slave, we do not need to permit traffic on udp port 53. All servers except for the two internal DNS servers are denied DNS access to the outside.

Firewall	Service	Source	Destination	Access
FW-1	dns	Outside	ExtDnsGrp	allow
FW-1	dns	IntDnsGrp	Outside	allow
FW-1	dns	MailGate	DnsIsp	allow
FW-1	dns	ftp	DnsIsp	allow
FW-1	dns	EmuMail	DnsIsp	allow
FW-2	dns	UserGrp	IntDnsGrp	allow
FW-2	dns	IntGrp	IntDnsGrp	allow
FW-1	dns	SrvGrp	Outside	deny

Ping Rules:

Whereas, ping is a powerful trouble-shooting tool, it can be used in DoS style attacks. At GIAC Enterprises ping is generally restricted to a local subnet, to prove basic network connectivity. Hosts on the IT subnet may ping any host anywhere on the GIAC network and externally as otherwise permitted. Otherwise, ping across a firewall is not permitted.

Firewall	Service	Source	Destination	Access
FW-1	ping	It	*	allow

http Rules:

There are four web and two squid servers in the GIAC network. These include the exterior or public web server; the Emu-mail server; the internal web server; the 9iAS production web server for business applications and the 9iAS development web server. A squid server acts as the front-end for all traffic to the public web server. The second squid server is used as a reverse proxy for all outgoing http(s) traffic. Direct public access to the external web server is not allowed. We deny all internal servers http(s) access to the outside.

Firewall	Service	Source	Destination	Access
FW-1	http	Outside	ExtProxy	allow
FW-1	https	Outside	ExtProxy	allow
FW-1	http	UserLanNet	ExtProxy	allow
FW-1	https	UserLanNet	ExtProxy	allow
FW-2	http	UserLanNet	IntWebGrp	allow
FW-2	http	UserLanNet	IntProxy	allow
FW-2	https	UserLanNet	IntProxy	allow
FW-2	http	IntProxy	Outside	allow
FW-2	https	IntProxy	Outside	allow
FW-2	http	NidsCon	Outside	allow
FW-1	http	IntProxy	Outside	allow
FW-1	https	IntProxy	Outside	allow
FW-1	http	NidsCon	Outside	allow

FW-1	http	SrvGrp	Outside	deny
FW-1	https	SrvGrp	Outside	deny

Secure Shell Rules:

The Secure Shell Server is used by all GIAC's customers, partners and suppliers for transfer of fortunes. Secure shell and secure copy are used for any remote server access within the GIAC network. Hosts on the IT subnet are allowed to ssh/scp to any server on the GIAC network. The Quality Control group has scp access to the Fortunes database server. We deny ssh access for most servers to the outside to reduce back channels.

Firewall	Service	Source	Destination	Access
FW-1	ssh/scp	Outside	SshSrv	allow
FW-2	ssh/scp	DbForSrv	SshSrv	allow
FW-1	ssh/scp	DbForSrv	SshSrv	allow
FW-1	ssh/scp	VpnSrv	EOSN	allow
FW-1	ssh/scp	VpnSrv	SrvGrp	allow
FW-2	ssh/scp	VpnSrv	SrvGrp	allow
FW-2	ssh/scp	QuaCon	DbForSrv	allow
FW-1	ssh/scp	SrvGrp	Outside	deny
FW-2	ssh/scp	It	SrvGrp	allow
FW-1	ssh/scp	It	EsnGrp	allow
FW-1	ssh/scp	It	Outside	allow

VPN Rules:

The VPN server is used exclusively by GIAC employees to access internal hosts from the Internet. Authenticated users on the VPN Secure Shell server are permitted the same access as users on the internal IT subnet, provided they have valid accounts on servers. Depending upon specific requirements of the collection agent we contract with for payment collections, we will need a VPN connection and presently assume it will be IPSec and ESP.

Firewall	Service	Source	Destination	Access
FW-1	isakmp	Outside	VpnSrv	allow
FW-1	esp (IP ID 51)	Outside	VpnSrv	allow
FW-1	isakmp	VpnSrv	ColAgent	allow
FW-1	esp (IP ID 51)	VpnSrv	ColAgent	allow

syslog Rules:

All GIAC servers and the boundary router send syslog information to the central syslog server cluster.

Firewall	Service	Source	Destination	Access
FW-1	syslog	Br	SyslogSrv	allow

FW-2	syslog	Br	SyslogSrv	allow
FW-1	syslog	EsnGrp	SyslogSrv	allow
FW-2	syslog	EsnGrp	SyslogSrv	allow
FW-2	syslog	DskTopSrvGrp	SyslogSrv	allow
FW-2	syslog	SrvGrp	SyslogSrv	allow

ftp Rules:

GIAC's public ftp server permits logons only from real users. Anonymous login is not permitted. Access is permitted from both the Internet and the GIAC User segments. IT users are permitted external ftp access. General GIAC users are not permitted external ftp services.

Firewall	Service	Source	Destination	Access
FW-1	ftp	Outside	ftp	allow
FW-2	ftp	UserGrp	ftp	allow
FW-1	ftp	It	Outside	allow

Oracle Rules:

Since we have located the Oracle 9iAS and Oracle 9i database servers on different networks off the internal firewall, we must allow some specific traffic.

Firewall	Service	Source	Destination	Access
FW-2	OrcList	OrcAs	BusDb	allow
FW-2	sqlnet	OrcAs	BusDb	allow

Miscellaneous Rules:

There are a few more rules needs for special situations. GIAC's IT group is permitted to attempt telnet sessions to hosts external to the GIAC network. GIAC will maintain a special IP address group of externally block sites. To permit handling of fragments properly, we will permit icmp unreachable messages to be returned. This is important if we decrease a host's MTU value to improve Internet performance.

Any traffic not explicitly permitted, will be dropped by the firewalls. We will open rules on the internal firewall (FW-2) for local devices only. We permit only the IT staff to telnet to the outside. ftp is permitted from the User Group to the ftp drop box and the IT staff can ftp anywhere externally. No other ftp traffic is permitted externally.

Our Oracle database servers produce reports that need to be printed in the User network. Any GIAC server may print to printers on the IT subnet. We want to log anything of interest in the remaining traffic, but if we do not filter out the multicast entries, our logs are going to be bloated. We deny all packets destined to a multicast address. A final deny log captures everything that falls out of the bottom of the rule set.

Firewall	Service	Source	Destination	Access
FW-1	telnet	It	Outside	allow
FW-1	telnet	SrvGrp	Outside	deny
FW-1	http	UserGrp	IpBlock	deny
FW-1	ICMP type 3 code 4	Outside	GiacNet	allow
FW-2	lpd	ForSrv	UserGrp	allow
FW-2	lpd	DevSrv	UserGrp	allow
FW-2	lpd	SrvGrp	It	allow
*	*	*	multicast	deny – no log
*	*	*	*	deny –log

A complete listing of the firewall policy and common objects is documented in Appendix B. This listing is the actual policy from the firewall used in Assignment III and was made with the following command line.

ssadm –x active > /tmp/appendix_b

Internal Firewall (FW-2):

Policy:

The internal firewall creates the various "subnets" in the Secured Server Zone. The intentions are to partition the servers from the users and themselves in a manner that only authorized individual workstations or servers have access to these central resources. It also provides an additional layer of defense from external attacks.

Internal Firewall Rules:

All rules for the internal firewall (FW-2) were described with the rules for the external firewall above.

VPN Server:

Policy:

Our VPN server provides IPSec services from remote hosts to the GIAC VPN gateway. Only GIAC employees will be authorized access via this network access path. The two groups of GIAC employees needing to use this service are the traveling sales team and the GIAC IT staff when working from home. In both situations, they will be using GIAC issued systems and are considered "road warriors" in that their IP addresses are not static, since they are coming through remote ISPs. If their IP address is assigned via DHCP, we may experience problems when the lease expires. In some cases the only option will be to recreate the IPSec session. Since the GIAC IT staff use LINUX 7.3 systems at home, making this IPSec solution work for them is simple. GIAC's traveling sales team uses Win2000 laptops. The native IPSec service on Win2000 is compatible with FreeS/WAN v 1.99.

We have selected the FreeSwan product as described earlier for our VPN service and will run it on Red Hat Linux 7.3, which is the latest Red Hat release supported. Our design calls for ESP instead of AH to obtain both authenticity and data encryption, which is not an option with authentication headers (AH). AH would not be a problem since the placement of the VPN server on the Exterior Services Network is not affected by our NAT services. ESP will retard man-in-the-middle style attacks because of the authentication process includes the original headers.

We will operate ESP in tunnel mode with 2048-bit RSA keys. Since this VPN service is partly to enable GIAC IT staff to work from home, we need strong security. Consequently, we want to use main mode for Internet Key Exchange (IKE), besides FreeSwan does not support aggressive mode. Triple DES (3DES) will be used for data encryption and key life will be restricted to best practice of 60 minutes. Finally, MD5 will be used for the HMAC.

FreeS/WAN provides two methods to build secured links, manual keying and automatic keying. The first requires secured transfer of a secret and automatic keying permits each system to authenticate each other and negotiate their own keys. We will use the automatic key generation using Internet Key Exchange (IKE) main mode. If a key is compromised, only the data transmitted until the next re-keying is compromised. With manual keying, the loss will continue until you learn about it and change keys. Manual key exchange tends to become a significant administrative load as the number of client systems increase. With public/private keys, we have to protect only our private key. The protection must be very strong. Additional advantage is that RSA authentication does not require static IP addresses. Remote systems can identify themselves to the server by name. This is critical for our “road warriors”.

We will need to assign a unique Fully Qualified Domain Name to each client system in the form of @it1.giac.com. This name does need to match any official Internet domain name.

Since we do not want any routing protocols on our VPN server we will use static routing only.

```
#Set default and static routes
route add default gw 223.223.223.33
route add –net 172.17.0.0 netmask 255.255.0.0
route add –net 172.16.0.0 netmask 255.255.0.0
route add –net 223.223.223.32 netmask 255.255.255.0
route add –net 223.223.223.64 netmask 255.255.255.0
```

FreeS/WAN's configuration file, /etc/ipsec.conf, is made up of three parts; the configuration setup; default connections and specific connections. Following is the configuration process for the FreeS/WAN ipsec.conf file.

```
#configuration setup for all connections
config setup
interfaces="ipsec0=eth0"
klipsdebug=none
plutodebug=none
plutoload=%search
plutostart=%search
```

In the configuration setup, we identified the VPN interface, eth0, and specified none for our debug modes for klipsec and Pluto.

The default connections section details the defaults for any connection. Basically, we will set the key exchange method as IKE, RSA for authentication and the key's lifetime. The keyingtries value means that re-keying negotiations are very persistent.

```
#Default connections
conn %default
keyingtries=0
keyexchange=ike
keylife=1h
authby=rsaig
auto=add
```

The specific connections sections is really the IPSec Security Policy Database (SPD). All unique details for any given connection can be detailed in this section. If some of our VPN connections were established with fixed IP addresses we would need two parts in this section. However, all of our VPN users are "road warriors" and can be covered in one part.

FreeS/WAN uses a notation of leftness and rightness in configuration, which system is which is not important, by convention, local is "left" and remote is "right".

The client side on the connection will be configured as a VPN gateway with the "network" comprised of the localhost. Consequently, we will tunnel as opposed to transport for our connection type.

```
#Specific connections
conn remote-giac
type=tunnel
leftid=223.223.223.41
left=223.223.223.41
```

```
leftnexthop=223.223.223.33
leftsubnet=223.223.223.0.24
leftrsaigkey=0x9df7erj23bn6op. . . . .

right=%any
rightid=0.0.0.0
spi=0x300
esp=3des-md5-96
espenckey=[192 bits]
espauthkey=[128 bits]
keyingtries=1
rightrsaigkey=0xg45mo76ap26ig90. . . . .
auto=add
```

For the server or left side we set the VPN server's IP address, it's RSA key, the border router's address and the local subnet network address. For our clients, the right side, we set encryption and authentication methods, key lengths, allow remote clients to initiate sessions and stop trying if the circuit or VPN server is down.

The remote clients /etc/ipsec.conf file will resemble the following. Left (local) is now the client, while "right" is the remote VPN server.

```
conn giac
leftid=@it1.giac.com
left=%defaultroute
leftnexthop=%defaultroute
leftrsasigkey=0s2jf63bd61os....

right=223.223.223.41
rightid=@giac.com
rightsubnet=223.223.223.0/24
rightrsasigkey=0sb0ui76sp82at... .
auto=add              # for the sales team
auto=start            # for the IT staff, connect on system boot
```

Secure Shell Server:

Policy:

All suppliers, customers and partners will use our secure shell server to upload or download fortunes, using secure copy (scp) v 3.5. Our secure shell server will be a very minimum build using Red Hat 8. Binaries for ssh, telnet, ftp, http, etc. will not be installed or else removed. ssh must be installed on the host for scp to work. The external firewall (FW-1) will block all packets from the secure shell server except syslog traffic directed to the internal syslog server and ntp traffic to the internal ntp servers.

Each external user will be assigned a unique user ID and strong eight byte password. When a user authenticates with the ssh server they will be held in their home directory by chroot. The only function a remote user requires is the ability to either upload or download files. Consequently, we do not need to provide a very extensive set of binaries. To set this up we need to follow a few steps.

First, we will compile the source code and enable static libraries.

```
./configure --enable-static
make
make install
```

Then we run ssh-chrootmgr as root adding the required user names.

```
ssh-chrootmgr customer1 supplier1 partner1 etc
```

Then edit /etc/ssh2/sshd_config.

```
ChRootUsers     customer1,supplier1,partner1,etc
```

This allows us the maximum control over user access, but is another step in managing user accounts. If we place all suppliers in one group. We just add the following line once to /etc/ssh2/sshd_config.

```
ChRootGroups    customers,suppliers,partners
```

Our final step is edit the /etc/passwd file to create a dummy shell (/bin/ssh-dummy-shell) for each remote user's shell.

In the /etc/ssh2/sshd_config file we will disallow w-window forwarding and tcpforwarding. This will prevent users from forwarding ports or tunneling to other GIAC systems. We will exclusively use ssh version 2 to avoid problems with version 1. We switched the listening port from 22 to 65322, this will require a port change on the ssh clients.

The /etc/ssh2/sshd_config file will be edited as shown below.

```
Port 65322
Protocol 2
Hostkey /etc/ssh2/ssh_host_dsa_key
KeyregistrationInterval 3600
ServerKeyBits 768
SyslogFacility AUTH
LogLevel INFO
```

```
LoginGraceTime 600
PermitRootLogin no
StrictModes yes
PublickeyAuthentication yes
RhostsAuthentication no
IgnoreRhosts yes
/etc/ssh2/ssh_known_hosts
RhostsRSAAuthentication no
HostbasedAuthentication no
PasswordAuthentication yes
PermitEmptyPasswords no
ChallenResponseAuthentication no
X11Forwarding no
X11DisplayOffset 10
PrintMotd yes
PrintLastLog no
KeepAlive yes
Banner /etc/issue.net
ReverseMappingCheck yes
AllowTcpForwarding no
Subsystem sftp /usr/lib/ssh/sftp-server
ChRootGroups    customers,suppliers,partners
```

Tripwire will be used as a host IDS to alert on any changes to the binaries and configuration files.

Use of TCPWrappers with ssh will provide additional security in use of /etc/hosts.allow and /etc/hosts.deny as well as the central logging back to our syslog server. ssh will have to compiled ssh to use TCPwrappers.

/etc/hosts.allow is configured as:

```
sshd : ALL
```

From a business perspective we will not know the IP address of our customers, partners or suppliers, thus filtering based on source IP is not practical.

/etc/hosts.deny is configured as:

```
sshd : ALL
```

The above security measures will lock down the secure shell server from unauthorized system or file access and prevent authorized users from using this system as a platform to perform other unauthorized activities. If

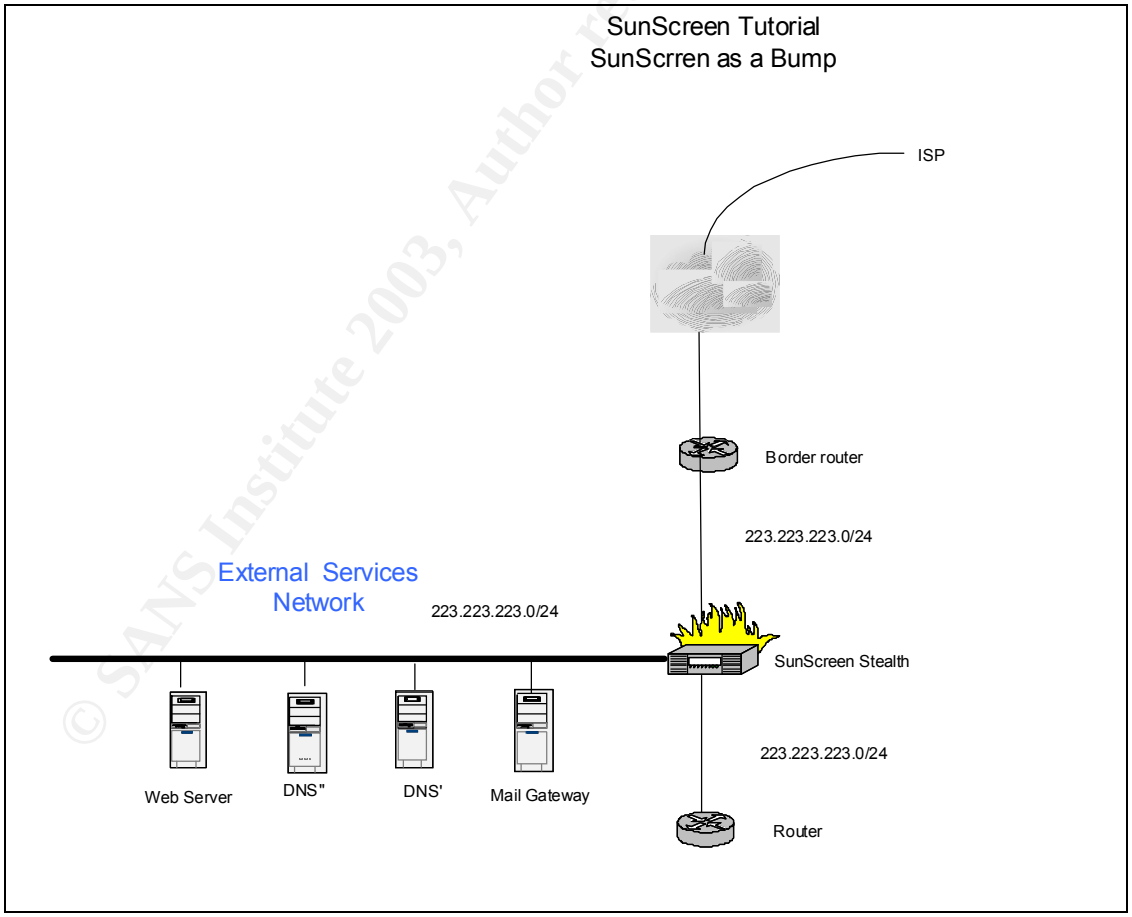
we still felt unsure about this criteria system, we could run Netfilter/IPTables on it as a host based firewall.

SunScreen Firewall Tutorial:

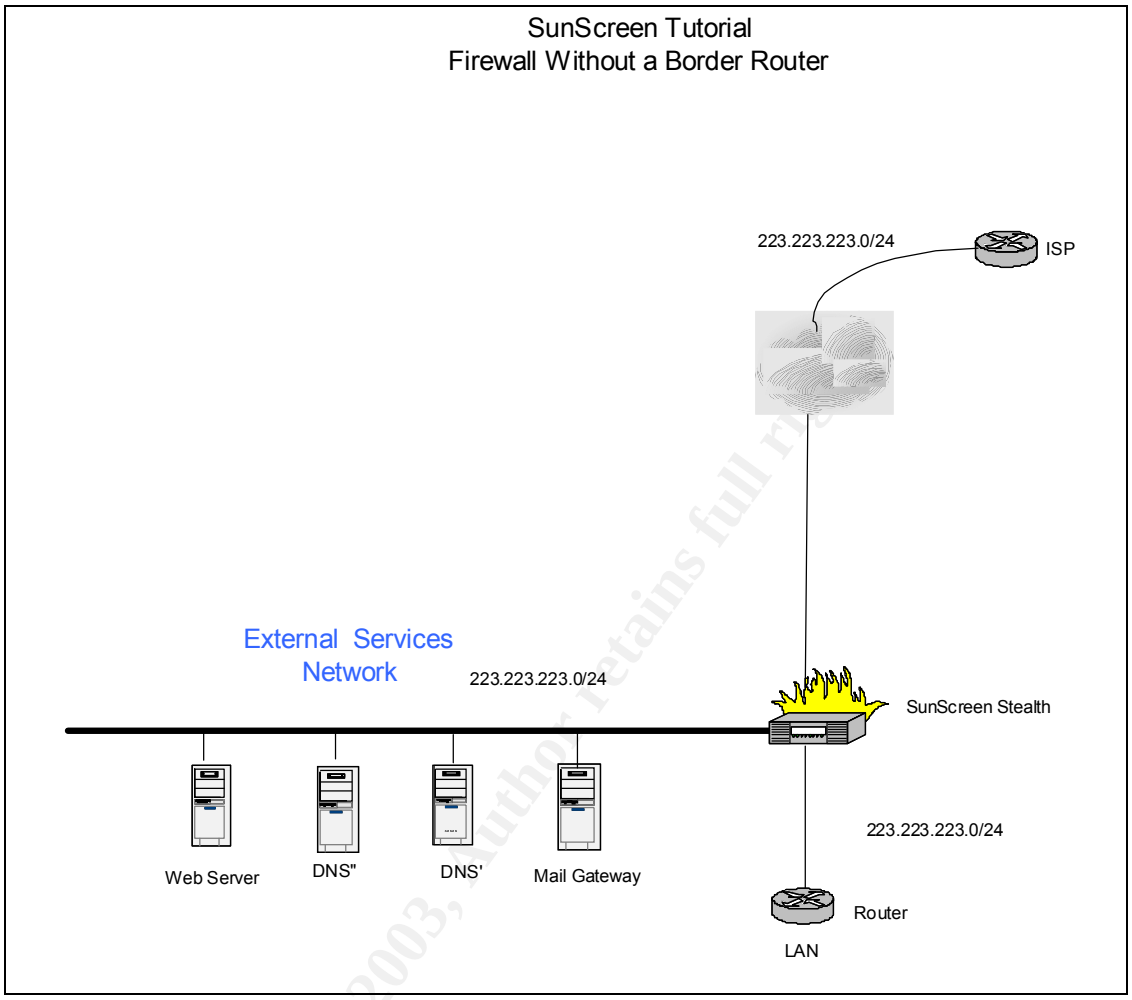
Network Design:

A SunScreen firewall maybe configured in three modes, routing, and stealth and mixed. Routing mode is based on packet filtering technology with assigned IP addresses and addressable ports. The routing mode permits inclusion of proxy rules for http, https, ftp, smtp and pop. Stealth mode is a statefull firewall without IP address assignments on the interfaces or addressable ports. Any of the modes may be configured with local or central management. Mixed mode allows simultaneous combinations of stealth and routable interfaces. This tutorial will focus exclusively on configuration in stealth mode with local management.

SunScreen firewalls appear to be stealth because they do not have IP addresses assigned to any interface, excluding an optional management port. In essence, they form a "bump" in the network, much as a bridge does. Consequently, it sits in the middle of a network segment, with the same network number and network mask on all interfaces.



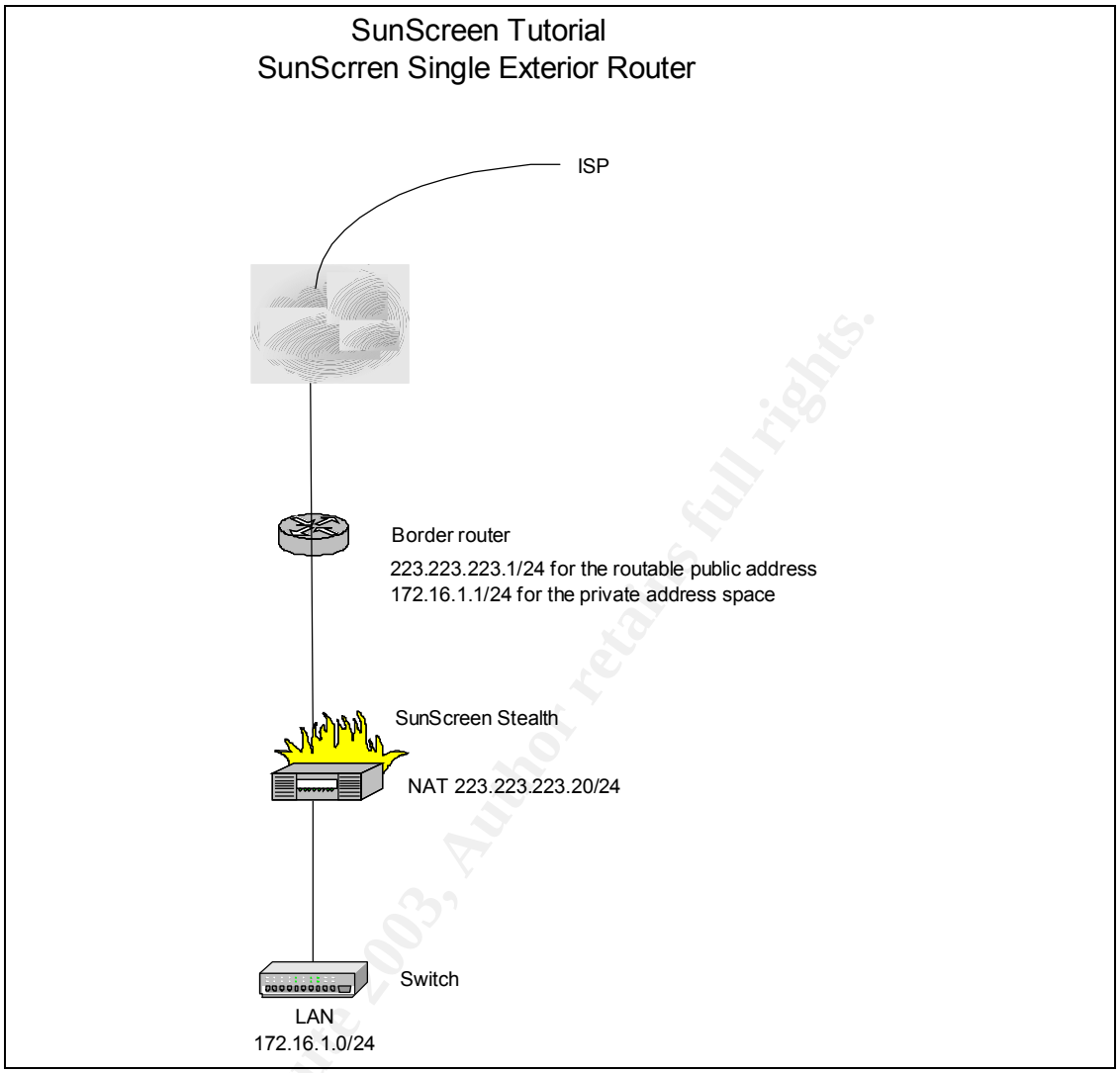
If the ISP provides sufficient IP addresses in the space on our side of their local router, we do not technically have to use a border router.



