GIAC Training & Certification

Firewalls, Perimeter protection and VPNs

GCFW Practical

Sans at Darling Harbour 2001

Phil Hale

February 2001

Table of Contents

Assignment 1 - Security Architecture (25 Points)

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

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).*

Assumptions:

- The access to GIAC's network by the Suppliers and the Partners is under contract, which includes a standard of network security as specified by GIAC Enterprises.
- Due to the volume of expected sales, this site requires high availability and minimal down time for maintenance.
- No limits have been placed on the implementation of the security architecture.
- Operating Systems will be hardened in accordance with this article.

Security Architecture.

The main facet in designing security architecture is balancing the network security, the business requirements and functionality. Indeed, controlling the access to the network while delivering high security at the highest network performance is no mean task. In fact it is the biggest and most complex task undertaken by IT Managers.

The size of the network would not allow for the network administrator to maintain both the network systems and the security issues involved with this site. Therefore a security team would be formed to maintain, support and develop the security architecture. The responsibility of the security team includes the maintenance of the security logs, patching of operating systems, security devices and other security equipment as required. The security team will maintain regular contact with email lists, such as bugtraq, SANS and other security mailing lists.

During the design phase we need to ask some questions, to accurately define the security policies, performance issues and the architecture to support the business requirements.

1. What services are accessed, from where and by whom?
2. How sensitive is the information on line?
3. Is the implementation going to affect performance?
4. Can the solution grow as the company grows?
5. Does it provide flexibility for the company's needs and address the security requirements across all sections of the business?

Service Access requirements

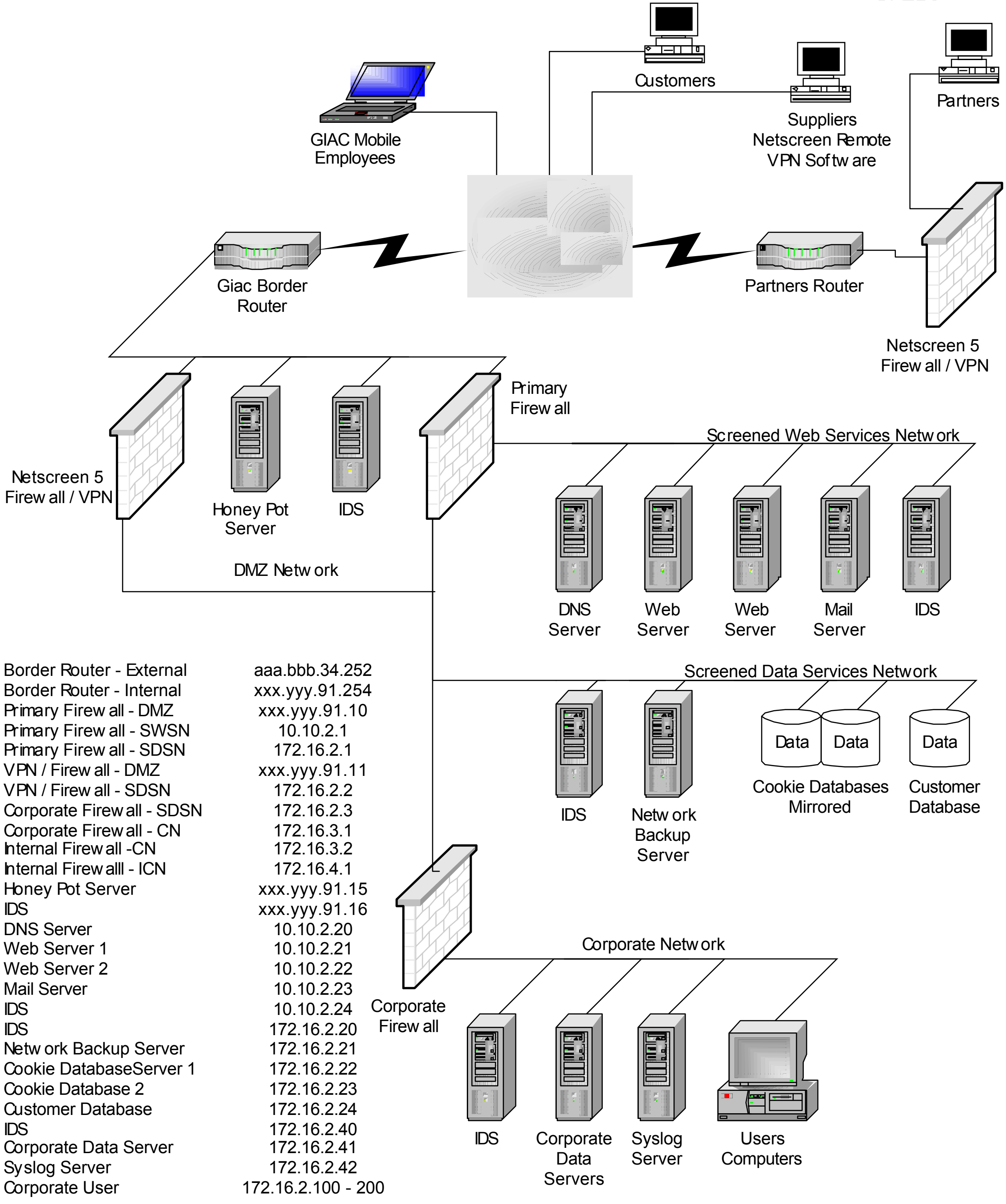
There are five main groups who require access, these being Customers, Suppliers, Partners, Mobile Users and the Internal Staff.

- ***Customers*** – Require access to our service via the internet using web based browsers supporting http and https protocols (http tcp port 80 and https (SSL) tcp port 443).
- ***Suppliers*** – Require access to the Screened Data Services Network via VPN Remote software to the cookie database.
- ***Partners*** - Require access to the Screened Data Services Network via Fixed VPN hardware devices to the cookie databases.
- ***Mobile Users*** – Require access to the Corporate Network via VPN Remote software, the corporate network resources accessed through static mappings.
- ***Internal Staff*** – Require access to the and the corporate network resources.

The critical resources of GIAC Enterprises include, the cookies database and the customer database. This one fact has determined the need to separate these servers in the Screened Data Services Network from the Web Servers in the Screened Web Services Network area via the primary firewall. Network Intrusion Detection Systems are implemented in each area to detect and notify the security staff of any potential threats.

Due to the high volumes of transactions required to maintain an estimated income of $200 million per year, the internet link installed is a E1 2 Mb Fastway DDS service. The design requires high availability with minimal down time. Therefore, both the web servers and the databases are required to allow for regular maintenance to be undertaken and to maintain the speed requirement.

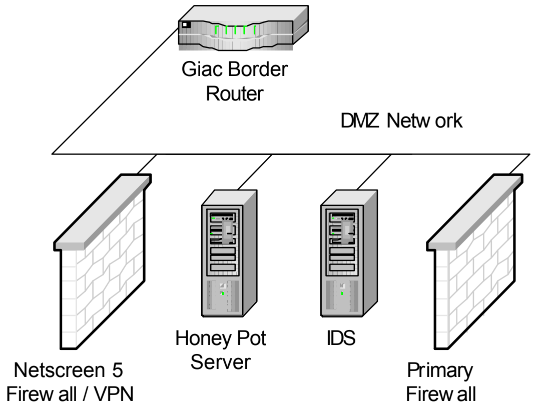
This security architecture is based on the answers to these questions. The architecture is of the defence by depth type, which caters for a possible security breach in one area of the network not affecting another area of the network. This is provided through the use of different firewalls to separate the different areas with in the design. See Figure1.



Border Router - External	aaa.bbb.34.252
Border Router - Internal	xxx.yyy.91.254
Primary Firewall - DMZ	xxx.yyy.91.10
Primary Firewall - SWSN	10.10.2.1
Primary Firewall - SDSN	172.16.2.1
VPN / Firewall - DMZ	xxx.yyy.91.11
VPN / Firewall - SDSN	172.16.2.2
Corporate Firewall - SDSN	172.16.2.3
Corporate Firewall - CN	172.16.3.1
Internal Firewall -CN	172.16.3.2
Internal Firewalll - ICN	172.16.4.1
Honey Pot Server	xxx.yyy.91.15
IDS	xxx.yyy.91.16
DNS Server	10.10.2.20
Web Server 1	10.10.2.21
Web Server 2	10.10.2.22
Mail Server	10.10.2.23
IDS	10.10.2.24
IDS	172.16.2.20
Network Backup Server	172.16.2.21
Cookie DatabaseServer 1	172.16.2.22
Cookie Database 2	172.16.2.23
Customer Database	172.16.2.24
IDS	172.16.2.40
Corporate Data Server	172.16.2.41
Syslog Server	172.16.2.42
Corporate User	172.16.2.100 - 200

Figure 1.

The security architecture is broken into five major components, these being the DMZ, Screened Web Services Network, Screened Data Services Network, Corporate LAN and the Internal Corporate LAN. The firewalls will use the policy of allow known and deny everything else.



Border Router - External	aaa.bbb.90.252
Border Router - Internal	xxx.yyy.91.254
Primary Firewall - DMZ	xxx.yyy.91.10
Primary Firewall - SWSN	10.10.2.1
Primary Firewall - SDSN	172.16.2.1
VPN / Firewall - DMZ	xxx.yyy.91.11
VPN / Firewall - SDSN	172.16.2.2
Honey Pot Server	xxx.yyy.91.15
IDS	xxx.yyy.91.16

The DMZ

The DMZ consists of a Cisco 4000 border router, used to filter the noise from the internet. Although the primary task of the border router is to route packets, most routers have packet filtering capabilities. It is generally accepted that routers should route and firewalls should filter the packets.

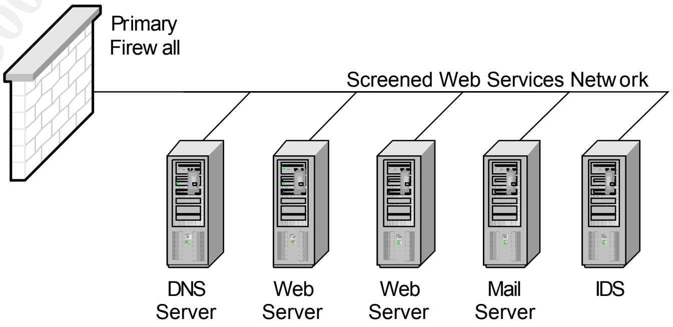
Having said this, the border router is configured to protect from IP spoofing, Source routing and ICMP Redirects. Netbios and Egress filtering is employed to ensure that all packets leaving the network comply with the legal IP network space allocated to GIAC Enterprises making us good internet neighbours.

An attacker will use scanning software to gather details of the network and to make an inventory of the system types. The IDS is used to detect, record any potential threats and watches for abnormal traffic patterns on the wire. The IDS is then able to email or page the security team member on call.

The Honey Pot server acts as a distraction for any would be hackers and allows for their work to be logged to the IDS and then analysed by the security staff to access any possible vulnerabilities.

