GCFW Practical v1.6a

Ma Huijuan

June 2002

Table of Contents

Assignment 1 Security Architecture

1.3 Introduction

In this part, security architecture for GIAC Enterprises is defined. GIAC Enterprises is a startup e-business which deals in the online sale of fortune cookie sayings. At present, its product is provided by 2 suppliers. The company has about 40 employees in various departments: Sales and Marketing, HR, Finance, IT and Management. The personnel in Sales and Marketing are traveling constantly and they need to securely connect to the internal network on traveling. The company has about 20 customers who purchase bulk online fortunes, and 2 international partners who translate and resell fortunes.

As an e-business enterprise, all the transactions are performed on-line with suppliers, customers and partners. Security plays a vital role for GIAC Enterprises in its survival and prosperity. Website defacement may undermine customer confidence. Data theft, downtime and virus may directly incur loss in revenue. How to achieve and maintain data confidentiality, integrity and availability while fulfilling business operation is the goal of this network design.

In the security architecture that we designed, access requirements and restrictions for the following are considered:

- Customers (the companies that purchase bulk online fortunes);
- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);
- Partners (the international partners that translate and resell fortunes);
- GIAC Enterprises (the employees located on GIAC's internal network).
- GIAC remote employees.

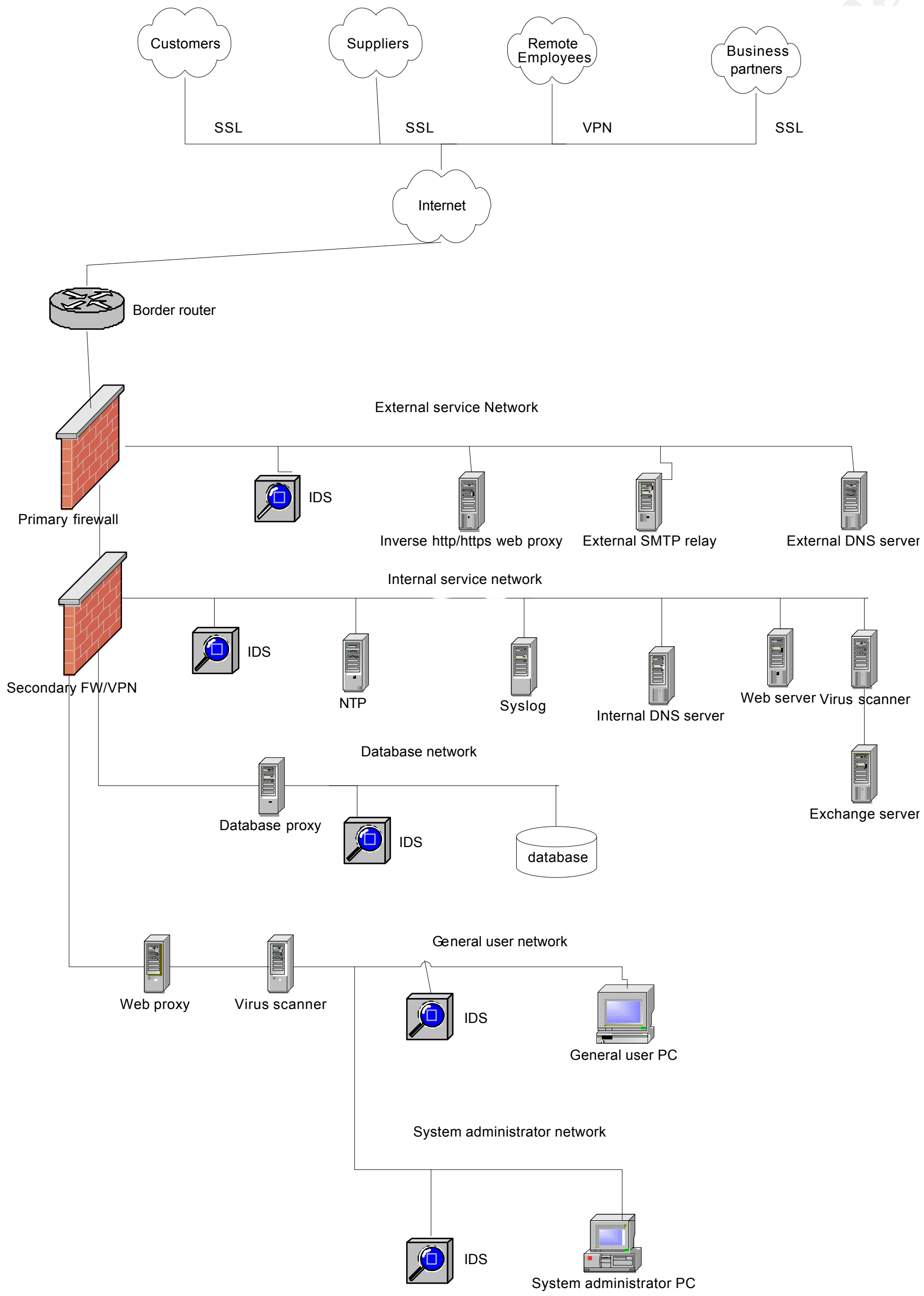
Our architecture includes the following components:

- filtering routers;
- firewalls;
- VPNs to business partners, customers, and suppliers;
- internal firewalls for additional, layered protection;
- secure remote access for remote employees.

In the following sections, access requirements and restrictions for customers, suppliers, partners, GIAC local employees and GIAC remote employees will be detailed first. After that, security features in various components of the design will be introduced. Following this, the components of perimeter defense will be detailed.

1.4 Access Requirements and Restrictions

In order to understand the access requirements and restrictions for various parties, some understanding of our network design is needed. Below is our network design diagram.



From the diagram, it can be seen that our network design includes the following components:

- Border router
- Primary firewall
- Secondary firewall/VPN
- External service network
- Internal service network
- General user network
- System administrator network.

Border router, primary firewall and secondary firewall/VPN are network devices with security functionality; no service is provided. In external service network, there are IDS, inverse http and https web proxy, external SMTP relay server and external DNS server. In internal service network, there are IDS, NTP server, syslog server, exchange server, internal DNS server and web server. In database network, there are database proxy, IDS and database server. In user network, there are IDS, virus scanning server, web proxy, and desktops of the general users except system administrators. In the system administrator network, there are IDS, virus scanning server, web proxy, and desktops of the system administrators.

The central business applications required by GIAC Enterprises are as follows:

Service	Application	Port
DNS	BIND	TCP/53, UDP/53
Mail	SENDMAIL, EXCHANGE	TCP/25
Web application	HTTP, HTTPS	TCP/80, TCP/443
Database	Oracle	TCP/1521

Accounting, auditing and logging applications:

Service	Application	Port
Logging	syslog	UDP/514
Time synchronization	Ntp	UDP/123

Below are the access requirements and restrictions for various parties for GIAC's business operation.

- Customers are the companies that purchase bulk online fortunes. They should be able to access and purchase the fortunes from the GIAC online securely. They require https connect to the inverse http/https web proxy in the external service network securely using SSL at tcp port 443. This kind of VPN implementation is used because it has no extra requirements to customer's software or hardware configuration. The inverse http/https web proxy will access the web server in internal service network, and the web server in internal service network will access the database proxy in the database network, then the database proxy will access the database server if needed.

- Suppliers are the authors of fortune cookie sayings that connect to supply fortunes. They should be able to submit fortunes to GIAC online. They require https connect to the inverse http/https web proxy in the external service network securely using SSL at tcp port 443. This kind of VPN implementation is used because it has no extra requirements to supplier's software or hardware configuration. The inverse http/https web proxy will access the web server in internal service network, and the web server in internal service network will access the database proxy in the database network, then the database proxy will access the database server if needed.

- Partners are the international partners that translate and resell fortunes. They should be able to access and update the internal resources. They require https connect to the inverse http/https web proxy in the external service network securely using SSL at tcp port 443. This kind of VPN implementation is used because it has no extra requirements to partner's software or hardware configuration. The inverse http/https web proxy will access the web server in internal service network, and the web server in internal service network will access the database proxy in the database network, then the database proxy will access the database server if needed.

- GIAC Enterprises employees located in the general user network need to surf the internet, receive and send email, as well as access and update internal resources. Thus they need access to the outside world at tcp port 80, 443, and 21, to the exchange server (at tcp port 25) and the internal DNS server (at tcp and udp port 53) in he internal service network, as well as to the oracle database proxy (at tcp port 1521) in the database network.

- GIAC remote employees should have the same access requirements and restrictions to the GIAC local employees. In their notebook, SecureClient is installed. They use their notebook to VPN connect to the secondary FW/VPN, the secondary FW/VPN connects them to the resources they require.

- System administrators in the system administrator network should have the same access to the general user. Besides, they need secure-shell access to the servers and network devices they are in charge of (at tcp port 22).

- All access not explicitly defined above should be denied.

1.3 Security Features in the Network Architecture

Border Router
Besides routing the packets, the router acts as the first layer of defense in-depth by providing static filtering function to prevent spoofing and DOS attack.

- Ingress filtering blocks inbound packets with source IP of internal network, loopback address, private address and non-routable address defined by the Internet Assigned Numbers Authority (IANA) in RFC 1918, as well as packets with destination IP of internal broadcast address Block.

- Egress filtering allow outbound packets with source IP of GIAC's address space and block and log all other traffic.
- Ingress/egress filtering blocks inbound and outbound packets to critical services such as NetBIOS traffic.
- The router also blocks some type of unusual packets, such as packets with source routing option set. This is to prevent backdoor exploits.

Primary Firewall

The second layer of perimeter defense is the primary firewall. This is needed, because static packet filtering provided by the border router evaluates each packet based on the packet header information itself, causing the following problem:

- It cannot maintain the packets that has been processed, thus may inadvertently allow through crafted packets where the attacker has set the ACK bit to 1.
- It cannot inspect packet payload, thus may cause problems with complex protocols such as FTP and DCOM.

In order to solve the above problem, this stateful primary firewall is used. It acts as the second layer of defense in-depth by providing stateful filtering and stateful inspection. By stateful filtering, a packet is matched against a connection table prior to rulebase processing. By stateful inspection, the firewall can analyze the payload, thus it can understand and handle complex protocols.

Secondary FW/VPN

The secondary FW/VPN combines the functionality of firewall with VPN. The secondary firewall is also stateful, same as the primary firewall. However, more security feature is provided by the secondary firewall, as a different product will be used. (In our design, the primary firewall uses IPTABLES, whereas the secondary firewall uses Check Point FW-1). This provides more security, as there is less probability that two different firewall has the same vulnerability. Thus, the probability that the hacker penetrates into internal network is greatly reduced.

The VPN here grants or block connections to the internal network from remote employees. Tunnel mode ESP will be used to provide maximum confidentiality, data integrity and user authentication.

The firewall also performs network address translation (NAT). Hide NAT will be used for the general user network, so that general user within GIAC can have access to Internet resources, while none of the systems are accessible from the internet.

Web Service

Customers, suppliers and partners do business with GIAC via the web service provided by the inverse http/https web proxy located in the external service network. They can browser the general information web page using http protocol. When confidential information is transmitted, https protocol is used.

The following security features are used regarding web service:

- Https protocol is used to provide data confidentiality, integrity and authentication.
- Users access to the inverse web proxy, not directly to the web server. This provides one more layer of protection.
- The inverse web proxy can filter out malformed traffic, virus, etc.

Email Service

GIAC employee sends and receives email through the exchange server located in the internal service network. Outside world uses the external SMTP server located in the external service network to send email to GIAC employees. The following security features are used regarding email service:

- The exchange server is protected by a virus scanner. Emails sent and received by GIAC employees are scanned first before processed.
- In order to send email to GIAC, the outside world must relay their email through the external SMTP relay server. They don't have direct access to the exchange server, thus adds one more layer of security.

