Global Information Assurance Certification Paper

SANS GCFW Practical Assignment

Charles Pham
July 2003
Version 2.0

Table of Contents

Table of Contents 2
Assignment #1: Security Architecture 4
Company Overview 4
Business Operations Overview 4
Customers 4
Suppliers 5
Partners 5
Public 5
GIACE Employees on internal network 5
GIACE Employees connecting remotely 6
GIACE Network Diagram 7
IP Addressing 7
IP Address Mappings 8
Filtering Router 9
External Firewall/VPN 9
External Servers Network 10
Web server 10
External DNS+NTP server 10
External SMTP server 10
Snort IDS 10
Internal Firewall 11
Internal Servers Network 11
Internal DNS+NTP server 11
Internal Mail server 12
Database server 12
Management Network 12
Intrusion Monitor+STEALTH control.domain server 12
Syslog+SNMP server 12
Management Workstation 13
Backup Facility 13
Defense-in-Depth 13
Assignment #2: Security Policy and Tutorial 14
Security Policy / Configuration for Border Router 14
Router management: 14
Access Control List: 16
Security Policy for NetScreen Firewall 17
Tutorial for implementing policy on the NetScreen Firewall 22
Security Policy for NetScreen VPN 24
Security Zone and Tunnel: 24
Addresses: 25
Services: 25
VPN: 25

Routes: 26
Policies: 26
NetScreen-Remote: 26
Security Policy for NetFilter Firewall 27
Assignment #3: Verify the Firewall Policy 33
Plan the validation: 33
Technical approach: 33
Risks and considerations: 35
Cost and Effort Level: 35
Conduct the validation: 36
Results of scans from the Internet segment: 37
Results of scans from the External Servers segment: 39
Results of scans from the Users segment: 45
Evaluate the results: 46
Recommendations: 46
Assignment #4: Design under Fire 48
Attack against the Firewall 48
A Distributed Denial of Service attack: 50
An attack plan to compromise an internal system: 51
Application server: 51
Appendix 53
Example Rule Base for NetFilter Firewall by Chris Brenton 53
References 60

Assignment #1: Security Architecture

Company Overview

GIAC Enterprises, an e-business company that deals in the online sale of fortune cookie sayings. Through strategic partnership, exceptional services and support, and well-managed suppliers relationships, the company has captured 60% of the worldwide market. The company, with a workforce of 40 employees, has managed to generate steady annual revenue of approximately $5 million. The core asset to make this all possible is the company comprehensive fortune cookie sayings database that stores millions of fortune. Needless to say, due diligence is required to protect this vital piece of the business.

Business Operations Overview

As with any profit generating business, GIAC Enterprises must observe its bottom line in order to keep itself in the black. Therefore, technology solution used to achieve revenues must be as cost-effective as possible. Inherently, the design goal is not absolute security but rather the current most effective layered defense approach for the least cost.

Additional business requirements:

- 24x7x365 high availability is a nice to have and not a must.
- Scalability is a nice to have and not a must.

Customers

-Customers of GIACE can purchase the saying in bulk from the company website via HTTPS link (https://customers.giacecookies.com) on TCP port 443. Customers info such as login, password, email address, credit card number are stored in the GIACE secure database.
-Potential clients can sign up through a HTTPS link (https://signup.giacecookies.com) from the main web page (http://www.giacecookies.com) accessible via HTTP on TCP port 80. Validation of client info by GIACE employees is required prior to account activation.
-Following all valid transaction, an automated email confirmation containing the transaction ID, quantity and cost is sent to the customers.
-All inputs into GIACE web-forms are automatically validated prior to processing.
-All other access conditions listed under the public's access requirements also apply.

Suppliers

-Suppliers of GIACE can upload the fortune cookie sayings to the company website via a HTTPS link (https://suppliers.giacecookies.com) on TCP port 443. Suppliers info such as login, password, email address, payment details are stored in the GIACE secure database.
-Each saying is validated against the stored sayings in the secure database and credit is given to only those that are acceptable.
-Following all valid transaction, an automated email confirmation containing the transaction ID, quantity and credit is sent to the supplier.
-All inputs into GIACE web-forms are automatically validated prior to processing.
-All other access conditions listed under the public's access requirements also apply.