Screened Web Services Network

The protected web services network employs the Linux IP Tables Firewall at the front end. The web services network is connected to the second interface of the firewall, which I have called the Screened Web Services Network. The first interface on the firewall is connected to the Border router via the hub. The third interface is connected to the Screened Data Services Network.



Primary Firewall - DMZ	xxx.yyy.91.10
Primary Firewall - SWSN	10.10.2.1
Primary Firewall - SDSN	172.16.2.1
DNS Server	10.10.2.20
Web Server 1	10.10.2.21
Web Server 2	10.10.2.22
Mail Server	10.10.2.23
IDS	10.10.2.24

Five servers are protected behind the firewall's screened web services network port, these include the DNS Server, SMTP Server, IDS Server and two Web Servers. The primary role for the firewall is to protect the Web servers on the Screened Web Services Network and the cookie databases on the Screened Data Services Network from being compromised and valuable data being stolen. Only the services required to support these servers should be allowed through the firewall, everything else is denied and logged.

The DNS service is isolated on a server of its own due to the number of vulnerabilities the BIND system has suffered. The service is running in a chrootd environment, that is, the service should not run with root privileges. The DNS server will not do recursive lookups, which will remove the possibility of DNS poisoning. Zone transfers are allowed only with the secondary DNS Server through the firewall using TCP port 53. The DNS Server address has a NAT address for DNS enquiries.

The Mail service has also been a source of vulnerabilities in the past and also warrants a separate server. The Sendmail service are not installed, instead Qmail is installed. This mail server is more secure and does not reveal its identity when the mail port is telented to. The mail server has a NAT address allowing internet mail to be received and sent. Mail service ports are allowed through the firewall to the Corporate Network to allow for access to Mail, TCP Ports 25 and POP3S, port 995.

The Web Servers run Redhat 7.0 Linux and Roxen Challenger HTTPD servers. The services provided through the firewall are TCP ports http (80) and https (443). All data that is considered to be sensitive is passed across any links use encrypted services, being Secure Socket Layers (SSL). No dynamic data is stored on the web servers but is accessed through the firewall from the databases located in the Screened Data Services Network.

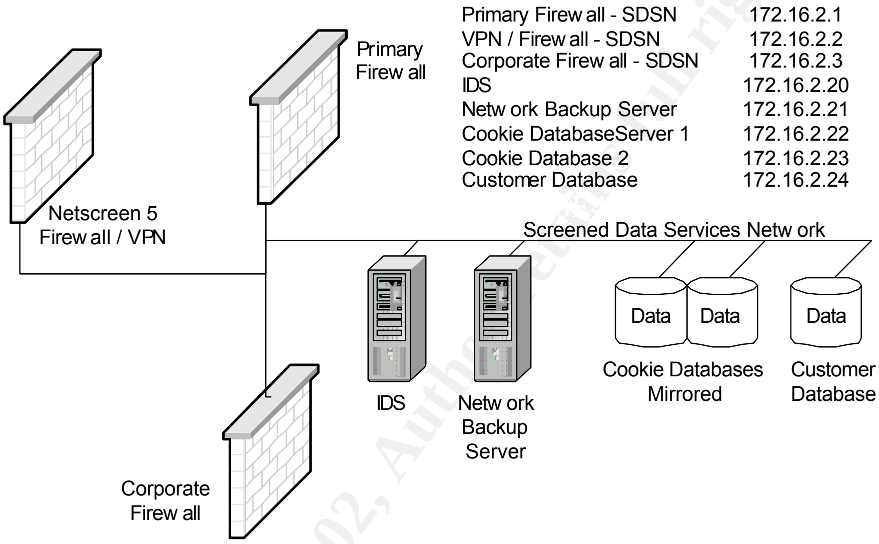
The IDS is used to detect, record any potential threats and watches for abnormal traffic patterns on the wire. The IDS is then able to email or page the security team member on call. The IDS server will log its output to the syslog server in the Screened Data Services Network through UDP port 514.

Screened Data Services Network.

The Data Services Network is connected to the third interface of the firewall.

On this network segment, we employ five servers, these being IDS, network backup, cookie database and customer database.

The IDS is used to detect, record any potential threats and watches for abnormal traffic patterns on the wire. The IDS is then able to email or page the security team member on call. The IDS server will log its output to the syslogd server in the Corporate Network on UDP port 514.



Primary Firew all - SDSN	172.16.2.1
VPN / Firew all - SDSN	172.16.2.2
Corporate Firew all - SDSN	172.16.2.3
IDS	172.16.2.20
Netw ork Backup Server	172.16.2.21
Cookie DatabaseServer 1	172.16.2.22
Cookie Database 2	172.16.2.23
Customer Database	172.16.2.24

Figure

The Netscreen 5 VPN is being used to provide access to the Corporate network for mobile users. The VPN tunnel is established through the Netscreen Remote VPN software on the mobile users laptops. A third firewall separates the Screened Data Services network from the Corporate LAN. This firewall uses IP Tables running on a bastion Redhat 7.0 Linux host.

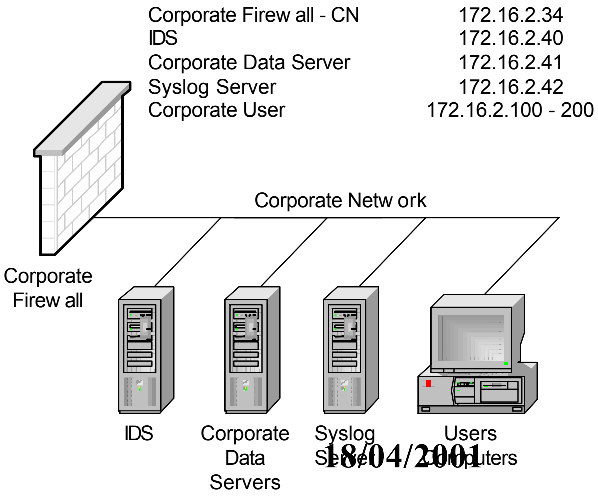
The Network Backup server provides backups of the critical data daily, this being the cookie and customer databases. The static web content and email facilities are incrementally backup daily with a fortnightly full backup procedure.

The database servers are based on i86 PC architecture running Redhat 7.0 and Mysql server. There is two cookie databases being replicated for high availability and are housed separately to the customer database to provide security for the customer database. This allows for a tighter user security to be employed when it comes to the Partner and Supplier access into this area of the network.

Corporate Network

This network segment contains three servers, IDS server, Corporate Data server and Syslogd server. A Linux based IP Tables firewall protects the Corporate LAN from the Screen Data Services Network. Two user groups are implemented here, these being Administration and Regular Users.

The IDS is used to detect, record any potential threats and watches for abnormal traffic patterns on the wire. The IDS is then able to email or page the security team member on call.



Corporate Firew all - CN	172.16.2.34
IDS	172.16.2.40
Corporate Data Server	172.16.2.41
Syslog Server	172.16.2.42
Corporate User	172.16.2.100 - 200

The IDS server will log its output to the syslogd server in the Corporate Network on UDP port 514. The IDS has been placed in this network segment to detect any potential internal threats, these being disgruntled employees or just inquisitative network browsers. For example, having people trying to access personnel files or financial information.

The security team is required to daily examine the syslog server logs from both the Screened Data Services Network and the Corporate Network. If these logs and alerts are monitored and checked daily, a compromise could go undetected.

Equipment Descriptions

Equipment	Description	Operating System
Border Router	Cisco 4000 M modular router	4000 Series IOS IP – Version 11.1
Primary Firewall	Netfilter 1.1.1	Redhat 7.0 – Kernel 2.4.3
Web Servers	Roxen Challenger 1.3.111	Redhat 7.0 – Kernel 2.4.3
DNS Server	Bind 8.2.3	Redhat 7.0 – Kernel 2.4.3
Mail Server	Qmail	Redhat 7.0 – Kernel 2.4.3
Cookie & Customer Database Servers	MYSQL 3.23.22	Redhat 7.0 – Kernel 2.4.3
Remote Access Firewall	Netscreen-5	
Corporate Firewall	Netfilter 1.1.1	Redhat 7.0 – Kernel 2.4.3

Assignment 2 – Security Policy

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

- *Border Router*
- *Primary Firewall*
- *VPN*

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

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, etc. (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 split-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 explanation 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 create 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".

The Security policies detailed in this section are in relation to the security architecture design from assignment 1. There are a few assumptions that need to detailed here.

The security team will be maintaining the security policies, all changes are update in the security policy documents and the staff, suppliers and partners are notified of changes in the security documents. All logs are checked daily for possible intrusion or internal threats, abnormal traffic patterns and abnormal access of information stores.

The secondary DNS server is housed at the ISP location and has IP address of aaa.bbb.90.25 and there is no other secondary DNS servers.

The IP Addresses of the different servers seen by the Internet are Network Translated Addresses as detailed.

Router Serial interface (Internet)		aaa.bbb.90.252
Router Ethernet interface (DMZ)		xxx.yyy.91.254/24
Primary Firewall		xxx.yyy.91.10/24
Mobile User Firewall / VPN		xxx.yyy.91.11/24
DNS Server	10.10.2.20/24	xxx.yyy.91.20/24
Web Server	10.10.2.21/24	xxx.yyy.91.21/24
Web Server	10.10.2.22/24	xxx.yyy.91.22/24
Mail Server	10.10.2.23/24	xxx.yyy.91.23/24

All the internal addressing is detailed in Assignment 1, Figure 1.

The security policy requires access to different areas of the network and therefore needs to be documented. The initial service access rights are detailed below.

Server or Service	Required By	Protocol	Ports
Web Servers	All	tcp	80 & 443
DNS Servers	All	udp	53
Zone Transfers	Secondary DNS Server	tcp	53
Mail Server	All	tcp	25 & 995
Database Servers	Web Servers	tcp	3306
Backup Servers	All Servers	tcp	22
Web, Intranet & Mail Servers	Internal Users	tcp	80,443,25 & 995
DNS lookups	Internal Users	udp	53
Syslog Server	IDS Servers	udp	514

I have decided to implement a allow known and deny everything else policy. This allows for easier detection of illegal traffic on the wire. Due to the traffic on different segments be known through the policies implemented, potential threats will be quickly identified.

The Border Router

The border router is a Cisco 4000 router running IOS 11.1(6) and will primarily route. The 4000 will be configure to do some packet filtering to filter out the noise from the internet, but still route. The border router is configured to protect from IP spoofing, Source routing and ICMP Redirects. Netbios and Egress filtering is employed to ensure that all packets leaving the network comply with the legal IP network space.

The Cisco IOS has inbuilt packet filtering in the form of Access Control Lists. Access Lists expresses the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router and packets that exit the routers outbound interface. There is two general types of filtering, standard and extended ACL's that can be configured to filter or test packets to determine whether to forward them to their destination or to drop the packet.

Access lists.

- **Standard IP** access lists examine the source IP address field in the packets IP header, which results in permitting or denying the packet movement.