DNS Service

Split DNS is used. Split DNS is the design of separating the internal DNS servers from the external DNS servers. The data in these servers should be completely different. The internal servers only contain internal DNS entries and the external server only contains external entries. Hackers love to find DNS servers that are not split and expose internal host to the Internet. They can use this information to get the IP address of internal hosts and find out specific information about the internal configuration of the network. If an attacker can see an hostname called fw01, he can easily assume it is a firewall and begin attacking it for vulnerabilities. The attacker may not have known the IP address of the firewall because ping usually is disabled by the firewall. But now the attacker knows the IP address of the firewall and many other network devices, that he can now launch attacks against." http://www.sans.org/infosecFAQ/DNS/sec_DNS.htm

GIAC's internal DNS server located in the internal service network is configured as slave to the external DNS server located in the external service network, thus the internal DNS server cannot request DNS lookups directly to Internet hosts. Zone transfers will not allow between internal and external DNS servers and zone transfers will not be allowed to external hosts.

Database

In order to penetrate to the database server, the outside world has to penetrate through the following devices: inverse web proxy, web server, then the database proxy, then the database server. Alternatively, they can penetrate to the database server by compromising the border router, primary firewall, secondary FW/VPN, database proxy, then to the database proxy. This adds several layer of protection to the database server.

IDS
GIAC Enterprise's IDS (Intrusion Detection System) devices are located in the external service network, internal service network, database network, general user network as well as system administrator network. IDS are used within the network to provide one more layer of security in case the perimeter device or the internal system does go wrong. IDS will alert the personnel involved to investigate and resolve the matter.

Logging
All the perimeter devices, IDS, and servers log to the syslog server located in the internal service network. Hackers will attempt to delete or modify systems logs in case they comprise the system, so they can go undetected. Remote logging can enable us to detect them.

An NTP server is located in the internal service network. All perimeter device and internal systems synchronize to this NTP server. This way, the logs will be more manageable.

Host-Based Security

- All perimeter devices, servers and desktops should install the latest OS with the latest patches applied.
- The systems should open only the ports that are needed by the business operation.
- Anti-virus software and host-based firewall should be installed in the desktop and remote employee's notebook. Virus definition and intrusion signature should be kept up-to-date.

Defense in Depth
The above constitutes the concept of defense-in-depth in the network design.

1.4 Perimeter Components

Border Router
GIAC Enterprises connects to the ISP via a Cisco 3640 router with a T1 CSU/DSU module for the PPP connection. This model is chosen because its capacity is suitable for the rate of traffic to and from the GIAC Enterprises network. It is running IOS 12.1 with the latest patches installed.

Primary Firewall
The primary firewall is running netfilter iptables v1.2.4 installed via a base install of Red Hat 7.2. Bastille Linux 1.3.pre10 on the firewall server is used to further secure and harden the operating system. For a more detailed view of Bastille, please visit Jay Beale's web site at http://www.bastille-linux.org

Secondary FW/VPN
This box uses a security-hardened and fully-patched Solaris 8.0 OS using the latest patch on a Sunblade 1000 Sparc server. Check Point version NG FW-1/VPN-1 application is used. The latest service packs and patches are applied on Checkpoint FW-1/VPN-1 application.

Inverse Http/Https Web Proxy
We uses the Solaris 8 on Sparc 10 Platform running Squid Proxy version 2.4 due to its support of proxying for both HTTP and HTTPS. Besides, extensive access controls are also available. Both the OS and the Squid are fully patched and hardened.

Web Proxy
The netcache proxy server is used.

Database Proxy
Gauntlet SQL proxy firewall 6.0 is used.

External SMTP Relay
SendMail version 8.12.3 on a fully hardened Sunblade 1000 Sparc server is used.

Virus Scanner
TrendMicro InterScan VirusWall is used.

IDS
We use Snort IDS 1.8.4 resides on fully hardened and patched RedHat 7.2.

Assignment 2 Security Policy

2.6 Introduction

This part is to provide a security policy for the following three components based on the security architecture defined in Assignment 1: border router, primary firewall and the VPN. For each component, the access requirements for internal users, customers, suppliers, and partners that we defined in Assignment 1 will be considered. The policies we define will accurately reflect those business needs as well as appropriate security considerations.

A tutorial is also included on how to implement the policy for the primary firewall. Screen shots, network traffic traces, firewall log information, and URLs are used to find further information and to clarify the instructions. The following are included:

- A general explanation of the syntax or format of the ACL, filter, or rule for the device.
- A general description of each of the parts of the ACL, filter, or rule.
- A general explanation of how to apply a given ACL, filter, or rule.
- For each ACL, filter, or rule in the security policy, the following info is described:
    - the service or protocol addressed by the rule, and the reason this service might be considered a vulnerability.
    - Any relevant information about the behavior of the service or protocol on

the network.

    - If the order of the rules is important, include an explanation of why certain rules must come before (or after) other rules.

After that, three sample rules from the primary firewall would be tested to make sure these three rules are working properly. Procedures to perform to test will be detailed.

2.7 Border Router Rules

The border router Cisco 3640 is our first line of defence, which performs static packet filtering besides routing. The function of reflexive access control is not utilised for performance reason. Only standard and extended access lists function will be used to perform filtering.

Ingress Filtering

- Block inbound packets with source IP of internal network to prevent spoofing.
- Block inbound packets with source IP of loopback address and private address to prevent spoofing and several DOS exploits attack.
- Block inbound packets with destination IP of internal broadcast address to prevent probes as well as to prevent being used as a SMURF amplifier.
- Block inbound packets with non-routable source IP address defined by the Internet Assigned Numbers Authority (IANA) in RFC 1918 to prevent spoofing.

```
access-list 101 deny ip 166.166.166.0 0.0.0.255 any log-input
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 169.254.0.0 0.0.255.255 any log-input
access-list 101 deny ip 172.16.0.0 0.15.255.255 any log-input
access-list 101 deny ip 192.0.2.0 0.0.0.255 any log-input
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log-input
access-list 101 deny ip 224.0.0.0 15.255.255.255 any log-input
access-list 101 deny ip 240.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 248.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 255.255.255.255 0.0.0.0 any log-input
access-list 101 deny ip 0.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 1.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 2.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 5.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 7.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 10.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 14.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 23.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 27.0.0.0 0.255.255.255 any log-input
```

```
access-list 101 deny ip 31.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 36.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 37.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 39.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 41.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 42.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 49.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 50.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 58.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 59.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 60.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 69.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 70.0.0.0 1.255.255.255 any log-input
access-list 101 deny ip 72.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 82.0.0.0 1.255.255.255 any log-input
access-list 101 deny ip 84.0.0.0 3.255.255.255 any log-input
access-list 101 deny ip 88.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 96.0.0.0 31.255.255.255 any log-input
access-list 101 deny ip 197.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 201.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 221.0.0.0 0.255.255.255 any log-input
access-list 101 deny ip 222.0.0.0 1.255.255.255 any log-input
access-list 101 deny ip 224.0.0.0 15.255.255.255 any log-input
access-list 101 deny ip 240.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 248.0.0.0 7.255.255.255 any log-input
access-list 101 deny ip 255.255.255.255  0.0.0.0 any log-input
access-list 101 accept any any
```

Egress filtering at the router

- Allow outbound packets with source IP of GIAC's address space 166.166.166.0.
- Block and log all other traffic to prevent spoofing and identify problem.

```
access-list 102 permit ip 166.166.166.0 0.0.0.255 any log-input
access-list 102 deny any log-input
```

Ingress/Egress Filtering

- Block and log inbound packets with TCP and UDP port 135-139, 445 and 111, TCP port 23, 512-514, 2049, 6000-6255 and UDP ports 69, 161-162 and 514. This is to protect internal critical services from being exploited by outside attacker.
- Block and log outbound packets with TCP and UDP port 135-139, 445 and 111, TCP port 23, 512-514, 2049, 6000-6255 and UDP ports 69, 161-162 and 514. This is to prevent internal user and comprised systems from exploiting these

services to attack outside systems, as well as to identify problem.

```
access-list 103 deny tcp any any range 135 139 log-input
access-list 103 deny udp any any range 135 139 log-input
access-list 103 deny tcp any any eq 445  log-input
access-list 103 deny udp any any eq 445  log-input
access-list 103 deny tcp any any eq 111  log-input
access-list 103 deny udp any any eq 111  log-input
access-list 103 deny tcp any any eq 23 log-input
access-list 103 deny tcp any any range 512 514  log-input
access-list 103 deny tcp any any eq 2049  log-input
access-list 103 deny tcp any any eq 6000 6255  log-input
access-list 103 deny udp any any eq 69 log-input
access-list 103 deny udp any any eq 161  log-input
access-list 103 deny udp any any eq 162  log-input
access-list 103 deny udp any any eq 514  log-input
access-list 103 permit  any any
```

Others

- Block all inbound and outbound packets with source routing option set to prevent backdoor exploits.

```
no ip source-route
```

Router Hardening

Besides configure the router to do routing and static packet filtering, as detailed above, the router itself should be hardened so that the router itself is not comprised.

- Restrict login access.

```
line con 0
login local
exec-timeout 5 0
line aux 0
login local
exec-timeout 0 1
no exec
```

- Encrypt the password.

```
config
enable secret
service password-encryption
```

- Disable SNMP, echo, discard, chargen, daytime services, finger service, HTTP and

BOOTP.

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

- Cisco discovery protocol is not needed, and we do not want to accept source routed packets. We are using static routes so the default RIP should be turned off.

```
no cdp run
no router rip
```

- Set the login banner

```
banner /
WARNING: Unauthorised access is not allowed!
/
```

- log all events to a remote logging host.

```
no logging console
logging buffered
logging <IP of syslog server>
```

2.8 Primary Firewall Rules

The first step is variable definition, module loading, parameter setting, and iptables initialization. The purpose of variable definition is to make the script more readable and portable. In case of IP or subnet changes, the script can be easily changed to suit the new environment. Module loading and parameter setting is to make the iptables work properly. Iptables initialization is to delete all existing user-defined chains and flush all chain rules.

```
#!/bin/sh
# variable definition

#inverse http/https web proxy
EXT_HTTP_IP="166.166.166.3"
#web server
INT_HTTP_IP=”166.166.14”

#external dns server
EXT_DNS_IP="166.166.166.5"
#internal dns server
INT_DNS_IP="166.166.166.13"

#external smtp relay server
EXT_SMTP_IP="166.166.166.4"
#exchange server
INT_SMTP_IP="166.166.166.12"

#NTP server
NTP_IP="166.166.166.10"

#syslog server
SYSLOG_IP="166.166.166.11"

#IDS in the external service netwrok
IDS_IP="166.166.166.2"

#interface of the FW to router
IFACE_ROUTER="eth0"
#interface of the FW to the external service network
IFACE_EXT="eth1"
#interface of the FW to the secondary FW/VPN
IFACE_INT="eth2"

IPTABLES="/usr/sbin/iptables"

#IP of the system administrator
SYSADMIN_IP=”166.166.166.25”
#firewall IP at the eth0
FW_IP=”166.166.166.33”
#External service network
EXT_SERVICE_NET=”166.166.166.0/29”
#Internal service network
INT_SERVICE_NET=”166.166.166.8/29”
```

```
# module loading
/sbin/depmod –a
/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state
```

```
# parameter set up.
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
# iptables initialization
iptables –F
iptables –X
```

The second step is to define the filtering rules, which is the core of the script. Default policies are defined first so that the default policy is to drop all.

```
IPTABLES -P INPUT DROP
IPTABLES -P OUTPUT DROP
IPTABLES -P FORWARD DROP
```

In iptables, there are three built-in chains, namely INPUT, OUTPUT and FORWARD. The chain INPUT is used to control all traffic sent to the firewall itself. The chain OUTPUT is used to control all traffic sent from the firewall. The FORWARD chain is where we define the policy for traffic attempting to pass through the firewall. This is the chain where most of the works are done.