Clearly, a border router in our control is highly desirable from a security point of view.

There are a couple of options to assign IP addresses. First, is to place a router on both the outside and inside interfaces, with required service hosts on one or more service networks. Second option is to use a single router as the border router and place a second IP address on the inside interface, omitting the interior router.

SunScreen runs on current Sparc based systems from Sun Microsystems. The number of installed NICs and amount of main memory limits number of network interfaces. Use of multiple quad NICs permits sufficient network segments for nearly any network design.



While this design option works, it is not exactly standard or obvious. Internet traffic will be routed to the 223.223.223.1 address and subsequently seen by the exterior stealth interface. The firewall will then run NAT on this traffic and direct all the translated private address traffic to the inside interface correctly. All the hosts on the LAN will use the 172.16.1.1router address as their default gateway. However, the firewall will translate all out bound packets to 223.223.223.20 before they reach the gateway. I prefer the dual router option for security and simplicity of the IP addressing scheme.

The main attraction of the above scheme is for small remote offices where SunScreen firewall acts as a VPN gateway for all inbound and outbound traffic routed to a central office facility.

Naming Standards:

All of the rules on a SunScreen firewall utilize common objects for individual hosts, groups of hosts, networks and services. Many standard services are already defined by default.

Tricks and Tips:

- Adopt a naming convention that is concise and descriptive.
- Use real hostnames where possible for individual systems and compound names for groups or networks.
- The object names are case sensitive, so a system where the first letter of each naming component is capitalized improves readability, e.g. ExtDns, IntNtp, etc.
- Use prefixes opposed to suffixes for groups of similar objects, because the pull down menus will be alphabetically sorted. This will keep groups together better, e.g. NetInside, NetOutside, etc.

One might consider using real host names as object names on the firewall as a security risk. If a hacker has gained enough access to the firewall to read the object name database, they also have enough snoop access to learn the same thing other ways. Therefore, I consider the risk to be far outweighed by the clarity of the firewall rules.

Failure to manage object names well can lead to duplicate names for the same physical resource or service. When duplication occurs, and you edit one object name to reflect a change, such as a new IP address, some rules may be wrong, because they are using a duplicate and forgotten object name.

Firewall Installation:

We will assume that Solaris 8 core distribution has already been installed and properly patched. The SunScreen Installation Manual lists a number of additional packages that must be installed with the core distribution. All of our configuration work will be done from the main console. Do not attach any network cables until the policy is completed and activated.

SunScreen firewall installation is straightforward and may be completed from either the GUI or command line. The command line option is much faster.

First, mount the distribution CD and load the packages.

```
pkgadd –d /cdrom/cdrom/sparc
sync
init 6
```

Accept the defaults for all questions. After the packages have been added we will modify the PATH and MANPATH variables in /etc/profile.

```
PATH=/opt/SUNWcgi:/SunScreen/bin:/usr/dt/bin:$PATH
MANPATH=$MANPATH:/opt/SUNWcgi/SunScreen/man
```

Now run install script.

```
ss_install
sync
init 6
```

You can accept the defaults, and hit "ENTER" when prompted for a certificate, unless you are configuring central management. Central management is a more complex install and probably should not be tried until you have successfully built a stand-alone system or screen. After the install finishes, reboot the system.

Depending on the requirements, a SunScreen firewall may have three or more interfaces. Use of quad NIC is necessary in these cases. Solaris will not automatically add or plumb interfaces on a quad NIC and the standard ifconfig qfe0 plumb will not work with a SunScreen. You must use a special script "plumbsunscreen" to prepare these extra interfaces.

Tricks and Tips:

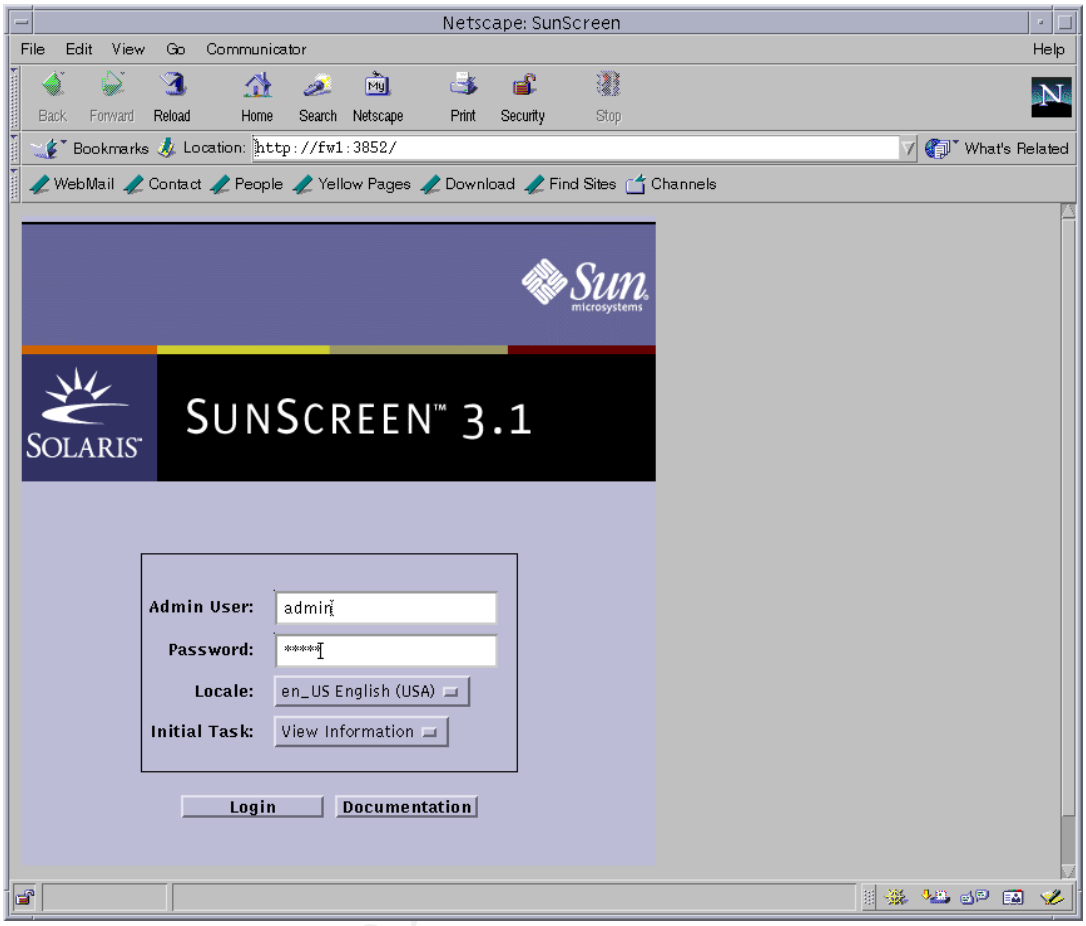
- Do not attempt to run other Solaris hardening scripts on a SunScreen system. Sun has already provided a custom hardening package, /usr/lib/sunscreen/lib/harden_os, other changes may have uncertain effects on system behavior and performance.

Getting Started:

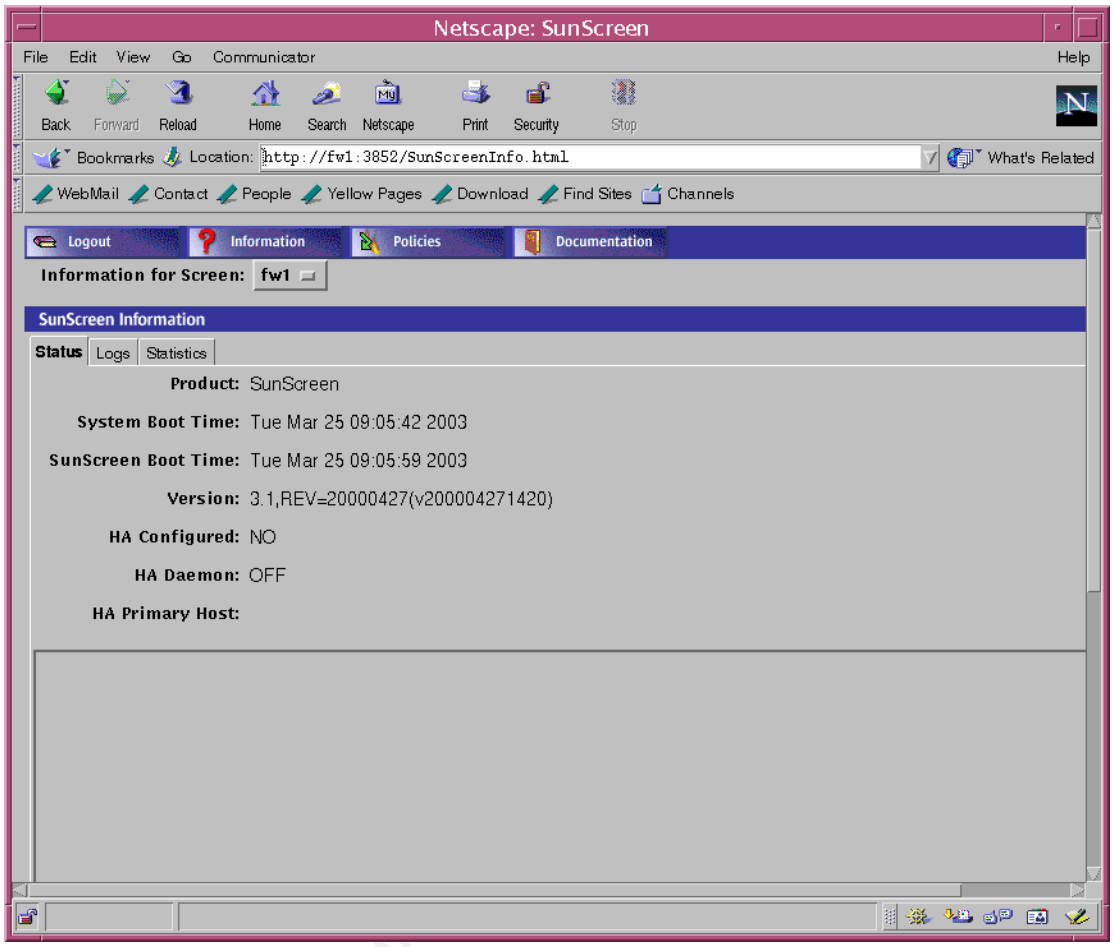
Sun offers two interface options. A standard command line interface (CLI) is very fast and efficient, when you understand what you are doing. Alternatively, you may use a web browser; Netscape is preferred, to perform all standard configuration functions. Some functions, such as backups and log transfers must be done from the CLI. This tutorial will cover the GUI version.

Logging In:

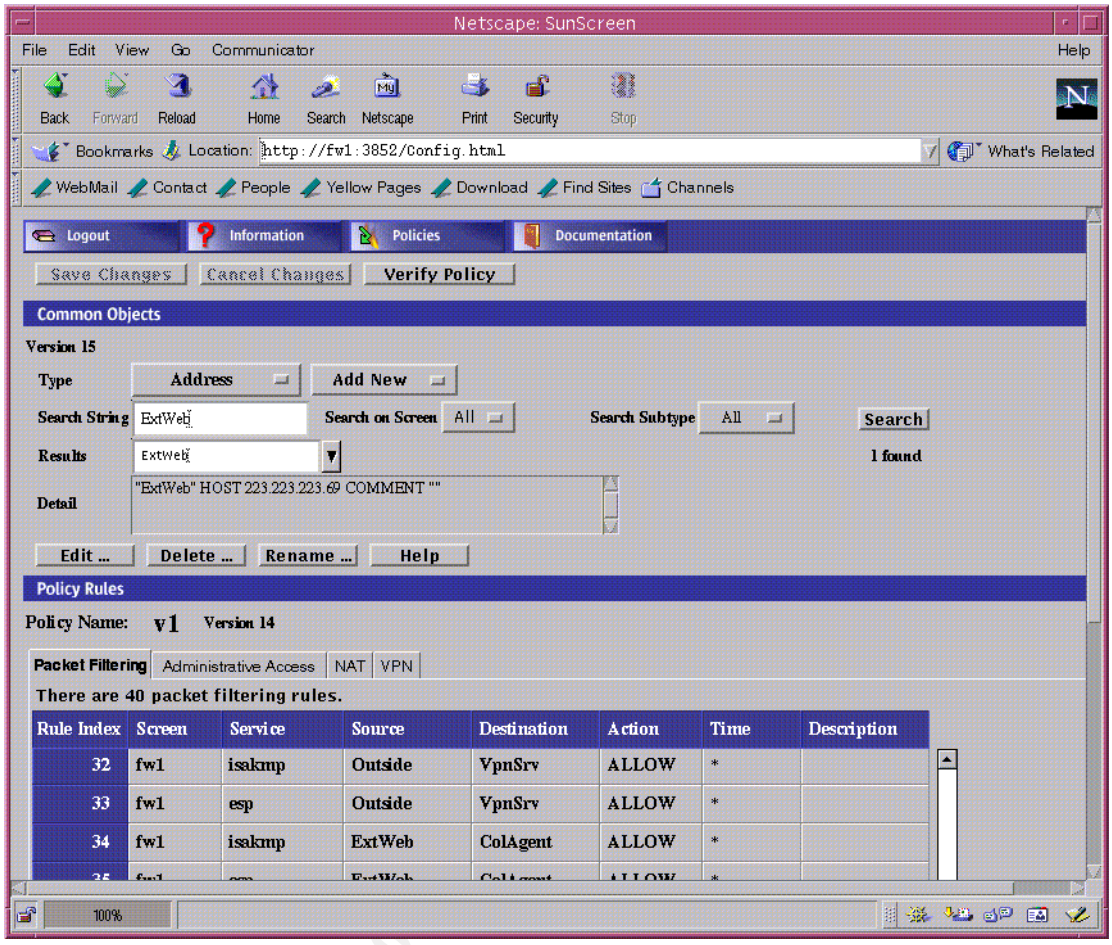
From a Netscape window, enter http://1.2.3.4:3852, where 1.2.3.4 is the IP address or hostname of the firewall. Enter "admin" for username and "admin" for the default password.



The main screen appears next.

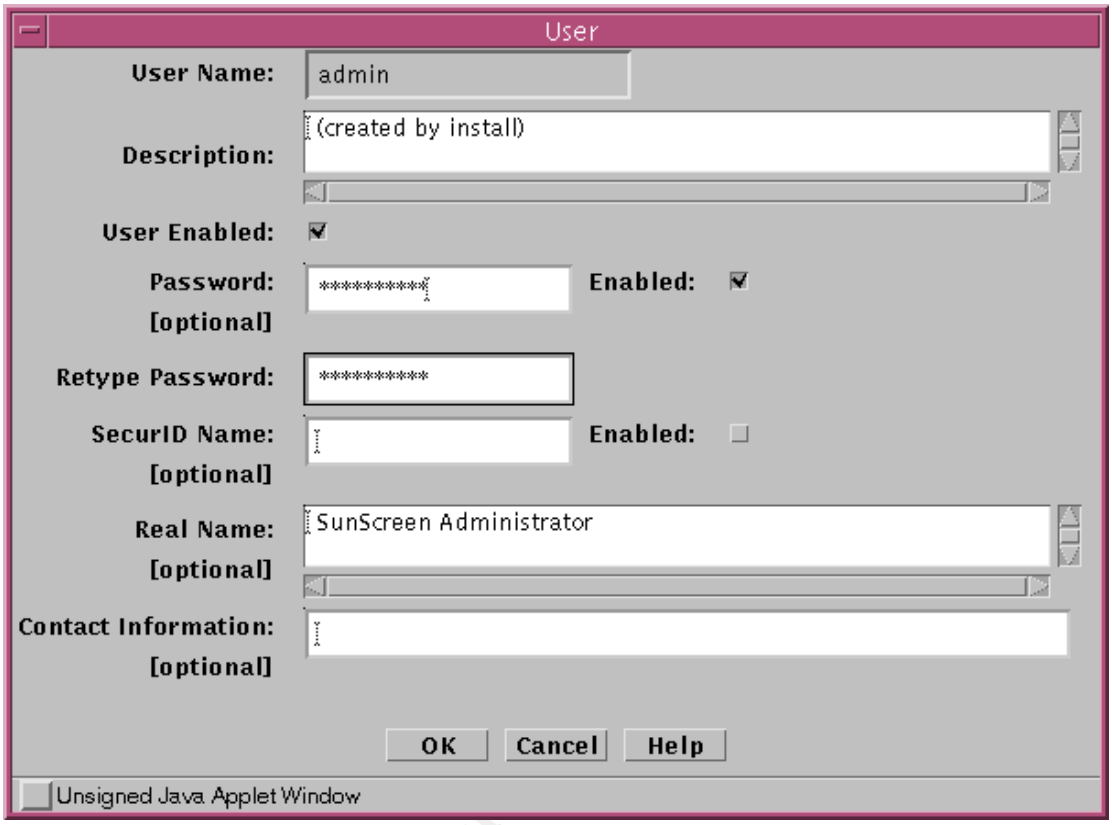


Select the “Policies” tab in the top blue strip. This will present the main Policy Window that is where most configuration steps are initiated. The “Information” tab provides GUI access to the firewall’s log files. The log files may be accessed and used with other scripts for analysis from the command line interface.



Changing the Default Password:

First thing we want to do is change the one default password. Click the selection button on the right side of the Address button. Then select “Admin User” in the type list and click the search button. Select “administrator” in the result field and select “Edit”. This will present the following screen. After making the password change, hit “Save” and then activate the new edited policy.

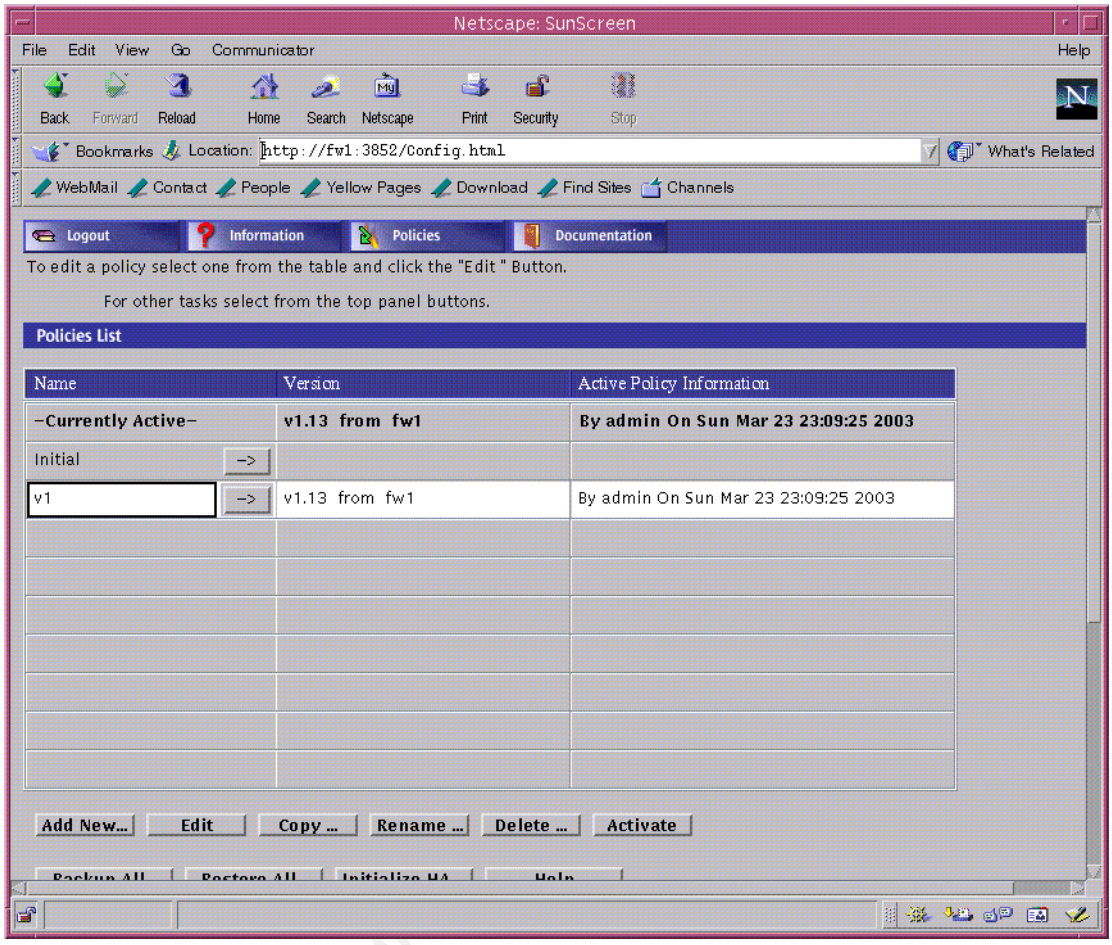


Trick and Tips:

- Make the Solaris root and SunScreen admin passwords different. This will permit non-firewall administrators system access to run snoop and perform limited system level operations.

Selecting a Base to Start editing:

At this time, the firewall is wide open and unfiltered traffic is permitted across all interfaces. Sun includes a default policy that can be used as a starting point. Select the default policy by left clicking in the left most column. Select the copy button on very bottom of the screen and save the default policy with a new name, such policy1 or v1.



Tricks and Tips:

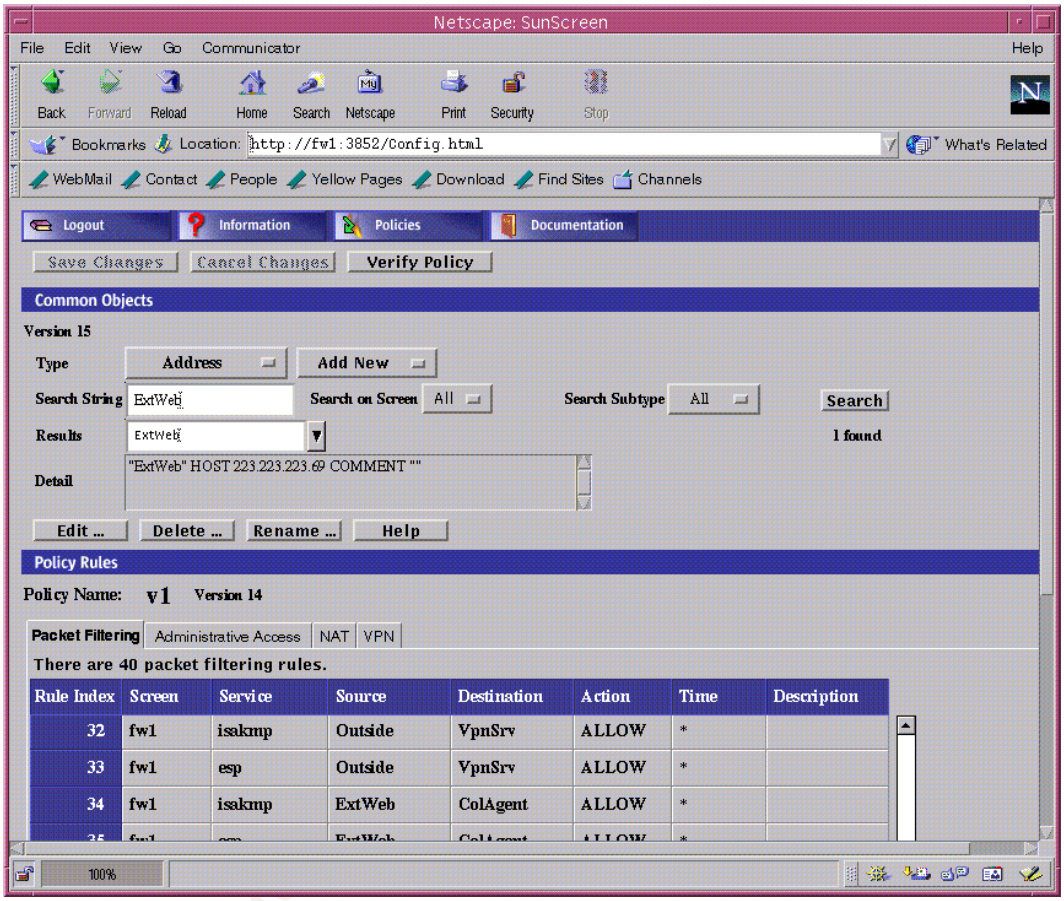
- Copy and save the last known working policy as the basis for a new policy when you will be making "significant" changes. Always leave a copy of a working policy behind your edits.
- Use policy names that indicate major changes in your rule set.
- Make sure you left click in the "currently active" column, otherwise the selected policy will open in read only mode.

Add Common Objects:

Near the top of the policy screen are two boxes, "Addresses and Add New". Under "Addresses" are categories for each kind of object we can create, such as Address, Services, Interface, Screen, etc. Under NEW will be a variety of options depending on the previous selection, typically New, Host, Group, Range, etc.

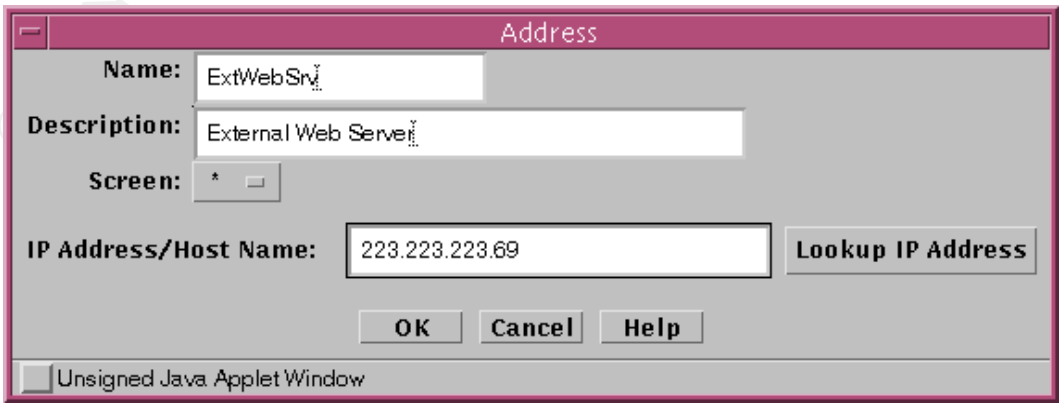
Right below these two boxes is blank search string field. To search any common object, first select the desired category, Address, Service, etc. above and then enter any portion of the object's name without wildcard characters, "*" and hit the "SEARCH" button to the far right. You will see the number of matches found and these will be displayed in the scrollable

RESULTS box. Then select the one you want and double-click the line. Details of the object will appear in the DETAIL box. At this point, you may Edit, Delete or Rename the object as required.