  Usage of the Standard Access List
  Access-list {*access-list-number*} {permit|deny} *source-addr* [*source-mask*] [log]
  The command consists of five main parts.
  1. The command : **access-list**
  2. Access-list-number identifies the list to which the entry belongs. Valid entry, numbers 1 to 99.
  3. **permit|deny** indicates whether this entry allows or blocks the packet from the source address
  4. The source address identity, can be ***any***, ***host xxx.xxx.xxx.xxx*** or ***xxx.xxx.xxx.xxx***
  5. The source mask, is usually the Cisco wildcard mask. This mask identifies which bits in the address field are matched, default value is 0.0.0.0 matching all bits. The mask looks like a standard netmask, in actual fact it is a reverse image of the netmask. For example: a typical netmask is 255.255.255.0 and the Cisco wilcard mask for the netmask is 0.0.0.255. Using this example, in the access list the first 24 bits on the source address must be the same if the rule is to match and act on the source address.
  6. The log statement. This is an optional statement and causes any matches with the filter to be logged. This can be locally on the router or can be sent to a syslog server using udp port 514.

  Applying the Standard Access List
  Router(config)#interface serial 0
  Router(config-if)#ip access-group *access-list-number* {in|out}
  1. Change to the interface the access list is to be applied.
  2. The command : **ip access-group**
  3. Access-list-number identifies the number of the access list to be applied to this interface
  4. **in|out** selects whether the access list is applied as an incoming or outgoing filter. If this is not

specified, then **out** is applied.

- **Extended IP** access lists examine both the source and destination addresses of the packets. They can also check specific port numbers and protocols, giving the administrator more flexibility in configuring the access list.

  Usage of the Standard Access List
  Access-list {access-list-number} {permit|deny} *protocol source-addr source-mask* [operator port] *destin-addr destin-mask* [operator port] [*established*] [log]
  The command consists of five main parts.

  1. The command : **access-list**
  2. **Access-list-number** identifies the list using a number in the range 100 to 199
  3. **permit|deny** indicates whether this entry allows or blocks the specified address
  4. The **protocol** can be either IP, TCP, UDP, ICMP, GRE, or IGRP
  5. The **source** and **destination address**, can be ***any***, ***host xxx.xxx.xxx.xxx*** or ***xxx.xxx.xxx.xxx***
  6. The **source** and **destination mask**, is usually the Cisco wildcard mask. This mask identifies which bits in the address field are matched, default value is 0.0.0.0 matching all bits. The mask looks like a standard netmask, in actual fact it is a reverse image of the netmask. For example: a typical Netmask 255.255.255.0 and the Cisco wilcard mask for the netmask 0.0.0.255. 0's indicate positions that must match; 1's indicate "don't care" positions.
  7. **operator port** can be lt (less than), gt (greater than), eq (equal to) or neq (not equal to) and a protocol port number.
  8. established is used for inbound TCP only. This allows TCP traffic to pass if the packet uses an established connection (for example, if it has ACK bits set).
  9. The log statement. This is an optional statement and causes any matches with the filter to be logged. This can be locally on the router or can be sent to a syslog server using udp port 514.

  Applying the Extended Access List
  Router(config)#interface serial 0
  Router(config-if)#ip access-group *access-list-number* {in|out}

  1. Change to the interface the access list is to be applied.
  2. The command : **ip access-group**
  3. Access-list-number identifies the number of the access list to be applied to this interface
  4. **in|out** selects whether the access list is applied as an incoming or outgoing filter. If this is not specified, then **out** is applied

The access lists can be applied as either Inbound or Outbound. Access lists can be applied to multiple interfaces, but there can only be one access list per protocol per interface per direction.

Access lists operate in a sequential and logical order, that is they evaluate packets from the top down, one statement at a time. If the packet matches one of the access list statements, it is permitted or denied as specified in the entry and all other statements that follow are skipped. If no match is found, the packet is denied by an implicit deny.

CAUTION.
If the ip access-group command has been applied to an interface before the access-list has been created, the result will be permit any. This makes the access-list live, adding access-list statements through ***config t*** could cause the interface to become deny most or even all, when the return key is pressed. This is due to the ***implicit deny any*** at the end of the access-list. Therefore, create any access-lists before applying them to any interface.

Configuration

The configuration of the border router is based on the Cisco 4000 router with a High Speed Serial port and IOS 11.1(6) ip feature set. The router has more than sufficient resources to allow for routing, ingress and egress filtering. The following configuration assumes the reader has a basic understanding of the Cisco IOS and only the relevant parts of the configuration will be discussed here.

Router Security

The router is the first main point of access to the network and is the first place for perimeter security is implemented. Therefore the router is armoured, that is, no unnecessary service left running that could cause some type of access vulnerability for a potential attacker. The router default passwords are change and encrypted. The small server and other services have been disabled.

Access to the router for configuration purposes is via the console port of the router. Due to vulnerabilities of telnet and passing of the username / password is in clear text form, telnet services have been denied in the access lists.

```
version 11.1
!
no service pad
no service udp-small-servers                    #The small servers have been disabled, no known vulnerabilities but to be safe.

no service tcp-small-servers
no service finger                       #Finger service is not require and gives info to a potential attacker, so disable.
no ip bootp server                              #bootp server and http server has been disabled. Stops alternate connections.
no ip source-route                              #Harmful packets can be routed using source routing. Disabled.
no ip subnet-zero
!
enable secret 5 $1$MHCa$FxPz/ryxKbFZIEPi238DD.      #Password for secret access is encrypted.
!
banner motd ^C WARNING: Authorised Access Only ^C   #warning banner stating it is unlawful to enter or attempt to enter
!                                                   #without the proper authorisation.
```

Ingress Access List

The ingress access list is used to block inbound spoofed packets. The private address space as detailed in RFC1918 are the common addresses used in this type of attack. This access list also blocks localhost, broadcast and multicast addresses.

```
access-list 101 deny   tcp 10.0.0.0 0.255.255.255 any log               #Deny RFC 1918 addresses
access-list 101 deny   tcp 127.0.0.0 0.255.255.255 any log      #Deny localhost addresses
access-list 101 deny   tcp 172.16.0.0 0.15.255.255 any log      #Deny RFC 1918 addresses
access-list 101 deny   tcp 192.168.0.0 0.0.255.255 any log      #Deny RFC 1918 addresses
access-list 101 deny   tcp 224.0.0.0 31.255.255.255 any log     #Deny Mulitcast addresses
access-list 101 deny   tcp 255.0.0.0 0.255.255.255 any log      #Deny broadcast address
access-list 101 deny   tcp xxx.yyy.91.0 0.0.0.255 any log               #Deny incoming packets of our internal address
access-list 101 deny   tcp host 0.0.0.0 any log                         #Deny any host from an invalid address
```

The ingress access list will filter for other services, these include NetBIOS, bootp. The NetBIOS service is used in denial of service attacks on windows based systems and is therefore blocked at the border router. The bootstrap protocol server and clients are blocked to stop potential allocation of IP addresses from the internal network.

```
access-list 101 deny   udp any any range bootps bootpc log              #Deny bootp services
access-list 101 deny   tcp any any eq 135 log                                   #Deny NetBIOS services
access-list 101 deny   udp any any range netbios-ns netbios-dgm log     #Deny NetBIOS services
access-list 101 deny   tcp any any eq 139 log                                   #Deny NetBIOS services
access-list 101 deny   tcp any any eq 445 log                           #Deny Microsoft Directory Services
access-list 101 deny   udp any any eq 445 log                                   #Deny Microsoft Directory Services
access-list 101 deny   icmp any any log                                 #Deny all ICMP traffic
```

The login services of the router is generally a high risk and should be disabled. The login services denied on the router include FTP, SSH, Telnet, Rlogin and exec commands. The X windows ports have also been denied on the external interface, as these port are a commonly scanned for, as they give remote access to the X windows interface.

```
access-list 101 deny   tcp any any range ftp telnet log     #Deny services FTP, SSH and Telnet
access-list 101 deny   tcp any any range exec shell log             #Deny services exec, login, who and shell
access-list 101 deny   tcp any any range 6000 6255 log              #Deny Xwindows ports
```

Allow specific access to the DNS servers for zone transfers, and allow traffic destined for the internal network.

```
access-list 101 permit tcp aaa.bbb.90.25 0.0.0.0 xxx.yyy.91.30 0.0.0.0 eq 53          #Allow DNS Zone transfer to Secondary
Server
access-list 101 permit udp any xxx.yyy.91.20 0.0.0.0 eq 53                             #Allow DNS Lookups
access-list 101 permit ip any xxx.yyy.91.0 0.0.0.255                          #Allow traffic destined for internal network
access-list 102 deny ip any any log                                           #Deny everything else and log attempts
```

We need to send the log data to the syslog server

```
logging trap information          #Set logging to information level
logging  xxx.yyy.91.40            #Direct logging to Syslog sever  using NAT address for 172.16.2.40
```

Apply the access list to the Serial interface and perform some additional IP filtering on the interface.

```
interface Serial 0                                    #Apply to external interface Serial 0
 ip address aaa.bbb.90.252 255.255.255.0      #Sets the ip address of the interface
 ip access-group 101 in                       #Applies the access list 101 to this interface for incoming packets
 no ip redirects                              #Prevents the malicious redirect commands from causing DoS problems
 no ip unreachables                           #Prevents the router giving out network information from ICMP errors
 no ip directed-broadcast                     #Prevents malicious directed broadcasts from causing DoS problems
```

The important access-list statements from the ingress filter are the following lines.

```
access-list 101 permit tcp aaa.bbb.90.25 0.0.0.0 xxx.yyy.91.30 0.0.0.0 eq 53          #Allow DNS Zone transfer to Secondary
Server
access-list 101 permit udp any xxx.yyy.91.20 0.0.0.0 eq 53                             #Allow DNS Lookups
```

These two lines should be the first two lines in the access-list, DNS lookups and Zone transfers are locked to the one host. This allows for faster DNS responses as the filter is not require to do a lot of processing to achieve a match.

```
access-list 101 permit ip any xxx.yyy.91.0 0.0.0.255                          #Allow traffic destined for internal network
access-list 102 deny ip any any log                                           #Deny everything else and log attempts
```

These lines should be the last two lines in the access-list, should these lines be place earlier in the access-list, they could negate the filters otherwise denying or allowing the packets. These lines act upon the IP protocol as a whole and therefore match tcp, udp, icmp, etc and therefore would be more easily matched.