The INPUT and OUTPUT chains are configured to manage the traffic to and from the firewall itself. System administrator has to be able to connect to the firewall using ssh.

```
IPTABLES -A INPUT -m state --state NEW,ESTABLISHED -p tcp -i eth2 -s
$SYSADMIN_IP -d $FW_IP --dport 22 -j ACCEPT
IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -p tcp -o eth2 -d
$SYSADMIN_IP -s $FW_IP --sport 22 -j ACCEPT
```

The system administrator also needs ICMP to determine the health of the firewall.

```
IPTABLES -A INPUT –i eth1 -s $SERVICE_NET -p icmp -j ACCEPT
IPTABLES -A OUTPUT -o eth1 -d $SERVICE_NET -p icmp -j ACCEPT
```

Localhost traffic should be allowed since it is the firewall communicating with itself.

```
IPTABLES -A INPUT -i lo -s 127.0.0.1 –j ACCEPT
IPTABLES -A OUTPUT -o lo -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

The rules below allow the firewall to perform time synchronization, name resolution and remote logging

```
#Allow syslog out
IPTABLES -A OUTPUT -m state --state NEW,ESTABLISHED -p udp -o eth2 -d
$SYSLOG_IP --dport 514 -j ACCEPT

#Allow DNS queries out and reply back in
IPTABLES -A OUTPUT -m state --state NEW,ESTABLISHED -p udp -o eth1 -d
$EXTDNS_IP --dport 53 -j ACCEPT
IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -p udp -i eth1 -s
$EXTDNS_IP --sport 53 -j ACCEPT

#Allow NTP queries out and reply back in
IPTABLES -A OUTPUT -m state --state NEW,ESTABLISHED -p udp -o eth1 -d
$NTP_IP --dport 123 -j ACCEPT
IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -p udp -i eth1 -s
$NTP_IP --sport 123 -j ACCEPT
```

All other traffic to or from the firewall itself are dropped and logged by the default policy for the chains INPUT and OUTPUT.

Using the above rule set, the firewall is invisible to the universe except the system administrator. This provides maximum protection to the firewall itself.

The FORWARD chain is configured to manage the traffic attempting to pass through the firewall. The following types of malformed tcp packets are blocked and logged: new connection with no SYN bit set, SYN/FIN packets and NULL packets.

```
IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j LOG   --log-
prefix "New not syn:"
IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j DROP

IPTABLES -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP

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

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

The system administrator needs to access the external service network using ssh.

```
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth2 -s
SYSADMIN_IP -o eth1 -d $EXT_SERVICE_NET --dport 22 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s
EXT_SERVICE_NET --sport 22 -o eth2 -d $SYSADMIN_IP -j ACCEPT
```

Allow through HTTP, HTTPS, SMTP and DNS from internet to the corresponding servers in the External Service Network.

```
#HTTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth0 -s
0.0.0.0/0 -o eth1 -d $EXT_HTTP_IP --dport 80 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s
$EXT_HTTP_IP -o eth0 -d 0.0.0.0/0 --sport 80 -j ACCEPT

#HTTPS
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth0 -s
0.0.0.0/0 -o eth1 -d $EXT_HTTP_IP --dport 443 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s
$EXT_HTTP_IP -o eth0 –d 0.0.0.0/0 --sport 443 -j ACCEPT

#DNS
#Tcp port 53 is not allowed to be accessed from outside.
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth0 -s
0.0.0.0/0 -o eth1 -d $EXT_DNS_IP --dport 53 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth1 -s
EXT_DNS_IP -o eth0 -d 0.0.0.0/0 --sport 53 -j ACCEPT

#SMTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth0 -s
0.0.0.0/0 -o eth1 -d $EXT_SMTP_IP --dport 25 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s
EXT_SMTP_IP -o eth0 -d 0.0.0.0/0 --sport 25 -j ACCEPT
```

Allow through SMTP, DNS from the corresponding servers in the external service network out to the internet.

```
#DNS
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth1 -s
$EXT_DNS_IP --sport 53 -o eth0 -d ! 0.0.0.0/0 --dport 53 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth0 -s 0.0.0.0/0 --
sport 53 -o eth1 -d $EXT_DNS_IP --dport 53 -j ACCEPT

#SMTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth1 -s
$EXT_SMTP_IP -o eth0 -d 0.0.0.0/0 --dport 25 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth0 -s 0.0.0.0/0 --
sport 25 -o eth1 -d $EXT_SMTP_IP -j ACCEPT.
```

Allows NTP server in the internal service network to go to the internet.

```
#NTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth1 -s
$NTP_IP --sport 123 -o eth0 -d 0.0.0.0/0 --dport 123 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth0 -s 0.0.0.0/0 --
sport 123 -o eth1 -d $NTP_IP --dport 123 -j ACCEPT
```

Allow through Syslog and NTP traffic from router and external service network to the syslog and ntp server.

```
#Syslog
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth0 -s
$ROUTER_IP -o eth1 -d $SYSLOG_IP --dport 514 -j ACCEPT
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth0 -s
$EXT_SERVICE_NET -o eth1 -d $SYSLOG_IP --dport 514 -j ACCEPT

#NTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth0 -s
$ROUTER_IP --sport 123 -o eth1 -d $NTP_IP --dport 123 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth1 -s $NTP_IP --
sport 123 -o eth0 -d $ROUTER_IP --dport 123 -j ACCEPT

IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth0 -s
$EXT_SERVICE_NET --sport 123 -o eth1 -d $NTP_IP --dport 123 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth1 -s $NTP_IP --
sport 123 -o eth0 -d $EXTERNAL_SERVICE_NET --dport 123 -j ACCEPT
```

Allow inverse http/https web proxy to web server.

```
IPTABLES –A FORWARD –m state –state NEW,ESTABLISHED -p tcp -i eth1 -s
$EXT_HTTP_IP -o eth2 -d $INT_HTTP_IP --dport 80 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth2 -s
$INT_HTTP_IP --sport 80 -o eth1 -d $EXT_HTTP_IP -j ACCEPT
```

Allow communication between external SMTP relay server with the exchange server, and between external DNS server and the internal DNS server.

```
#SMTP
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth1 -s
$EXT_SMTP_IP -o eth2 -d $INT_SMTP_IP --dport 25 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth2 -s
$INT_SMTP_IP --sport 25 -o eth1 -d $EXT_SMTP_IP -j ACCEPT

IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth2 -s
$INT_SMTP_IP -o eth1 -d $EXT_SMTP_IP --dport 25 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s
$EXT_SMTP_IP --sport 25 -o eth2 -d $INT_SMTP_IP -j ACCEPT

#DNS
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -i eth2 -s
$INT_DNS_IP --sport 53 -o eth1 -d $EXT_DNS_IP --dport 53 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -i eth1 -s
$EXT_DNS_IP --sport 53 -o eth2 -d $INT_DNS_IP --dport 53 -j ACCEPT

IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p udp -o eth2 -d
$INT_DNS_IP --dport 53 -i eth1 -s $EXT_DNS_IP --sport 53 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p udp -o eth1 -d
$EXT_DNS_IP --dport 53 -i eth2 -s $INT_DNS_IP --sport 53 -j ACCEPT
```

Allow user connection to the internet through the web proxy server.

```
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tdp -o eth0 -d
0.0.0.0/0 --dport 80 -i eth2 -s $WEB_PROXY_IP --sport any -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tdp -o eth2 -d
$WEB_PROXY_IP --dport any -i eth0 -s 0.0.0.0/0 --sport 80 -j ACCEPT

IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tdp -o eth0 -d
0.0.0.0/0 --dport 443 -i eth2 -s $WEB_PROXY_IP --sport any -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tdp -o eth2 -d
$WEB_PROXY_IP --dport any -i eth0 -s 0.0.0.0/0 --sport 443 -j ACCEPT
```

All others not explicitly defined above are dropped by the default policy.

2.9 VPN Rules

In our design, Check Point FW-1/VPN-1 resides on the same machine. The VPN-1 is responsible for establishing and maintaining the connections from remote employee to the internal service network and the database network. RemoteClient is installed in remote employee's notebook to connect to GIAC using VPN.

Split Tunnelling

According to http://searchnetworking.techtarget.com/tip/1,289483,sid7_gci783169,00.html, split tunneling refers to having separate paths. For example, let's assume a VPN tunnel from your head-end VPN concentrator in your DMZ terminates on a remote user's laptop, which is directly connected to the Internet. If split tunneling is enabled, the traffic from the user's laptop to the Internet will go from the users laptop to the Internet. If split tunnels are disabled, this same traffic will be forced to go across the Internet through the VPN tunnel to your head-end concentrator in your DMZ, where it will do a hairpin turn and go back out through your firewall to its Internet destination.

The downside to using split tunnels is security. If your remote users are allowed to access the Internet, you know they'll be downloading all kinds of high-risk material. And you know that even with a "personal firewall" the average user's laptop is far from secure. With a split tunnel, if an intruder can find just one vulnerability in a remote desktop, they can bypass your firewall and access your internal network with the same privileges that the real user has. This is a frightening prospect indeed. By disabling split tunnels, you prevent any direct communication with the Internet (other than your VPN), which goes a long way towards securing the device. This is critical when the users are using their personal machines from home, which are typically "always on" via a cable-modem.

Due to the above reason, split tunneling is not allowed in our design.

IPSec key exchange parameters

ISAKMP

Protocol: DES
Authentication: MD5
Duration for key change: 720 minutes

IPSec

Protocol: ESP
Mode: Tunnel
Encryption: 3DES 168-bit
Authentication: HMAC MD5
Duration for key change: 30 minutes

AH or ESP, Tunnel Mode or Transport Mode

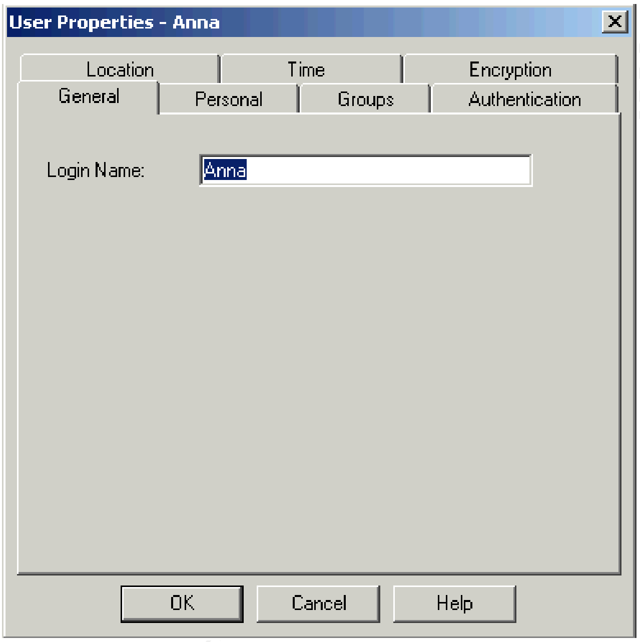
ESP in tunnel mode in chosen.

- Tunnel mode is chosen because it encapsulates the entire IP packet, whereby transport mode is used only to protect data in IP packets.
- ESP is chosen because AH does not provide for data confidentiality.