Hosts:

Pull down on the first box for "Addresses" and select "Single" under the NEW menu. This will allows us to create a new object for a single address.

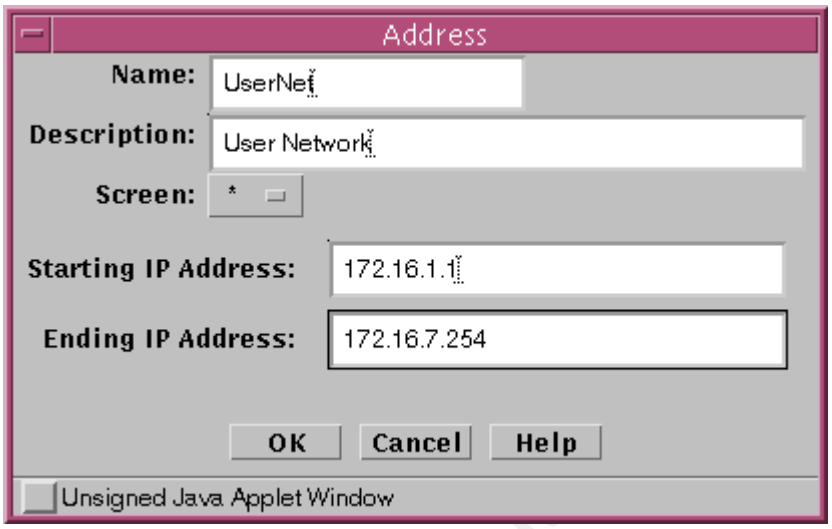


Groups:

Once all of the individual hosts are defined, create any useful groups of hosts. Examples would include a group for all external DNS servers. This way a single rule will cover all members of this group. Group members may be individual hosts and other groups in any combination. You may also define a group to be another single group or address range, optionally excluding one or more common objects. You may want to create a rule for all hosts on a service network except for the VPN server.

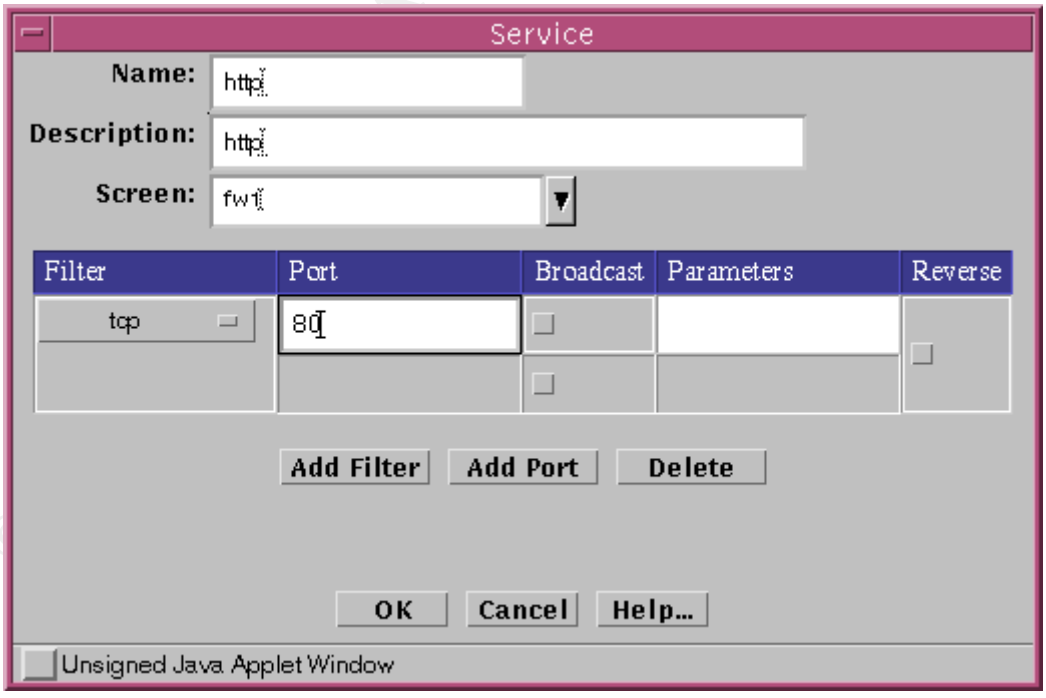
Ranges:

Ranges are useful for creating objects referring to entire networks or portions of a subnet. Typically, this may include all possible addresses expected on a single network interface. These objects will be used latter in setting up rules to prevent address spoofing.



Services:

Many required services are defined by default, e.g. http, smtp, ftp, ssh, etc. Frequently, you will need to create a service for traffic between web application and database servers. For example, you may want to define a service for Oracle using TCP/1521 or TCP/1630. You may add multiple protocols and/or ports into a single service.



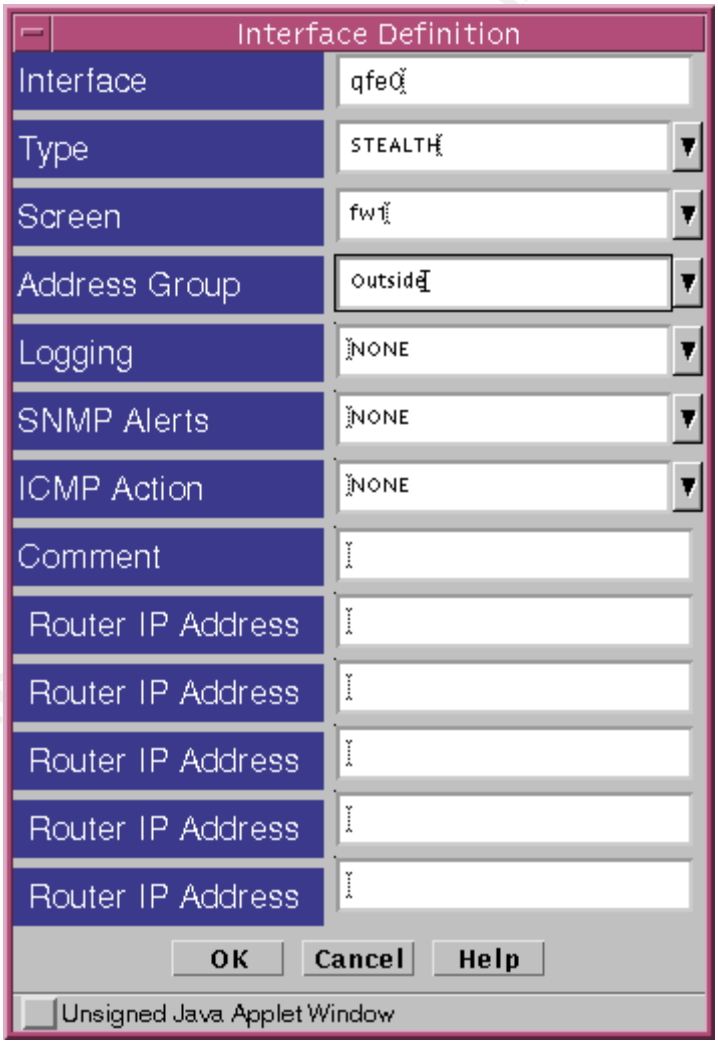
Tricks and Tips:

- Create a table with all of your host names and their IP addresses before starting configuring the firewall.

- Also, set up all of the logical groups of hosts and networks in table format. A logical group might include all external DNS servers or all internal ntp servers, etc.
- If you will use any non-standard services, you will want to create these tables as well.
- Remember there is no concept of subnets or network masks when configuring a stealth SunScreen firewall.

Configure Interfaces:

Interface configuration involves associating IP addresses with a physical interface. In doing so, the firewall permits packets from only the defined IP addresses. This becomes a very powerful anti-spoofing facility.



Tricks and Tips:

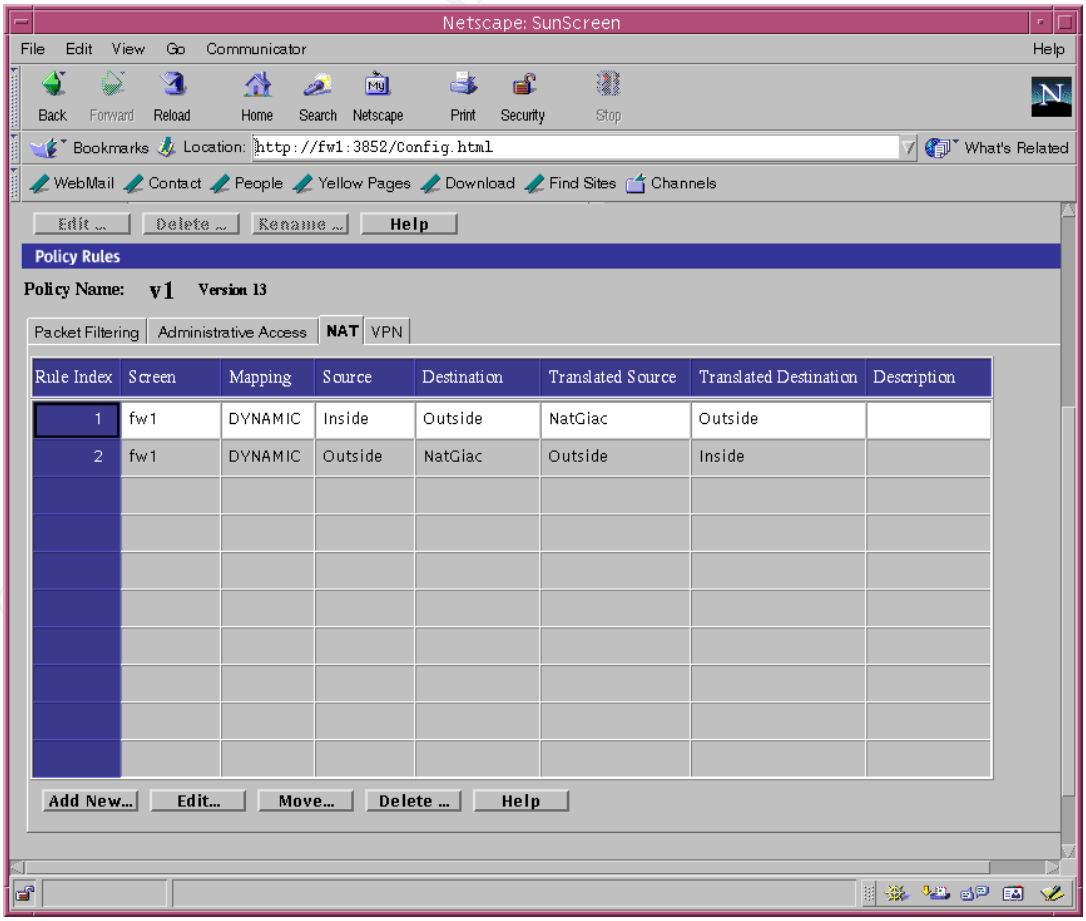
- For the outside interface assign all possible IP addresses, denoted by a "*", excluding the internal address space and any other address

spaces desired. This construct is similar to ingress ACLs on a border router.

- Associate only an address group not an entire network range, e.g. ExtSrv, for all hosts in the external services network(s).
- Associate all and only the private IP address space on the inside interface. This will prevent firewall updates every time a new desktop is added to the internal network, e.g. 172.16.1.1 – 172.16.255.254.
- Associate group names with interfaces connected to external service networks, where security needs are highest and physical changes are minimal.
- When a group object is associated with an interface, you must edit the group definition anytime a physical system is either added or deleted on the interface.
- For the exterior and inside interfaces, add the IP address of the exterior gateway on the first "Router IP Address" line.

Configure NAT:

For small to medium sized networks, a single dynamic NAT address should be sufficient. We need to translate the real private address space IPs to a single translated public IP address.

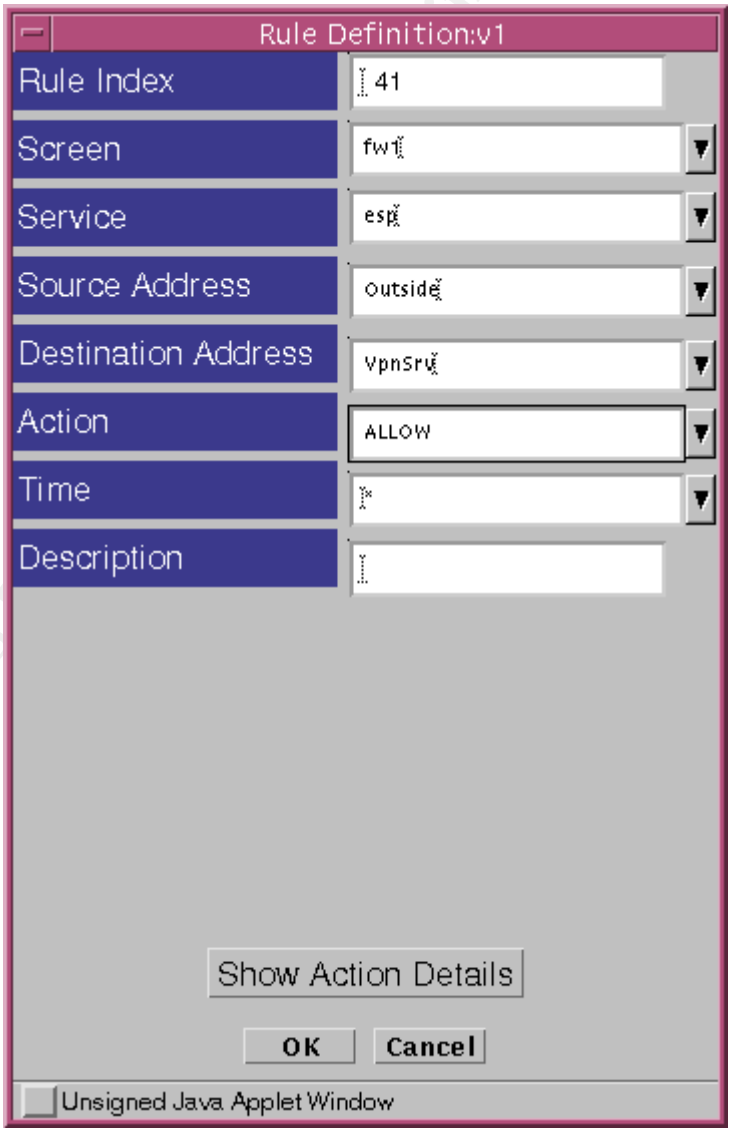


Tricks and Tips:

- Use a prefix of "Nat" for all translated NAT addresses.
- Typical use of static NATs is for permitting IPSec sessions to be established from an internal host to some external device.

Add New Rules:

Adding rules follows the same basic steps as creating a new rule. Bring up the policy to which the rule will be added and select NEW at the bottom of the screen. While completing the rule fields you may select any rule number desired to insert the new rule or add it at the end of the policy. Pay attention to your rule order policy. This screen is used to create all of the basic ACLs used in a SunScreen firewall. In a multiscreen or firewall design, the "Screen" field allows you to specify to which firewall(s) the rule applies. Service may be a predefined service, smtp, isakmp, ssh, etc. or any custom defined service.



Rule Order:

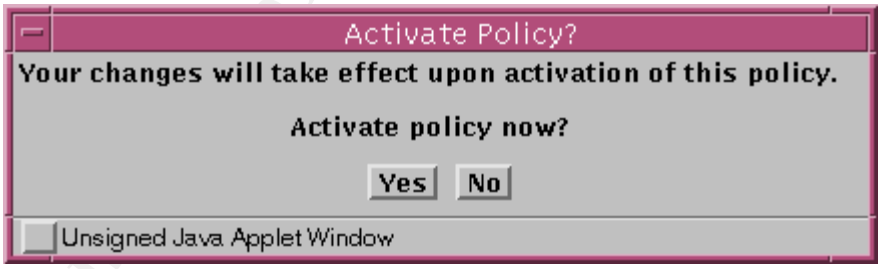
Packets will be passed or dropped on a first match rule opposed to a best match rule. Consequently, rule order for SunScreen firewalls need to follow a more to less specific sequence. Otherwise, most frequently used rules such as http(s) should precede less urgent protocols such as smtp. There is no Quality-of-Service (QoS) feature in the SunScreen product.

Trick and Tips:

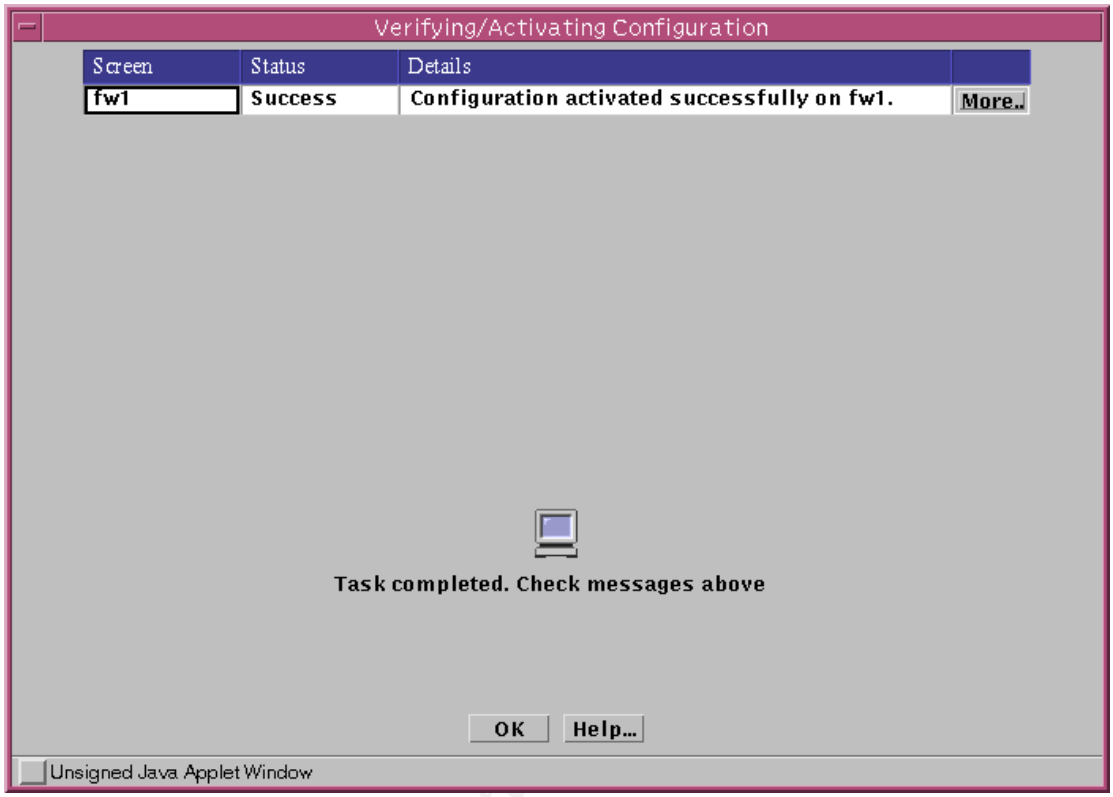
- Grouping rules by protocol first, specificity next and then by host will usually meet security policy requirements and be easier to read and maintain.
- Typically, you will want a "deny all" rule type at the end to log all packets that falls out of the bottom of the rule set. Just preceding the "deny all" rule, include a "deny * icmp" rule to avoid filling the log file with icmp unreachable messages, unless you really care about these. Presumably, the border router will handle these ICMP messages anyway.

Save and Activate Rules:

The last step is to save and activate the new policy or rule set. To do hit the "save and activate" key near the top of the screen. A second window will open requesting confirmation to activate the policy, say YES. Once a "success" message is displayed, the firewall is running the new policy. You may also save a new policy without activation or activate any existing rule set.



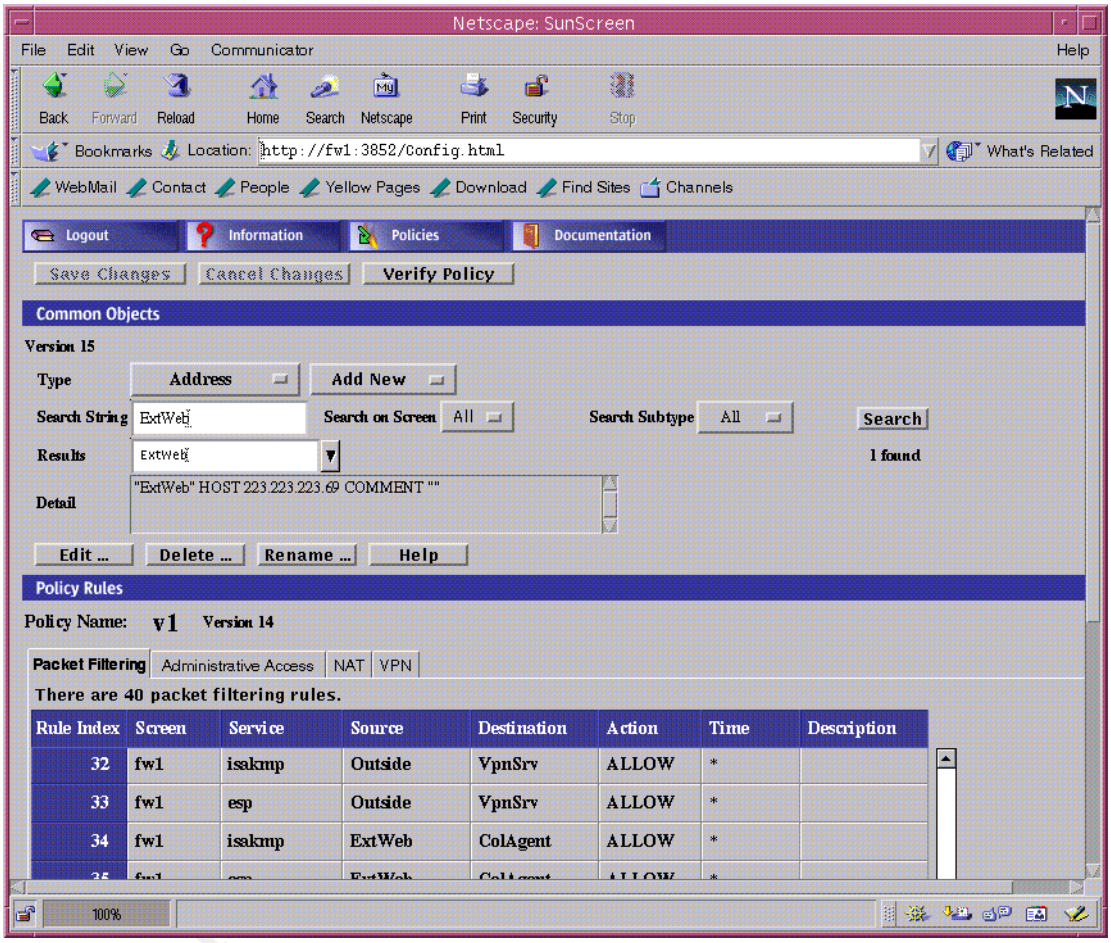
The following screen will verify successful compilation and activation of the policy. If compilation fails it is denoted with an "error" message in the "Status" field, and the "More..." button will provide useful debugging information.



The above screen shows the successful compilation and activation of the policy. In a multi-firewall design with Central Management, there will be a line for each screen or firewall.

Edit Common objects:

Common objects may require occasional edits to reflect changes in the physical network and/or security policy. To do this first perform a search as described earlier and select the EDIT button. A new window will appear very similar to the window used to create new common objects allowing whatever changes needed.

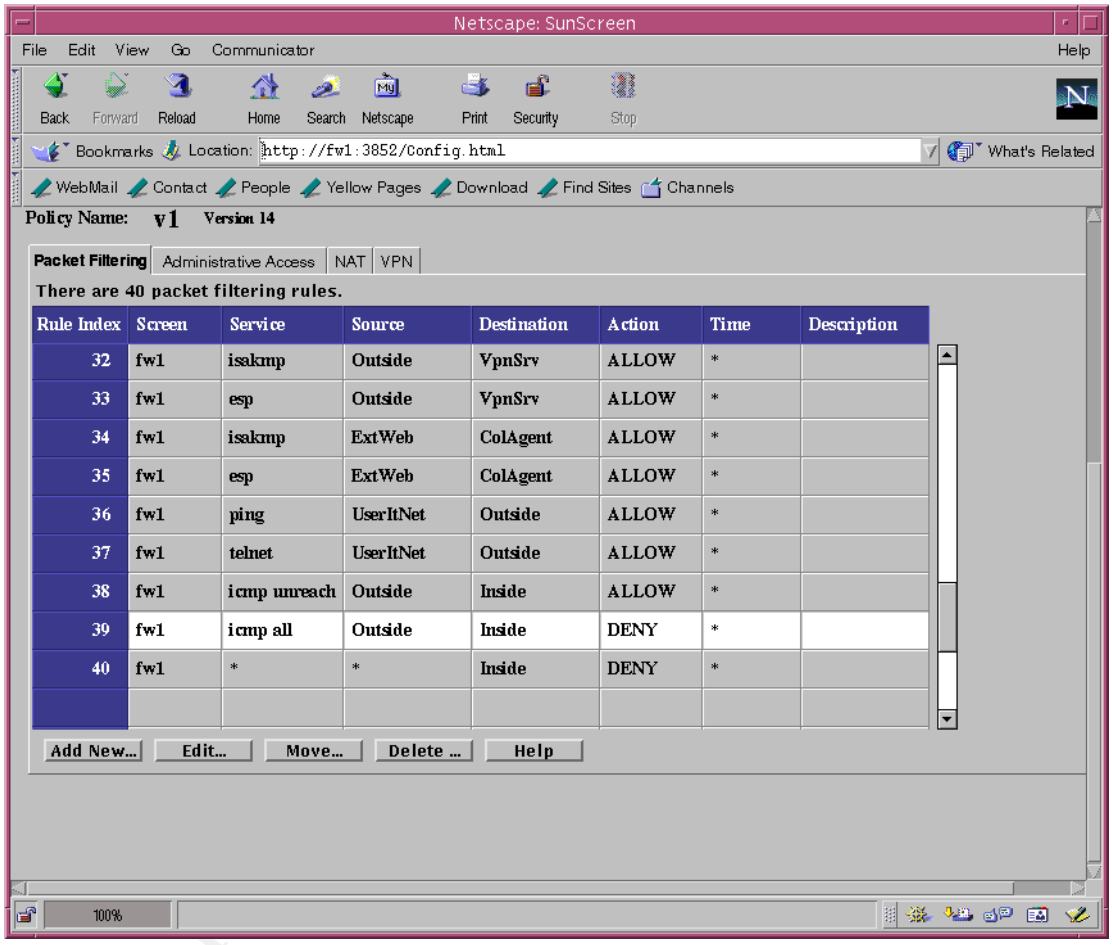


Tricks and Tips:

- In performing a search, the notion of a wild card is implicit. If you want to search for all objects starting with Dns, just enter “Dns” without the typical “*” character. Under the search button, you will see the number of matches found. These will be listed in a pull-down menu. Then fine-tune your search by selecting any of these options.
- All policy sets utilize the same named object database. Be careful in modifying objects, which may have a different definition in an older policy that might be needed again. The rule set may not activate at all or not according to your security policy.
- Remember to “Save and Activate” any changes you make.
- You cannot rename an object while in the edit window, all other fields may be changed. Use the RENAME button instead.