Egress Access List

The egress access list is used to prevent our internal network from sending any spoofed packets. This ensure that packets leaving our network belong to our network address space. Again the private address space as detailed in RFC 1918 are blocked. We will block all internal hosts from send icmp packets and allow only packets from our network address. Everything else will be logged.

```
access-list 102 deny   tcp 10.0.0.0 0.255.255.255 any log                 #Deny RFC 1918 addresses
access-list 102 deny   tcp 172.16.0.0 0.15.255.255 any log        #Deny RFC 1918 addresses
access-list 102 deny   tcp 192.168.0.0 0.0.255.255 any log        #Deny RFC 1918 addresses
access-list 102 deny   tcp any 10.0.0.0 0.255.255.255 log                 #Deny RFC 1918 addresses
access-list 102 deny   tcp any 172.16.0.0 0.15.255.255 log        #Deny RFC1918 addresses
access-list 102 deny   tcp any 192.168.0.0 0.0.255.255 log        #Deny RFC 1918 addresses
access-list 102 deny   icmp any any log                           #Deny any icmp traffic
access-list 102 permit ip xxx.yyy.91.0 0.0.0.255 any              #Permit only packets from our network
access-list 102 deny ip any any log                               #Deny everything else and log attempts
```

Apply the access list to the Serial interface and perform some additional IP filtering on the interface.

```
interface Ethernet 0                          #Apply to external interface Ethernet 0
 ip address xxx.yyy.91.254 255.255.255.0      #Sets the ip address of the interface
 ip access-group 102 in                       #Applies the access list 102 to this interface for incoming packets
 no ip unreachables                           #Prevents the router giving out network information from ICMP errors
```

The important access-list statements from the egress filter are the following lines.

```
access-list 102 permit ip xxx.yyy.91.0 0.0.0.255 any           #Permit only packets from our network
access-list 102 deny ip any any log                            #Deny everything else and log attempts
```

These lines should be the last two lines in the access-list, should these lines be place earlier in the access-list, they could negate the filters otherwise denying or allowing the packets. These lines act upon the IP protocol as a whole and therefore match tcp, udp, icmp, etc and therefore would be more easily matched.

> ***CAUTION.***
> The Cisco IOS is not very forgiving, in that, when the access-lists are being loaded in through *config t* you can not insert missed rules. Therefore the order in which the rules are loaded is important. The rules are listed in this assignment in the order as they would be in the configuration of the router, if displayed using the command *sh config* from the router prompt. The IOS has no way of being able to insert a statement into the list, it can only append to the list. If a statement is missed, the access-list needs to be removed and reloaded with the correct statement order.

Primary Firewall

A firewall prevents any direct unathourised attempts at access, and encryption protects transmissions from authorised remote users. To implement a firewall, you provide a series of rules that govern the kind of access allowed into the network. The system's firewall capability can effectively protect the network for outside attacks.

Packet filtering is the process of deciding whether a packet received at the firewall should be passed to the network. The packet filtering software checks the source and the destination address of the packet, as well as other tcp/ip header items and compares this with the filter rules to decided if the packet should be allowed.

The firewall detailed in this assignment is a bastion Redhat Linux / Netfilter system.

Redhat Configuration

The Firewall server is based on a Redhat 7.0 server using kernel 2.4.3. This server has been armourised using the Bastille projects, Bastille Linux. One advantage to compiling the kernel is the ability to customise its configuration, selecting particular devices and network support required. The 2.4.3 kernel no longer offer IP masquerade support as a kernel compile time option. Instead, you should select the network packet filtering option:

```
Networking options  --->
[*] Network packet filtering (replaces ipchains)
[*] Socket Filtering
[*] TCP/IP networking
[*]   IP: advanced router
[*]       IP: fast network address translation
[*]     IP: verbose route monitoring
[*]   IP: TCP syncookie support (disabled per default)
IP: Netfilter Configuration  --->
---
<*> 802.1d Ethernet Bridging
QoS and/or fair queueing  --->
```

The 802.1d Ethernet Bridging option has been complied into the kernel as a loadable module. This allows the kernel to be used as bridge instead of a router. The primary firewall is configured to be a gateway router. The other firewalls will be configured as bridges.

The other options selected above have been selected for either there ability to provide security filtering or routing support in the kernel. The Netfilter configuration options, as selected, are shown below. These were selected to provide kernel support of Network Address Translation and dedicated match filtering. The logging target support is also selected for Iptables firewall support.

```
IP: Netfilter Configuration  --->
<*>Connection tracking (required for masq/NAT)
<*>   FTP protocol support
<*> IP tables support (required for filtering/masq/NAT)
<*>   limit match support
<*>   MAC address match support
<*>   Multiple port match support
<*>   Connection state match support
<*>   Packet filtering
<*>   Full NAT
<*>     REDIRECT target support
<*>   LOG target support
```

CAUTION.
You should retain a copy of your current kernel, in case something goes wrong in the compilation and install of the new kernel. This is completed by backing up the file vmlinuz-*version*, where version is the version number attached. The System.map file should also be backed up as it supplies kernel symbols needed by the modules to start.

Netfilter Configuration.

Netfilter implements firewalls using a packet filtering tool called iptables. With IP Tables, different tables of rules can be applied to select packets according to differing criteria. Packet filtering is implemented using a filter table that contains rules for dropping or accepting packets. Network Address translation is implemented using the NAT table that contains IP masquerading rules. Instead of all the packets checking one large table, they only access the table of rules that relate to their traffic pattern.

IP Table rules are managed using the ***iptables*** command. The default is the filter table, which need not be specified. In iptables commands, chain names have to be entered in uppercase. The main chains to which this relates, is the INPUT, FORWARD and OUTPUT chains. The filter rules used by iptables is very similar to those of the ipchains with a few exceptions. To add a new rule the –N option is used and as already stated the default chain names need to be in uppercase.

The default list of commands follows:

```
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 rule 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]            List 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
  --destination -d [!] address[/mask]
                                  destination specification
  --in-interface -i [!]           input name[+]
                                  network interface name ([+] for wildcard)
  --jump  -j target               target for rule
  --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
  --exact  -x                     expand numbers (display exact values)
[!] --fragment       -f           match second or further fragments only
[!] --version        -V           print package version.
```

Additional command options include items like,

--state	specifying connection states ,eg NEW, INVALID, RELATED and ESTABLISHED
--tcp-flags	tcp flags, eg SYN, ACK, FIN, RST, URG, PS and ALL for all flags
--limit	Specify the rate of matches, matching a given number of times per second.

Network Address Translation (NAT)

NAT is the process whereby a system will change the source or destination of the packets as they pass through the system. To make this work the system is also required to remember these changes, so that the source and destination for any reply packets is change back to the original addresses.

Packet selection rules for NAT operations are added to the NAT table managed through iptables command. The rules are added to the ***nat*** table through the command ***iptables –t nat***. In addition, there are two types of NAT operations, the source NAT and the destination NAT.

- Source NAT
  This is when you alter the source address of the first packet: ie you are changing where the connection is coming from. Source NAT is always done post-routing, just before the packet goes out onto the wire.
- Destination NAT
  This is when you alter the destination address of the first packet: ie you are changing where the connection is going. Destination NAT is always done pre-routing, when the packet first comes of the wire.

Three rules in the ***nat*** table are used by the kernel for NAT processing. These are PREROUTING, POSTROUTING and OUTPUT. You would implement IP masquerading by adding a MASQUERADE rule to the POSTROUTING rule.

Firewall Configuration.

The first operation required on the firewall system is to create the aliases for the Network Address Translation of the web services, that is, the web servers, DNS server and mail server. The ip aliases are loaded from the network script during startup.

```
fw101$ ifconfig eth0:0 xxx.yyy.91.20        #NAT for DNS server 10.10.2.20
fw101$ ifconfig eth0:1 xxx.yyy.91.21        #NAT for Web server 10.10.2.21
fw101$ ifconfig eth0:2 xxx.yyy.91.22        #NAT for Web server 10.10.2.22
fw101$ ifconfig eth0:3 xxx.yyy.91.23        #NAT for Mail server 10.10.2.23
```

Next, the ***nat*** table is created and the filter rules for the NAT operations are loaded in to the ***nat*** table. These two steps are required to be completed first in order to get the NAT operations working, then the other firewall rules are applied. The security policy used on this firewall is deny everything except for the known services. This rules are applied in the input and output table as shown in the configuration shown below. The implementation of the firewall is through a IP Tables script.

```
# # Firewall Gateway configuration, three ethernet interfaces,
# # eth0  -  DMZ interface : address xxx.yyy.91.10
# # eth1  -  Screened Web Services Network interface : address 10.10.2.1
# # eth2  -  Screened Data Services Network interface : address 172.16.2.1

# Turn off IP forwarding
echo 0 > /proc/sys/net/ipv4/net/ip_forward

# The command to set up the nat table to hold NAT rules
iptables -N nat

# Flush chain rules
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
iptables -t nat -F

# set default policies
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD ACCEPT

# configure the forward table to use nat
iptables -A FORWARD -j nat
```

```
# setup NAT for the servers
iptables -t nat -A PREROUTING -i eth0 -d xxx.yyy.91.20 -j DNAT --to 10.10.2.20
iptables -t nat -A POSTROUTING -o eth0 -s 10.10.2.20 -j SNAT --to xxx.yyy.91.20
iptables -t nat -A PREROUTING -i eth0 -d xxx.yyy.91.21 -j DNAT --to 10.10.2.21
iptables -t nat -A POSTROUTING -o eth0 -s 10.10.2.21 -j SNAT --to xxx.yyy.91.21
iptables -t nat -A PREROUTING -i eth0 -d xxx.yyy.91.22 -j DNAT --to 10.10.2.22
iptables -t nat -A POSTROUTING -o eth0 -s 10.10.2.22 -j SNAT --to xxx.yyy.91.22
iptables -t nat -A PREROUTING -i eth0 -d xxx.yyy.91.23 -j DNAT --to 10.10.2.23
iptables -t nat -A POSTROUTING -o eth0 -s 10.10.2.23 -j SNAT --to xxx.yyy.91.23
iptables -t nat -A PREROUTING -i eth2 -d xxx.yyy.91.20 -j DNAT --to 10.10.2.20
iptables -t nat -A POSTROUTING -o eth2 -s 10.10.2.20 -j SNAT --to xxx.yyy.91.20
iptables -t nat -A PREROUTING -i eth2 -d xxx.yyy.91.21 -j DNAT --to 10.10.2.21
iptables -t nat -A POSTROUTING -o eth2 -s 10.10.2.21 -j SNAT --to xxx.yyy.91.21
iptables -t nat -A PREROUTING -i eth2 -d xxx.yyy.91.22 -j DNAT --to 10.10.2.22
iptables -t nat -A POSTROUTING -o eth2 -s 10.10.2.22 -j SNAT --to xxx.yyy.91.22
iptables -t nat -A PREROUTING -i eth2 -d xxx.yyy.91.23 -j DNAT --to 10.10.2.23
iptables -t nat -A POSTROUTING -o eth2 -s 10.10.2.23 -j SNAT --to xxx.yyy.91.23
iptables -t nat -A POSTROUTING -o eth0 -s 172.16.2.0/24  -j MASQUERADE

# IP Spoofing denial from the internal networks that have external source addresses
iptables -A INPUT -j LOG -i eth1 \! -s 10.10.2.0/24
iptables -A INPUT -j DROP -i eth1 \! -s 10.10.2.0/24
iptables -A INPUT -j LOG -i eth2 \! -s 172.16.2.0/24
iptables -A INPUT -j DROP -i eth2 \! -s 172.16.2.0/24
# IP Spoofing denial of packets not on eth1 or eth2 with internal network addresses
iptables -A INPUT -j LOG \! -i eth1 -s 10.10.2.0/24
iptables -A INPUT -j DROP \! -i eth1 -s 10.10.2.0/24
iptables -A INPUT -j LOG \! -i eth2 -s 172.16.2.0/24
iptables -A INPUT -j DROP \! -i eth2 -s 172.16.2.0/24
# IP Spoofing denial of outside packets to the localhost address
iptables -A INPUT -j LOG -i \! lo -s 127.0.0.0/8
iptables -A INPUT -j DROP -i \! lo -s 127.0.0.0/8

# allow connection to web servers on port 80 and 443
iptables -A INPUT -j ACCEPT -p tcp -i eth0 --dport www -d 203.34.90.21
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 --dport www -s 203.34.90.21
iptables -A INPUT -j ACCEPT -p tcp -i eth0 --dport www -d 203.34.90.22
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 --dport www -s 203.34.90.22
iptables -A INPUT -j ACCEPT -p tcp -i eth0 --dport 443 -d 203.34.90.21
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 --dport 443 -s 203.34.90.21
iptables -A INPUT -j ACCEPT -p tcp -i eth0 --dport 443 -d 203.34.90.22
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 --dport 443 -s 203.34.90.22
# deny new connections from the web servers to the internal network
iptables -A OUTPUT -m state --state NEW -o eth2 -p tcp --sport www -d 172.16.2.0/24 -j DROP

# allow connection to the DNS Server
iptables -A INPUT -j ACCEPT -p udp -i eth0 --dport 53 -s xxx.yyy.91.20
iptables -A OUTPUT -j ACCEPT -p udp -o eth0 --sport 53 -d xxx.yyy.91.20
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 -s aaa.bbb.57.5 --dport 53 -d xxx.yyy.91.20

# allow mail to be received and sent
iptables -A INPUT -j ACCEPT -p tcp -i eth0 --dport 25 -d xxx.yyy.91.23
iptables -A OUTPUT -j ACCEPT -p tcp -o eth0 --dport 25 -s xxx.yyy.91.23
iptables -A INPUT -j ACCEPT -p tcp -i eth2 --dport 995 -d xxx.yyy.91.23
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 --dport 995 -s xxx.yyy.91.23
# deny all other connections to Mail server
iptables -A INPUT -j DROP -i eth0 -s xxx.yyy.91.23 -d xxx.yyy.91.23
```

```
# Allow connections from Web Servers to the databases
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.21 -d 172.16.2.22
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.22 -d 203.34.91.21
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.22 -d 172.16.2.22
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.22 -d 203.34.91.22
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.21 -d 172.16.2.23
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.23 -d 203.34.91.21
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.22 -d 172.16.2.23
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.23 -d 203.34.91.22
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.21 -d 172.16.2.24
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.24 -d 203.34.91.21
iptables -A INPUT -j ACCEPT -ip tcp -i eth2 --dport 3306 -s 203.34.91.22 -d 172.16.2.24
iptables -A OUTPUT -j ACCEPT -p tcp -o eth2 -s 172.16.2.24 -d 203.34.91.22

 # allow IDS to send logs to syslog server
iptables -A INPUT -j ACCEPT -p udp -i eth2 --dport 514  -d 172.16.2.40

# turn back on IP Forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

exit 0
```