VPN Rules and Implementation

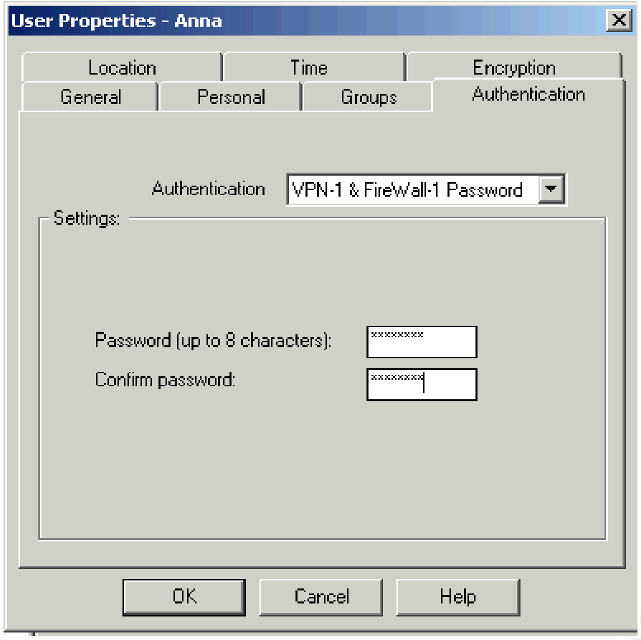
a. First, the remote employees are defined so that they can authenticate to a Policy Server.

From the Check Point Policy Editor, select Manage > Users > New > User by Template > Default, under the General tab, enter the user name Anna.



Login Name: Anna

Under the Authentication tab, choose VPN-1 & FireWall-1 Password, then enter and confirm a password for user Anna.

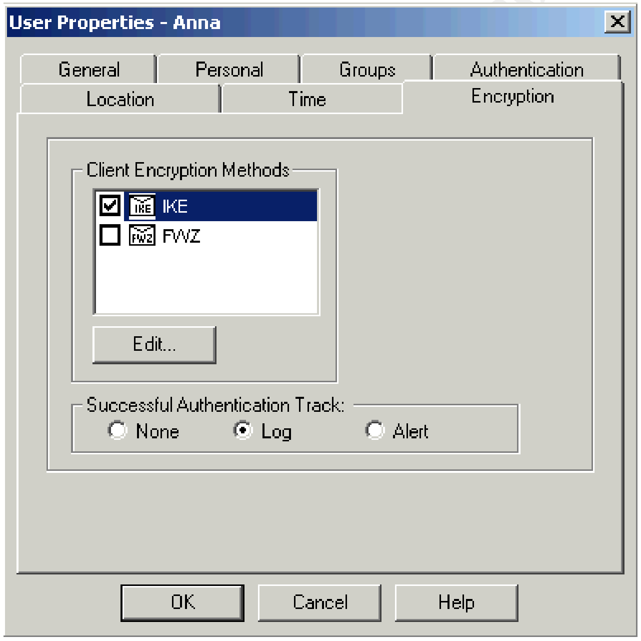


Authentication: VPN-1 & FireWall-1 Password

Password (up to 8 characters): ********

Confirm password: ********

Under the Encryption tab, from the Client Encryption Methods field, select the IKE option; Select Log from the Successful Authentication Track field.



Client Encryption Methods

☑ IKE

☐ FWZ

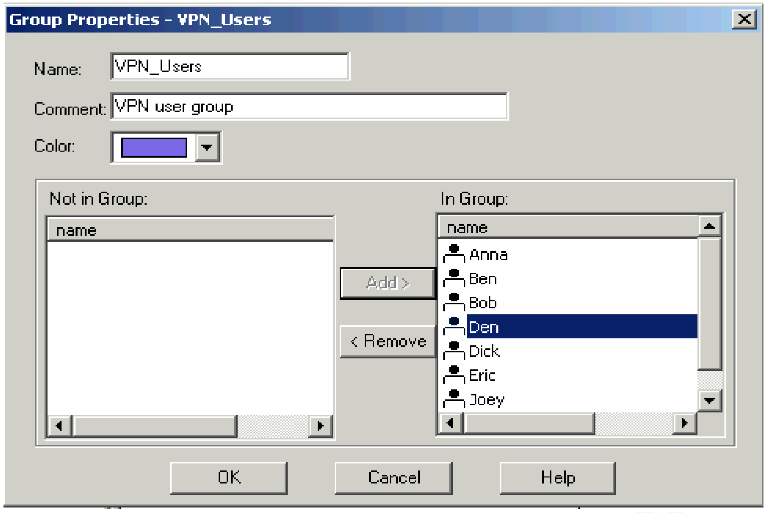
Successful Authentication Track:

○ None ● Log ○ Alert

Click OK. The user Anna is added.

Use the same method to add all the remote employees.

From the Users screen, create a new group called VPN_Users, then add all the users into this group, as shown in the figure below.



Group Properties - VPN_Users

Name: VPN_Users

Comment: VPN user group

Color:

Not in Group:

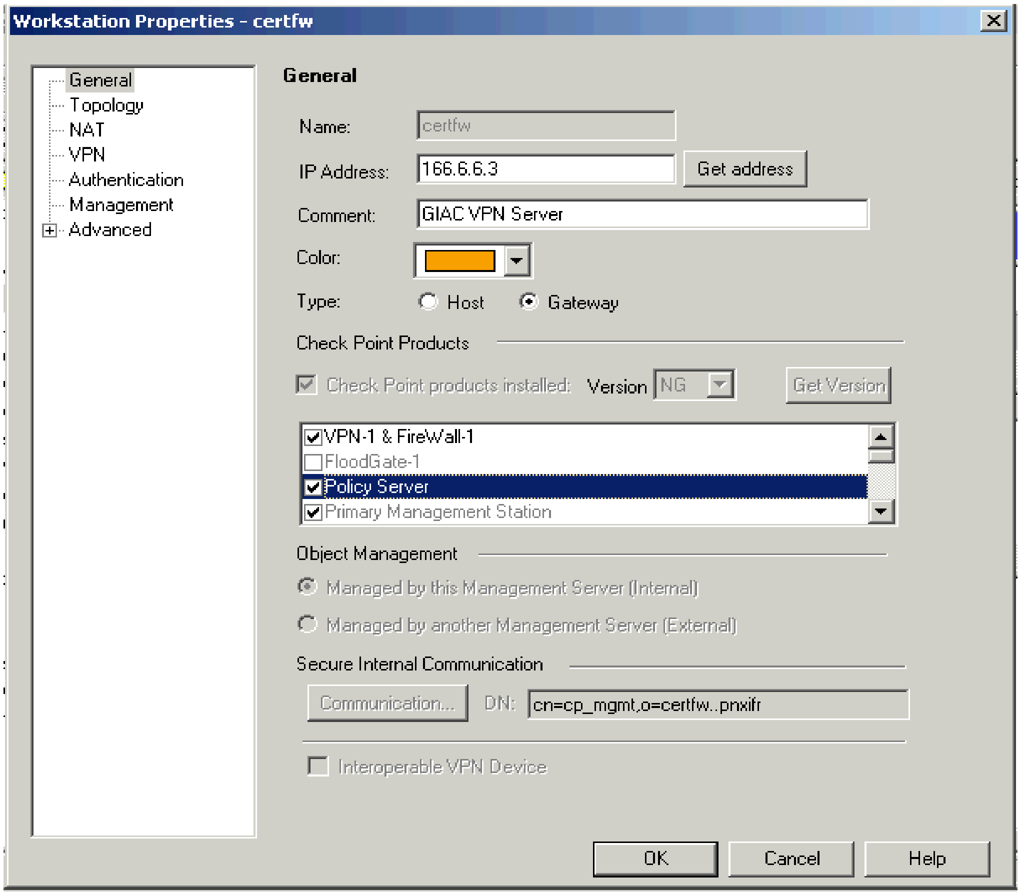
In Group: Anna, Ben, Bob, Den, Dick, Eric, Joey

From the Check Point Policy Editor, select Policy > Install Users Database.

b. Configure the Policy Server

Configure the Workstation Properties screen to enable the use of a VPN-1/FireWall-1 NG module as a Policy Server.

Under the General Tab, click on the option Policy Server.



General

Name: certfw

IP Address: 166.6.6.3

Comment: GIAC VPN Server

Color:

Type: ○ Host ● Gateway

Check Point Products

☑ Check Point products installed: Version NG

☑ VPN-1 & FireWall-1
□ FloodGate-1
☑ Policy Server
☑ Primary Management Station

Object Management

● Managed by this Management Server (Internal)
○ Managed by another Management Server (External)

Secure Internal Communication

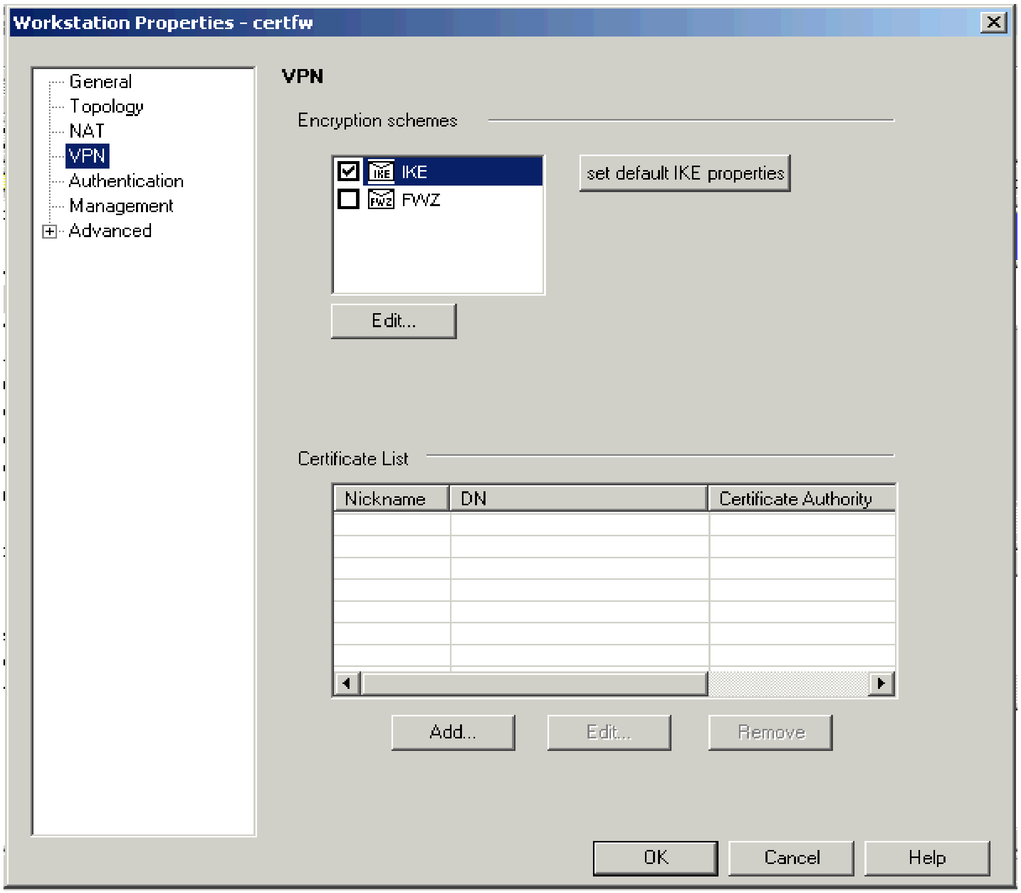
DN: cn=cp_mgmt,o=certfw..pnxifr

□ Interoperable VPN Device

Under the Topology tab, in the VPN Domain section of the screen, verify that the Exportable for SecuRemote option is selected.

Under Authentication tab, verify that VPN-1 & FireWall-1 Password is selected. In the Policy Server section of this screen, click the Users drop-down menu, then select VPN_Users group.

Under the VPN tab, verify that the IKE option in the Encryption schemes field is selected.



Then add the Internal Certificate into the Certificate List. After that, click OK.

Until now, the Policy Server is configured.

c. Create Rules

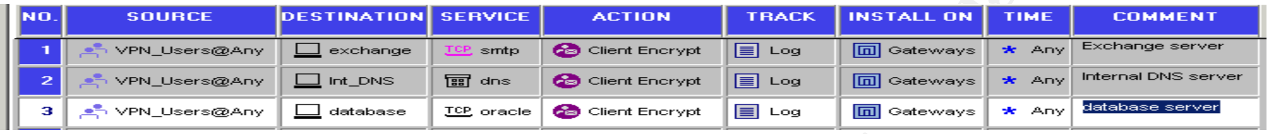
From the Check Point Policy Editor, select Rules ->Add Rules ->Top to add rules. The following three rules are added at the top.