Edit Rules:

Any rule maybe edited while in the policy screen. To do so, select the individual rule number and hit "Edit" at the bottom of the screen. An edit window appears that is very similar to the screen used to add new rules. You may change rule order by selecting the "Move" button at the bottom of the screen.



Rule Index	Screen	Service	Source	Destination	Action	Time	Description
32	fw1	isakmp	Outside	VpnSrv	ALLOW	*	
33	fw1	esp	Outside	VpnSrv	ALLOW	*	
34	fw1	isakmp	ExtWeb	ColAgent	ALLOW	*	
35	fw1	esp	ExtWeb	ColAgent	ALLOW	*	
36	fw1	ping	UserItNet	Outside	ALLOW	*	
37	fw1	telnet	UserItNet	Outside	ALLOW	*	
38	fw1	icmp unreach	Outside	Inside	ALLOW	*	
39	fw1	icmp all	Outside	Inside	DENY	*	
40	fw1	*	*	Inside	DENY	*	

Delete Common objects:

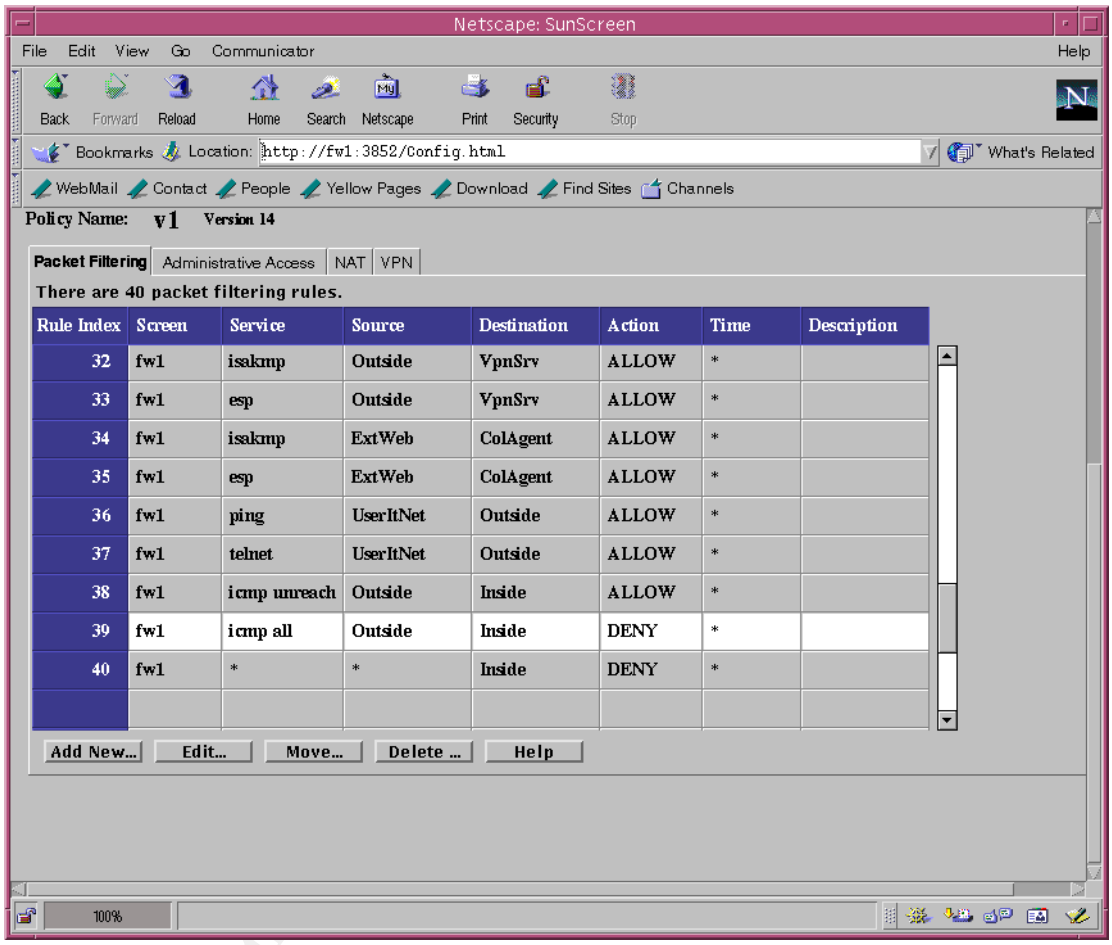
First performing and then selecting the required object from the search result list may delete any named object. Just hit the DELETE key below the selection window as shown above.

Tricks and Tips:

- Make sure you delete any rules referencing a named object before deleting the named object itself. Otherwise, policy compilation will fail.

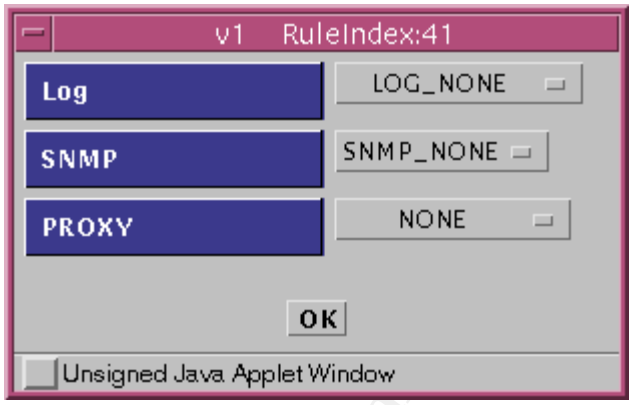
Delete Rules:

To delete a specific rule select the policy and then scroll down to the desired rule, select the row and then hit the “Delete” button at the bottom of the screen. All subsequent rules are renumbered automatically.



Adding Logging:

Logging can be added to any rule. To do so, just hit the LOG button at the end of the rule creation window. Logging may be done at the summary or detail level. Summary level is usually sufficient. I was unable to capture an example of the detailed logging window.

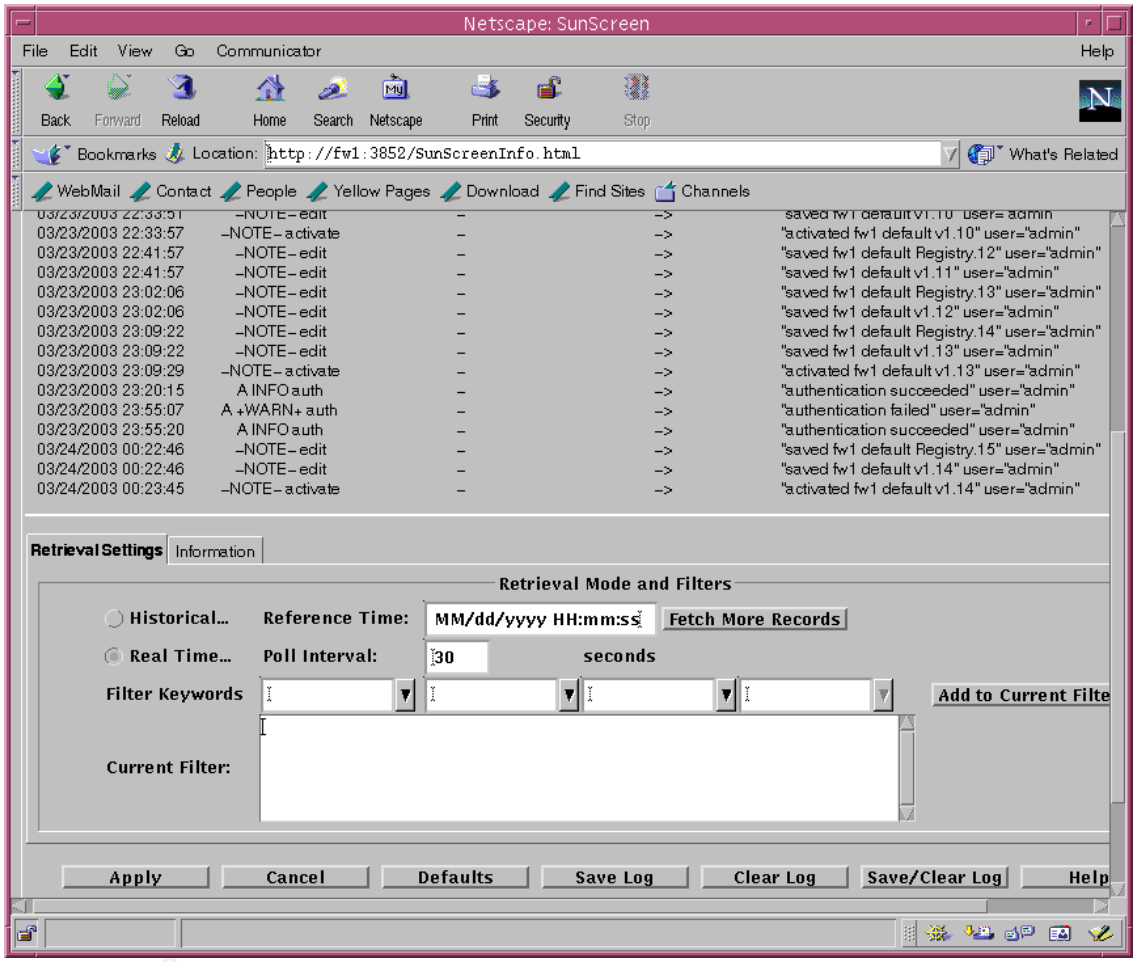


Tricks and Tips:

- The logging file will rollover after it is full, preventing the firewall from performance degradation in heavy traffic situations.
- If you need to see logs for packets forwarded, add an allow rule and turn on logging for that rule.
- Size of the log file is set at time of product installation. However, it may be changed by a command line instruction.

Monitoring the Log Files:

To view the logs go back to the main window and select the “Information” tab with the red question mark. By default, the most current snapshot of the log file is displayed. In the lower portion of the window, you may select any date/time block you want to view the log file and apply a variety of filters to select particular addresses or protocols.



The above log window was made during the installation process.

ASSIGNMENT III: SECURITY ARCHITECTURE AUDIT:

Introduction:

Prior to bringing any host or security device into production, we will perform extensive testing and analysis of the security policy and its implementation. Even with extensive planning, small typing errors or other oversights can produce miss-configured systems and networks. In some cases, errors may manifest themselves in obvious malfunctions. Other times all may appear correct until harm has already happened. Security policy verification can be accomplished by following a well-defined process where each aspect of the policy is tested and results analyzed for correct responses, followed by exception correction and retesting as required.

Scope of Assignment:

This assignment is to design a test, which will validate the exterior firewall as described in Assignments I & II. Under a real-life situation, all elements of the security system, including staff and the security policy and the devices they protect would be tested.

Primary emphasis of our tests will be to verify that only traffic explicitly permitted by the security policy is seen across firewall interfaces. Network descriptions, tables and drawings in Sections 2.9.x details what is permitted. What we will test is the firewall security policy described in Assignment II. Since GIAC Enterprises is fully operational and all business requirements have been certified, we will not test for normal or expected network traffic.

Planning:

Success of this audit will largely be how well we plan the work and work the plan. Carelessness here can mask the very problems we are trying to identify. The planning and audit steps include:

- Identify the Audit Team
- Develop and maintain accurate notes and results
- Meet regularly with the Audit Team as required
- Clearly identify the objectives and how the objectives will be measured
- Work closely with the GIAC security specialist
- Select a day and time for the test
- Research for known vulnerabilities in the products being tested
- Design a series of tests and expected results
- Assemble and configure necessary hardware/software
- Notify all required individuals of the testing schedule
- Conduct the tests
- Analysis results
- Draw conclusions
- Prepare a Findings Document, including recommendations for any problems noted

Time and Cost Estimates:

12^2 Computer Security is proposing to conduct the requested validation test according the estimates shown in the following table.

Description	Quantity	Total Cost US Dol
Planning	8	$1,200
Research and setup	4	600
Conducting the test	12	1,800
Data analyses	2	300
Finding preparation	1	150
Subtotal		$4,050
Contingency	20%	$ 810
Total Costs		$4,860

Not included in the above total are direct and indirect costs to GIAC Enterprises, these are estimated in the table below.

Description	Quantity	Total Cost US Dol
12^2 Computer Security		$4,860
Staff time for tests	36 hrs x $40	1,440
Staff time for prep.	12 hrs x $40	480
Management time	6 hrs x 60	360
Total Test Cost		$7,140

We have estimated costs for the technical staff time required for special system backups and other preparatory tasks, presence of the GIAC IT staff during the tests and management's time for planning and post review activities. All required hardware and testing software is already owned by 12^2 Computer Security or GIAC Enterprises.

Scheduling:

In order to minimize impact to GIAC's 24 x 7 worldwide operating schedule, we will conduct the actual testing starting at 6:00 P.M. local time on a Saturday evening. Even through this will hit normal business hours for some customers, GIAC feels this will cause the minimum business impact. Starting two weeks ahead of the scheduled test a notice will be posted on the GIAC web server advising customers and partners of a possible network outage. E-mail will also be sent to the GIAC sales team, GIAC's ISP, suppliers, partners and major customers advising them of the same.

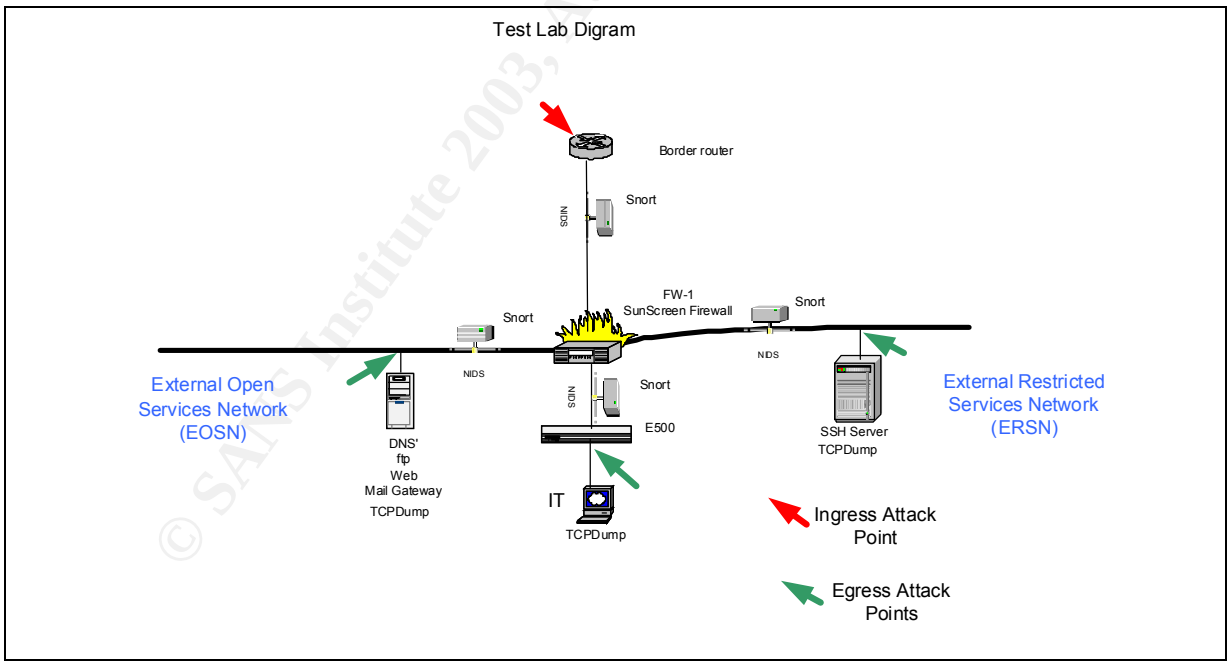
Network testing can sometimes be very noising and create significant loads. To help reduce this affect we will setup a test lab to perform portions of the desired tests. This will reduce impact to the business units, and permit use some of the more dangerous test options if so desired. The test lab will duplicate the firewall and its rule set exactly with dummy hosts on each interface to simulate some of

the real hosts. Since, the exterior firewall is a SunScreen running in stealth mode we must have a real host with at least one IP attached to each interface. To reduce the number of real hosts required we would assign multiple IP addresses to the three internal dummy hosts with a script like the one below. The port numbers that the real hosts listen on will be opened on these dummy hosts to simulate expected connections or other host responses. This technique would not be acceptable if our objective was to test end-to-end security, since each address would be listening on all opened ports. This detail makes no real difference, when the test objective is only the firewall configuration.

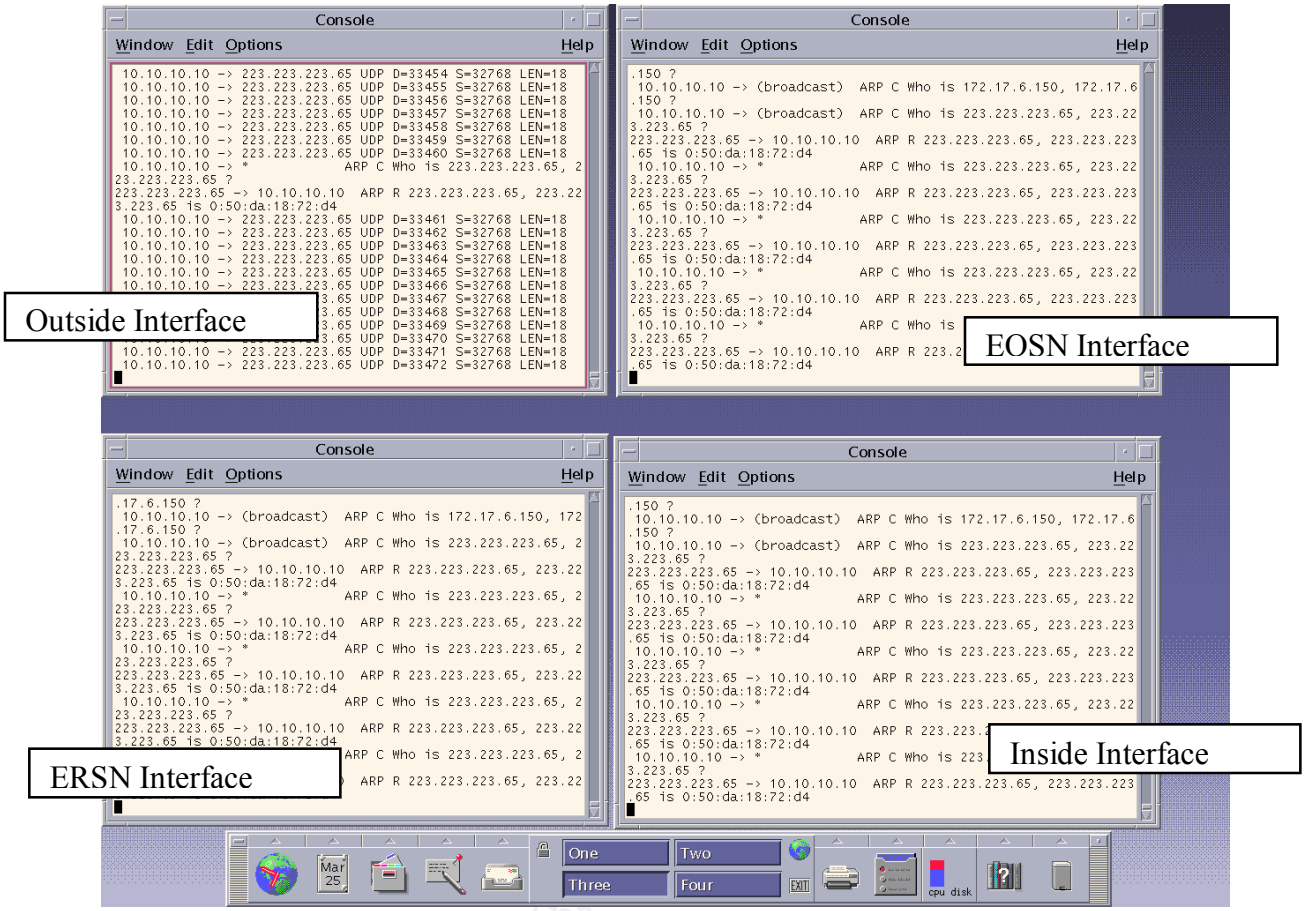
```
# Configure multiple IPs on a single interface
ifconfig eth0:65 223.223.223.65
ifconfig eth0:66 223.223.223.66
ifconfig eth0:70 223.223.223.70
...clip...
```

Since we are not testing ACLs or other security features provided by the border router, we will reconfigure the border router in the test lab to pass all traffic. This will allow us to place devices outside of the router without concern what the router is doing to the test results.

A schematic diagram of the test lab is shown below.



In addition to the monitoring of the NIDS systems and the firewall logs, we will run tcpdump on each of three hosts internal to the firewall. One of the most effective means to test with a SunScreen firewall is to run snoop on each interface as a different window on the firewall's monitor. This allows us to observe in real time if the rules are working correctly.



Finial testing will be done on GIAC's live network with an emphasis on how specific hosts respond to test traffic and not what is blocked at the firewall. We must be mindful that while we testing with a live system, we could observe the traces of real scanning and attack patterns mixed into our own traffic.

The lab setup for this assignment did not include SunScreen Central Management because I did not have sufficient hardware and it does not add directly to the quality of the security policy.

Preparation for Testing:

Part of the verification test contract includes a signed clause from GIAC Enterprises that 12^2 Computer Security is authorized to perform the specified tests and is not responsible for any damage or loss incurred by GIAC Enterprises. It also clearly states that 12^2 Computer Security will not disclose any information obtained from the planning or testing process to any other party.

GIAC's IT staff is responsible for completion of full backups for every active host or device on the network during the test period. They must also schedule sufficient resources to recover any system(s) that might be altered during the test process. 12^2 Computer Security will make every attempt to provide a list of

impacted systems prior to and after testing is completed. The IT staff will provide assurance that no devices will shunt any testing traffic, e.g. NIDS sending resets back to the testing devices.

Just prior to starting the test period, we will confirm with senior GIAC management and GIAC's ISP of the pending test. The GIAC security specialist, networking and operating system managers must be present during the entire testing process. This will help advert reacting to alarm conditions created by our testing procedures.

Prescribed Tests:

If we were performing a full perimeter review, we might run the following individual tests to validate the security policy.

- Update and patch history documentation review
- Physical access
- TCP and UDP port scans, NMAP
- Ping tests
- Passive OS fingerprinting
- Vulnerability scans using Nessus
- Bad TCP flag combinations
- Spoofed addresses, external and internal
- Denial of service attack
- Verification of correct normal operations

However, this assignment is restricted to reviewing the security policy of the external firewall only. Consequently, we will select a much more tailored list of testing tools. Besides using GIAC's NIDS systems, we will use tcpdump, snort and perhaps ethereal, on Linux systems attached to each firewall interface.

- TCP and UDP port scans, NMAP
- Ping tests
- Bad TCP flag combinations
- Spoofed addresses, external and internal
- smtp
- nslookup, old but suitable for this job
- web browser
- telnet
- ntpdate

Since this is already a production network, GIAC Enterprises will self verify normal business traffic flows and that all systems work correctly from a business perspective.

Test Locations:

To fully test ingress and egress policies, we will test from four different network locations:

- Outside to internal network (inside) and both external service networks (EOSN and ERSN).
- Internal network (inside) to the outside and both external service networks.
- EOSN to the outside; ERSN and the internal network.
- ERSN to the outside; EOSN and the internal network

Testing Strategies:

Testing strategy for this exercise differs from penetration testing in that we are starting with a current network map, the high-level security policy and detailed device level security policies. Even through this is not a penetration test, we will annotate why a cracker might look for the vulnerabilities we are testing for.

If this were a penetration test, we would start by getting as much DNS domain information as possible; build a network map; identify operating systems and running services; look for vulnerabilities and then try to exploit those vulnerabilities.

Since the objective is to test the firewall, not the entire perimeter, we will attach our testing and monitoring devices inside of the border router and the other three "subnets" created by the external firewall (FW-1).

Testing Tools:

In this section, we will describe the tools and techniques we will use in the verification tests.

Update and Patch Documentation Review:

We will review the update and patch history logs kept by GIAC to ensure that local/best practices are being followed. Network devices and hosts will be crosschecked against the logs for verification purposes.

Physical Security:

GIAC's local security policy states several physical security requirements. We will verify those pertaining to the router and firewall has been met.

DNS Domain Testing:

The only devices that should be addressable by the Internet include:

- Exterior DNS servers
- Squid proxy server

- Emu-mail server
- Secure shell server
- VPN server
- ftp drop box.

nslookup:

nslookup will be used to test for DNS testing. We will test if we can learn about any other devices in the external or internal networks, including attempts for a zone transfer.

hping2:

Hping2 will be used to test for responses from ICMP and other pings. If GIAC's hosts respond to hping, we may be able to fill-in portions of a network map.

nmap:

nmap is one of the most powerful general-purpose network probing tools available, http://www.namp.org/. The nmap man pages explain the wide range of options. Our primary interest will be to use nmap to test for open ports. Since the SunScreen firewall is operating in stealth mode with no IP addresses or ports, we do not expect to find any open ports on the firewall itself. Any responding ports will be those of the servers behind the firewall.