Corporate Firewall

The Corporate firewall isolates the Screened Data Services Network from the Corporate Network. This firewall is base on Redhat 7.0 with kernel 2.4.3 and configured as a bridge, IP Tables packet filtering is used to do the fire walling.

Redhat Configuration

The Firewall server is based on a Redhat 7.0 server using kernel 2.4.3, which is the default kernel with Redhat 7.0. This server has been armourised using the Bastille projects, Bastille Linux. One advantage to compiling the kernel is the ability to customise its configuration, selecting particular devices and network support required. The 2.2.16 kernel will be recompiled to include the feature set we require for the bridge and firewall use. Instead, you should select the network options shown below:

CAUTION.

> You should retain a copy of your current kernel, in case something goes wrong in the compilation and install of the new kernel. This is completed by backing up the file vmlinuz-*version*, where version is the version number attached. The System.map file should also be backed up as it supplies kernel symbols needed by the modules to start.

Netfilter Configuration.

Netfilter implements firewalls using a packet filtering tool called iptables. With IP Tables, different tables of rules can be applied to select packets according to differing criteria. Packet filtering is implemented using a filter table that contains rules for dropping or accepting packets. Network Address translation is implemented using the NAT table that contains IP masquerading rules. Instead of all the packets checking one large table, they only access the table of rules that relate to their traffic pattern.

IP Table rules are managed using the ***iptables*** command. The default is the filter table, which need not be specified. In iptables commands, chain names have to be entered in uppercase. The main chains to which this relates, is the INPUT, FORWARD and OUTPUT chains. The filter rules used by iptables is very similar to those of the ipchains with a few exceptions. To add a new rule the –N option is used and as already stated the default chain names need to be in uppercase. The command set is listed above in the Primary Firewall section and will not be repeated here

Firewall Configuration.

The first operation required on the firewall system is to create the aliases for the Network Address Translation of the web services, that is, the web servers, DNS server and mail server. The ip aliases are loaded from the network script during startup.

```
# # Firewall Gateway configuration, three ethernet interfaces,
# # eth0  -  Screened Data Services Network : address 0.0.0.0
# # eth1  -  Corporate Network interface : address 0.0.0.0
# # fw201  -  Network Bridge : address 172.16.2.32

# Turn off IP forwarding
echo 0 > /proc/sys/net/ipv4/net/ip_forward

# setup the bridge - fw201
brctl addbr fw201
# add ethernet cards to the bridge
brctl addif fw201 eth0
brctl addif fw201 eth1
# assign an ethernet address to the bridge
ifconfig fw201 172.16.2.32 netmask 255.255.255.0 broadcast 172.16.2.255

# The command to set up the bridge chain to enable firewall on the bridge
iptables -N fw201
```

```
# Flush tables rules
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
iptables -t fw201 -F

# set default policies
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD ACCEPT

# allow connection to web servers on port 80 and 443
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport www -d 203.34.90.21
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport www -s 203.34.90.21
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport www -d 203.34.90.22
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport www -s 203.34.90.22
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport 443 -d 203.34.90.21
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport 443 -s 203.34.90.21
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport 443 -d 203.34.90.22
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport 443 -s 203.34.90.22
# deny new connections from the web servers to the internal network
iptables -A fw201 -m state --state NEW -o eth2 -p tcp --sport www -d 172.16.2.0/24 -j DROP

# allow connection to the DNS Server
iptables -A fw201 -j ACCEPT -p udp -i eth0 --dport 53 -s xxx.yyy.91.20
iptables -A fw201 -j ACCEPT -p udp -o eth0 --sport 53 -d xxx.yyy.91.20
iptables -A fw201 -j ACCEPT -p tcp -o eth0 -s aaa.bbb.91.25 --dport 53 -d xxx.yyy.91.20

# allow mail to be received and sent
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport 25 -d xxx.yyy.91.23
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport 25 -s xxx.yyy.91.23
iptables -A fw201 -j ACCEPT -p tcp -i eth0 --dport 995 -d xxx.yyy.91.23
iptables -A fw201 -j ACCEPT -p tcp -o eth0 --dport 995 -s xxx.yyy.91.23
# deny all other connections to Mail server
iptables -A fw201 -j DROP -i eth0 -s xxx.yyy.91.23 -d xxx.yyy.91.23

# Allow connections from Corporate Network to the databases
iptables -A fw201 -j ACCEPT -ip tcp -i eth0 --dport 3306 -d 172.16.2.22
iptables -A fw201 -j ACCEPT -p tcp -o eth0 -s 172.16.2.22
iptables -A fw201 -j ACCEPT -ip tcp -i eth0 --dport 3306 -d 172.16.2.23
iptables -A fw201 -j ACCEPT -p tcp -o eth0 -s 172.16.2.23
iptables -A fw201 -j ACCEPT -ip tcp -i eth0 --dport 3306 -d 172.16.2.24
iptables -A fw201 -j ACCEPT -p tcp -o eth0 -s 172.16.2.24

 # allow IDS to send logs to syslog server
iptables -A fw201 -j ACCEPT -p udp -i eth0 --dport 514  -d 172.16.2.40

# turn back on IP Forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

exit 0
```

VPN / FIREWALL

The Netscreen 5 VPN is being used to provide access to the Corporate network for mobile users. The VPN tunnel is established through the Netscreen Remote VPN software on the mobile users laptops. The Netscreen 5 utilises the most stringent security techniques - such as DES and Triple DES encryption, IKE key Management for secure key exchange, MD5 and SHA-1 authentication.

The implementation used for the VPN configuration is base on Triple DES encryption and MD5 authentication. There is three VPN tunnels provided, these being, one for the partners, one for the suppliers and on for the mobile users. More tunnels can be added as the need requires.

VPN Configuration

```
Total Config size 4728:
# configure authority access to user database with idle time out
set auth type 0
set auth timeout 10

# configure time zone GMT
set clock zone 10

# configure local admin users database
set admin auth timeout 10
set admin auth type Local
set admin name security_one
set admin password nKVUM2rwMUzPcrkG5sWIHdCtqkAibn
set admin manager-ip 172.16.2.0 255.255.255.0
set admin sys-ip 172.16.2.250

# configure email notification of alarms
set admin mail alert
set admin mail server-name xxx.yyy.91.23
set admin mail mail-addr1 security@giac.com
set admin mail traffic-log

# configure the trusted and untrusted interfaces
set interface trust ip 172.16.2.254 255.255.255.0
set interface untrust ip xxx.yyy.91.11 255.255.255.0
set interface trust bandwidth 10000
set interface untrust bandwidth 10000
set interface untrust gateway xxx.yyy.91.254
set interface trust manage ssl

# configure domain
set domain giac.com

# configure time synchronisation with other servers
set ntp server 172.16.0.3
set ntp interval 120

# configure the address book
#contains the ip addreses of hosts that can have their traffic either allowed, blocked, encrypted or authenticated
set address untrust "ISP" xxx.yyy.90.1 255.255.255.255 "ISP ACCESS ROUTER"
set address trust "Database Access" 172.16.2.22 255.255.255.255 "Customer Database One"
set address trust "Database Access 2" 172.16.2.23 255.255.255.255 "Customer Database Two"
set address trust "Corporate LAN" 172.16.2.0 255.255.255.0 "Corporate LAN  for Mobile Users"
set group address trust "Cookie Databases" comment "Database access to cookies"
set group address trust "Cookie Databases" add "Database Access"
set group address trust "Cookie Databases" add "Database Access 2"

# configure specialised filter for mySQL databases
set service "mySQL" protocol tcp src-port 3306 dst-port 3306 group "other"
```

```
# configure firewall options on the VPN device
set syn-threshold 200
set firewall tear-drop
set firewall syn-flood
set firewall ip-spoofing
set firewall ping-of-death
set firewall src-route
set firewall land
set firewall icmp-flood
set firewall udp-flood
set firewall winnuke
set firewall port-scan
set firewall ip-sweep
set firewall applet
set firewall default-deny
set firewall syn-flood alarm-threshold 1024
set firewall syn-flood queue-size 10240
set firewall log-self

# configure VPN users
set user "Mobile" ike-id "laptop"
set user "Mobile" "enable"
set user "Partners" ike-id "Par1"
set user "Partners" "enable"
set user "Supplier" ike-id "Supp1"
set user "Supplier" "enable"

# configure the VPN access
set dialup "Dial-Up VPN Tunnel" + "Partners"
set dialup "Dial-Up VPN Tunnel" + "Supplier"
set dialup "Dial-Up VPN Tunnel" + "Mobile"

# configure IKE gateway parameters
# chooses an appropiate Phase 1 proposal for negotiating the building of the tunnel
set ike gateway "Partners_in" dialup "Dial-Up VPN Tunnel" Main preshare "p4rtn3rs" proposal "pre-g2-3des-md5"
set ike gateway "Suppliers_in" dialup "Dial-Up VPN Tunnel" Main preshare "suppl13rs" proposal "pre-g2-3des-md5"
set ike gateway "Mobile_in" dialup "Dial-Up VPN Tunnel" Main preshare "m0b1les" proposal "pre-g2-3des-md5"
set ike policy-checking
set ike respond-bad-spi 1

# associate a remote gateway with a Phase 2 Proposal
# describes how the data will encrypted through the tunnel
set vpn "Supp_tunnel" id 13 gateway "Suppliers_in" replay tunnel idletime 0 proposal "nopfs-esp-3des-md5"
set vpn "Supp_tunnel" monitor
set vpn "Part_tunnel" id 14 gateway "Partners_in" replay tunnel idletime 0 proposal "nopfs-esp-3des-md5"
set vpn "Part_tunnel" monitor
set vpn "Mobil_tunnel" id 15 gateway "Mobile_in" replay tunnel idletime 0 proposal "nopfs-esp-3des-md5"
set vpn "Mobil_tunnel" monitor
set ike id-mode subnet
set traffic-shaping ip_precedence 7 6 5 4 3 2 1 0

# configure the policy for access to the network via the VPN
# These policies require authentication - auth
# count and log gives an indication of the amount of traffic moving accross these links
set policy id 6 name "Supplier_Access" outgoing "Cookie Databases" "Dial-Up VPN" "mySQL" Tunnel vpn \
              "Supp_tunnel" id 21 auth log count
set policy id 8 name "Partner_Access1" outgoing "Cookie Databases" "Dial-Up VPN" "mySQL" Tunnel vpn \
             "Part_tunnel" id 22 auth log count
set policy id 10 name "Mobile_Access" outgoing "Corporate LAN" "Dial-Up VPN" "ANY" Tunnel vpn \
             "Mobil_tunnel"  id 20 auth log count
```

```
# configure the syslog features to log to Syslog server in the Corporate network
set syslog config 172.16.2.40 local7 alert
set syslog enable
set syslog traffic

# enable the secure command shell, SSH compatible client management
set scs enable
ns5->
```