Source	Destination	Service	Action	Track	Install On	Commen t
VPN_Users@Any	166.166.166.12	Tcp/25	Client Encrypt	Log	Gateways	Exchange server
VPN_Users@Any	166.166.166.13	Tcp/53; udp/53	Client Encrypt	Log	Gateways	Internal DNS server
VPN_Users@Any	166.166.166.18	Tcp/1521	Client Encrypt	Log	Gateways	database server

In the above table, the first rule is for VPN_Users to access the exchange mail server in the internal service network, the second rule is for VPN_Users to access to the internal

DNS server in the internal service network, while the third rule is for VPN_Users to access to the database server in the database network. The service "smtp" and "dns" exists by default in Check Point NG, whereas for the service "oracle", we defined it as tcp port 1521.

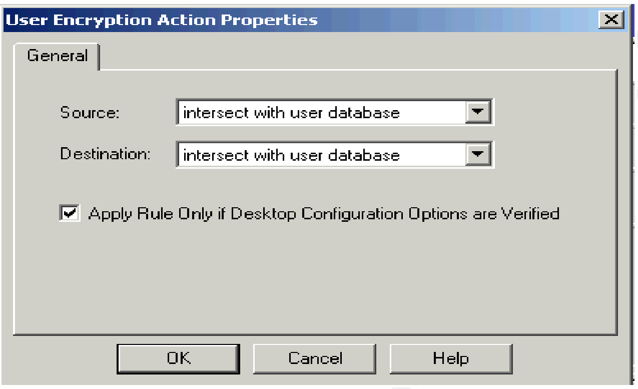
In the policy editor, the three rules should look like this:



NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
1	VPN_Users@Any	exchange	TCP smtp	Client Encrypt	Log	Gateways	* Any	Exchange server
2	VPN_Users@Any	Int_DNS	dns	Client Encrypt	Log	Gateways	* Any	Internal DNS server
3	VPN_Users@Any	database	TCP oracle	Client Encrypt	Log	Gateways	* Any	database server

Below these three rules, the normal firewall rules will follow, which are not detailed here.

Right-click the Client Encrypt icon in the Action column, select the Edit Properties option from the drop-down menu, the User Encryption Action Properties screen appears. Verify that Source and Destination are set to intersect with the user database, and the option "Apply Rule Only if Desktop Configuration Options are Verified" is checked.



User Encryption Action Properties

General

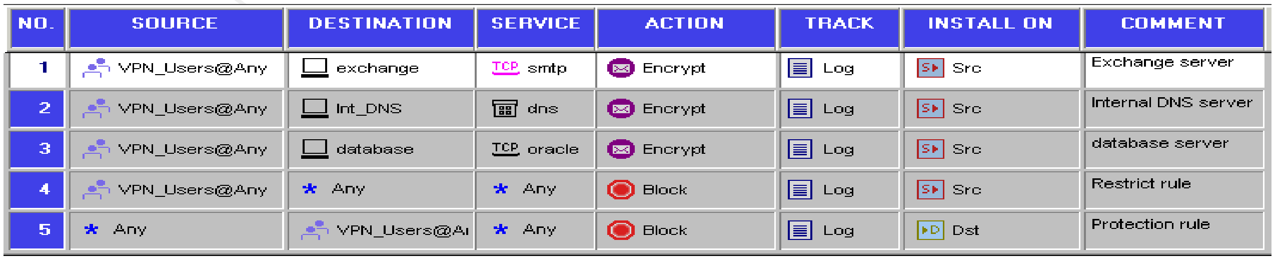
Source: intersect with user database

Destination: intersect with user database

☑ Apply Rule Only if Desktop Configuration Options are Verified

Click OK to return.

From Check Point Policy Editor, click the Desktop Security tab, then click Rules -> Add Rule -> Top, then configure the rule so that it looks like this:



NO.	SOURCE	DESTINATION	SERVICE	ACTION	TRACK	INSTALL ON	COMMENT
1	VPN_Users@Any	exchange	TCP smtp	Encrypt	Log	Src	Exchange server
2	VPN_Users@Any	Int_DNS	dns	Encrypt	Log	Src	Internal DNS server
3	VPN_Users@Any	database	TCP oracle	Encrypt	Log	Src	database server
4	VPN_Users@Any	* Any	* Any	Block	Log	Src	Restrict rule
5	* Any	VPN_Users@Ai	* Any	Block	Log	Dst	Protection rule

In the above, the first rule is for VPN_Users to access the exchange mail server encryption domain, the second rule is for VPN_Users to access to the internal DNS server encryption domain, the third rule is for VPN_Users to access to the database server encryption domain, the fourth is to restrict VPN_Users to access any location other than the ones specified in the first three rules, and the last one is to disallow an intruder to connect to the user's machine while the users are using VPN.

2.10 Tutorial

Below is an excellent link to the tutorial of iptables.
http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html

Here are more links.
http://www.telematik.informatik.uni-karlsruhe.de/lehre/seminare/LinuxSem/downloads/netfilter/iptables-HOWTO-1.html

http://www.oofle.com/iptables/whatis.htm

Introduction

IPTables is a linux command line program that sets up packet filtering in the linux kernel based on what rules you input into it. IPTables is the replacement for ipchains in the newer linux kernels of version 2.4 or higher.

Syntax

Type in the command "iptables –h" to get help on the syntax of the program. The output is as follows:

```
iptables v1.2.4

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

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

Parts of the Rule

Here are some example rules used in our network design, and an explanation of the parts of the rules.

IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not syn:"

In this rule, the part “-A FORWARD” means that this rule should append to the FORWARD chain, “-p tcp” means tcp protocol, “!--syn” means without SYN flag, “-m state --state NEW" means non-established connection, "-j LOG --log-prefix "New not syn:"" means to log with prefix "New not syn:". This rule will then append to the FORWARD chain, and log with prefix "New not syn:" all the tcp packets that do not have SYN flag set, and do not belong to any established connections.

Another example.

```
IPTABLES -A FORWARD -p tcp --tcp-flags SYN,FIN  -j DROP
```

This rule will append to FORWARD chain and drop any tcp packets with SYN and FIN flags set simultaneously.

One more example.

```
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth2 -s
$SYSADMIN_IP -o eth1 -d $EXT_SERVICE_NET --dport 22 -j ACCEPT
```

This rule will append to FORWARD chain, and accept all new and established tcp packets from eth2 to eth1, with source IP of $SYSADMIN_IP and destination of $EXT_SERVICE_NET and destination port of 22.

How to Apply Rules

In Linux kernel 2.4 and higher, iptables can be applied easily. In the section 2.3, policies for the primary firewall, the rules are written in script format. Executing the script will enable the rules applied.

The following commands will flush and delete all the rules configured.

```
iptables –F
iptables –X
```

Test Rules

Before testing the rules as detailed below, it is important to note that due to the default policy defined, all the connections not explicitly allowed will be dropped.

```
IPTABLES -A INPUT -m state --state NEW,ESTABLISHED -p tcp -i eth2 -s
$SYSADMIN_IP -d $FW_IP --dport 22 -j ACCEPT
IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -p tcp -o eth2 -d
$SYSADMIN_IP -s $FW_IP --sport 22 -j ACCEPT
```

The first rule in this pair is to allow new and established tcp packets going from eth2 with source IP of the system administrator's IP to the firewall external IP at port 22. The second rule in this pair is to allow established and related tcp packets from firewall external IP and port 22 to the system administrator's system. Combined together, this pair is to allow system administrator to connect to the firewall using ssh.

To test this rule, we connect to the firewall from the system administrator's computer using ssh. We get through. Using other computer other than the system administrator's system, we do not get through. This verifies that this pair of rules are working as expected.

```
IPTABLES -A FORWARD -m state --state NEW,ESTABLISHED -p tcp -i eth0 -s 0.0.0.0/0 -o eth1 -d $EXT_HTTP_IP --dport 80 -j ACCEPT
IPTABLES -A FORWARD -m state --state ESTABLISHED -p tcp -i eth1 -s $EXT_HTTP_IP -o eth0 -d 0.0.0.0/0 --sport 80 -j ACCEPT
```

The first rule in this pair is to allow new and established tcp packets going from eth0 to eth1 with destination IP of the external http server at port 80; the second rule in the pair is to allow established tcp packets from eth1 with source IP of the external http server at port 80 to eth0. Combined together, this pair of rules allow the outside world to access the external http server at port 80.

To test this pair of rules, connect to the external http server from internet, we get the web page. This verifies that this pair of rules are working properly.