We will also use nmap to test for non-standard TCP Flag combinations.

telnet:

Telnet will be useful to see what responses we can get from the web and mail servers on any listening ports. We will also try to telnet to other IP addresses and port 23, just to verify the telnet service is shut down.

ntpdate:

ntpdate will be used to verify the ntp suite of rules. We are looking to see if any GIAC device other than the two internal ntp servers can obtain a response from any other ntp server across the firewall.

syslog:

In the GIAC, network all devices with the exception of the Sunscreen firewalls, logs to a common syslog facility. The firewalls log to their own independent central console system. We will check these central logs for evidence of our testing procedures.

tcpdump:

tcpdump is the standard packet capture tool. We will run it on each attacking and target host.

IDS Systems:

The GIAC Snort NIDS will also be checked for evidence of our testing.

snoop:

Solaris snoop will be used on the SunScreen. By setting up four snoop windows, we can monitor traffic on every interface at one time. This will be the most obvious way to see if any unauthorized traffic makes it through the firewall.

Tools not used:

The requirement of this assignment is to test the external firewall's security policy. We might normally use other tools such as Phonesweep, http://www.sandstorm.net/, whisker, http://www.wiretrip.net/ or screamingcobra, http://www.dachb0den.com/projects/screamingcobra.html. Whisker and screamingcobra are both cgi script-testing tools intended for web server testing. Since the GIAC network prohibits modems on networked devices, there is little reason to include these tools, because they will not test the firewall's rules in this assignment.

dsniff from http://naughty.monkey.org/~dugsong/dsniff/ might be used if were do a real penetration test, but this tool is not required in this effort.

In a real test, we might use tools like John The Ripper v 1.6, to try guessing UNIX passwords or L0phtcrack v 3.0 for Window passwords. We are not expecting to obtain password files in this exercise. We note that John the Ripper v 1.6 can crack both UNIX and Window passwords.

Nessus, http://www.nessus.org/ uses nmap for testing of open ports. However, Nessus will try to identify the type of service running behind a listening port and attempt to exploit certain vulnerabilities. Some tests in the Nessus suite are potentially dangerous to run, e.g. might crash a host, and are one of the reasons to perform extra backups prior to testing with Nessus.

Test Results:

One of the first observations was that arp requests were seen on every interface for any traffic generated on the subnet that included the firewall. This is of some concern, in that internal or private IP addresses can be seen on the outside interface(s). This information could be used to help map the network and fingerprint that a SunScreen firewall in stealth mode existed on the subnet. This would still require someone with snooping access on the local segment.

In the following sections we will summarize the firewall policy being tested, show the command used to test the policy, test results and if the results match

expectations. Please note, that since GIAC Enterprises verifies ability to perform all normal business operations, we will not test policies for "allowed" traffic, primarily those policies that "deny" access.

DNS Domain Testing:

Source	Permission	Destination
External hosts	access	GIAC's external DNS servers
External hosts	deny	Zone transfers
External hosts	deny	Internal DNS servers
Internal hosts	deny	External DNS servers

First, we attempt a zone transfer from the external DNS servers.

```
ls -d giac.com
ls: connect: No error
***Can't list domain giac.com: Unspecified error
```

Since tcp/53 is blocked, we get no useful information. If we perform nslookups for each server by name we get the expected IP address, but only those hosts listed on the external DNS servers.

We know the internal DNS server's hostnames and IP addresses and test access to these from the outside using nslookup.

```
nslookup server dns3.giac.com
***Can't find address for server dns3.giac.com: Non-existent domain
```

Even if we use the correct IP address for dns3.giac.com we get a DNS request timeout error.

Using a Linux host within the GIAC internal network, we issue:

```
nslookup server outside.dns.server
connection timed out; no servers could be reached
```

GIAC hosts cannot perform resolve host names with external DNS servers. We have tested all DNS denied cases and have found everything works as expected.

Ping and Hping2:

Source	Permission	Destination
External hosts	deny	Any GIAC host
Internal IT subnet hosts	allow	External hosts
Internal hosts	deny	External hosts

We will test a range of IP addresses, including hosts on the external service networks and the internal GIAC network. The first ping was from each test point using the broadcast address.

```
ping –b 223.223.223.255
```

The firewall blocked the pings across all interfaces as expected. This test was mainly used to confirm full connectivity and operation of the sniffers.

ICMP ping from the outside:

```
ping 223.223.223.34          # Exterior interface of E-500
-  - 233.233.233.34 ping statistics - - -
24 packets transmitted, 0 received, 100% lost, time 22998ms

ping 223.223.223.65          # External DNS' server
-  - 233.233.233.65 ping statistics - - -
26 packets transmitted, 0 received, 100% lost, time 23365ms

ping 223.223.223.40          # Secure shell server
-  - 233.233.233.40 ping statistics - - -
15 packets transmitted, 0 received, 100% lost, time 12489ms

ping 223.223.223.50          # GIAC NAT address
-  - 233.233.233.50 ping statistics - - -
20 packets transmitted, 0 received, 100% lost, time 21734ms
```

ICMP ping from an internal host on the IT subnet:

```
ping 1.2.3.4                 # A known external address

Pinging xyz.com [1.2.3.4] with 32 bytes of data:

Reply from 1.2.3.4: bytes=32 time=241ms TTL=118
Reply from 1.2.3.4: bytes=32 time=219ms TTL=118
Reply from 1.2.3.4: bytes=32 time=206ms TTL=118
Reply from 1.2.3.4: bytes=32 time=213ms TTL=118

Ping statistics for 1.2.3.4:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

This ping worked because the IT segment is the only GIAC network permitted to ping outside of the firewall.

ICMP ping from an internal host not on the IT subnet:

```
ping 1.2.3.4                          # A known external address
-  - 1.2.3.4 ping statistics - - -
30 packets transmitted, 0 received, 100% lost, time 27649ms
```

Next we will use hping2 to firewalk the external SunScreen firewall. If we receive ICMP type 11 error message in response to an ICMP ping with a time to live (TTL) of one, we will know a host behind the firewall is listening on that port. If we do not get a type 11 error message, we know either the port is blocked or ICMP error messages are not allowed out of the firewall. The SunScreen firewall will not respond to the packet.

Our first test is the web based Emu-mail server, which listens on port 443.

```
hping –S –c 1 –p443 –t 1 223.223.223.71

[root@loki root]# hping -S -c 1 -p 443 -t 1 223.223.223.71
HPING 223.223.223.71 (eth1 223.223.223.71): S set, 40 headers + 0
data bytes
ICMP Host Unreachable from ip=10.10.10.10 get hostname...
--- 223.223.223.71 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss round-trip
min/avg/max = 0.0/0.0/0.0 ms
```

As expected we did not get an ICMP type 11 error message, because we block all outbound ICMP traffic, except pings and returned ICMP unreachable. If we repeat the test to a blocked port on the external DNS server, we get the same thing, but this time because the port is blocked at the firewall.

```
hping –S –c 1 –p1 –t 1 223.223.223.65

root@loki root]# hping -S -c 1 -p 1 -t 1 223.223.223.65
HPING 223.223.223.65 (eth1 223.223.223.65): S set, 40 headers + 0
data bytes
--- 223.223.223.65 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

All ping tests worked as expected, only the IT subnet can ping out and since all pings fail from the outside, common ping network mapping and reverse mapping techniques fail.

ICMP Unreachable Error:

Crackers may use traceroute to help create network maps. The firewall policy prohibits ICMP unreachable errors as an egress rule. Therefore,

from the outside we will run traceroute against both a host in the external services network and the internal network. We do not expect any useful information to be returned.

```
traceroute 223.223.223.65                    # Exterior DNS server

 [root@loki root]# traceroute 223.223.223.65
traceroute to 223.223.223.65 (223.223.223.65), 30 hops max, 38-byte
packets
 1  * * *
 2  * * *
 3  * * *
 …snip…
28  * * *
29  * * *
30  * * *
```

As expected the traceroute timed out.

Next we attempt a traceroute to an external host from an internal, non-IT, host. We will repeat this test from each internal network segment.

```
traceroute 1.2.3.4                    # Desktop not in the IT group

[root@inside root]#  traceroute 10.10.10.10
traceroute to 10.10.10.10 (10.10.10.10), 30 hops max, 38 byte packets
 1  * * *
 2  * * *
 3   * * *
…snip…
28  * * *
29  * * *
30  * * *
```

These tests results show that ICMP unreachable error messages are not passed outside of the firewall and non-IT hosts cannot successfully run traceroute commands to external addresses.

Telnet Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host
Internal IT subnet hosts	allow	External hosts
Internal hosts	deny	Any host

In this test, we will use the standard telnet command from two access points.

Userhost# telnet 1.2.3.4 # From an internal GIAC host

Trying 1.2.3.4...
telnet: connect to address 1.2.3.4: No route to host

ExternalHost# telnet 172.16.1.10 # From an external host to the external DNS server.
[root@loki root]# telnet 172.16.1.10
Trying 172.16.1.10...
telnet: connect to address 172.16.1.10: No route to host

This test is repeated for each host on the external networks and the NAT address. In all cases access is denied as expected.

smtp Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host except mail gateway
Internal hosts	deny	Any host except mail server

For these tests, we will use nmap to test establishing a connection from both the outside and inside networks. Our first test from an internal host to an external mail relay failed as expected, since smtp traffic is only permitted from the mail server to the outside.

nmap -p25 –sT –P0 1.2.3.4 # ISP mail rely host

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (1.2.3.4):
Port State Service
25/tcp filtered smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 47 seconds

In nearly all of our nmap tests we will omit the leading ping test by using "-P0", since these pings should fail and would this prevent the rest of the test from succeeding. The "-sT" option establishes a full TCP connection, which is needed. Our destination is port 25 on a mail relay server, 1.2.3.4 at our ISP.

The next test is conducted from outside the firewall to verify that only the external mail gateway will accept inbound smtp traffic.

```
nmap –p25 –sT –P0 223.223.223.65-71          # Each host on both
nmap –p25 –sT –P0 223.223.223.40-41            external service
                                               networks

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (223.223.223.70):
Port       State       Service
25/tcp     filtered    smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 47
seconds
```

This test is repeated for each GIAC host in the external services networks and the dynamic NAT address on the firewall. In all cases, except for the mail gateway, the test failed to connect with an open port.

ntp Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host
Internal hosts	deny	Any host except internal ntp servers

In this test, we are verifying that only the two internal ntp servers can connect with an external ntp clock server and external hosts cannot connect with any internal host on port 123. This will not test what the border router will do when receiving an ntp request packet on its outside interface.

From the external service networks and the inside, we will try to use ntpdate to attempt synchronizing with an external ntp clock. We make sure ntpd is not running on the test platform first, or else we will see that the ntp socket is in use. The –q option does the query without actually changing the local host's clock.

```
ntpdate –q 140.221.9.20              # From an internal host

[root@ersn root]# ntpdate -q 140.221.9.20
server 140.221.9.20, stratum 0, offset 0.000000, delay 0.00000
25 Mar 20:03:06 ntpdate[1363]: no server suitable for synchronization
found
```

Next we perform the same test from an external host and get:

```
ntpdate –q 223.223.223.65            # From an external host
```

```
[root@loki root]# ntpdate -q 223.223.223.65
server 223.223.223.65, stratum 0, offset 0.000000, delay 0.00000
25 Mar 20:08:09 ntpdate[11282]: no server suitable for synchronization
found
```

All ntp test results are as expected since TCP/123 is permitted outbound only from the two internal ntp servers.

http Rules:

Source	Permission	Destination
External hosts (80)	deny	Any GIAC host except squid server
External hosts (443)	deny	Any GIAC host except EmuMail
Internal hosts	deny	Any host except internal proxy server

We will use nmap to verify that only the squid proxy server on tcp/80, and Emu-mail server on tcp/443 will respond from the outside. Again we do not want a preceding ping attempt and want to test for completion of a TCP session.

```
nmap –P0 -sT –p80 223.223.223.65-71          # From an external
                                               host
nmap –P0 -sT –p80 223.223.223.40-41

[root@loki root]# nmap -P0 -sT -p80 223.223.223.67

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (223.223.223.67):
Port       State      Service
80/tcp     filtered   http

Nmap run completed -- 1 IP address (1 host up) scanned in 13
seconds

nmap –P0 –sT –p443 223.223.223.40-41    # From an external host
nmap –P0 –sT –p443 223.223.223.65-71

[root@loki root]# nmap -P0 -sT -p443 223.223.223.40

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (223.223.223.40):
Port       State      Service
443/tcp    filtered   https

Nmap run completed -- 1 IP address (1 host up) scanned in 46
seconds
```

This test is repeated for each host on the external service networks and the NAT address. Next, we test if a user can connect directly to an external web site by reconfiguring their web browser setting to bypass the internal proxy server.

All http(s) rules work as expected.

Secure Shell Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host except ssh server
Internal hosts	allow	Any external ssh host
Internal hosts	deny	Any external ssh host

We will use a system with ssh installed to test each server in the external services networks and externally from the inside.

OutsideHost# ssh user1@223.223.223.65 #From outside to DNS

[root@loki root]# ssh it1@223.223.223.65
ssh: connect to address 223.223.223.65 port 22: Connection timed out

We repeat this test for each server in the external services network. Next, we use a user's host and an internal GIAC server to try to connect with a known external ssh server.

InsideHost# ssh user1@1.2.3.4 #From inside to outside

[root@ersn root]# ssh it1@10.10.10.10
ssh: connect to address 10.10.10.10 port 22: Connection timed out

tcpdump listing:

20:22:29.792720 223.223.223.40.32775 > 10.10.10.10.ssh: S 3822801289:38228012890) win 5840 <mss 1460,sackOK,timestamp 42828707 0,nop,wscale 0> (DF)
20:22:53.792721 223.223.223.40.32775 > 10.10.10.10.ssh: S 3822801289:38228012890) win 5840 <mss 1460,sackOK,timestamp 42840995 0,nop,wscale 0> (DF)

The tcpdump listing shows no return traffic. All secure shell rules work as expected.

VPN Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host except VPN server
Internal hosts	deny	Any host

The first part of establishing an IPSec VPN session requires UDP/500. If we do not get a response from UDP/500 on the target system, IPSec will fail. We will test this with an hping command. The -2 option is for UDP packets.

hping -2 –p500 223.223.223.65

Results were as expected, blocked at the firewall.

ftp Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host except ftp server
Internal IT subnet	allow	Any external ftp server
Internal hosts	deny	Any GIAC host except ftp server

To test this we will setup a test system outside of the firewall to determine if any system other than the ftp drop box, will respond. Likewise, we will test a user's host to determine if they can connect to an external ftp server.

```
Oustidehost# ftp 223.223.223.69          # From outside to web
                                           server

[root@loki root]# ftp 223.223.223.65
ftp: connect: Connection timed out
ftp>bye

tcpdump listing:
20:30:36.357011 10.10.10.10.32794 > 223.223.223.65.ftp: S
46032604:46032604(0) win 5840 <mss 1460,sackOK,timestamp
11604389 0,nop,wscale 0> (DF)

The tcpdump listing shows no return traffic.

InsideHost# ftp 1.2.3.4                  # From inside to outside

ftp: connect: Connection timed out
ftp>bye
```

tcpdump listing:
20:30:36.357011 10.10.10.10.32794 > 1.2.3.4.ftp: S
7035674:17035674 (0) win 5840 <mss 1460,sackOK,timestamp
13645361 0,nop,wscale 0> (DF)

The tcpdump listing shows no return traffic.

This test is repeated against each server in the external service networks and non-IT internal subnets with the same results.

The ftp rules work as expected.

syslog Rules:

Source	Permission	Destination
External hosts	deny	Any GIAC host

To test this rule we will use hping2 from the outside to verify we can not connect with the syslog server.

hping -sU –P0 –p514 172.17.1.243

root@loki root]# nmap -sU -P0 -p514 172.17.1.243

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (172.17.1.243):
Port State Service
514/udp open syslog

Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds

The test is blocked at the firewall as expected.

IP Address Spoofing:

We have configured each firewall interface with both a range of expected IP addresses (XxxNet) or a group (XxxGrp) of allowed hosts. Consequently, the firewall will not permit traffic to pass unless it is from one of the specified hosts known to a given interface. This will stop spoofing attempts and other miss-configuration problems. However, it does cannot stop problems if a physical device substitution is made on a given network segment.

In this test, we will place a system on each of the external service networks, the inside and the outside interface segments and configure the test system with a plausible IP address from the subnet being tested, but

not included in the firewall interface configuration. For the outside interface, we will use an internal IP address. The actual test must use a packet with a destination port that would normally be permitted, smtp will work well for the external and service networks.

```
nmap –P0 –p25 223.223.223.70                # From an external host

[root@loki root]# nmap -P0 -p25 223.223.223.70

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (223.223.223.70):
Port       State       Service
25/tcp     filtered    smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 46
seconds
```

We repeat this test from each of the four segments and find that rules are correct.

Next we want to repeat this test using reserved IP addresses, non-routable addresses, IP of 0.0.0.0 and broadcast addresses. In each case, we confirm the firewall drops the packets.

```
nmap –p25 –S10.1.1.1 223.223.223.65          # To external DNS
nmap –p25 –S172.16.1.1 223.223.223.65        # To external DNS
nmap –p25 –S192.168.1.1 223.223.223.65       # To external DNS

[root@loki root]# hping -a 224.0.0.1 -p 25 223.223.223.65
HPING 223.223.223.65 (eth1 223.223.223.65): NO FLAGS are set, 40
headers + 0 data bytes

--- 223.223.223.65 hping statistic ---
46 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

tcpdump listing:
21:37:48.997367 224.0.0.1.1910 > 223.223.223.65.smtp: . win 512
21:37:49.997310 224.0.0.1.mtp > 223.223.223.65.smtp: . win 512
21:37:50.997242 224.0.0.1.1912 > 223.223.223.65.smtp: . win 512
21:37:51.997177 224.0.0.1.1913 > 223.223.223.65.smtp: . win 512
21:37:52.997111 224.0.0.1.1914 > 223.223.223.65.smtp: . win 512
```

Host on Wrong Interface:

The last series of tests includes reconnecting a device assigned to the EOSN on the ERSN. We expect and find that the firewall drops these

packets, since it knows that this address belongs to a different firewall interface.

Sanity Test:

We will run another nmap test against all TCP and UDP ports from outside the firewall to see if any unexpected results show up. This is a very long test and the results are even longer and have been omitted for space issues.

```
nmap –P0 –p1-65535  223.223.223.34        #GIAC's NAT address
nmap –sU –P0 –p1-65535 223.223.223.34
```

Port Tests:

We will use nmap to verify access to the ports and unusual TCP flag combinations. First test is for open ports from the outside targeting the full range of addresses used in the external networks. This range does include vacant addresses. In the first pass we will just use the well-known ports to conserve time. If we see unexpected results, we will repeat at least some of the tests with all 65,535 ports. In this assignment we will not use the –O option to fingerprint a target's OS, because we already know the network structure. Even through we ran only abbreviated versions of each of the following port scans, we feel extremely confident the results would have not have changed if we had extended the test to all possible hosts and the full 65535 ports.

TCP connect Scan:

This option is used to create an open session with the target. All GIAC's 223.223.223.0 addresses are in the 34-71 range except for the border router, which is 223.223.223.33. The –n option prevents reverse DNS lookups to save time.

```
nmap –sT –P0  -n 223.223.223.34-71  –p 1-1024  # TCP port connect scan

[root@loki root]# nmap -sT -P0 -n -p1-1024 223.223.223.34-71

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
caught SIGINT signal, cleaning up
[root@loki root]# nmap -sT -P0 -n -p1-1024 223.223.223.34

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
All 1024 scanned ports on  (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 1268 seconds
```

UDP Scan:

If we receive an ICMP port unreachable error message we know the port is closed. No response may mean the port is either open or the packet was dropped.

nmap –sU –P0 –n 223.223.223.34-71–p 1-1024 # UDP port scan

[root@loki root]# nmap -sU -P0 223.223.223.34 -p1-100

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 100 scanned ports on (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 120 seconds

ACK Scan:

nmap –sA –P0 –n 223.223.223.34-71 –p 1-1024 # Ack scan

[root@loki root]# nmap -sA -P0 -n 223.223.223.34 -p1-100

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 100 scanned ports on (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 264 seconds

tcpdump listing:
22:45:15.052866 10.10.10.10.56935 > 223.223.223.34.24: . ack 2481792015
win 2048
22:45:15.052889 10.10.10.10.56935 > 223.223.223.34.linuxconf: . ack 2481792015 win 2048
22:45:15.052941 10.10.10.10.56935 > 223.223.223.34.87: . ack 2481792015
win 2048
22:45:15.052956 10.10.10.10.56935 > 223.223.223.34.msp: . ack 2481792015 win 2048
22:45:15.053005 10.10.10.10.56935 > 223.223.223.34.51: . ack 2481792015
win 2048
22:45:15.053020 10.10.10.10.56935 > 223.223.223.34.netrjs-3: . ack 2481792015 win 2048

FIN Scan:

nmap –sF –P0 –n 223.223.223.34-71 –p 1-1024 # Fin scan

[root@loki root]# nmap -sF -P0 -n 223.223.223.34 -p1-100

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 100 scanned ports on (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 120 seconds

tcpdump listing:
22:53:12.750946 10.10.10.10.41049 > 223.223.223.34.10: F 0:0(0) win 409622:53:12.750996 10.10.10.10.41049 > 223.223.223.34.8: F 0:0(0) win 4096
22:53:12.751011 10.10.10.10.41049 > 223.223.223.34.32: F 0:0(0) win 409622:53:12.751061 10.10.10.10.41049 > 223.223.223.34.bootpc: F 0:0(0) win
4096
22:53:12.751075 10.10.10.10.41049 > 223.223.223.34.93: F 0:0(0) win 4096

Xmas Scan (Fin, Urg & Push):

nmap –sX –P0 –n 223.223.223.34-71 –p 1-1024 # Xmas scan

[root@loki root]# nmap -sX -P0 -n 223.223.223.34 -p1-100

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 100 scanned ports on (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 120 seconds

tcpdump listing:
22:54:22.003737 10.10.10.10.35050 > 223.223.223.34.time: FP 0:0(0) win 3072 urg 0
22:54:22.003798 10.10.10.10.35050 > 223.223.223.34.msp: FP 0:0(0) win 3072 urg 0
22:54:22.003859 10.10.10.10.35050 > 223.223.223.34.99: FP 0:0(0) win 3072 urg 0
22:54:22.003919 10.10.10.10.35050 > 223.223.223.34.90: FP 0:0(0) win 3072 urg 0

Null Scan:

nmap –sN –P0 –n 223.223.223.34-71 –p 1-1024 # Null scan

[root@loki root]# nmap -sN -P0 -n 223.223.223.34 -p1-100

Starting nmap V. 3.00 (www.insecure.org/nmap/)
All 100 scanned ports on (223.223.223.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 120 seconds

tcpdump listing:
22:56:53.523280 10.10.10.10.61025 > 223.223.223.34.whois++: . win 1024
22:56:53.537489 10.10.10.10.61025 > 223.223.223.34.ftp: . win 1024
22:56:53.537595 10.10.10.10.61025 > 223.223.223.34.97: . win 1024
22:56:53.537655 10.10.10.10.61025 > 223.223.223.34.netrjs-3: . win 1024
22:56:53.537715 10.10.10.10.61025 > 223.223.223.34.rje: . win 1024
22:56:53.537773 10.10.10.10.61025 > 223.223.223.34.99: . win 1024
22:56:53.537831 10.10.10.10.61025 > 223.223.223.34.rlp: . win 1024

In all port scans we observed that the firewall blocks this traffic since it is not part of an established session. The SYN Scan will work against real hosts in the external service networks.

Next we ran the above tests again with packet fragments, adding the –f option to nmap. Results were the same.

We will expect essentially the same results when we run these tests from each of the two external service networks and from the inside segment. The statefull nature of the firewall should not permit any packet through that is not an initial SYN packet or part of an established session. Examination of the tcpdump logs does confirm our expectations

Oracle Rules:

In this network design we choose not to permit the Oracle 9iAS server to communicate directly with an internal Oracle 9i database server, e-mail is used instead. This also means we did not have to open ports 1521 or 1630 for Oracle traffic, depending on where the Oracle Listener might have been placed.

Testing Analysis:

The firewall reacted to all of our tests exactly as expected, blocking all spoofed or otherwise improper IP addresses properly. The firewall's state tables properly blocked Port scans in all directions. A complete policy listing of the firewall used in this test is included as Appendix B.