CAUTION.
The default user and password should be changed, as this is one of the major items usually forgotten, and leaves the device wide open to attack. Do not print out the configuration of this device, as the preshare key is stored in the device config as clear text.

Assignment 3 - Audit Your Security Architecture

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

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 the 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 (and 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 nmap or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.*

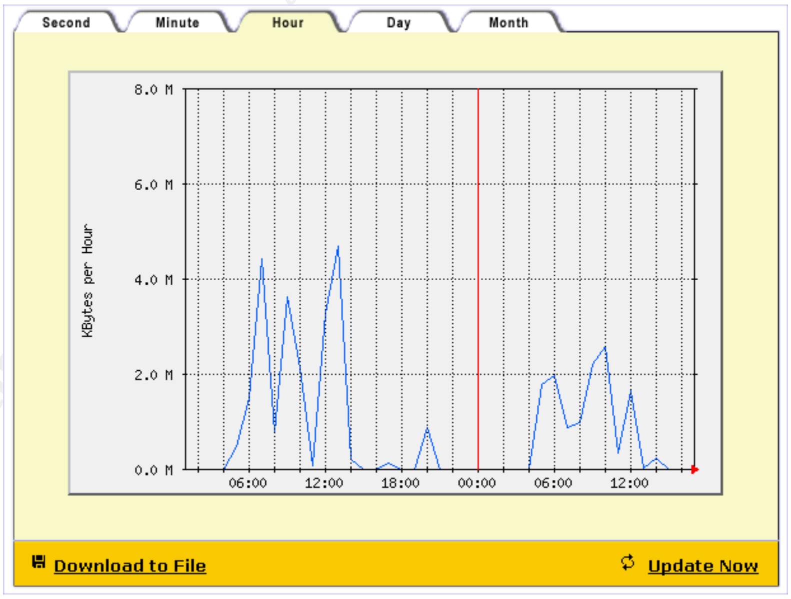
The Audit Plan

The audit plan should include all levels of security, from the border router to the last firewall, and everything in between. This plan should be based on the initial security strategies and when planning an audit assessment of the security architecture there are several items that need to be considered.

- During testing the network performance will be compromised
- Interruptions to normal business activities
- Potential disruptions to the web servers and site access

So in order to strategise an audit plan these items need to be taken into account. After monitoring the network and its usage, it has become apparent that the best time to undertake this testing is late Sunday night, starting from about 10:30pm and going through until Monday morning, 5am. This will minimise the impact on the business and maintain the business requirements for operational status. This can bee seen from the graph.

This time frame would allow enough time to have several tests running on different parts of the network. The tests would produce logs, which can be further analysed away from the site, therefore causing minimal disruptions to the business operations.



The Testing Procedure

The testing procedure will contain six steps, these being:

1. To scan and identify the external interfaces for services that respond.
2. Attempt to compromise the services using common vulnerabilities
3. Check the firewall rules are working correctly
4. Check the IDS logging systems, to identify failures and successes
5. Compare logs and testing results against the known status of the network
6. Make recommendations and changes to make the network more secure.

What to test

The main security devices in the network are as follows and these are the item to be tested. The testing will be carried out using standard Linux programs and utilities, these being netcat, nslookup, dig and the program NMAP, version 2.53 will also be utilised.

1. The Border Router
2. The Primary Firewall
3. The VPN / Firewall

These three devices form the outer perimeter defense of the network and therefore are more critical to the security of the network. Therefore the scans detailed in the table will test for opening in this layer of defense. A ping sweep is also undertaken from the outside world to indicate what servers are available.

Security Device	IP address Tested	Test Tool	Type of tests
Border Router	aaa.bbb.90.252	NMAP	FIN Stealth Scan UDP Port Scan connect() Scan
Primary Firewall	xxx.yyy.91.10	NMAP	SYN Stealth Scan UDP Port Scan connect() Scan
VPN / Firewall	xxx.yyy.91.11	NMAP	SYN Stealth Scan UDP Port Scan connect() Scan

There is also the matter of access to particular servers and services, which needs to be considered. These rules need to be tested to complete the audit. This testing will be in relation to the initial service access rights detailed earlier. The server or services to be tested, would include the Corporate Firewall, DNS, Mail, Web and Database servers.

Therefore in assessing the firewall's implementation of it rules, testing would be carried out using NMAP, ping and nslookup. The result we are looking for are :

Does any server answer requests to port 80 and 443 except for the Web Servers
Does any server answer requests to port 53 other than the DNS server
Does any server answer requests to port 25 and 995 other than the Mail server
Does any server answer requests to port 3306 except for the database servers
Does any server answer requests to port 514 except for the syslog server
Do the firewalls drop icmp packets and is the IDS logging

SCAN TYPES

-sT ***TCP connect() scan:*** This is the most basic form of TCP scanning. The connect() system call provided by your operating system is used to open a connection to every interesting port on the machine. If the port is listening, connect() will succeed, other-wise the port isn't reachable. One strong advantage to this technique is that you don't need any special privileges. Any user on most UNIX boxes is free to use this call.

This sort of scan is easily detectable as target host logs will show a bunch of connection and error messages for the services which accept() the connection just to have it immediately shutdown.

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

-sF -sX -sN

Stealth FIN, Xmas Tree, or Null scan modes: There are times when even SYN scanning isn't clandestine enough. Some firewalls and packet filters watch for SYNs to restricted ports, and programs like Synlogger and Courtney are available to detect these scans. These advanced scans, on the other hand, may be able to pass through unmolested.

The idea is that closed ports are required to reply to your probe packet with an RST, while open ports must ignore the packets in question (see RFC 793 pp 64). The FIN scan uses a bare (surprise) FIN packet as the probe, while the Xmas tree scan turns on the FIN, URG, and PUSH flags. The Null scan turns off all flags.

Unfortunately Microsoft (like usual) decided to completely ignore the standard and do things their own way. Thus this scan type will not work against systems running Windows95/NT. On the positive side, this is a good way to distinguish between the two platforms. If the scan finds open ports, you know the machine is not a Windows box.

If a -sF,-sX,or -sN scan shows all ports closed, yet a SYN (-sS) scan shows ports being opened, you are probably looking at a Windows box. This is less useful now that nmap has proper OS detection built in. There are also a few other systems that are broken in the same way Windows is. They include Cisco, BSDI, HP/UX, MVS, and IRIX. All of the above send resets from the open ports when they should just drop the packet.

-sU ***UDP scans:*** This method is used to determine which UDP (User Datagram Protocol, RFC 768) ports are open on a host. The technique is to send 0 byte udp packets to each port on the target machine. If we receive an ICMP port unreachable message, then the port is closed. Otherwise we assume it is open.

Some people think UDP scanning is pointless. I usually remind them of the recent Solaris rcpbind hole. Rpcbind can be found hiding on an undocumented UDP port somewhere above 32770. So it doesn't matter that 111 is blocked by the firewall. But can you find which of the more than 30,000 high ports it is listening on? With a UDP scanner you can! There is also the cDc Back Orifice backdoor program which hides on a configurable UDP port on Windows machines. Not to mention the many commonly vulnerable services that utilize UDP such as snmp, tftp, NFS, etc.

Unfortunately UDP scanning is sometimes painfully slow since most hosts implement a suggestion in RFC 1812 (section 4.3.2.8) of limiting the ICMP error message rate. For example, the Linux kernel (in net/ipv4/icmp.h) limits destination unreachable message generation to 80 per 4 seconds, with a ¼ second penalty if that is exceeded. Solaris has much more strict limits (about 2 messages per second) and thus takes even longer to scan. *nmap* detects this rate limiting and slows down accordingly, rather than flood the network with useless packets that will be ignored by the target machine.

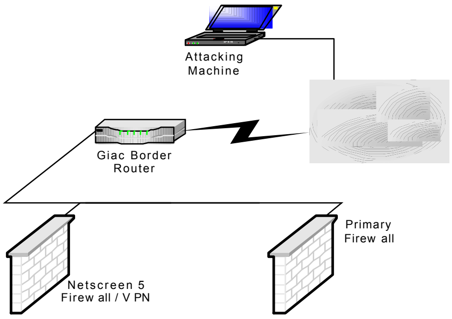
As is typical, Microsoft ignored the suggestion of the RFC and does not seem to do any rate limiting at all on Win95 and NT machines. Thus we can scan all 65K ports of a Windows machine very quickly.