```
IPTABLES -A FORWARD -p tcp --tcp-flags SYN,FIN  -j DROP
```

This rule is to drop any tcp packets attempting to traverse the firewall with SYN and FIN flags set simultaneously.

To test this rule, we send a packet with SYN and FIN flags set simultaneously to the external http server. In the firewall, using the tcpdump command to capture the traffic, we see this packet. In the http server, we also use the tcpdump command to capture the traffic, however, we didn’t see this packet. It verifies that this rule is working properly.

Assignement 3 Audit the Primary Firewall

3.4 Introduction

Regular audit on all the systems are of great importance to the security of the network. In this part, only the audit on the primary firewall for GIAC Enterprises is described.

In order to conduct the audit, we will need to:

1. Plan the audit. Technical approaches are recommend to assess the firewall. Considerations such as what shift or day we would do the assessment are included. Costs and level of effort are estimated. Risks and considerations are identified.
2. Conduct the audit. Using the approaches in step 1, the fact that the primary firewall is actually implementing GIAC Enterprises' security policy is validated.
3. Evaluate the audit. Based on our assessment, the perimeter defense is analyzed and recommendations for improvements or alternate architectures are made.

3.5 Plan the audit

Coordination

Before the audit can be conducted, support from various groups within the company must be requested and agreed upon.

- Seek management agreement. The management must be informed of the schedule, scope and budget of the audit, and then approve the plan.
- The IT department needs to allocate two staff to help in the audit in case of emergency.

Schedule

Friday will not be a good day to conduct the audit, since people are some how in the holiday mood. Besides, if something really goes wrong and extra work is needed on weekend, some help may not be found.

Daytime is not recommended, as the audit may disrupt normal business operation.

Monday night seems good to conduct the auditing.

Scope and Tools

The objective of the audit is to validate that the firewall is actually implementing the GIAC's security policy.

- Scan the vulnerability of the firewall to validate the OS and iptables hardening measure. Nessus will be used for this.

- Port-scan all the interfaces of the firewall to validate the ACL on the traffic to or from the firewall. This part of the ACL is defined by the chains INPUT and OUTPUT. Nmap will be used.
- Port-scan the networks behind the firewall from all directions to ensure that firewall's ACL on the passing traffic is working properly. This part of the ACL is defined by the chains FORWARD and user-defined chains. Again nmap will be used.
- Check the logs in the firewall, IDS and syslog server to ensure that the logging mechanism is working properly. No tool is needed for this part of the audit.

Budget

There will be no hardware or software costs associated with this exercise, since all the tools are publicly available. An existing Linux laptop loaded with nmap and nessus will be used to run the test.

One audit staff with two IT staff are required to work around 3 hours for the vulnerability and port scan, 3 hours for checking the log, and another 3 hours to analyse the result and report to the management. If serious problems are found in the vulnerability scan, some more time is needed by the two IT staff, depending on the nature of the problem.

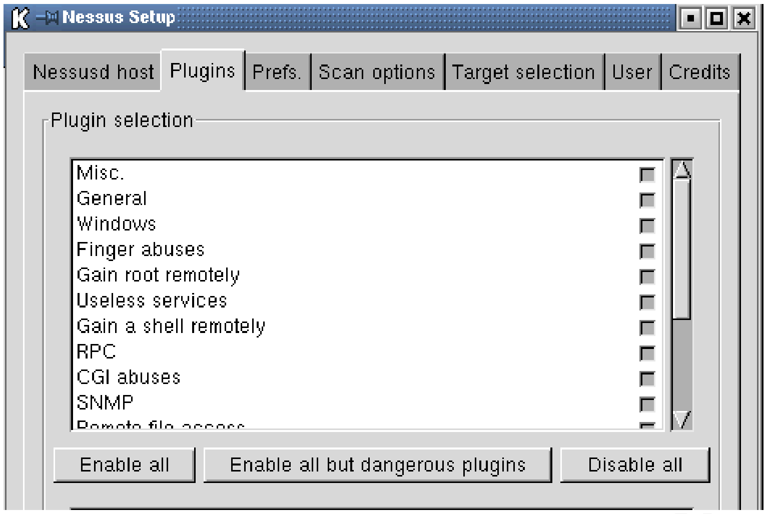
3.3 Conduct the Audit

Scan the Vulnerability of the Firewall

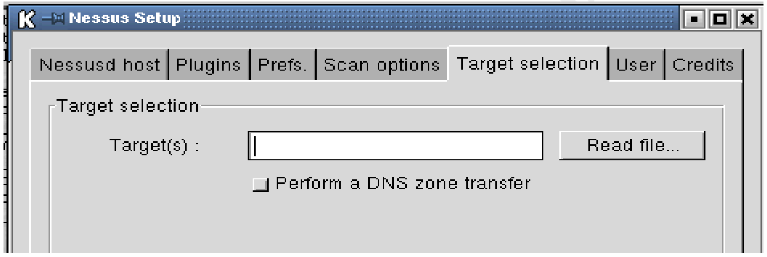
Nessus 1.2.0 is used to scan the vulnerability of the firewall so that OS and iptables hardening measures can be validated. Version 1.2.0 is the latest stable version currently. The homepage of it is http://www.nessus.org/. According nessus.org, Nessus is free, powerful, up-to-date and easy to use remote security scanner. A security scanner is a software which will audit remotely a given network and determine whether bad guys (aka 'crackers') may break into it, or misuse it in some way. Nessus does not take anything for granted. That is, it will *not* consider that a given service is running on a fixed port - that is, if you run your web server on port 1234, Nessus will detect it and test its security. It will not make its security tests regarding the version number of the remote services, but will really attempt to exploit the vulnerability. Nessus is very fast, reliable and has a modular architecture.

The nessus scan in this part is supposed to scan for the OS level vulnerabilities of the firewall, so rules are added in the firewall to allow for all connections from the nessus scanner to the firewall and all established and related connections. Without these rules, all the packets from the scanner is dropped, no connection can be made, so no real vulnerability scanning can be done.

After installing nessus, create an account by the command “nessus-adduser”. Then start the daemon by the command “nessusd –D’. After that, the program can be started by the command “nessus”. After logging in, the following interface will appear at this stage.

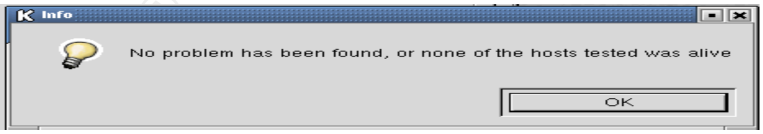


Click on the “Plugins” tab and choose the vulnerabilities to scan for. Click on the “Target selection” tab and fill in the IP of the firewall.



Then click the button “start” and the scan starts.

At the end of the scan, the window below pops up.



No problem has been found, or none of the hosts tested was alive

The above procedure is repeated for the two other interfaces as well. No vulnerabilities are found. This validated the OS and iptables hardening measures.

Port-scan All the Interfaces of the Firewall

This part is to validate the ACL on the traffic to or from the firewall. This part of the ACL is defined by the chains INPUT and OUTPUT. Nmap http://www.nmap.org will be used to verify that the firewall is dropping all attempted connections to the firewall.

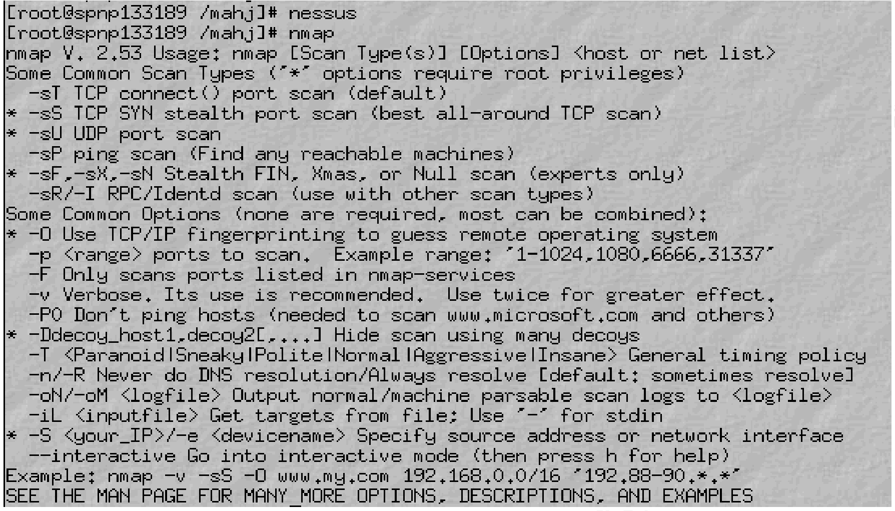
According to nmap.org, this tool is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers, and both console and graphical versions are available. Nmap is free software, available with full source code under the terms of the GNU GPL.

Nmap is ...

- **Flexible**: Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, pings sweeps, and more.
- **Powerful**: Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- **Portable**: Most operating systems are supported, including Linux, Open/Free/Net BSD, Solaris, IRIX, Mac OS X, HP-UX, Sun OS, and more.
- **Easy**: While Nmap offers a rich set of advanced features for power users, you can start out as simply as "nmap -O -sS *targethost*". Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.
- **Free**: The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the GNU General Public License (GPL).
- **Well Documented**: Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, and tutorials.
- **Supported**: While Nmap comes with no warranty, you can write the author (fyodor@insecure.org) if you experience any problems.
- **Acclaimed**: Nmap has won numerous awards, including "Information Security Product of the Year" by both Info World and Codetalker Digest. It has been featured in hundreds of magazine articles.

- **Popular**: Thousands of people download Nmap every day, and it is included with many operating systems (Redhat Linux, Debian Linux, FreeBSD, OpenBSD, etc). It is among the top ten (out of 15,000) downloads at the Freshmeat repository. This is important because it lends Nmap its vibrant development and user support communities.

Below is the description for all the options of the nmap.



```
[root@spnp133189 /mahj]# nessus
[root@spnp133189 /mahj]# nmap
nmap V. 2.53 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
  -sT TCP connect() port scan (default)
* -sS TCP SYN stealth port scan (best all-around TCP scan)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/-oM <logfile> Output normal/machine parsable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY_MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

Before perform the scan, the rules added in the firewall for the nessus scan is removed at this stage.

Scan on tcp port of the external interface of the firewall is first performed.

```
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 166.166.166.33

Starting nmap V. 2.53 ( www.insecure.org/nmap/ )
Interesting ports on 166.166.166.33:
(The 65534 ports scanned but not shown below are in state: filtered)
```

Then scan on ucp port of the external interface of the firewall is performed.

```
[root@spnp133.189 root]# nmap -P0 -sU -p 1-65535 166.166.166.33

Starting nmap V. 2.53 ( www.insecure.org/nmap/ )
Interesting ports on 166.166.166.33:
(The 65534 ports scanned but not shown below are in state: filtered)
```

These scans perform both TCP as well as UDP connect scan for all ports from 1 to 65535 (i.e. all 64k ports).

The above procedure is then repeated for the other two interfaces of the firewall. No ports are found open.

Running tcpdump on the primary firewall itself, it can be verified that the dropped network packets of the scan did reach the firewall machine.

The above procedure validated the ACL on the traffic to or from the firewall.

Port-scan the Network Behind the Firewall From All Directions

This part is to ensure that firewall's ACL on the passing traffic is working properly. This part of the ACL is defined by the chains FORWARD and user-defined chains. Again nmap will be used.

First scan the inverse web proxy 166.166.166.3.

```
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 www.giac.com
Starting nmap V. 2.53 ( www.insecure.org/nmap/ )
Interesting ports on www.giac.com
(The 65534 ports scanned but not shown below are in state: filtered)

Port      State      Service
80/tcp     open       http

[root@spnp133.189 root]# nmap -P0 -sU -p 1-65535 www.giac.com
Starting nmap V. 2.53 ( www.insecure.org/nmap/ )
Interesting ports on www.giac.com
(The 65534 ports scanned but not shown below are in state: filtered)
```

Scans for other servers in the external service network are also performed.

```
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 dns.giac.com
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 dns.giac.com
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 smtp.giac.com
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 smtp.giac.com
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 ids.giac.com
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 ids.giac.com
```

The scans are then performed for the internal service network, database network and system administrator network.

```
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 166.166.166.0/29
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 166.166.166.0/29
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 166.166.166.8/29
```

```
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 166.166.166.8/29
[root@spnp133.189 root]# nmap -P0 -sT -p 1-65535 166.166.166.16/29
[root@spnp133.189 root]# nmap -P0 –sU -p 1-65535 166.166.166.16/29
```

Check The Logs

Check the logs in the firewall, IDS and syslog server to ensure that the logging mechanism is working properly by looking at whether they log the above scan properly. No tool is needed for this part of the audit.

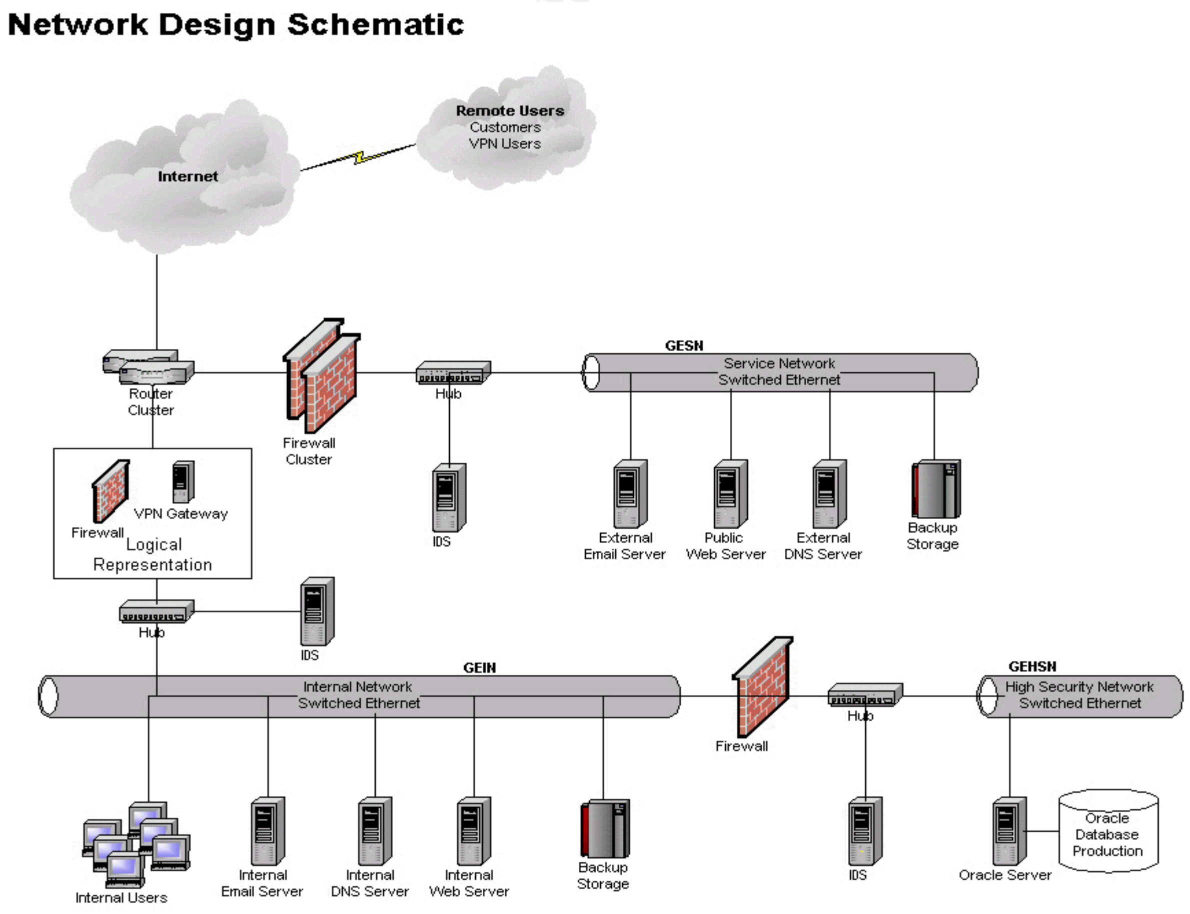
3.4 Evaluate the audit

The audit conducted above demonstrates that the primary firewall of the GIAC enterprises is actually implementing the GIAC's security policy.