Partners

-Partners of GIACE can connect and download the sayings through the company website via a HTTPS link (https://partners.giacecookies.com) on TCP port 443. Partners info such as login, password, email address, accounts are stored in the GIACE secure database.
-Annual access fee is charged to the individual partner based on the number of sayings downloaded.
-Following all valid transaction, an automated email confirmation containing the transaction ID, and quantity is sent to the partner.
-All inputs into GIACE web-forms are automatically validated prior to processing.
-All other access conditions listed under the public's access requirements also apply.

Public

-GIACE host a public website (http://www.giacecookies.com) accessible via HTTP port 80. This is the main page and contains info such as about GIACE and its business, employment opportunities, and links to the customers, suppliers, and partners section.
-Email communication with GIACE is through SMTP on TCP port 25.
-GIACE will be authoritative for its own domain giacecookies.com, and thus inbound DNS query on UDP port 53 is openly accessible.

GIACE Employees on internal network

All employees of GIACE can access the Internet through HTTP on TCP port 80, HTTPS on TCP port 443, FTP on port 20 and 21 after being authenticated by the NetScreen firewall internal user database.

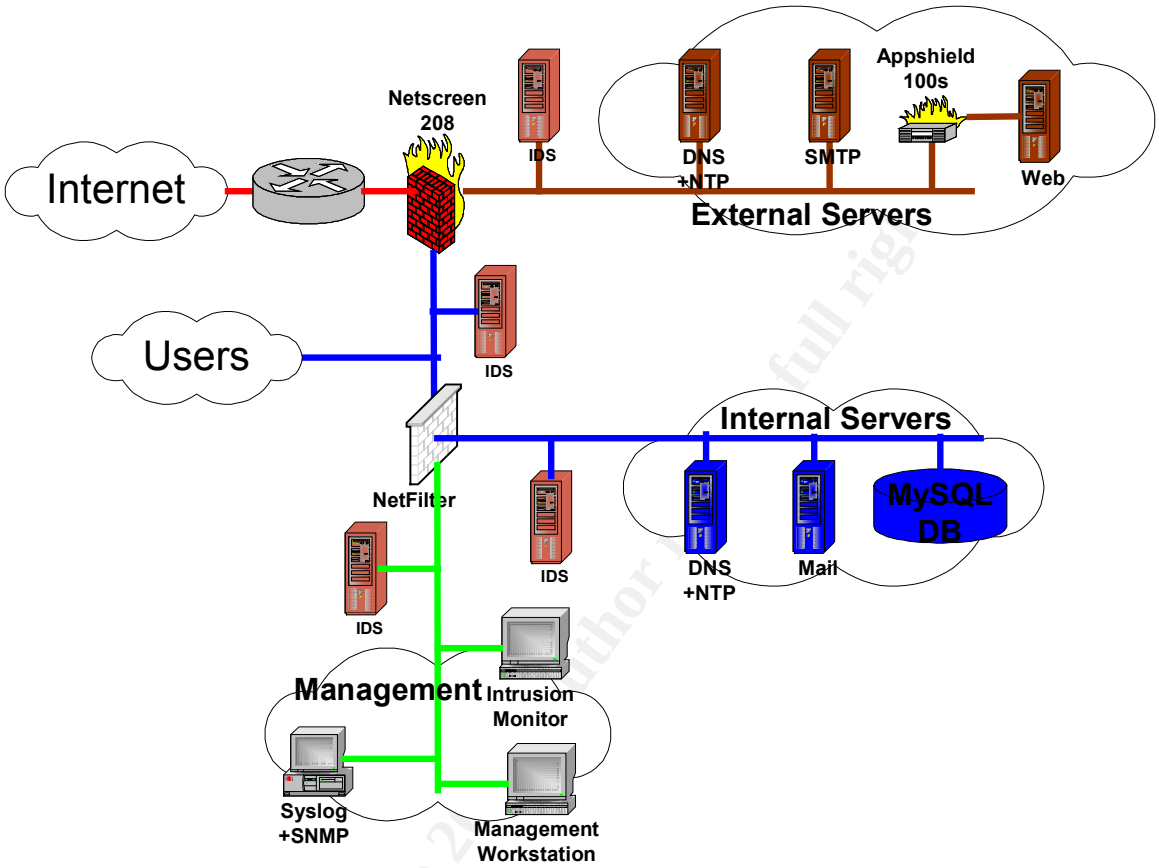
Email via SMTP on TCP port 25 and POP3 on TCP port 110 and DNS lookup on UDP port 53 are handled via the Internal Servers network through the NetFilter firewall.