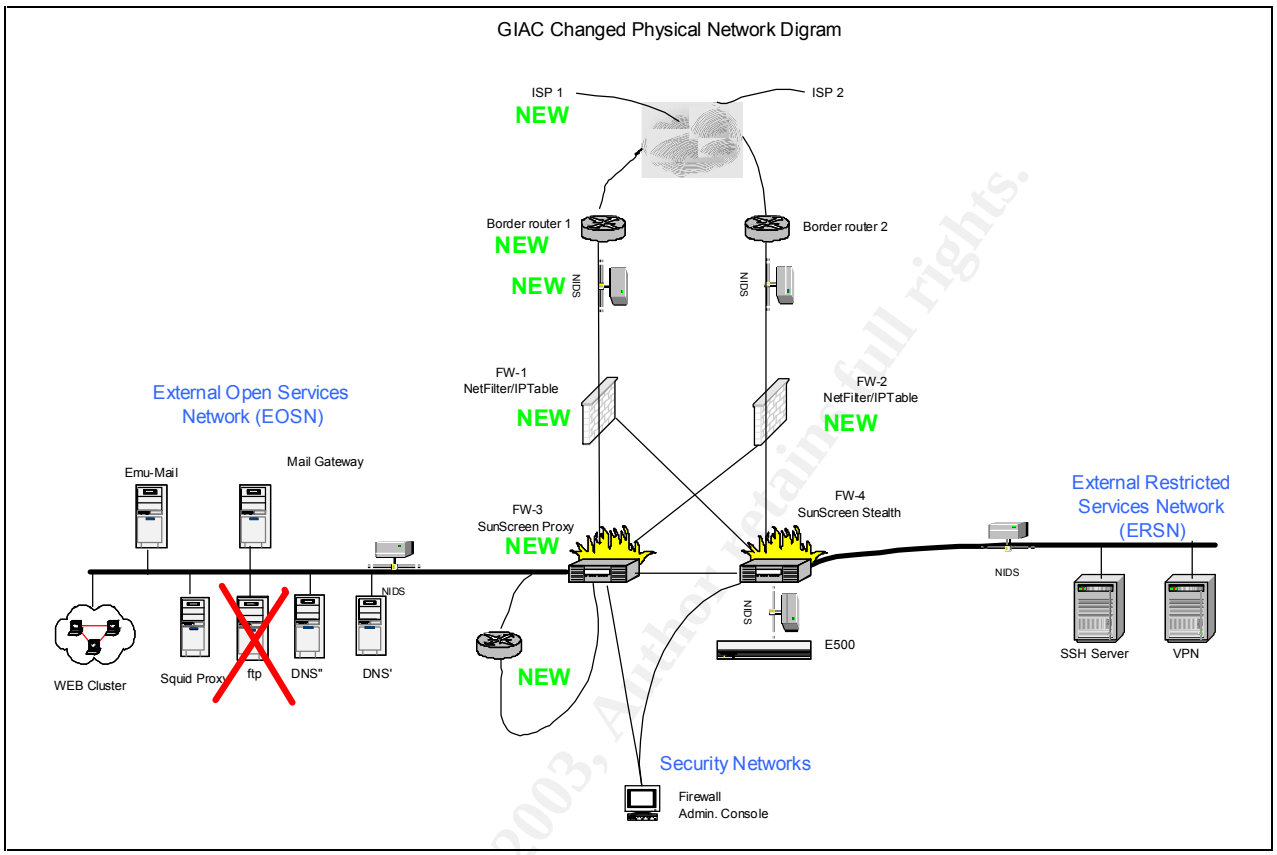
Improvements:

Through out the above assignments, we have explained several alternate solutions for a variety of network design considerations. Recommendations for changes should be partially based on risk mediation. Without testing from the outside all the way to individual hosts, I believe most recommendations are premature. Some of the design changes I would consider making include:

- Adding a Netfilter/IPTable firewall in front of the external SunScreen firewall to provide two different firewall solutions are the perimeter.
- Add access to a second ISP to provide alternate Internet access paths and improve availability. This would require a much more complex firewall and router configuration including RGP as an exterior routing protocol.
- Move the exterior DNS servers to a dedicated service segment. There is no real requirement to these on the same network segment with the more vulnerable web, mail and ftp servers.
- Eliminate the exterior ftp drop box is at all possible. This might mean increasing the size of allowed in-bound e-mail messages. It is not clear that the business gain of the ftp drop box is worth the added risks and complexities.
- Add a proxy firewall in-series with the Exterior Open Services Network, web, ftp and mail servers. This would make back channel attacks more difficult because of possible authentication requirements at the proxy level.
- Consider adding Quality-of-Services (QoS) features such as Network-Based Application Recognition (NBAR) to ensure a desired distribution of bandwidth between Internet services, such in-bound http and ssh verses outgoing web traffic and e-mail. There is no point in implementing such features unless there is a demonstrated problem that needs to be fixed.
- Evaluate some of the protocol related ACLs to determine if they are necessary with the SunScreen firewall in a statefull and stealth mode. The redundancy may not be worth the extra time in processing the added rules. A key factor to consider will be actual network loads and the load distributions over time.

Each of these options help to improve security in certain respects at the added cost of increased network complexity and management, which may decrease overall security. Any changes to a production needs to be reviewed extensively and preferably tested in a lab environment prior to production roll out. These proposed changes are roughed out in the following revised physical network drawing. Many unchanged components were omitted to increase emphasis on

the changes. Please note the third router attached to the SunScreen proxy firewall (FW-3) is required when configuring the SunScreen in mixed mode, stealth and proxy.

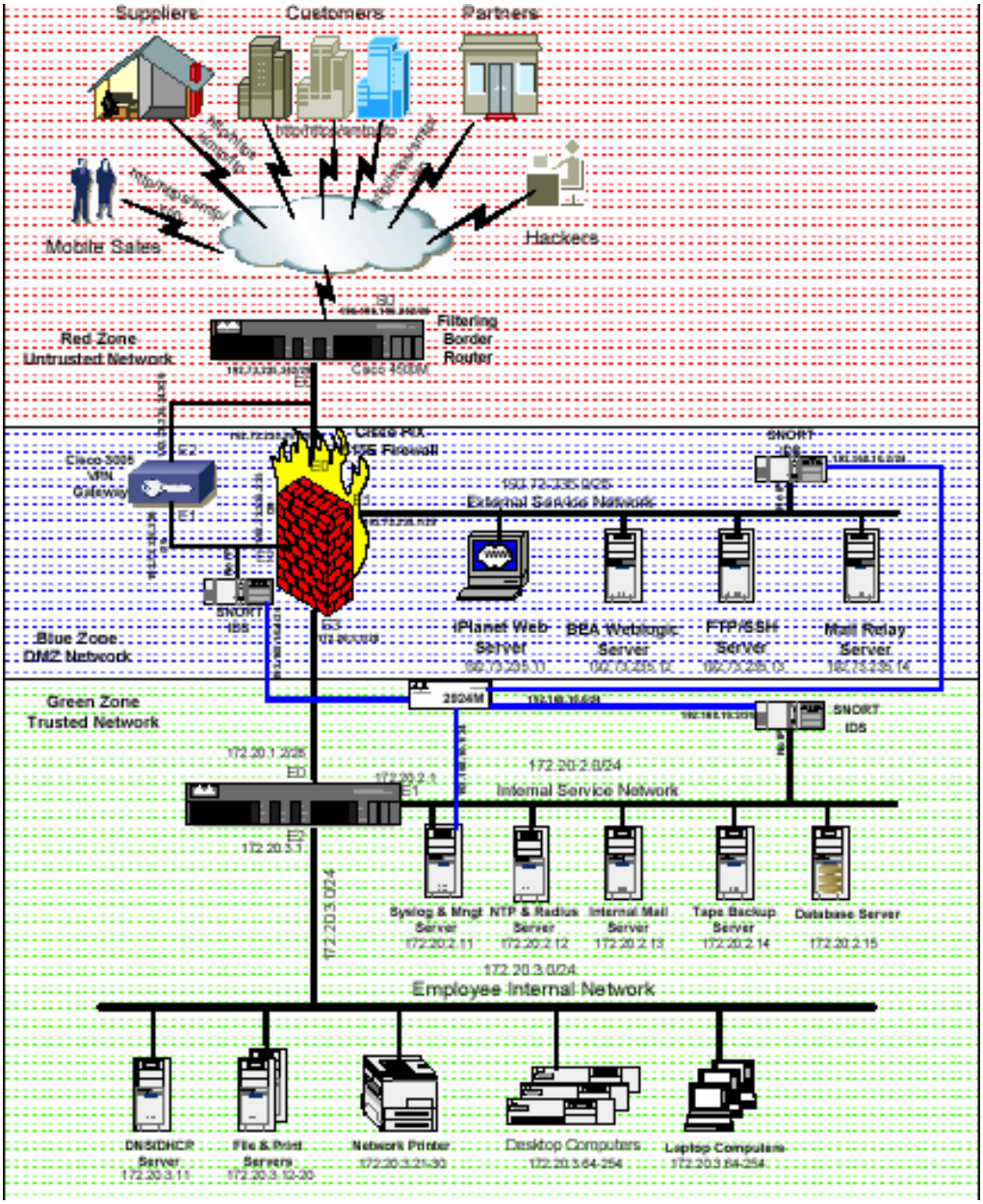


This is a much more complicated network design with higher support requirements and I would not recommend it for a smaller organization. There is a great deal to be said about simplicity and better security.

ASSIGNMENT IV: DESIGN UNDER FIRE:

Introduction:

I have selected a network design for GIAC Enterprises by William Chan published November 30, 2002, http://www.giac.org/practical/ .



Attack the Firewall:

Firewall Vulnerability:

Preparation for an attack typically includes steps such as: reconnaissance, network mapping, target selection, vulnerability research, determining a method past perimeter defenses and other network and host protection.

In this situation, reconnaissance and network mapping was accomplished by selecting and reviewing William Chan's practical assignment.

We searched for vulnerabilities of the CISCO PIX firewall 6.2 at several standard web sites such as CERT, ISS X-Forace, Mitre's CVE service, SecurityFocus and http://www.cisco.com/go/psirt. Presently, there are only a few vulnerabilities for this version of Cisco PIX firewall. Of those we found the "Cisco PIX Firewall Telnet/SSH Subnet Handling Denial of Service Vulnerability" appear to give us the best chance of a compromise against William's network design, BugTrak ID 6110 and described in SecurityFocus, http://www.securityfocus.org/bid/6110.

Preparation:

The Security Focus vulnerability discussion describes that if TCP SYN packets are repeatedly sent to the subnet address, the firewall may suffer a denial of service condition, http://www.securityfocus.org/bid/6110/discussion.

In William's external firewall policy table, he indicates that all internal subnets have telnet and ssh access to the external service networks, the border router and external hosts. No specific mention was made of the external firewall. However, in light of the above policies and a general lack of specific host configuration and policy information, we believe that the external firewall was probably configured with telnet/SSH access enabled for internal hosts.

The firewall address is known to be 192.73.235.241 with a 29-bit mask. Therefore, the firewall's subnet address is 192.73.235.240/29. This is the IP address we need to attack with TCP/SYN packets.

The vulnerability literature is not specific if the TCP/SYN packets must be sent to a particular interface for this exploit to work. We would want to test this issue on a lab network to confirm how the exploit really works. However, for this effort we will assume any interface is suitable. I do not presently have access to the required equipment to run this experiment.

Another design question is do we have to maintain the stream of TCP/SYN packets to make the firewall unresponsive or will it crash and have to be rebooted after a short burst of such traffic. If it will crash, we might want to set up a simple cron job to create the desired traffic pattern every couple of hours. If we do this, we will at least want to vary the spoofed address. Preferably, we would use a TFN2K style DoS to do the work for us, thus hiding our true identity a little better.

Attack Design:

We want to hit the network address of the firewall with a continuous stream of TCP/SYN packets. This could be done with a variety of packet generators. Since we do not need to receive any return packets in this attack, we will spoof an address to hide our real identity. During the attack, we could use a regular web browser to determine if GIAC's firewall has been made unresponsive.

To set up this attack we will use Hping2, with the following command line:

hping -S –i u10000 –a 1.2.3.4 192.73.235.240

- -S for TCP SYN
- -I u10000 to send 10 packets/second
- -a 1.2.3.4 for a spoofed address (apologies for anyone who owns 1.2.3.4)

Countermeasures:

Unless it is necessary, disable the telnet/SSH access feature for internal hosts. In Security Focus, Cisco recommended to "… use the ACL option to filter TCP directed broadcasts specific to the subnet.", http://www.securityfocus.org/bid/6110/solution. The other option is to upgrade to 6.2.2.111 of the CISCO PIX firewall OS.

Results:

GIAC's firewall should be unresponsive due to the continued TCP/SYN flood directed to its subnet address. I do not have access to the required equipment to test this setup.

Denial of Service Attack:

Attack Design:

We have elected to use a standard SYN flood style DoS attack using TFN2K against the public web server. Regardless of the countermeasures taken, if enough systems are used in the attack, the target is almost certain to become unresponsive due to exhaustion of network bandwidth. We will compromise 50 PCs connected to the Internet via DSL or cable modems to obtain higher performance and greater probability they will be on-line when needed. Similar DoS attack tools include Trinoo, TFN and Stacheldraht.

The TFN2K attack is based on amplification from one TFN2K Master to an arbitrary number of TFN2K daemons, all directing SYN packets to the targeted host. Communication between the master and daemons can be done with TCP, UDP or ICMP packets using random ports or a port chosen at execution time.

Getting the IP address of the web server should be trivial, by doing either a zone transfer from the GIAC DNS servers or just guessing that it's www.giac.com. Next we will establish a web session with www.giac.com just to confirm IP address and port numbers.

The nature and capabilities of TFN2K is explained in "CERT Advisory CA-1999-17 Denial-of-Service Tools".

> "Like TFN, TFN2K is designed to launch coordinated denial-of-service attacks from many sources against one or more targets simultaneously. It includes features designed specifically to make TFN2K traffic difficult to recognize and filter, to remotely execute commands, to obfuscate the true source of the traffic, to transport TFN2K traffic over multiple transport protocols including UDP, TCP, and ICMP, and features to confuse attempts to locate other nodes in a TFN2K network by sending "decoy" packets.
>
> TFN2K is designed to work on various UNIX and UNIX-like systems and Windows NT.
>
> TFN2K obfuscates the true source of attacks by spoofing IP addresses. In networks that employ ingress filtering as described in [1], TFN2K can forge packets that appear to come from neighboring machines.
>
> Like TFN, TFN2K can flood networks by sending large amounts of data to the victim machine. Unlike TFN, TFN2K includes attacks designed to crash or introduce instabilities in systems by sending malformed or invalid packets."

Our objective is to shut down GIAC's Internet ISP connection. This may be accomplished by consuming their ISP bandwidth capacity and/or resource consumption on their servers. William has specified the Internet connection as a 5Mbps fractional T3 circuit. Our 50 compromised PC daemons should be able to flood this circuit.

TFN2K may be obtained in a zipped, tar format from http://www.defcon.tv/distributed/tfn2k.tgz.

In preparation for the attack we have listed the 50 compromised PCs in a file we named "attack.hosts", and set up an 8 to 32-character password. The attack command will be:

Attack Master# tfn –f attack.hosts –c5 -1 192.73.235.11

- -c5 is command ID for a SYN flood that may fill connection tables on servers
- -c4 is for UDP flood, may cause ICMP port unreachable messages
- -c6 ICMP echo reply or ping attack, send large ping packets and the target responds with packets of equal size
- -c7 SMURF attack, uses amplification and pings
- -c8 MIX attack, uses a straight ratio of UDP, SYN and ICMP packets

Countermeasures:

There are no absolute means to stop a TFN2K DoS attack, besides disconnecting from the Internet, but then the game's over. Some of the recommended measures include:

- Use application proxies to filter TFN2K packets if possible.
- Use application proxy exclusive firewalls if possible.
- DoS aware firewalls, that can detect and drop DoS packets.
- Block all UDP, TCP and ICMP traffic that is not required.
- Use bandwidth management tools.
- Use anti-spoofing rules on border routers.
- Block all non-essential ports.
- On service hosts decrease the timeout value, so partially established sessions timeout faster.
- Increase memory allocated for establishing and maintaining connections.
- Take advantage of any capabilities on firewalls, which may detect and block SYN floods.
- Be a good Internet neighbor and use effective egress filtering so your systems are not used in DoS attacks. Such as detecting hosts generating more than a certain number of SYN packets per second, and block those IPs.

Following is typical tcpdump trace of a SYN flood.

20:34:12.829501 spoofsrc.4601>targetdest.http: S 459328027: 459328027(0) win 65535
20:34:12.829788 targetdest.http> spoofsrc.4601: S 788402512: 788402512(0) ack 459328028 win 5840 <mss 1460>(DF)

20:34:12.995412 spoofsrc.3728>targetdest.http: S 459328027: 459328027(0) win 65535
20:34:12.996799 targetdest.http> spoofsrc.3728: S 78848736: 78848736(0) ack 459328028 win 5840 <mss 1460>(DF)

Results:

As expected, the server becomes unresponsive very quickly and remains in that condition until we stop the TFN2K attack

Attack an Internal System:

Target Selection:

We have selected to attack the internal sendmail server using the newly found (March 3, 2003) Remote Sendmail Header Processing Vulnerability, CAN-2002-1337, SANS Alert 2003-03-03, http://wiki.sans.org/tiki-index.php?page=SendmailExploit. Presently, this vulnerability is believed to affect nearly all implementations of sendmail between 5.2 and 8.12.7, http://nipc.gov/warnings/advisories/2003/03-004.htm. Some sources say sendmail 5.79 and 8.12.7. CERT and SecurityFocus both published notes describing which sendmail vendor implementations are vulnerable, http://www.securityfocus.org/archive/1/31757/2003-03-01/2003-03-07/0 and http://nipc.gov/warnings/advisories/2003/03-004.htm.

William was not explicit about the OS for many of the servers in his network plan. There is little evidence of any significant use Window systems for servers. He is explicit about using Snort v 1.8.6 on a Red Hat Linux 7.2 platform. From the lack of contrary information, we will suspect his internal mail server also runs on Red Hat 7.2. He did state this server runs sendmail. By sending e-mail with a bad "To:" address, like webmater@giac.com, we might get a mailbox unavailable message. If we are lucky, we might get the sendmail version and names of the servers that processed the message at GIAC Enterprises. There maybe useful information imbedded in this undeliverable message.

Since both SecurityFocus and CERT confirmed that Red Hat 7.2 sendmail implementation is vulnerable to this flaw. We believe there is an excellent chance that the targeted mail server will be vulnerable. His implementation of Snort is vulnerable to the Snort RPC Preprocessing Vulnerability this may be of interest later in this attack, http://www.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21951.

This sendmail vulnerability involves a decrementing counter problem when multiple brackets <> are used in the "From:" line in either the smtp command line or when "From :" is embedded in a message. The embedded option is much better because we can include about 2k bytes worth of code opposed to only 256 bytes in the command line option.

Attack Process:

With an exploit tool similar to one released by a Polish group called the Last Stage of Delirium (LSD) or proof of concept code published by SecurityFocus, http://www.securityfocus.org/archive/1/31757/2003-03-01/2003-03-07/0 we could create an email message with a header where the "From::" line includes the exploit code and send it to any legal address at GIAC Enterprises. There is no issue about getting our deadly message through their firewall and other security tools, since this is a message-oriented and not a connection-oriented vulnerability, http://www.cert.org/advisories/CA-2003-07.html. Getting a valid address should be simple by checking their web site for the web master's address or any other "mail this person for help" type of text. Perhaps GIAC has even published their employee phone list and e-mail list on their web site to be customer friendly. Ideally, we want to use an addresses that would appear to be a one character typo error, such as webmater@giac.com. This obscurity may be wasted if GIAC closely examines their non-deliverable mail.

This attack will permit us to run arbitrary code with the permissions of the sendmail deamon, likely root. One of the recently published e-mails suggesting to have working exploit code, claimed to have returned a root prompt from a compromised host. If this is true, it may be not be difficult to get an external root prompt from the sendmail server on William's network. He states in the external firewall policy that all internal networks may initiate telnet, ftp, ssh, http and https sessions with any host on the Internet. Any internal host may also ping any external host. Assuming he has left this large hole open in his firewall, we will assume that at least some of these tools were left on many servers in his network. We would like to use telnet, ssh/scp and ftp.

If we are lucky and get a root prompt, we want to stop logging, get the /etc/password/shadow, /etc/services, /etc/inetd.conf and /etc/ssh(2)/sshd_config files.

If we fail to get a prompt back, our first test will be to mail ourselves a copy of the /etc/password/shadow, /etc/services, /etc/inetd.conf and /etc/ssh(2)/sshd_config files. If we get these files back we will know our exploit was successful. Our next intention will be to use John The Ripper, on our site, to crack at first only the root password, which we do in 18 hours. The contents of /etc/services and /etc/inetd will tell us great deal about what services and ports may be available on the GIAC network. The content of /etc/ssh/sshd_config will tell us if root is permitted to login with secure shell and possibly a list of other ssh privileged users in the "allowed users" line. William did not provide sshd_config file details. Depending upon the results we may want to crack one these ssh

privileged accounts. We could use it to ssh to another system and then su to root. We are hoping that GIAC system administrators have used the same passwords for root and another one for themselves on various internal and possible external systems.

The security policy permits us use secure copy (scp) or ftp out of the GIAC network to download additional software, such as rootkits, Hping2 as a backdoor using the -9 option, port forwarders and other remote control software. Perhaps a bit slow for a control process, we could take the redirected results of these actions and include them in mail messages back to ourselves. Another option, if we can use ftp or scp, is to install a copy of Hping2 or similar tool. We might be able to use the data payload portion of an hping2 packet as a covert channel. Even if we could not compromise other GIAC systems, we could always issue a simple line such as:

```
cd /; rm -rf *
```

GIAC Enterprises would at least know something very bad happened and would give them a good excuse to test their recovery procedures.

Unless the Snort systems have been maintained with up-to-date rules, this attack most likely would go undetected at the border router, firewalls and NIDS, because we are using what would otherwise be acceptable e-mail. Even if GIAC got and installed the Snort rule for this sendmail vulnerability early, we may still be able to blind them. By crafting an RPC packet on the sendmail server and sending it to the iPlant web server, we might take out two of the three Snort boxes, by capitalizing on the new Snort RPC Preprocessing Vulnerability. The third Snort system is inside of the VPN server and will never see any of our traffic. It is doubtful we could get any externally generated RPC packets through the external firewall to blind the Snort probes, so we need to first compromise an internal system.

At this point it is checkmate- game over.

Countermeasures:

This type of vulnerability is similar to Code Red and Nimda, in that the vulnerability is installed by default in a widely deployed product. The best defense is to either install one of the vendor's patches for the current version of sendmail or upgrade to 8.12.8, which is not vulnerable. Presently, there are no other known measures to block traffic exploiting this vulnerability.

Another option is to migrate from sendmail to a MTA product designed with security features built-in. Sendmail is a mature product that has been

subjected to extensive reviews and believed to have been stable for a number of years, but it is very large and complex.

Other fatal flaws included use of the same passwords for root and other privileged accounts across many of the internal systems. Using different passwords at least within a given security zone might limit the extent and or speed of compromises. Being able to use scp and ftp from any of the internal servers permitted us to install our favorite tools. It is best to limit availability and network policy of tools such as ftp, telnet, ping, http(s), ssh and scp to the fewest number of hosts possible and only to those networks or subnets as is required.

Join and read Internet lists about the NIDS system and keep the NIDS signature files up-to-date so new exploits and vulnerabilities might be detected early.

Results:

Not only was the mail server root compromised, a significant portion of the GIAC network fell because of repeated use of the same passwords across many servers for the same ID.