Assignment 4 Design Under Fire

For this part, Dennis Pickett's practical is chosen. This assignment resides at http://www.giac.org/practical/Dennis_Pickett_GCFW.zip . The primary firewall used is Firewall-1 4.0 build 4094 running on Nokia IP440s with IPSO 3.2.1 (IPSO build 13).

His network design:



4.1 Three Primary Firewall Vulnerabilities

Check Point Firewall-1 Fast Mode TCP Fragment Vulnerability

This vulnerability was published on http://online.securityfocus.com/bid/2143 on Dec 14, 2000. According to securityfocus, Check Point Software's VPN-1 and Firewall-1 products contain a vulnerability in their "Fast Mode" option that may allow an attacker to bypass access control restrictions and access certain blocked services. Fast Mode is a setting that turns off analysis of packets in tcp sessions after the TCP 3-way handshake has completed for speed-crtitical services. If this setting is enabled on a firewall, it may be possible for a remote attacker to access blocked services on the host protected by the firewall using fastmode. It is also reportedly possible to access hosts at least one hop away on the same interface as the target host being protected. In order for this to be possible, at least one TCP service on a host protected by the firewall must be accessible by the attacker to which a SYN can be sent legitimately. The vulnerability is due to a failure to handle malformed fragmented TCP segments. This vulnerability may allow attackers to access vulnerable services normally protected by the firewall ruleset.

Check Point Software contacted SecurityFocus with an update regarding this issue and sent out an alert on http://www.checkpoint.com/techsupport/alerts/fastmode.html. According to Check Point, the described vulnerability is completely resolved in VPN 1/FireWall-1 4.1 SP3. In addition, an immediate workaround is available for all versions: disable Fastmode. Note that Fastmode is disabled by default, and would only turned on if the administrator manually enabled this feature. Also note that VPN-1/FireWall-1 includes a "Fast Mode" option that provides higher performance for TCP protocols. Because VPN-1/FireWall-1 performance is continually improved with each release, Fast Mode was most relevant for older versions of the product. Based on the current performance of VPN-1/FireWall-1, Fast Mode no longer provides a significant performance benefit. Consequently, Check Point will discontinue the Fast Mode option with the next major release of VPN-1/FireWall-1.

An exploit for this vulnerability is available at http://downloads.securityfocus.com/vulnerabilities/exploits/fm.c. This exploit is provided by Thomas Lopatic lopatic@tuv.net. The site http://www.soldierx.com/exploits/os/hardware/firewalls/firewall-1/fm.c contains a detailed description of the vulnerability and exploit. According to this site, if a certain service is defined to be a Fastmode service, then all non-SYN packets with a source or destination port equal to the Fastmode service will be accepted by the firewall. Only SYN packets are still passed through the inspection engine. Version 4.1 SP2 does not include a minimal length check for the first fragment of a TCP packet anymore. Instead, when examining TCP ports and TCP flags, it copies the TCP header from the linked list of fragments to a contiguous memory buffer. Thus, if we fragment the 20 byte TCP header into three 8 byte + 8 byte + 4 byte fragments, FW-1 will still interpret the TCP header correctly. This is the major difference to prior versions. In prior versions, the inspection engine made sure that the first fragment had a length of at least 40 bytes and then performed the rulebase checks (TCP ports, TCP flags) directly in the part of the first fragment. No copying.

The exploit is developed taking advantage of the above problem. The attack needs two things in order to succeed: a) a Fastmode service and b) an open port at a certain IP address. Let us assume that we have a web server with port 80 open to the public. Suppose that the administrator has made port 80 a Fastmode service, in order to improve firewall performance. We now send two fragmented TCP packets, packet A and packet B. Fragment #1 of these packets contains the first 8 bytes of the respective TCP header, fragment #2 contains the next 8 bytes, and fragment #3 contains the remaining 4 bytes. Packet A is an ACK packet with a source port equal to the Fastmode service, i.e. a source port of 80. The destination port of this packet is the blocked service that we want to get a SYN to. Let us assume it is 32775. Suppose A1, A2 and A3 are the three fragments of packet A. They now contain the following information.

A1: ports (80 -> 32775)
A2: flags (ACK)
A3: ...

This packet will be accepted, because the source port is a Fastmode service and it is not a SYN packet. Packet B is a SYN packet with a non-privileged source port, e.g. 1024. The destination port of this packet is the service which is open to the outside world, i.e. 80. So, the fragments of packet B contain the following information.

B1: ports (1024 -> 80)
B2: flags (SYN)
B3: ...

This fragment will be accepted, because it is accepted by the rulebase.

For both fragment sets we choose the same IP id. And what we want to end up with is that the destination host of the fragments drops A2, B1, and B3. Because then the firewall will accept two harmless packets that will be combined into a single not so harmless packet at the destination, as in

A1: ports (80 -> 32775)
B2: flags (SYN)
A3: ...

So, we have to somehow malform A2, B1, and B3. However, the fragments must not be malformed when we send them. Otherwise the intermediate routers between us and the final destination would detect the malformation and drop our fragments. Therefore we use a timestamp IP option that will overflow right at the destination host. In this way, all intermediate routers between us and the destination will see intact packets with a valid timestamp option. The destination, however, will see that the timestamp IP option has been completely used up by the previous hop and thus consider the option to be invalid and drop the fragment.

We can do this for any non-first fragment. For first fragments FW-1 ensures that they start with 0x45, i.e. that they do not contain any options.

Now we can make the destination drop A2 and B3. And with BSD semantics, a second fragment that has the same offset as a fragment in the reassembly queue will be overlapped by the fragment in the reassembly queue, i.e. it will potentially be discarded. Hence, if we send packet A before packet B, B1 will be dropped because A1 already exists in the reassembly queue and has the same offset and length. For destination hosts which overlap fragments the other way around, we would have to send packet B before packet A.

And that is basically it. We sneak a SYN through the firewall from a Fastmode port to any other port at the same IP address as the port that is open to the outside. All remaining non-SYNs will be accepted, because they contain a Fastmode service as their source port (our packets) or destination port (reply packets).

To extend the attack to hosts that are at least one hop away from the firewall, we can use source routing to have the hop behind the firewall rewrite the destination address of fragment B2 to anything we want. Thus we can redirect the SYN fragment to any IP address after it has passed the firewall.

RDP Communications Issue

This problem is published on http://online.securityfocus.com/bid/2952/discussion/ on Jul 09, 2001. According to SecurityFocus, a problem has been discovered with the firewall that allows traversal. It is possible for a remote user to pass packets across the firewall via port 259 by using false RDP headers on UDP packets. This makes it possible for remote users to gain access to restricted information systems

QinetiQ SHC Research reported this issue to Check Point. Check Point has become aware of a condition with RDP Protocol in VPN-1/ FireWall-1 4.1 and Next Generation (NG) that may affect system stability. Details can be found at http://www.checkpoint.com/techsupport/alerts/rdp_comms.html. According to Check Point, if the error occurs on a 4.1 module, certain management functions, such as logging and administrator communications, will halt. On NG modules, encryption key processing may be briefly interrupted. At no point is security compromised, and the firewall continues to enforce the security policy and allows appropriate traffic. No unauthorized access, information leakage or breach of security occurs. Check Point knows of no organizations that have had systems affected by this issue.

A fix for VPN-1/ FireWall-1 4.1 and Next Generation (NG) on all platforms is available for immediate download at http://www.checkpoint.com/techsupport/index.html

One exploit is available at http://www.securiteam.com/exploits/5SP0B154UY.html.

IP Fragment-driven Denial of Service Vulnerability

This problem was discovered by Lance Spitzner at lance@spitzner.net. It was published on http://online.securityfocus.com/bid/1312 on Jun 06, 2000. It's bugtraq id is 1312, cve number CVE-2000-0482. According to SecurityFocus, by sending illegally fragmented packets directly to or routed through Check Point FireWall-1, it is possible to force the firewall to use 100% of available processor time logging these packets. The FireWall-1 rulebase cannot prevent this attack and it is not logged in the firewall logs.

Details of this vulnerability can also be found at http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html. According to Check Point, a stream of large IP fragments can cause the FireWall-1 code that logs the fragmentation event to consume most available host system CPU cycles. It should be noted that no unauthorized access, information leakage, or fragment passing occurs. Testing by Check Point indicates that versions 4.0 and 4.1 of FireWall-1 can be impacted (versions earlier then the 4.0 version were not tested). For security reasons (e.g., overlay attacks) FireWall-1 reassembles all IP fragments of a datagram prior to inspection against the security policy. After reassembly, the packet is processed by the FireWall-1 Stateful Inspection engine, and if allowed by the security policy to proceed, the packet is refragmented and forwarded. To identify and audit attacks such as Ping of Death, Check Point added a mechanism to FireWall-1 - outside of its standard logging capability - to log certain events that occur during the
FireWall-1 virtual reassembly process. This fragmentation logging takes place on the gateway itself and not on the management station (relevant for distributed management deployments).

The binaries in Service Pack 2 of FireWall-1 version 4.1 for 4.1 users, and Service Pack 6 Hot Fix for FireWall-1 version 4.0 users were released to solve this problem. As an interim workaround, customers can disable the console logging, thereby mitigating this issue by using the following command line on their FireWall-1 module(s):

```
$FWDIR/bin/fw ctl debug -buf
```

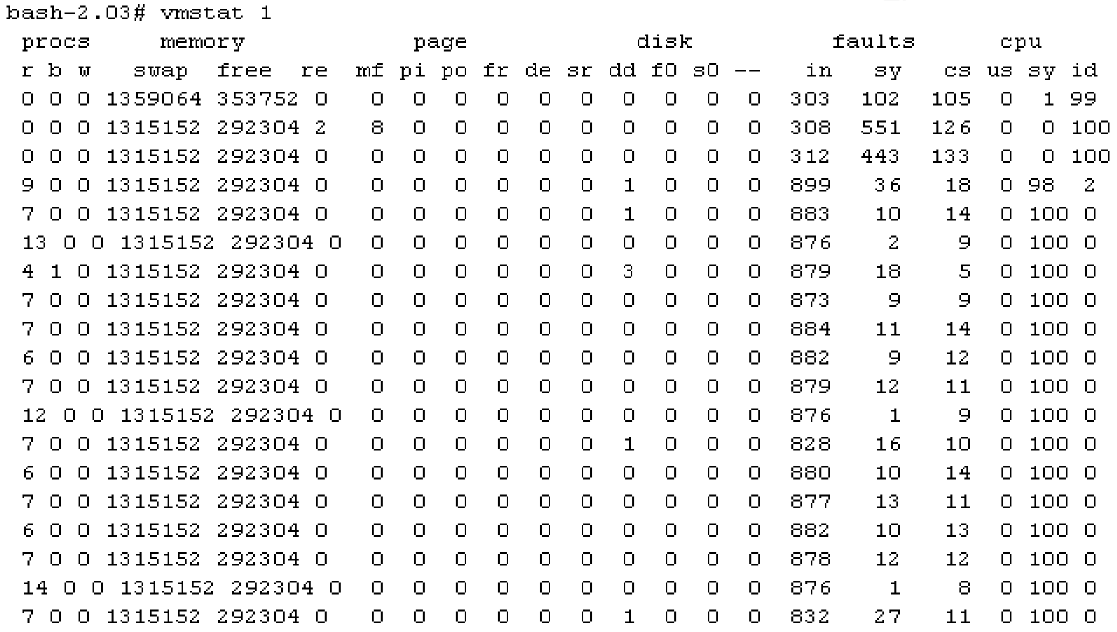
This takes effect immediately. This command can be added to the $FWDIR/bin/fw/fwstart command in order to be enabled when the firewall software is restarted. It should be noted that although this command will disable fragmentation console output messages, standard log messages (e.g., Long, Short, control messages, etc.) will continue to operate in their traditional way.