The proceeding descriptions of the Scan Types is quoted from the NMAP man page

How to test the perimeter

The Border Router, Primary Firewall and the VPN Firewall are the devices to be tested as detailed above. A ping sweep will also show all the servers behind the firewall. The syntax of these tests are as follows:

This is the configuration of the equipment while the testing of the Perimeter Equipment. The attacking machine was running Redhat Linux 7.0 kernel 2.4.3 and had nmap and snort installed.



Border Router

FIN Steallth Scan to the Border Router

nmap -sF -O -PO -R aaa.bbb.90.252

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered. I would expect the finger print of the Cisco IOS to be detected

Acutal Results for FIN Stealth Scan

This output from NMAP shows the border router ip address was scanned, and the result was all 1062 port are filtered as per the expected result.

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 All 1062 scanned ports on  (aaa.bbb.90.252) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1298 seconds
```

ACL logging of the FIN Stealth scan attack as recorded by the syslog server. This shows the border router denying access and dropping the packet, and a log entry is recorded. The log entry is date and time stamped, the type of packet is indicated with the source and destination addresses with the ports following each address in the brackets.

```
Apr 17 21:13:05 xxx.yyy.91.254 7014: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(61391) -> aaa.bbb.90.252(6007), 1 packet
Apr 17 21:13:11 xxx.yyy.91.254 7015: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(61392) -> aaa.bbb.90.252(6007), 1 packet
Apr 17 21:13:17 xxx.yyy.91.254 7016: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(61391) -> aaa.bbb.90.252(6144), 1 packet
Apr 17 21:13:17 xxx.yyy.91.254 7017: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(61391) -> aaa.bbb.90.252(512), 1 packet
```

UDP Port Scan to the Border Router

nmap -sU -O -PO -R aaa.bbb.90.252

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find a udp port open for DNS lookup (53) and the IOS to be detected

Actual Results for UDP port scan

This output from NMAP shows the border router ip address was scanned using the UDP port scan, and the result was all 975 port are filtered. This was not the expected result but the filter is working as zone transfer are able to be completed. Due to there being no open tcp ports the operating system could not be fingerprinted, again not the expected result.

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Warning:  No TCP ports found open on this machine, OS detection will be MUCH
less reliable
All 975 scanned ports on  (aaa.bbb.90.252) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 1397 seconds
```

ACL logging of the port scan attack as recorded by the syslog server. This shows the border router denying access and dropping the packet, and a log entry is recorded. The log entry is date and time stamped, the type of packet is indicated with the source and destination addresses with the ports following each address in the brackets

```
Apr 17 20:24:23 xxx.yyy.91.254 6425: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56900) -> aaa.bbb.90.252(445), 1 packet
Apr 17 20:24:30 xxx.yyy.91.254 6426: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56901) -> aaa.bbb.90.252(445), 1 packet
Apr 17 20:26:48 xxx.yyy.91.254 6427: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56900) -> aaa.bbb.90.252(137), 1 packet
Apr 17 20:26:54 xxx.yyy.91.254 6428: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56901) -> aaa.bbb.90.252(137), 1 packet
Apr 17 20:27:12 xxx.yyy.91.254 6429: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56900) -> aaa.bbb.90.252(138), 1 packet
Apr 17 20:27:18 xxx.yyy.91.254 6430: %SEC-6-IPACCESSLOGP: list 101 denied udp
aaa.bbb.90.3(56901) -> aaa.bbb.90.252(138), 1 packet
```

Connect() Scan to the Border Router
nmap -sT -O -PO -R aaa.bbb.90.252

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result
To find all port to be closed or at least filtered. I would expect the finger print of the Cisco IOS to be detected

Actual Results for Connect() scan
This output from NMAP shows the border router ip address was scanned using the Connect() scan, and the result was all 1062 port are filtered. This was not the expected result but the filter is working as zone transfer are able to be completed. Due to there being no open tcp ports the operating system could not be fingerprinted, again not the expected result

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Warning:  No TCP ports found open on this machine, OS detection will be MUCH
less reliable
All 1062 scanned ports on  (aaa.bbb.90.252) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 1434 seconds
```

ACL logging of the port scan attack as recorded by the syslog server. This shows the border router denying access and dropping the packet, and a log entry is recorded. The log entry is date and time stamped, the type of packet is indicated with the source and destination addresses with the ports following each address in the brackets

```
Apr 17 21:22:21 xxx.yyy.91.254 7043: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(2887) -> aaa.bbb.90.252(6002), 1 packet
Apr 17 21:22:27 xxx.yyy.91.254 7044: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(2904) -> aaa.bbb.90.252(6002), 1 packet
Apr 17 21:22:45 xxx.yyy.91.254 7045: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(2950) -> aaa.bbb.90.252(6110), 1 packet
Apr 17 21:22:51 xxx.yyy.91.254 7046: %SEC-6-IPACCESSLOGP: list 101 denied tcp
aaa.bbb.90.3(2968) -> aaa.bbb.90.252(6110), 1 packet
```

Primary Firewall

FIN Stealth Scan to the Primary Firewall

nmap -sF -O -PO -R xxx.yyy.91.10

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered. I would have excepted to have the LinuX OS identified.

Acutal Results for FIN Stealth Scan

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

UDP Port Scan to the Primary Firewall

nmap -sU -O -PO -R xxx.yyy.91.10

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find a udp port open for DNS lookup (53) and I would have excepted to have the LinuX OS identified.

Actual Results for UDP port scan

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

Connect() Scan to the Primary Firewall

nmap -sT -O -PO -R xxx.yyy.91.10

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered. I would have excepted to have the LinuX OS identified.

Actual Results for Connect() scan

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

VPN / Firewall

FIN Stealth Scan to the VPN / Firewall

nmap -sF -O -PO -R xxx.yyy.91.11

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered.

Actual Results

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

UDP Port Scan to the VPN / Firewall

nmap -sU -O -PO -R xxx.yyy.91.11

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered.

Actual Results

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

Connect() Scan to the VPN / Firewall

nmap -sT -O -PO -R xxx.yyy.91.11

the options	-O	Operating System Detection
	-PO	Do not ping the test host
	-R	Resolve the port owner

Expected Result

To find all port to be closed or at least filtered.

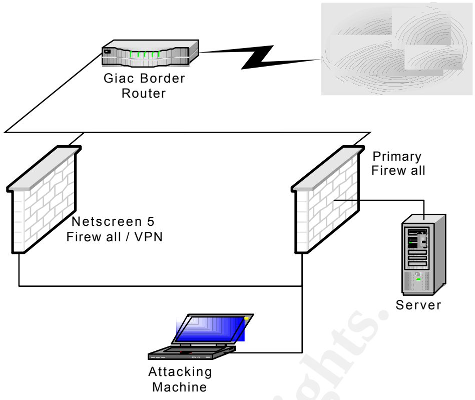
Actual Results

Due to the result being the same for the VPN / Firewall as the primary firewall and the Border Router, I have not included them again.

Assessing the Firewall

In the process of validating the implementation of the Primary firewall, the use of the following programs will confirm the firewall rules. The programs are ping, snort, nslookup and nmap. The rules used in assignment 2 for the security policy are used here to generate the firewall validation process.

This is the configuration of the network for the testing and validation of the firewall policies. The attacking machine was configured with Redhat Linux 7.0 and kernel 2.4.3. The machine was loaded with nmap and snort, and the usual ip utils load as a part of Redhat.



Does any server answer requests to port 80 and 443 except for the Web Servers
The tool of choice here is nmap, as it can scan the network looking for servers listening explicitly for the ports 80 and 443. The syntax for the command is

nmap -PT -p80,443 203.34.91.0-255

```
[root@localhost /root]# nmap -PT -p80,443 xxx.yyy.91.0-255
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
The 1 scanned port on (xxx.yyy.91.20)is: closed
Interesting ports on  (xxx.yyy.91.21):
(The 1 port scanned but not shown below is in state: closed)
Port        State       Service
80/tcp      open        http
443/tcp     open        https
Interesting ports on  (xxx.yyy.91.22):
Port        State       Service
80/tcp      open        http
443/tcp     open        https
The 1 scanned port on (xxx.yyy.91.23)is: closed
The 1 scanned port on (xxx.yyy.91.27)is: closed
All 2 scanned ports on  (xxx.yyy.91.254) are: closed
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 47 seconds
[root@localhost /root]#
```

The output of the scan shows 6 hosts as being available, of these two server responded with open ports. The two web servers responded, although the other server responded with at least one of these ports being closed. this is an acceptable risk as their address are changed through the NAT process an the port is port 80.

Does any server answer requests to port 53 other than the DNS server
The tool of choice here is nmap, as it can scan the network looking for servers listening explicitly for the ports 53. The syntax for the command is

nmap - sU -PO -p53 203.34.91.0-255

```
[root@localhost /root]# nmap -sU -p53 xxx.yyy.91.0-255
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (xxx.yyy.91.20):
Port        State       Service
53/udp     open        domain
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 45 seconds
[root@localhost /root]#
```

The output of the scan shows 6 hosts as being available, of these one server responded with open port 53.

Does any server answer requests to port 25 and 995 other than the Mail server

The tool of choice here is nmap, as it can scan the network looking for servers listening explicitly for the ports 25 and 995. The syntax for the command is

nmap -PT -p25,995 203.34.91.0-255

```
[root@localhost /root]# nmap -PT -p25,995 xxx.yyy.91.0-255
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (xxx.yyy.91.22):
Port       State       Service
25/tcp     open        smtp
995/tcp    open        pop3s
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 43 seconds
[root@localhost /root]#
```

The output of the scan shows 6 hosts as being available, of these one server responded with open port 25 and 995.

Does any server answer requests to port 3306 except for the database servers

The tool of choice here is nmap, as it can scan the network looking for servers listening explicitly for the ports 3306. The syntax for the command is

nmap -PT -p3306 203.34.91.0-255

```
[root@localhost /root]# nmap -PT -p3306 xxx.yyy.91.0-255
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 43 seconds
[root@localhost /root]#
```

The output of the scan shows 6 hosts as being available, none of the servers responded with this port. This is what was expected due to the fact that the database servers are in the Screened Data Services Network and therefore protect by the Primary firewall policies.

Does any server answer requests to port 514 except for the syslog server

The tool of choice here is nmap, as it can scan the network looking for servers listening explicitly for the ports 514. The syntax for the command is

nmap - sU -PO -p514 203.34.91.0-255

```
[root@localhost /root]# nmap -sU -p514 xxx.yyy.91.0-255
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (xxx.yyy.91.27):
Port       State       Service
514/udp    open        syslog
The 1 scanned port on  (xxx.yyy.91.254) is: closed
Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 45 seconds
[root@localhost /root]#
```

The output of the scan shows 6 hosts as being available, of these two servers responded with open port 514. The first was the Syslog server, which was expected, as was the border router but with the port closed. The IDS should not have responded as they are on the private addresses and not in the scan range.

Does the firewalls drop icmp packets and is the IDS logging

The tool of choice here is ping, as it can send an icmp echo request to the firewall ip addresses looking for an echo reply packet. Snort is used to verify the packet has reached the firewall, and is indeed dropping the packet. The syntax for the command is

ping 203.34.91.10

```
[root@localhost /root]# ping xxx.yyy.91.10
PING xxx.yyy.91.10 (xxx.yyy.91.10) from 172.16.2.66 : 56(84) bytes of data.