APPENDIX A
Custom Solaris Hardening Script

```
#! /bin/sh
#  Script Name:       sys.setup
#  Shell              sh
#  Author:           Fred Gross
#  Arguments:          None
#  Purpose:           To perform some of the basic system setup functions for a new
server
#
#  Created:           FG 5/8/2002
#  Last Modify:
#
echo "All files being modified are being copied as xxxx.bk in the same directory."

echo "Checking for default router"
cd /etc
 if [ -f /etc/defaultrouter ]; then
  echo "Value for the default route is `cat /etc/defaultrouter`"
 fi

echo "Create links in root"
cd /
ln -s /usr/local/scripts /scripts
ln -s /usr/local/sys_dir /sys_dir
ln -s /usr/local/sys_reports /sys_reports
ln -s /usr/local/tars /tars

echo "Create directories in /usr/local"
cd /usr/local
 mkdir scripts sys_dir sys_reports tars

echo "Create directories for ntp"
cd /var
 mkdir lock
 cd lock
 mkdir subsys
cd /etc
 mkdir drift
echo "Set new PATH"
 PATH=$PATH:/usr/local/scripts:/usr/local/sys_dir
 export PATH
```

```
echo "Fix PATH in /etc/profile
cd /etc
 cp /etc/profile /etc/profile.bk
 sed -e '/PATH=/c\
 PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/openwin/bin:/usr/local/scripts; export PATH' <
/etc/profile > tmp
 mv tmp /etc/profile


echo "Creating empty hosts.equiv, /.rhosts and /.netrc files
 touch /.rhosts /.netrc /etc/hosts.equiv
 chmod 0 /.rhosts /.netrc /etc/hosts.equiv

echo "Creating control files for cron"
cd /etc/cron.d
 echo "root" > cron.allow
 echo "oracle" > cron.allow
 chown root cron.allow
 chmod 600 cron.allow

 cp -p cron.allow at.allow

 cat /etc/passwd | cut -d: -f1 | grep -v root | grep -v oracle > cron.deny
 chown root cron.deny
 chmod 600 cron.deny

 cp -p cron.deny at.deny

echo "Setting unmask values for all startup scripts to 022"
cd /etc/init.d
 echo "unmask 022" > unmask.sh
 cdhmod 700 unmask.sh
 for d in /etc/rc2.d
  do
   ln /etc/init.d/unmask.sh $d/S00unmask.sh
  done

echo "Verify logging is correct"
cd /var/adm
 touch loginlog
 chown root loginlog
 chgrp sys loginlog
 chmod 600 loginlog

 touch sulog
 chown root sulog
```

```
 chgrp sys sulog
 chmod 600 sulog

 touch tcpdlog
 chown root tcpdlog
 chgrp sys tcpdlog
 chmod 600 tcpdlog

 chmod 600 messages

echo "Disbale NIS"
 cd /opt
 if [ -d /opt/SUNWnisu ]; then
  echo "Removing NIS packages"
  rm -r SUNWnisu
  rm -r SUNWnisr
 fi

echo "add RC script to ensure /tmp/has sticky bit"
cd /etc/rc3.d
 echo "#!/bin/sh" > S79tmpfix
 echo "if [ -d /tmp ]" >> S79tmpfix
 echo "then" >> S79tmpfix
 echo "/usr/bin/chmod 1777 /tmp" >> S79tmpfix
 echo "/usr/bin/chgrp sys /tmp" >> S79tmpfix
 echo "/usr/bin/chown sys /tmp" >> S79tmpfix
 echo "fi" >> S79tmpfix

 chmod 750 ./S79tmpfix
 chgrp sys ./S79tmpfix

echo "If not a router then touch /etc/notrouter"
 touch /etc/notrouter

echo "Set aset to high for the Automated Security Enhancement Tool"
 /usr/aset/aset -l high

echo "Create /etc/ftpusers"
cd /etc
 cat /etc/passwd | cut -d: -f1 | grep -v root | grep -v oracle | grep -v isb* > /etc/ftpusers
 chown root /etc/ftpusers
 chmod 600 /etc/ftpusers

echo "Fix default values in /etc/default"
cd /etc/default
 cp /etc/default/passwd /etc/default/passwd.bk
```

```
 sed -e 's/MAXWEEKS=/&12/' </etc/default/passwd > /etc/default/tmp
  mv /etc/default/tmp /etc/default/passwd
 sed -e 's/MINWEEKS=/&2/' </etc/default/passwd > /etc/default/tmp
  mv /etc/default/tmp /etc/default/passwd
 sed -e 's/PASSLENGTH=6/PASSLENGTH=8/' </etc/default/passwd > /etc/default/tmp
  mv /etc/default/tmp /etc/default/passwd

echo "Configure RFC 1948 for TCP sequence numbers"
cd /etc/default
 cp /etc/default/inetinit /etc/default/inetinit.bk
 sed -e 's/TCP_STRONG_ISS=1/TCP_STRONG_ISS=2/' < /etc/default/inetinit >
/etc/default/tmp
  mv /etc/default/tmp /etc/default/inetinit

cd /etc/rc2.d
cp S69inet _bkS69inet
 echo "Making changes to /etc/rc2.d/S69inet"
 sed -e '/Configure default IPv4 routers/i\
# Change LOTS of network parameters.  This should help to secure\
# the system against some types of Denial Of Service attacks, and\
# intrusion attempts.  It will also keep us from forwarding Denial\
# Of Service attacks to other networks.\
#\
# Combat ARP DOS attacks by flushing entries faster.\
/usr/sbin/ndd -set /dev/arp arp_cleanup_interval 60000\
/usr/sbin/ndd -set /dev/ip ip_ire_arp_interval 60000\
#\
# Combat ICMP DOS attacks by ignoring them.\
/usr/sbin/ndd -set /dev/ip ip_respond_to_echo_broadcast 0\
/usr/sbin/ndd -set /dev/ip ip6_respond_to_echo_multicast 0\
/usr/sbin/ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0\
/usr/sbin/ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0\
#\
# Ignore redirect requests.  These change routing tables.\
/usr/sbin/ndd -set /dev/ip ip_ignore_redirect 1\
/usr/sbin/ndd -set /dev/ip ip6_ignore_redirect 1\
#\
# Don't send redirect requests.  This is a router function.\
/usr/sbin/ndd -set /dev/ip ip_send_redirects 0\
/usr/sbin/ndd -set /dev/ip ip6_send_redirects 0\
#\
# Dont respond to timestamp requests.  This may break rdate\
# on some systems.\
/usr/sbin/ndd -set /dev/ip ip_respond_to_timestamp 0\
#\
# If a packet isnt for the interface it came in on, drop it.\
```

```
/usr/sbin/ndd -set /dev/ip ip_strict_dst_multihoming 1\
/usr/sbin/ndd -set /dev/ip ip6_strict_dst_multihoming 1\
#\
# Dont forward broadcasts.\
/usr/sbin/ndd -set /dev/ip ip_forward_directed_broadcasts 0\
#\
# Dont forward source routed packets.\
/usr/sbin/ndd -set /dev/ip ip_forward_src_routed 0\
/usr/sbin/ndd -set /dev/ip ip6_forward_src_routed 0\
#\
# Combat SYN flood attacks.\
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q0 8192\
#\
# Combat connection exhaustion attacks.\
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q 1024\
#\
# Dont forward reverse source routed packets.\
/usr/sbin/ndd -set /dev/tcp tcp_rev_src_routes 0\
#\
# Combat IP DOS attacks by decreasing the rate at which errors\
# are sent.\
/usr/sbin/ndd -set /dev/ip ip_ICMP_err_interval 1000\
/usr/sbin/ndd -set /dev/ip ip_ICMP_err_burst 5\
#\
#' < /etc/rc2.d/S69inet > tmp
  mv /etc/rc2.d/tmp /etc/rc2.d/S69inet


echo "Edit /etc/rc2.d/S72inetsvc to limit creation of core files"
cd /etc/rc2.d
 cp /etc/rc2.d/S72inetsvc /etc/rc2.d/_bkS72inetsvc
 sed -e 's/\/usr\/sbin\/inetd -s/& -t/' < /etc/rc2.d/S72inetsvc > tmp
  mv /etc/rc2.d/tmp /etc/rc2.d/S72inetsvc
 sed -e '/inetd -s/i\
ulimit -c 0' < /etc/rc2.d/S72inetsvc > tmp
  mv /etc/rc2.d/tmp /etc/rc2.d/S72inetsvc

echo "Disable network root logins"
cd /etc/default
 cp /etc/default/login /etc/default/login.bk
 sed -e '/CONSOLE=/c\
CONSOLE=\/dev\/console' < /etc/default/login > tmp
  mv /etc/default/tmp /etc/default/login

echo "Limit use of the su command"
cd /etc
```

```
 cp /etc/group /etc/group.bk
 /usr/sbin/groupadd -g 13 admin
 /usr/bin/chgrp admin /usr/bin/su /sbin/su.static
 /usr/bin/chmod 4550 /usr/bin/su /sbin/su.static
 sed -e '/admin::13:/c\
admin::13:root,oracle,isb_02,isb_03,isb_08,isb_10' < /etc/group > tmp
  mv /etc/tmp /etc/group

echo "Set /etc/shells"
cd /etc
 cp /etc/shells /etc/shells.bk
 touch /etc/shells
 echo "sh" >> /etc/shells
 echo "bash" >> /etc/shells
 echo "sendmail" >> /etc/shells

echo "Set unmask values in /etc/profile, /etc/skel/local.cshrc & local.profile"
cd /etc
 cp /etc/.login /etc/.login.bk
 cp /etc/profile /etc/profile.bk
 cp /etc/skel/local.profile /etc/skel/local.profile.bk
 cp /etc/skel/local.login /etc/skel/local.login
 sed -e '$a\
unmask 027' < /etc/.login > tmp
  mv /etc/tmp /etc/.login

 sed -e '/umask 022/c\
unmask 027' < /etc/profile > tmp
  mv /etc/tmp /etc/profile

cd /etc/skel
sed -e '$a\
unmask 027' < /etc/skel/local.profile > tmp
  mv /etc/tmp /etc/skel/local.profile

sed -e '$a\
unmask 027' < /etc/skel/local.login > tmp
  mv /etc/tmp /etc/skel/local.login


echo "Over write ftp and telnet version banners"
cd /etc/default
 echo "" > /etc/default/ftpd
 echo "" > /etc/default/telnetd

echo "Fix /etc/issue & motd"
```

```
 if [ -f /etc/nmed.warning ]; then
  cp /etc/nmed.warning /etc/issue
  cp /etc/nmed.warning /etc/motd
 else
  echo "/etc/nmed.warning not found."
 fi

echo "Change password length to 8 characters"
cd /etc/default
 cp /etc/default/passwd /etc/default/passwd.bk
 sed -e '/PASSLENGTH/c\
PASSLENGTH=8' < /etc/default/passwd > tmp
  mv /etc/default/tmp /etc/default/passwd

 sed -e '/MAXWEEKS/c\
MAXWEEKS=12' < /etc/default/passwd > tmp
 mv /etc/default/tmp /etc/default/passwd

 sed -e '/MINWEEKS/c\
MINWEEKS=2' < /etc/default/passwd > tmp
   mv /etc/default/tmp /etc/default/passwd

echo "Fix optons in /etc/mail/sendmail.cf"
cd /etc/mail
  cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.bk
  sed -e '/SmtpGreetingMessage=/c\
O SmtpGreetingMessage=' < /etc/mail/sendmail.cf > tmp
  mv tmp /etc/mail/sendmail.cf

sed -e '/PrivacyOptions=/c\
O PrivacyOptions=authwarnings,noexpn,novrfy' < /etc/mail/sendmail.cf > tmp
  mv tmp /etc/mail/sendmail.cf

# Clean up startup scripts
/usr/local/sys_dir/sys.rc.cleanup

echo "Remove uneeded links"
 rm /usr/bin/yppasswd

echo "Setup authorized ssh users"
cd /usr/local/etc
 sed -e '/AllowUsers/c\
AllowUsers    isb_02 isb_03 isb_08 isb_10' < /usr/local/etc/sshd_config > tmp
  mv tmp /usr/local/etc/sshd_config

echo "Fix version given from BIND"
```

```
cd /etc
 if [ -f /etc/named.conf ]; then
   sed -e '/directory/a\
                        version "DNS";' < /etc/named.conf > tmp
   mv tmp /etc/named.conf
 fi

echo "Disallow syslogd from acting as the syslod server"
cd /etc/rc2.d
 sed -e '/\/usr\/sbin\/syslogd >\/dev\/msglog 2>&1 &/c\
              \/usr\/sbin\/syslogd -t >\/dev\/msglog 2>&1 &' < /etc/rc2.d/S74syslog > tmp
 mv tmp /etc/rc2.d/S74syslog

echo "Completed sys.setup"
echo "Make sure ntp is installed and running"
```

APPENDIX B
FIREWALL POLICY LISTING

```
//SunScreenConfig 3.0
//*********Do NOT edit this file.********//
//**It is generated by SunScreen tools.**//


//Internal State Engines
"realaudio"    STATE_ENGINE;
 "rsh"  STATE_ENGINE;
 "sqlnet"       STATE_ENGINE;
 "ftp"   STATE_ENGINE;
 "tcp"  STATE_ENGINE;
 "dns" STATE_ENGINE;
 "udp" STATE_ENGINE;
 "stateless"    STATE_ENGINE;
 "pmapudp"   STATE_ENGINE;
 "pmaptcp"    STATE_ENGINE;
 "ip"    STATE_ENGINE;
 "ping"STATE_ENGINE;
 "nfsro"        STATE_ENGINE;


//State Engines
"nfsro"STATE_ENGINE    "nfsro" "port" FORWARD  11     0      0xffff  3      86400
1 0 ;
 "nis"  STATE_ENGINE    "udp" "program"     FORWARD  10     0      0x7fffffff
        3       60 1 1 ;
 "pmap_nis"  STATE_ENGINE    "pmapudp" "program"      FORWARD  9      0
        0x7fffffff      2       60 -1 ;
 "pmap_udp" STATE_ENGINE    "pmapudp" "program"      FORWARD  8      0
        0x7fffffff      2       60 3600 ;
 "pmap_tcp"  STATE_ENGINE    "pmaptcp" "program"      FORWARD  7      0
        0x7fffffff      2       60 3600 ;
 "rpc_tcp"      STATE_ENGINE    "tcp" "program"      FORWARD  6      0
        0x7fffffff      1       86400 ;
 "rpc_udp"      STATE_ENGINE    "udp" "program"      FORWARD  5      0
        0x7fffffff      3       60 1 0 ;
 "realaudio"  STATE_ENGINE    "realaudio" "port"    FORWARD  4      1      0xffff
        1       3600 ;
 "rsh"  STATE_ENGINE    "rsh" "port"    FORWARD  4      1      0xffff  1      86400
;
 "sqlnet"       STATE_ENGINE    "sqlnet" "port"      FORWARD  4      1      0xffff
        1       3600 ;
```

```
 "ftp"   STATE_ENGINE    "ftp" "port"    FORWARD  4       1       0xffff  3       600
600 0 ;
 "tcp"   STATE_ENGINE    "tcp" "port"    FORWARD  4       1       0xffff  1       86400
;
 "tcpall"        STATE_ENGINE    "tcp" "port"    FORWARD  4       0       0xffff  1
        86400 ;
 "dns" STATE_ENGINE    "dns" "port"   FORWARD  3       2       0xffff  2       60 1 ;
 "ntp"   STATE_ENGINE    "udp" "port"   FORWARD  3       2       0xffff  3       60 1 0
;
 "udp_stateless"      STATE_ENGINE    "stateless" "port"    FORWARD  3       1
        0xffff  0       ;
 "udp_datagram"     STATE_ENGINE    "stateless" "port"    FORWARD  3       1
        0xffff  0       ;
 "udp" STATE_ENGINE    "udp" "port"   FORWARD  3       1       0xffff  3       60 1 0
;
 "udpall"        STATE_ENGINE    "udp" "port"   FORWARD  3       0       0xffff  3
        60 1 0 ;
 "ping"STATE_ENGINE    "ping" "type"  FORWARD  2       1       0xffff  1       10 ;
 "icmp"          STATE_ENGINE    "stateless" "type"    FORWARD  2       0       0xff
        0       ;
 "ipmobile"    STATE_ENGINE    "ip" "type"    FORWARD  1       3       0xff    2
        3600 0 ;
 "iptunnel"    STATE_ENGINE    "ip" "type"    FORWARD  REVERSE  1       2
        0xff    2       60 0 ;
 "ipfwd"         STATE_ENGINE    "ip" "type"    FORWARD  1       1       0xff    2
        60 1 ;
 "ip"    STATE_ENGINE    "stateless" "type"    FORWARD  1       0       0xff    0
        ;
 "ether"         STATE_ENGINE    "stateless" "type"    FORWARD  0       0       0xffff
        0       ;


//Services
"echo"SERVICE     tcp     7 ;
 "discard"      SERVICE     tcp     9 ;
 "systat"       SERVICE     tcp     11 ;
 "daytime"      SERVICE     tcp     13 ;
 "quote"        SERVICE     tcp     17 ;
 "chargen"      SERVICE     tcp     19 ;
 "ftp"   SERVICE     ftp     21 ;
 "telnet"       SERVICE     tcp     23 ;
 "smtp"         SERVICE     tcp     25 ;
 "time"SERVICE     tcp     37 ;
 "whois"        SERVICE     tcp     43 ;
 "nicname"      SERVICE     tcp     43 ;
 "dns" SERVICE     tcp     53      dns     53 ;
```

```
"tftp"  SERVICE     udp     69      PARAMS      ( 60 -1 7  ) ;
"gopher"      SERVICE     tcp     70 ;
"finger"      SERVICE     tcp     79 ;
"www"         SERVICE     tcp     80 ;
"pop" SERVICE     tcp     109 - 110 ;
"imap"        SERVICE     tcp     143 ;
"auth"SERVICE     tcp     113 ;
"ntp"  SERVICE     udp     123 ;
"nntp"SERVICE     tcp     119 ;
"snmp"        SERVICE     tcp     161     udp     161 ;
"snmp traps"SERVICE     udp_datagram        162 ;
"rlogin"      SERVICE     tcp     513 ;
"rsh"  SERVICE     rsh     514 ;
"sqlnet"      SERVICE     sqlnet 1521 ;
"syslog"      SERVICE     udp_datagram        514 ;
"printer"     SERVICE     tcp     515 ;
"rip"   SERVICE     udp_datagram        520 520       BROADCAST ;
"archie"      SERVICE     udp     1525   PARAMS     ( 360 -1 0  ) ;
"certificate discovery"       SERVICE     udp     1640   PARAMS      ( 60 1 1  ) ;
"remote administration"     SERVICE     tcp     3852 - 3854 ;
"HA administration"SERVICE     tcp     3853 ;
"HA heartbeat"      SERVICE     ping    8 ;
"HA"  SERVICE_GROUP "HA heartbeat"        "HA administration";
"SecurID PIN"       SERVICE     tcp     3855 ;
"securid"     SERVICE     udp     5500 ;
"securidprop"       SERVICE     tcp     5510 ;
"real audio"  SERVICE     realaudio     7070 ;
"traceroute"  SERVICE     udp_datagram        33430 - 34000  REVERSE          icmp
        11  REVERSE         icmp    3 ;
"tracert"     SERVICE     ping    8  REVERSE          icmp    11 ;
"icmp echo-reply"   SERVICE     icmp    0 ;
"icmp unreach"      SERVICE     icmp    3 ;
"icmp quench"       SERVICE     icmp    4 ;
"icmp redirect"     SERVICE     icmp    5 ;
"icmp echo-request"       SERVICE     icmp    8 ;
"router announcement"     SERVICE     icmp    9 9     BROADCAST ;
"router solicitation" SERVICE     icmp    10 10  BROADCAST ;
"icmp exceeded"     SERVICE     icmp    11 ;
"icmp params"       SERVICE     icmp    12 ;
"icmp info"   SERVICE     icmp    13 14 15 16 17 18 ;
"ping"SERVICE     ping    8 ;
"router discovery"  SERVICE     icmp    10 10  BROADCAST  REVERSE icmp    9 9
        BROADCAST ;
"rstat"SERVICE     rpc_udp       100001        pmap_udp    100001 ;
"rusers"      SERVICE     rpc_udp       100002        pmap_udp    100002 ;
"nfs prog"    SERVICE     pmap_udp    100003        udp     2049   tcp     2049 ;
```

```
"nfs readonly prog" SERVICE     pmap_tcp     100003       pmap_udp    100003
        nfsro   2049 ;
"ypserv"      SERVICE     nis     100004       pmap_nis     100004       pmap_nis
        100004       BROADCAST ;
"mountd"      SERVICE     rpc_tcp       100005       rpc_udp       100005
        pmap_tcp     100005       pmap_udp    100005 ;
"ypbind"      SERVICE     rpc_udp       100007       pmap_udp    100007 ;
"wall" SERVICE     rpc_udp       100008       pmap_udp    100008 ;
"yppasswd"  SERVICE     rpc_udp       100009       pmap_udp    100009 ;
"rquota"      SERVICE     rpc_tcp       100011       rpc_udp       100011
        pmap_tcp     100011       pmap_udp    100011 ;
"spray"       SERVICE     rpc_udp       100012       pmap_udp    100012 ;
"rex"  SERVICE     rpc_udp       100017       pmap_udp    100017 ;
"nlm" SERVICE     rpc_tcp       100021       rpc_udp       100021       pmap_tcp
        100021       pmap_udp    100021  REVERSE rpc_tcp       100021  REVERSE
        rpc_udp       100021  REVERSE pmap_tcp     100021  REVERSE pmap_udp
        100021 ;
"status"      SERVICE     rpc_tcp       100024       rpc_udp       100024
        pmap_tcp     100024       pmap_udp    100024 ;
"ypupdate"  SERVICE     rpc_udp       100028       pmap_udp    100028 ;
"nfs acl"     SERVICE     rpc_tcp       100227       rpc_udp       100227
        pmap_tcp     100227       pmap_udp    100227 ;
"nfs"  SERVICE_GROUP "mountd"      "nfs prog"     "rquota"       "nlm"   "status"
        "nfs acl";
"nfs readonly"       SERVICE_GROUP "mountd"      "nfs readonly prog"  "rquota"
        "nlm"   "status"       "nfs acl";
"nis"  SERVICE_GROUP "ypserv"      "yppasswd"   "ypupdate"    "ypbind";
"ospf"SERVICE     ip      89      BROADCAST 89 ;
"skip" SERVICE     iptunnel      57 79 ;
"esp" SERVICE     iptunnel      50 ;
"ah"   SERVICE     iptunnel      51 ;
"isakmp"      SERVICE     udp     500 ;
"ipsec"       SERVICE_GROUP "esp"  "ah"    "isakmp";
"ipv6 tunnel" SERVICE     iptunnel      41 ;
"icmp all"    SERVICE     icmp    *   *    BROADCAST ;
"ip all"      SERVICE     ip      *  ;
"ip mobile"   SERVICE     ipmobile      *  ;
"ip tunnel"   SERVICE     iptunnel      *  ;
"ip forward"  SERVICE     ipfwd   *  ;
"udp all"     SERVICE     udpall  *  ;
"tcp all"     SERVICE     tcpall  0 - 3850 3855 - 65535 ;
"rpc all"     SERVICE     rpc_udp       *  ;
"rpc tcp all"  SERVICE     rpc_tcp       *  ;
"pmap udp all"      SERVICE     pmap_udp     *   *    BROADCAST ;
"pmap tcp all"      SERVICE     pmap_tcp     *  ;
```

```
 "common"    SERVICE_GROUP "tcp all"        "udp all"        "syslog"        "dns"  "rpc
all"     "nfs prog"     "icmp all"     "rip"    "ftp"    "rsh"   "real audio"   "pmap udp all"
         "pmap tcp all"        "rpc tcp all"   "nis"   "archie"        "traceroute"   "ping";
 "common services" SERVICE_GROUP "tcp all"        "udp all"        "syslog"        "dns"
         "rpc all"        "nfs prog"     "icmp all"     "rip"    "ftp"    "rsh"   "real audio"   "pmap
udp all"        "pmap tcp all"        "rpc tcp all"   "nis"   "archie"        "traceroute"   "ping";
 "X11" SERVICE     tcp     6000 - 6063 ;
 "pcnfsd"        SERVICE     pmap_tcp     150001        pmap_udp     150001
         rpc_tcp        150001        rpc_udp        150001 ;
 "automount" SERVICE     pmap_tcp     300019        pmap_udp     300019
         rpc_tcp        300019        rpc_udp        300019 ;
 "ypxfrd"        SERVICE     pmap_tcp     100069        pmap_udp     100069
         rpc_tcp        100069        rpc_udp        100069 ;
 "exec"          SERVICE     tcp     512 ;
 "wais"SERVICE     tcp     210 ;
 "uucp"          SERVICE     tcp     540 ;
 "irc"   SERVICE     tcp     6670   tcp     6680 ;
 "VDOLive"   SERVICE     tcp     7000   tcp     7010 REVERSE     udp     32649 ;
 "CU See Me"         SERVICE     udp_datagram        7648 - 7652 ;
 "Vosaic"        SERVICE     tcp     1235  REVERSE     udp_datagram        61801 -
61820  REVERSE   udp_datagram        20000 - 20020 ;
 "StreamWorks"       SERVICE     udp_datagram        1558  REVERSE
         udp_datagram        1558 ;
 "CoolTalk"   SERVICE     tcp     6499 - 6500  udp_datagram        13000  REVERSE
         udp_datagram        13000 ;
 "Backweb"   SERVICE     udp     370    PARAMS     ( 60 0 3  ) ;
 "radius"        SERVICE     udp     1645 ;
 "ssl"   SERVICE     tcp     443 ;
 "who" SERVICE     udp_datagram        513     BROADCAST ;
 "netstat"       SERVICE     tcp     15 ;
 "biff"  SERVICE     udp_datagram        512     BROADCAST ;
 "bootp"         SERVICE     udp     67      BROADCAST        PARAMS     ( 60 0 3  ) ;
 "kerberos"   SERVICE     udp     88 ;
 "ntp-tcp"       SERVICE     tcp     123 ;
 "netbios name"      SERVICE     udp     137 137        BROADCAST ;
 "netbios datagram" SERVICE     udp_datagram        138 138        BROADCAST ;
 "netbios session"   SERVICE     tcp     139 ;
 "netbios"       SERVICE_GROUP "netbios name"        "netbios datagram"  "netbios
session";
 "lpd"   SERVICE     tcp     2766 ;
 "mosaic"        SERVICE_GROUP "www""ssl"    "gopher"        "ftp"    "archie";
 "echo-udp"   SERVICE     udp     7 ;
 "echo group"SERVICE_GROUP "echo""echo-udp";
 "discard-udp"         SERVICE     udp     9 ;
 "discard group"       SERVICE_GROUP "discard"       "discard-udp";
 "time-udp"    SERVICE     udp     37 ;
```

```
"time group" SERVICE_GROUP "time" "time-udp";
"daytime-udp"       SERVICE     udp     13 ;
"daytime group"     SERVICE_GROUP "daytime"     "daytime-udp";
"tcp-high-ports"    SERVICE     tcp     1024 - 65535 ;
"udp-high-ports"    SERVICE     udp     1024 - 65535 ;
"http" SERVICE      tcp     80 0 ;
"https"         SERVICE     tcp     443     SCREEN      "fw1";
"ssh" SERVICE       tcp     22 0    SCREEN      "fw1";


//Addresses
"fw1_eri0"      ADDRESS     "*";
"SshSrv"        ADDRESS     223.223.223.40      COMMENT     "";
"ExtDns1"       ADDRESS     223.223.223.65      COMMENT     "External DNS Prim";
"ExtDnsS"       ADDRESS     223.223.223.66      COMMENT     "";
"MailGate"      ADDRESS     223.223.223.70      COMMENT     "Mail Gateway";
"EmuMail"       ADDRESS     223.223.223.71      COMMENT     "";
"ExtProxy"      ADDRESS     223.223.223.68      COMMENT     "";
"ExtWeb"        ADDRESS     223.223.223.69      COMMENT     "";
"ExtFtp"        ADDRESS     223.223.223.67      COMMENT     "";
"VpnSrv"        ADDRESS     223.223.223.41      COMMENT     "";
"NatGiac"       ADDRESS     223.223.223.50      COMMENT     "";
"IntDns1"       ADDRESS     172.17.1.103        COMMENT     "Internal DNS Prim";
"IntDns2"       ADDRESS     172.17.1.103        COMMENT     "";
"ExtE500"       ADDRESS     223.223.223.34      COMMENT     "External interface E500";
"IntE500"       ADDRESS     172.20.1.1      COMMENT     "";
"EOSN"      ADDRESS     "EmuMail"       "ExtDns1"       "ExtDnsS"       "ExtFtp"
        "ExtProxy"      "ExtWeb"        "MailGate"      COMMENT     "";
"ERSN"      ADDRESS     "SshSrv"        "VpnSrv"        COMMENT     "";
"SwiExt"        ADDRESS     172.20.1.2      COMMENT     "External switch interface";
"DbBus"         ADDRESS     172.17.1.100        COMMENT     "Oracle Business Server";
"OasSrv"        ADDRESS     172.17.1.11     COMMENT     "";
"MailSrv"       ADDRESS     172.17.1.13     COMMENT     "";
"IntProxy"      ADDRESS     172.17.1.14     COMMENT     "";
"IntWebSrv" ADDRESS     172.17.1.15     COMMENT     "";
"DevWebSrv"         ADDRESS     172.17.1.241        COMMENT     "";
"   DevOas"     ADDRESS     172.17.1.243        COMMENT     "";
"DevDbSrv"      ADDRESS     172.17.1.242        COMMENT     "";
"DbForSrv"      ADDRESS     172.17.1.200        COMMENT     "";
"SyslogSrv"     ADDRESS     172.17.1.231        COMMENT     "";
"AlarmSrv"      ADDRESS     172.17.1.232        COMMENT     "";
"NidsAdmSrv"        ADDRESS     172.17.1.221        COMMENT     "";
"UserMangNet"       ADDRESS     172.16.2.10 - 172.16.2.30   COMMENT     "";
"UserMarkNet"       ADDRESS     172.16.3.10 - 172.16.3.30   COMMENT     "";
"UserRevNet"        ADDRESS     172.16.5.10 - 172.16.5.30   COMMENT     "";
"UserPayNet"        ADDRESS     172.16.6.10 - 172.16.6.30   COMMENT     "";
```

```
"UserSaleNet"       ADDRESS    172.16.7.10 - 172.16.7.30  COMMENT   "";
"UserQcNet"ADDRESS    172.16.4.10 - 172.16.4.30  COMMENT   "";
"UserLanGrp"        ADDRESS    "UserMangNet"      "UserMarkNet"
        "UserPayNet"        "UserQcNet" "UserRevNet"        "UserSaleNet"
        COMMENT   "";
"UserItNet"   ADDRESS    172.16.1.10 - 172.16.1.40  COMMENT   "";
"UserLanNet"        ADDRESS    "UserItNet"   "UserLanGrp"        "UserMangNet"
        "UserMarkNet"       "UserPayNet"        "UserQcNet" "UserRevNet"
        "UserSaleNet"       COMMENT   "";
"DevGrp"     ADDRESS    "DevDbSrv"   "  DevOas"   "DevWebSrv"        COMMENT
        "";
"SyslogGrp" ADDRESS    "AlarmSrv"    "SyslogSrv"   COMMENT   "";
"DeskTopAppSrv"  ADDRESS    172.16.8.11  COMMENT   "";
"DeskTopDataSrv" ADDRESS    172.16.8.12  COMMENT   "";
"DeskTopOsSrv"    ADDRESS    172.16.8.10  COMMENT   "";
"DeskTopGrp"        ADDRESS    "DeskTopAppSrv"   "DeskTopDataSrv"
        "DeskTopOsSrv"     COMMENT   "";
"losn" ADDRESS    "IntProxy"     "IntWebSrv"  "MailSrv"      "OasSrv"      COMMENT
        "";
"lrsn" ADDRESS    "IntDns1"      "IntDns2"      "DbBus"       COMMENT   "";
"SrvGrp"      ADDRESS    "DeskTopGrp"        "DevGrp"      "DbForSrv"    "losn"  "lrsn"
        "SyslogGrp"  COMMENT   "";
"Inside"        ADDRESS    172.16.1.1 - 172.17.1.255  COMMENT   "GIAC internal
networks";
"Outside"     ADDRESS    "*"      !"EOSN"      !"ERSN"      !"inside.qfe3" COMMENT
        "";
"ExtDnsGrp" ADDRESS    "ExtDns1"     "ExtDnsS"     COMMENT   "";
"IntDnsGrp"  ADDRESS    "IntDns1"      "IntDns2"      COMMENT   "";
"IntNtp1"      ADDRESS    172.17.1.103          COMMENT   "";
"IntNtp2"      ADDRESS    172.17.1.104          COMMENT   "";
"IntNtpGrp"  ADDRESS    "IntNtp1"      "IntNtp2"      COMMENT   "";
"DnsIsp"       ADDRESS    1.2.3.4          COMMENT   "ISPs DNS server";
"NtpClocks"  ADDRESS    1.2.3.4 - 1.2.3.8      COMMENT   "External NTP servers";
"EsnGrp"     ADDRESS    "EOSN"        "ERSN"        COMMENT   "External NTP
servers";
"BR"  ADDRESS    223.223.233.33      COMMENT   "Border Router";
"ColAgent"   ADDRESS    1.2.3.10       COMMENT   "Collection Agency";
"IpBlock"     ADDRESS    1.2.3.99       COMMENT   "Blocked IPs";
"TestSystem"        ADDRESS    223.223.223.72      COMMENT   "";
"inside.qfe3"ADDRESS    "ExtE500"     "Inside"        COMMENT   "";


//Screens
"fw1" SCREEN ADMIN_IP 172.16.1.250 CERTIFICATE "fw1.admin"
CERTIFICATE_DISCOVERY ROUTING SPF 223.223.223.0 255.255.255.0 COMMENT
"Exterior Firewall";
```

```
//Interfaces
"eri0"  INTERFACE ADMIN        "fw1_eri0"                    ;
 "qfe1"INTERFACE SPF   "EOSN"                                 SCREEN     "fw1";
 "qfe2"INTERFACE SPF   "ERSN"                                 SCREEN     "fw1";
 "qfe0"INTERFACE SPF   "Outside"                              SCREEN     "fw1";
 "qfe3"INTERFACE SPF   "inside.qfe3"                          SCREEN     "fw1";


//Certificates
"fw1.admin"  CERTIFICATE        NSID 8 MKID
"0x22120caf4294526cef6d3562f307516b";
 "remote"      CERTIFICATE        NSID 8 MKID "0x";
 "admin-group"        CERTIFICATE        "remote";


//Time objects

//Data gathered from ezdb databases
@db vars { name:ascii="LogSeverity"; value:ascii="INFO"; description:ascii="global log
severity limit"; enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="log"; name:ascii="LogPort"; value:ascii="3853";
description:ascii="global log daemon UDP port"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="log"; name:ascii="LogSize"; value:ascii="100";
description:ascii="global log capacity (MB)"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="auth"; name:ascii="userImmediateDomain"; value:ascii="default";
description:ascii="map user from immediate to domain"; _value_default:ascii="default";
enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="auth"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, authentication"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSHolddown"; value:ascii="300";
description:ascii="seconds to ignore a non-responsive RADIUS server";
enabled:unsigned32=1; tstamp:utime=1048470073; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSNodeSecret"; enabled:unsigned32=0;
value:ascii="....-....1....-....2....-....3.."; description:ascii="shared secret for (this) RADIUS
client"; tstamp:utime=1048470073; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSRetryPasses"; value:ascii="3";
description:ascii="how many times to try each RADIUS server"; enabled:unsigned32=1;
tstamp:utime=1048470073; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSServers"; enabled:unsigned32=0;
values:struct={ host:ascii="server1"; host:ascii="server2"; host:ascii="1.2.3.4"; };
```