An exploit of this vulnerability exists at /data/vulnerabilities/exploits/jolt2.c. Although this exploit was coded for a different vulnerability, it has proven to be effective in demonstrating this vulnerability as well.

The authors used jolt2 to send a stream of extremely large IP fragments to a FireWall-1 gateway, which in some cases can cause the write mechanism to grab all host CPU resources. There is no fragmentation tracking resource that is exhausted; it is the case that the fragmentation logging process is the cause of this issue.

Running the IP Fragment-driven DoS Attack Against the Firewall
First download jolt2.c from <u>/data/vulnerabilities/exploits/jolt2.c</u>. Then compile it using the command: gcc –o jolt2 jolt2.c. The file 'jolt2' is then generated as an executable file.

Start the attack by running this command: *jolt2 ip_of_the_firewall*. Run 'vmstat -1' on the firewall to report the memory and CPU usage on it. Below is the snapshot from the attacked firewall.



```
bash-2.03# vmstat 1
 procs     memory            page            disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr dd f0 s0 --   in   sy   cs us sy id
 0 0 0 1359064 353752 0   0  0  0  0  0  0  0  0  0  0  303  102  105  0  1 99
 0 0 0 1315152 292304 2   8  0  0  0  0  0  0  0  0  0  308  551  126  0  0 100
 0 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  312  443  133  0  0 100
 9 0 0 1315152 292304 0   0  0  0  0  0  0  1  0  0  0  899   36   18  0 98  2
 7 0 0 1315152 292304 0   0  0  0  0  0  0  1  0  0  0  883   10   14  0 100 0
 13 0 0 1315152 292304 0  0  0  0  0  0  0  0  0  0  0  876    2    9  0 100 0
 4 1 0 1315152 292304 0   0  0  0  0  0  0  3  0  0  0  879   18    5  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  873    9    9  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  884   11   14  0 100 0
 6 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  882    9   12  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  879   12   11  0 100 0
 12 0 0 1315152 292304 0  0  0  0  0  0  0  0  0  0  0  876    1    9  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  1  0  0  0  828   16   10  0 100 0
 6 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  880   10   14  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  877   13   11  0 100 0
 6 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  882   10   13  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  0  0  0  0  878   12   12  0 100 0
 14 0 0 1315152 292304 0  0  0  0  0  0  0  0  0  0  0  876    1    8  0 100 0
 7 0 0 1315152 292304 0   0  0  0  0  0  0  1  0  0  0  832   27   11  0 100 0
```

It can be seen from above that in the first 3 seconds, attack has not started, cpu idle time is 99 or 100. The attack starts at the 4th second, and the cpu system utilization immediately jumps to 98 while cpu idle time drops to 2. Starting from the 5th second, cpu system utilization remains at 100 due to the attack. No free cpu is available. This is an successful DoS attack against the firewall.

Below is a sample of the entry generated by running tcpdump on the attacked firewall.

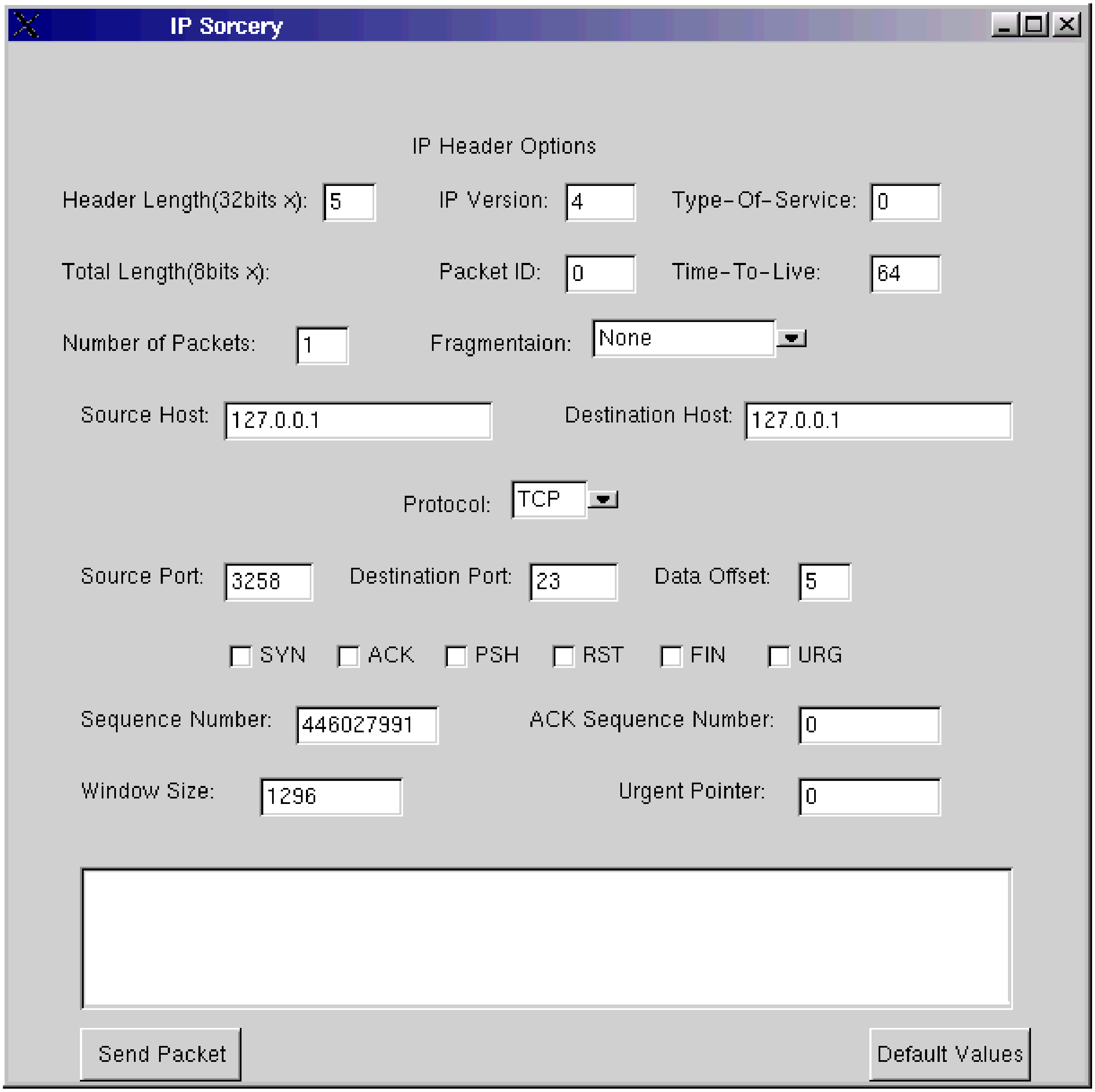
```
21:31:21.267384 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267502 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267581 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267657 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267735 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267812 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267885 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.267958 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268033 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
```

```
21:31:21.268102 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268173 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268245 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268327 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268403 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268478 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
21:31:21.268551 spnp133189.spnp.nus.edu.sg > certfw: (frag 1109:9@65520)
```

4.2 Denial of Service Attack

There are numerous Denial of Service tools on the internet. For example, on the website http://packetstormsecurity.nl/DoS/ , source codes for hundreds of DOS tools can be downloaded. For this practice, IP Sorcery 1.6 is used. IP Sorcery is a TCPIP packet generator which allows you to send TCP, UDP, and ICMP packets with a GTK+ interface. The ability to specify number of packets in the GUI version was added in version 1.6, along with the ability to send up to 25 RIP entry tables in the console. It can be downloaded from http://packetstormsecurity.nl/UNIX/misc/ipsorc-1.6.tar.gz

After compiling and installing it on a Linux machine, type in the command 'magic', the following window will appear.



IP Header Options

Header Length(32bits x): 5   IP Version: 4   Type-Of-Service: 0

Total Length(8bits x):   Packet ID: 0   Time-To-Live: 64

Number of Packets: 1   Fragmentaion: None

Source Host: 127.0.0.1   Destination Host: 127.0.0.1

Protocol: TCP

Source Port: 3258   Destination Port: 23   Data Offset: 5

□ SYN □ ACK □ PSH □ RST □ FIN □ URG

Sequence Number: 446027991   ACK Sequence Number: 0

Window Size: 1296   Urgent Pointer: 0

To perform DOS attack, first specify the source and destination address (the source address can be spoofed), then specify the Number of Packets to send out. Tcp SYN attack can be performed by check the SYN option. ICMP and UDP attack can be performed by choose the ICMP or UDP pull-down option.

Suppose each of the 50 compromised cable modem/DSL systems have about 128 Kbps upstream connections. Combined together, they have a 128 Kbps*50=6.4 Mbps capacity to attack. Each SYN packet has a length of 40 bytes only, since it contains on data and no other IP or TCP options. The 50 compromised systems thus can send out 6.4 / (8 * 40) = 0.02 million SYN packets per second, i.e., 20 k packets per second. FireWall-1 on IPSO, by default, is capable of handling 25,000 concurrent connections. Besides, in Dennis Pickett's design, two IP 440 are used to provide for high availability. So, after 25 * 2 / 20 = 2.5 seconds, 25, 000 * 2 concurrent connection limit will be reached, i. e., a successful DOS attack is performed.

References

http://www.sans.org/infosecFAQ/DNS/sec_DNS.htm
http://www.bastille-linux.org
http://searchnetworking.techtarget.com/tip/1,289483,sid7_gci783169,00.html
http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html
http://www.telematik.informatik.uni-

karlsruhe.de/lehre/seminare/LinuxSem/downloads/netfilter/iptables-HOWTO-1.html
http://www.oofle.com/iptables/whatis.htm
http://www.nessus.org/
http://www.nmap.org
http://www.giac.org/practical/Dennis_Pickett_GCFW.zip
http://online.securityfocus.com/bid/2143
http://www.checkpoint.com/techsupport/alerts/fastmode.html
http://downloads.securityfocus.com/vulnerabilities/exploits/fm.c
http://www.soldierx.com/exploits/os/hardware/firewalls/firewall-1/fm.c
http://online.securityfocus.com/bid/2952/discussion/
http://www.checkpoint.com/techsupport/alerts/rdp_comms.html
http://www.checkpoint.com/techsupport/index.html
http://www.securiteam.com/exploits/5SP0B154UY.html
http://online.securityfocus.com/bid/1312
http://www.checkpoint.com/techsupport/alerts/ipfrag_dos.html
/data/vulnerabilities/exploits/jolt2.c
http://packetstormsecurity.nl/DoS/
http://packetstormsecurity.nl/UNIX/misc/ipsorc-1.6.tar.gz