--- xxx.yyy.91.10 ping statistics ---
32 packets transmitted, 0 packets received, 100% packet loss
[root@localhost /root]#
```

The output of the ping shows a result of 100 % packet loss, with 32 packets being sent and none received. This is a good indicator of the firewall dropping the echo request. The snort log shows the echo leaving the interface never to return.

```
04/17-23:03:28.532375 0:10:A4:F5:71:CC -> 0:E0:29:18:E0:6F type:0x800 len:0x62
172.16.2.66 -> xxx.yyy.91.10 ICMP TTL:64 TOS:0x0 ID:38051 IpLen:20 DgmLen:84
Type:8  Code:0  ID:46369   Seq:0  ECHO
20 3F DC 3A 79 1F 08 00 08 09 0A 0B 0C 0D 0E 0F   ?.:y...........
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/17-23:03:29.531492 0:10:A4:F5:71:CC -> 0:E0:29:18:E0:6F type:0x800 len:0x62
172.16.2.66 -> xxx.yyy.91.10 ICMP TTL:64 TOS:0x0 ID:38052 IpLen:20 DgmLen:84
Type:8  Code:0  ID:46369   Seq:256  ECHO
21 3F DC 3A 06 1C 08 00 08 09 0A 0B 0C 0D 0E 0F  !?.:............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/17-23:03:30.531489 0:10:A4:F5:71:CC -> 0:E0:29:18:E0:6F type:0x800 len:0x62
172.16.2.66 -> xxx.yyy.91.10 ICMP TTL:64 TOS:0x0 ID:38053 IpLen:20 DgmLen:84
Type:8  Code:0  ID:46369   Seq:512  ECHO
22 3F DC 3A 07 1C 08 00 08 09 0A 0B 0C 0D 0E 0F  "?.:............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Assignment 4 - Design Under Fire (25 Points)

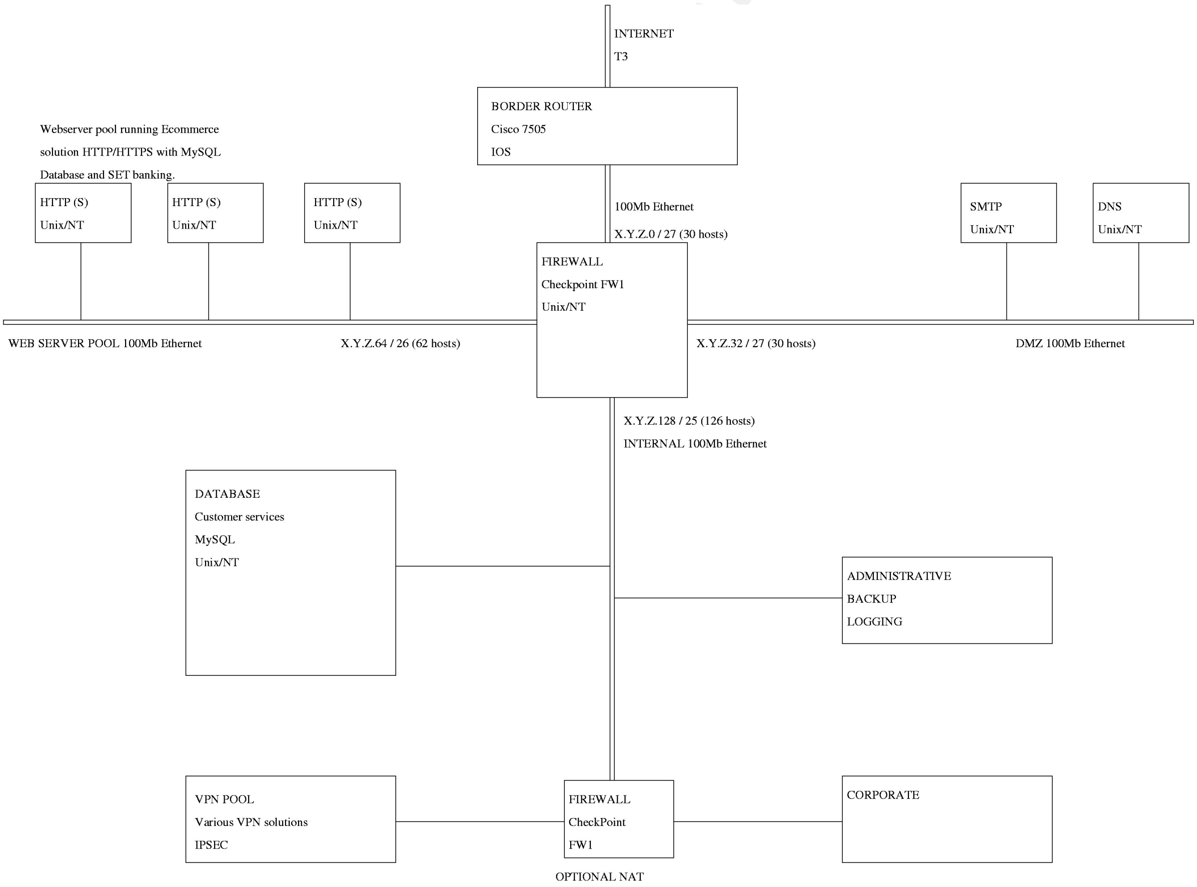
The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

1. *An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.*
2. *A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.*
3. *An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.*

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.*

For the requirements of the design under fire, I have chosen to use Vince Berk's practical. Vince's practical can be located at http://www.sans.org/y2k/practical/Vince_Berk_GCFW.zip. The reasons for choosing this practical is the primary firewall is a Checkpoint Firewall 1 solution, and through research on the web, there are known vulnerabilities with this firewall.



The Check Point, Firewall 1 software has multiple vulnerabilities, these are listed at at the Tech support site for CheckPoint. These vulnerabilities are listed below:

1. ACK Dos Attack
2. IP Fragment DoS Vuknerability
3. Passive FTP Vulnerability
4. Fast Mode Vulnerability
5. Denial of Service reported on RealSecure Netwrok Sensor.

The type of attack, which is most devastating to the architecture, which Vince has chosen is the IP Fragment driven Denial of Service vulnerability. This is the attack chosen, due all traffic to travel across this device. The Firewall 1 product reassembles all IP fragments of datagram prior to the inspection of the packet by the firewall policies. This allows a large stream of IP fragments to cause the firewall 1 code programming that logs the fragments and compiles the fragments to consume most if not all of the CPU time. Therefore stopping the movement of any other traffic into or out of the internal network to the internet.

When an attack of this type is required the program of choice would be the jolt2, which sends a stream of extremely large packets to the firewall which inturn consumes large amounts of CPU time and eventually may even consume 100% of the CPU time. To develop an attack of the Distributed Denial of Service type, there are other programs of choice which all operate basically on the same principle. These progams will deploy a slave or client process on a compromised computer which talks to a master or server process.

The program of choice here is the TFN2K, which stands for TFN 2000 the successor to the original TFN. This DDos tool also deploys a client process on the compromised computer. To implement this type of an attack, you would need to compromise numerous hosts on the internet, and because the home users are not as security aware these systems would be an easy mark, particularly the user using cable modems and the like which is required for the speed on the links. As these are compromised then the slave process is installed.

TFN2K does not communicate on fixed ports like some of the DDoS tools, but can utilise randomised communications on ports and encryption. This cause a few problems as prevention can not be implemented at the border router and the encryption method can negate the Network Intrusion Dection Systems, therefore making this DDoS tool a nasty bit of work.

To prevent this attack, Checkpoint have built in two features into there stateful packet inspect engine, which can be enabled to fend off the SYN flood attacks. These features are the SYNDefender Relay and the SYNDefender Gateway, both of these features can work in parallel. The SYNDefender Relay ensures there has been an actual three way hand shake completed before the connection is passed into the internal network. The SYNDefender Gateway is used to protect against the shear volumes of packets used in this attack, by moving legitimate packets through the queue and considers it an open connection, much like the Syn Cookies of the Linux Kernel.

Other countermeasures would be to bastille the server the Checkpoint Firewall1 is running on, by eliminating any no essential services and by marinating patches and permissions up to date. By using rate filters which can limit the number of connections per second to the firewall would protect the firewall. An example of this is implemented in the Cisco IOS 12 and later, which is called the ICMP rate filter.

To compromise an internal server on this network would require access through both a Cisco Border router and the Checkpoint Firewall 1. Because of this fact I have chosen to attack one of the Web servers as there is already holes in both the router and the firewall for web based protocols.

The initial strategy is to use tool like nmap to run a TCP Connect scan and a UDP port scan to gain some sort of knowledge about the server, what services are running on it, what ports are open. The other way to gather information about the server is to look at the web page sources coming off the server. Do they include CGI scripts, ASP scripts or any other scripts that could be carelessly written with vulnerabilities. Is the web server running Front Page extensions? Microsoft's Front page extensions are a known vulnerability on IIS servers. Depending on what is found on the server would dedicate the style of attack.

If front page extensions were found to be enabled on the server, then this would be the point of attack. Front page extensions have poor security and if misconfigured, access is gained into the root directory of the website. With this sort of access, I would load a script, which would create a new user and password onto the server then giving access to the server and the launching pad to the entire network.

REFERENCES

Brenton Chris. Firewalls 101: Perimeter Protection with Firewalls. 2.2. Sans Institute

Brenton Chris. Advanced Perimeter Protection and Defense. 2.3. Sans Institute

Anonymous. Maximum Linux Security. Indianapolis, Indiana. Sams Publishing. 1999

Ogletree Terry. Prctical Firewalls. Indianapolis, Indiana. Que

Odom Wendell. Cisco CCNA Certification Guide. Indianapolis, Indiana. Cisco Press 2000

Scambray Joel, McClure Stuart, Kurtz George. Hacking Exposed : Network Security Secrets & Solutions. Berkley. Osborne / McGraw-Hill. 2000

Northcutt Stephen, Novak Judy. Network Intrusion Detection: An Analyst's Handbook. Indianapolis, Indiana. New Riders. 2000

Petersen Richard. Linux: The Complete Reference. Berkley Osborne / McGraw-Hill. 2001

NetScreen-5 User's Guide. USA. NetScreen Technologies 2000

Checkpoint Firewall-1 Technical Support April 2001
http://www.checkpoint.com/techsupport/alerts/

NMAP -- The Network Mapper April 2001
http://www.insecure.org/nmap/

Linux Ethernet Bridging Code
http://www.math.leidenuniv.nl/~buytenh/bridge/

Netscreen Product Range
http://www.netscreen.com/products/

Linux netfilter Hacking HOWTO
http://netfilter.filewatcher.org/unreliable-guides/netfilter-hacking-HOWTO/

http://www.boingworld.com/workshops/linux/iptables-tutorial/rc.firewall.txt

Linux Kernel Source 2.4.3
http://www.kernelnotes.org/

SANS Reading Room: Firewalls & Perimeter Protection
http://www.sans.org/infosecFAQ/firewall/firewall_list.htm

Screening Router Access List
http://pasadena.net/cisco/secure.html