description:ascii="RADIUS server name(s) / address(es) to query";
tstamp:utime=1048470073; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSService"; value:ascii="radius";
description:ascii="RADIUS service / port # at which to query server(s)";
enabled:unsigned32=1; tstamp:utime=1048470073; };
@db vars { prg:ascii="auth"; name:ascii="RADIUSTimeout"; value:ascii="5";
description:ascii="seconds to await each RADIUS server response";
enabled:unsigned32=1; tstamp:utime=1048470073; };
@db vars { prg:ascii="edit"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, editor"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="ftpp"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, FTP proxy"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="ftpp"; name:ascii="N_Sessions"; value:ascii="100";
description:ascii="limit on # of concurrent proxy sessions"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, HTTP proxy"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="N_Sessions"; value:ascii="333";
description:ascii="limit on # of concurrent proxy sessions"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="TargetSvcs"; values:struct={ svc:ascii="www";
}; description:ascii="restriction on TCP port #s URIs can target"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="JavaExtensions"; values:struct={
suffix:ascii=".jar"; }; description:ascii="Java extensions, HTTP proxy - *reserved*";
enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="JavaMagicExtensions"; values:struct={
suffix:ascii=".class"; }; description:ascii="Java *magic* extensions, HTTP proxy -
reserved"; enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="JavaMagicMIMETypes"; values:struct={
name:ascii="application/octet-stream"; }; description:ascii="Java *magic* MIME types,
HTTP proxy - *reserved*"; enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="JavaMIMETypes"; values:struct={
prefix:ascii="application/x-java-"; }; description:ascii="Java MIME types, HTTP proxy -
reserved"; enabled:unsigned32=1; tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="OtherXExtensions"; enabled:unsigned32=0;
values:struct={ suffix:ascii=".cab"; suffix:ascii=".dll"; suffix:ascii=".js"; suffix:ascii=".scf";
suffix:ascii=".vbr"; suffix:ascii=".vbs"; }; description:ascii="Other-X extensions, HTTP
proxy - *reserved*"; tstamp:utime=1048470072; };
@db vars { prg:ascii="httpp"; name:ascii="OtherXMIMETypes"; values:struct={
name:ascii="application/octet-stream"; prefix:ascii="application/ole";
prefix:ascii="application/scriptlet"; name:ascii="application/x-msdownload";
prefix:ascii="application/x-ole"; prefix:ascii="application/x-scriptlet"; };

```
description:ascii="Other-X MIME types, HTTP proxy - *reserved*";
enabled:unsigned32=1; tstamp:utime=1048470073; };
@db vars { prg:ascii="httpp"; name:ascii="RspBodySave"; value:ascii="0";
description:ascii="response body save silo (MB), HTTP proxy"; enabled:unsigned32=1;
tstamp:utime=1048470073; };
@db vars { prg:ascii="httpp"; name:ascii="FtpPwdDomain"; enabled:unsigned32=0;
value:ascii="UNCONFIGURED-DOMAIN"; description:ascii="default domain in ftp
method anonymous password"; tstamp:utime=1048470073; };
@db vars { prg:ascii="httpp"; name:ascii="FtpPwdUser"; value:ascii="anonymous";
description:ascii="default user in ftp method anonymous password";
enabled:unsigned32=1; tstamp:utime=1048470073; };
@db vars { prg:ascii="httpp"; name:ascii="FtpUser"; value:ascii="webproxy";
description:ascii="default user in ftp method"; enabled:unsigned32=1;
tstamp:utime=1048470073; };
@db vars { prg:ascii="smtpp"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, SMTP proxy"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db vars { prg:ascii="telnetp"; name:ascii="LogSeverity"; value:ascii="INFO";
description:ascii="global log severity limit, Telnet proxy"; enabled:unsigned32=1;
tstamp:utime=1048470072; };
@db authuser { name:ascii="admin"; auth_method:struct={ type:ascii="password";
crypt_password:ascii="F0q.NqNcf.6t6"; enabled:unsigned32=1; };
enabled:unsigned32=1; };
@db authuser { name:ascii="admin1"; auth_method:struct={ type:ascii="password";
crypt_password:ascii="3yieqKLei7J9k"; enabled:unsigned32=1; };
enabled:unsigned32=1; };
@db proxyuser { name:ascii="admin"; auth_user_name:ascii="admin";
description:ascii="initial SunScreen administrator"; type:ascii="simple";
enabled:unsigned32=1; };
@db proxyuser { name:ascii="admin-group"; type:ascii="group";
description:ascii="SunScreen administrators"; enabled:unsigned32=1;
member_name:ascii="admin"; };
@db proxyuser { name:ascii="anonymous"; backend_user_name:ascii="anonymous";
description:ascii="unauthenticated user, for anonymous FTP, etc."; type:ascii="simple";
enabled:unsigned32=1; };
@db proxyuser { name:ascii="ftp"; backend_user_name:ascii="anonymous";
description:ascii="unauthenticated user, for anonymous FTP, etc."; type:ascii="simple";
enabled:unsigned32=1; };
@db proxyuser { name:ascii="radius"; ext_auth_type:ascii="radius";
description:ascii="default, external, non-specific RADIUS proxy_user";
type:ascii="simple"; enabled:unsigned32=1; };
@db proxyuser { name:ascii="securid"; ext_auth_type:ascii="securid";
description:ascii="default, external, non-specific SecurID proxy_user";
type:ascii="simple"; enabled:unsigned32=1; };
//SunScreenConfig 3.0
//*********Do NOT edit this file.********//
```

```
//**It is generated by SunScreen tools.**//


//Rules
RULE "ip all"  "Inside"  "IpBlock"  DENY LOG SUMMARY     SCREEN     "fw1";
 RULE"dns"  "Outside"  "ExtDnsGrp"  ALLOW   SCREEN     "fw1";
 RULE"dns"  "IntDnsGrp"  "Outside"  ALLOW   SCREEN     "fw1";
 RULE"dns"  "MailGate"  "DnsIsp"  ALLOW     SCREEN     "fw1";
 RULE"dns"  "ExtFtp"  "DnsIsp"  ALLOWSCREEN     "fw1";
 RULE"dns"  "EmuMail"  "DnsIsp"  ALLOW     SCREEN     "fw1";
 RULE"http"  "Outside"  "ExtProxy"  ALLOW     SCREEN     "fw1";
 RULE"https"  "Outside"  "ExtProxy"  ALLOW   SCREEN     "fw1";
 RULE"http"  "UserLanNet"  "ExtProxy"  ALLOW       SCREEN     "fw1";
 RULE"https"  "UserLanNet"  "ExtProxy"  ALLOW      SCREEN     "fw1";
 RULE"http"  "IntProxy"  "Outside"  ALLOW     SCREEN     "fw1";
 RULE"https"  "IntProxy"  "Outside"  ALLOW    SCREEN     "fw1";
 RULE"http"  "NidsAdmSrv"  "Outside"  ALLOWSCREEN     "fw1";
 RULE"ssh"  "Outside"  "SshSrv"  ALLOW       SCREEN     "fw1";
 RULE"ssh"  "DbForSrv"  "SshSrv"  ALLOW     SCREEN     "fw1";
 RULE"ssh"  "VpnSrv"  "EOSN"  ALLOWSCREEN     "fw1";
 RULE"ssh"  "VpnSrv"  "SrvGrp"  ALLOW       SCREEN     "fw1";
 RULE"ssh"  "UserItNet"  "SrvGrp"  ALLOW    SCREEN     "fw1";
 RULE"ssh"  "UserItNet"  "EsnGrp"  ALLOW    SCREEN     "fw1";
 RULE"ssh"  "UserItNet"  "Outside"  ALLOW    SCREEN     "fw1";
 RULE"ntp"  "IntNtpGrp"  "NtpClocks"  ALLOW SCREEN     "fw1";
 RULE"ntp"  "EsnGrp"  "IntNtpGrp"  ALLOW     SCREEN     "fw1";
 RULE"ntp"  "BR"  "IntNtpGrp"  ALLOW  SCREEN     "fw1";
 RULE"syslog"  "BR"  "SyslogSrv"  ALLOW     SCREEN     "fw1";
 RULE"syslog"  "EsnGrp"  "SyslogSrv"  ALLOW       SCREEN     "fw1";
 RULE"smtp"  "Outside"  "MailGate"  ALLOW   SCREEN     "fw1";
 RULE"smtp"  "MailGate"  "ExtE500"  ALLOW   SCREEN     "fw1";
 RULE"smtp"  "ExtE500"  "MailGate"  ALLOW   SCREEN     "fw1";
 RULE"smtp"  "MailGate"  "Outside"  ALLOW   SCREEN     "fw1";
 RULE"ssl"  "Outside"  "EmuMail"  ALLOW       SCREEN     "fw1";
 RULE"pop"  "EmuMail"  "ExtE500"  ALLOW     SCREEN     "fw1";
 RULE"isakmp"  "Outside"  "VpnSrv"  ALLOW   SCREEN     "fw1";
 RULE"esp"  "Outside"  "VpnSrv"  ALLOW       SCREEN     "fw1";
 RULE"isakmp"  "ExtWeb"  "ColAgent"  ALLOW       SCREEN     "fw1";
 RULE"esp"  "ExtWeb"  "ColAgent"  ALLOW     SCREEN     "fw1";
 RULE"ping"  "UserItNet"  "Outside"  ALLOW   SCREEN     "fw1";
 RULE"telnet"  "UserItNet"  "Outside"  ALLOW  SCREEN     "fw1";
 RULE"icmp unreach"  "Outside"  "Inside"  ALLOW     SCREEN     "fw1";
 RULE"icmp all"  "Outside"  "Inside"  DENY LOG SUMMARY SCREEN     "fw1";
 RULE"*"  "*"  "Inside"  DENY LOG SUMMARY SCREEN     "fw1";
```

```
//Nat
NAT   DYNAMIC "Inside" "Outside" "NatGiac" "Outside"      SCREEN     "fw1";
 NAT  DYNAMIC "Outside" "NatGiac" "Outside" "Inside"      SCREEN     "fw1";


//Local Access
ACCESS "admin"    ALL;


//Remote Access
ACCESS "admin"    ALL "*"         SKIP_VERSION_2 "admin-group" "DES-CBC" "DES-
CBC" "MD5" "NONE";


//VPN Nodes


@origin fw1 default v1.28
```

BIBLIOGRAPHY

PRINTED MATERIALS:

Akin, Thomas, Hardening Cisco Routers, North Sebastopol, CA: O'Reilly (2002)

Brenton, Chris and Abuhoff, Bob, Mastering Cisco Routers 2nd Ed., Alameda, CA: SYBEX (2000)

Doyle, Jeff and Carroll, Jennifer, Routing TCP/IP Vol. II, Indianapolis, ID: CISCO Press (2001)

Hucaby, David and McQuerry, Steve, CISCO Field Manual: Router Configuration, Indianapolis, ID: CISCO Press (2002)

The SANS Institute, Firewalls 101: Perimeter Protection with Firewalls, SANS Firewalls Track 2.2, Bethesda, MD: SANS Press (2002)

The SANS Institute, Firewalls 102: Perimeter Protection Defense In-Depth, SANS Firewalls Track 2.3, Bethesda, MD: SANS Press (2002)

The SANS Institute, Network Design and Performance, SANS Firewalls Track 2.5, Bethesda, MD: SANS Press (2002)

The SANS Institute, Securing Cisco Routers Step-by-Step, Version 1.0 (August 2002)

The SANS Institute, Solaris Security Step by Step version 2.0 (2001)

The SANS Institute, TCP/IP for Firewalls, SANS Firewalls Track 2.1, Bethesda, MD: SANS Press (2002)

The SANS Institute, VPNs and Remote Access, SANS Firewalls Track 2.4, Bethesda, MD: SANS Press (2002)

The SANS Institute, Windows NT Security Step by Step version 3.03 (February, 2001)

The SANS Institute, TCP/IP for Firewalls, SANS Firewalls Track 2.1, Bethesda, MD: SANS Press, (2002)

The SANS Institute, Firewalls 101: Perimeter Protection with Firewalls, SANS Firewalls Track 2.2, Bethesda, MD: SANS Press, (2002)

The SANS Institute, Firewalls 102: Perimeter Protection Defense In-Depth, SANS Firewalls Track 2.3, Bethesda, MD: SANS Press, (2002)

The SANS Institute, VPNs and Remote Access, SANS Firewalls Track 2.4, Bethesda, MD: SANS Press, (2002)

The SANS Institute, Network Design and Performance, SANS Firewalls Track 2.5, Bethesda, MD: SANS Press, (2002)

INTERNET REFERENCES and SOURCES:

Barlow, Jason. Thrower, Woody "TFN2K - An Analysis" 10 February 2000
URL: http://www.ussrback.com/docs/distributed/TFN2k_Analysis-1.3.txt (March 7, 2003)

Bindview-Razor Team. "Strategies for Defeating Distributed Attacks"
URL: http://razor.bindview.com/publish/papers/strategies.html (February 2003)

CERT. "CERT® Advisory CA-1999-17 Denial-of-Service Tools" 3 March 2000
URL: http://www.cert.org/advisories/CA-1999-17.html (March 7, 2003)

CERT. "CERT®Advisory CA-1999-17 Denial-of-Service Tools" (March 3, 2000)
URL: http://www.cert.org/advisories/CA-1999-17.html (March 3, 2003)

CERT. "Vulnerability Note VU#398025"
URL: http://www.kb.cert.org/vuls/is/398025 (March 6, 2003)

CERT. "CERT Advisory CA-2003-07 Remote Buffer Overflow in Sendmail" (March 4, 2003)
URL: http://www.cert.org/advisories/CA-2003-07.html (March 6, 2003)

Chadd, Adrian. "Squid 2.4 Stable1 Configuration Manual"
URL: http://squid.visolve.com/squid24s1/contents.htm (February 2003)

Cisco Corp. "Cisco 7606 Router"
URL: http://www.cisco.com/en/US/products/hw/routers/ps368/ps371/index.html (February 2003)

Cisco Corp. "Cisco Catalyst 6506 Switch"
URL: http://www.cisco.com/en/US/products/hw/switches/ps708/ps710/index.html (March 2003)

Cisco Corp. "Cisco IOS Command References Master Index" 12 October 2001
URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122mindx/crfindx.htm (February 2003)

Cisco Systems Product Security Incident Response Team.

URL: http://archives.neohapsis.com/archives/cisco/2002-q2/0017.html (March 22, 2003)

Cisco Systems. “CISCO 6500 Catalyst Series Switches”
URL: http://www.cisco.com/en/US/products/hw/switches/ps708/index.html (March 2003)

Cisco Systems.
URL: http://www.cisco/go/psirt (March 2003)

Dept. of Homeland Security. “Remote Sendmail Header Processing Vulnerability” (March 3, 2003)
URL: http://nipc.gov/warnings/advisories/2003/03-004.htm (March 6, 2003)

FreeS/WAN Project Team. “FreeS/Wan Documentation”
URL: http://www.freeswan,org/freeswan_trees/freeswan-1.95/doc/index.html (February 2000

Fyodor. “Remote OS detection via TCP/IP Stack FingerPrinting” 10 April 1999
URL: http://www.nmap.org/nmap/nmap-fingerprinting-article.html (February 2003)

Fyodor. “Nmap Stealth Port Scanner” 1 April 2002
URL: http://www.nmap.org (February 2003)

IANA. “Internet Protocol V4 Address Space”
URL: http://www.iana.org/assignments/ipv4-address-space (February 2003)

Institute for Security Technology Studies. “Jeanne: Reverse Proxy Server” 2001
URL: http://www.ists.dartmouth.edu/IRIA/projects/jeanne.htm (February 2003)

Internet Software Consortium. “Berkeley Internet Name Domain” 2 February 2002
URL: http://www.isc.org/products/BIND/ (February 2003)

“iPass Corporate Access”
URL: http://www.ipass.com/services/services_corpaccess.html (March 4, 2003)

ISS X-Force. “Snort RPC Preprocessing Vulnerability” (March 3, 2003)
URL: http://www.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21951 (March 6, 2003)

Keeney, Frank. “Screening Router Access List” 30 December 1998
URL: http://pasadena.net/cisco/secure.html (February 2003)

Lasser, Jon, Beale, Jay. “Bastille Linux Hardening System” (February 2003)

URL: http://www.bastille-linux.org/ (February 2003)

Microsoft Corp. “Microsoft Solution for Securing Windows 2000 Server”
URL:
http://www.microsoft.com/technet/security/prodtech/Windows/SecWin2k/Default.asp (March 3, 2003)

MITRE Corp. “Common Vulnerabilities and Exposures-Search Page” 2002
URL: http://www.cve.mitre.org/ (February 2003)

National Security Agency, United State of America. “Router Security Configuration Guide” (February 2003).
URL: http://www.nsa.gov/snac/cisco/guides/cis-2.pdf (March 2003)

Oracle Corp. “General Notes for Oracle9i Application Server Enterprise Edition”
URL: http://metalink.oracle.com/metalink/plsql/cert_views.platform_selection
(February 2003)

Red Hat. “An Overview of Red Hat Advanced Server V2.1 Reliability, Availability, Scalability and Manageability (RASM) Features” (2002)
URL: http://www.redhat.com/pdf/as/as_rasm.pdf (March 2003)

Red Hat. “Linux.Slapper.Worm—What Red Hat customers can do about it”
URL: http://www.redhat.com/support/alerts/linux_slapper worm.html

Red Hat. “Red Hat Linux Advanced Server 2.1”
URL: http://www.redhat.com/advancedserver/

Red Hat. “Stronghold Enterprise Apache Solutions”
URL: http://www.redhat.com/software/stronghold/index.html

Reed, Nicj. “Practical Assignment v 1.7” (January 2002)
URL: http://www.giac.org/practical/ (March 10, 2003)

Sanfilippo, Salvatore “hping”
URL: http://www.hping.org/ (February 2003)

SANS Institute. “SANS/FBI Top 20 List, Version 3.21” (October 17, 2002)
URL: http://www.cisecurity.org/scanning_tool.html (February 2003)

SANS Institute. “Commonly Probed Ports” May 4, 2000
URL: http://www.sans.org/y2k/ports.htm (February 2003)

Security Focus Corp. 2002
URL: http://www.securityfocus.com (February 2003)

Security Focus. "CISCO PIX Firewall Telnet/SSH Subnet Handling Denial of Service Vulnerability" (November, 11, 2002)
URL: http://www.securityfocus.org/bid/6110 (March 7, 2003)

Security Focus. "Technical analysis of the remote sendmail vulnerability" (March 4, 2003)
URL: http://www.securityfocus.org/archive/1/31757/2003-03-01/2003-03-07/0 (March 6, 2003)

Silicon Defense.
URL:http://www.silicondefense.com (February 2003)

Snort. "Snort: The Open Source Network Intrusion Detection System"
URL:http://www.snort.org (February 2003)

Snort. "SNORT FAQ Version 1.8 – July 10 2001 v1.8.1"
URL: http://snort.sourcefire.com/docs/faq.html (March 3, 2003)

Spitzner, Lance. "Armoring Linux"
URL: http://www.enteract.com/~lspitz/papers.html (February 2003)

Squid. "Squid Web Proxy Cache"
URL: http://www.squid-cache.org/ (March 5, 2003)

"SSH Secure Shell for Servers: User Restrictions"
URL: http://www.ssh.com/support/faq/secureshellserver/qa_191_687.html

Steamballoon. "Open Source Solutions for a Competitive World"
URL: http://www.steamballoon.com/

The SANS Institute. "Mistakes People Make that Lead to Security Breaches" (October 23, 2001)
URL: http://www.sans.org/resources/mistakes.php (March 10, 2003)

The SANS Institute. "Sendmail Exploit"
URL: http://wiki.sans.org/tiki-index.php?page=SendmailExploit (March 3, 2003)

Trendmicro. "InterScan AppIertTrap" (2003)
URL: http://www.trendmicro.com.en.products/gateway/evaluate/overview.html (February 2003)

Trendmicro. "InterScan Messaging Security Suite-Features" (2003)
URL: http://www.trendmicro.com.en.products/gateway/ismss/evaluate/features (February 2003)

Trendmicro. "InterScan ServerProtect to Linux" (2003)

URL: http://www.trendmicro.com.en.products/file-server/sp-linux/evaluate (February 2003)

Trendmicro. "Policy-based Antivirus and Content Security for the Messaging Gateway" (June 2002)
URL: http://www.trendmicro.com/ (February 2003)

Tripwire, Inc. "Tripwire Open Source Project"
URL: http://www.tripwire.org (February 2003)

University of Delaware. "Time Synchronization Server"
URL: http://www.eecis.udel.edu/~ntp (February 2003)

Winters, Scott. "Securing the Perimeter with Cisco IOS 12 Routers"
URL: http://rr.sans.org/firewall/blocking_cisco.php (February 2003)

"World Wide Internet Access"
URL: http://www.net-roamer.com/index.htm (March 5, 2003)

http://www.chkrootkit.org (March 2003)

ftp.wu-ftpd.org.

http://eudora.com

http://freshmeat.net/projects/snortsnarf/?topic_id=245%2C43

http://ralphb.net/IPSubnet/restr.html/

http://www.blackhat.com/presentations/bh-usa-01/JayBeale/23

http://www.eecis.udel.edu/

http://www.emumail.com/

http://www.eweek.com/print_article/0,3668,a=34326,00.asp

http://www.integrityns.com/

http://www.isc.org

http://www.L0pht.com/advisories/L0pht-watch.tar.gz

http://www.lids.org/

http://www.mcafee.com/myapps/mss/default.asp

http://naughty.monkey.org/~dugsong/dsniff/

http://www.oit.ucsb.edu/~eta/swatch/swatch-3.0.4.tar.gz

http://www.openwall.com/john/

http://www.sun.com/solutions/blueprints/browsesubject.html#jumpstart

http://www.wu-ftpd.org/

http://www.yassp.org/