System administrators are provided SSH access on TCP port 22 to all servers from the Management Workstation in the Management Network. Firewall administrators are provided with HTTPS access on TCP port 443 and SSH access on TCP port 22 to the NetScreen firewall, and SSH access on TCP port 22 to the NetFilter firewall from the Management Workstation. Database administrators and a selected group of individuals are provided with MySQL access on TCP port 3306 from the Management Workstation.

GIACE Employees connecting remotely

GIACE employees connecting from a remote site have the same access as those located on the internal network through a VPN connection. The NetScreen-Remote VPN client machine is protected with a GIACE configured integrated personal firewall. VPN connectivity utilizes XAUTH to authenticate the user to the NetScreen firewall/VPN in addition to IPSec authentication. Access controls are applied at the NetScreen firewall where the VPN terminate.

GIACE Network Diagram



IP Addressing

GIACE purchased a class C subnet of 999.999.999.0/27 (it's not real for this paper purpose), which allows for 30 valid Internet addresses. This is adequate for GIACE at this time considering the company size and the associate cost.

Network address translation is used to enable communication between the company internal networks and the Internet. Servers and devices (with the exception of the IDS sensor) on the External segment have a one to one mapping of Internet routable addresses. The remaining Internet routable addresses are mapped through dynamic network address translation for the Users segment.

GIACE uses the 192.168.0.0/16 on the company internal networks. The assignment are listed in the table below:

GIACE Internal Network	Subnet	Default GW
Border Router to External Firewall	192.168.0.0/30	
External Servers Network	192.168.1.0/24	192.168.1.1
Users Network	192.168.2.0/24	192.168.2.1
Internal Servers Network	192.168.3.0/24	192.168.3.1
Management Network	192.168.4.0/24	192.168.4.1

IP Address Mappings

Router, Firewalls	Interface	Internal IP Address	External IP Address
Cisco Border Router	External NetScreen FW		999.999.999.1
External NetScreen Firewall	Cisco Border Router		999.999.999.2
	External Servers Network	192.168.1.1	
	Users Network	192.168.2.1	
Internal NetFilter Firewall	Users Network	192.168.2.254	
	Internal Servers Network	192.168.3.1	
	Management Network	192.168.4.1	
External Servers			
Appshield 100s		192.168.1.15	999.999.999.4
Apache Web Server		192.168.1.200	
External DNS+NTP Server		192.168.1.5	999.999.999.3
External Mail Server		192.168.1.25	999.999.999.5
Snort IDS		192.168.1.253	
Users Network			
User Computers		192.168.2.2-251	
WebAuth IP		192.168.2.252	
Snort IDS		192.168.2.253	
Internal Servers			
Internal DNS+NTP Server		192.168.3.5	

Internal Mail Server		192.168.3.25	
MySQL Database		192.168.3.200	
Snort IDS		192.168.3.253	
Management Network			
Intrusion Monitor+STEALTH control.domain		192.168.4.50	
Management Workstation		192.168.4.99	
Syslog+SNMP Server		192.168.4.150	999.999.999.6
Snort IDS		192.168.4.253	

Filtering Router

GIACE uses a Cisco model 2691 with IOS version 12.3(1) for connection between the Internet and the NetScreen firewall to provide the first line of defense. Ingress and Egress filtering is used to filter out illegitimate packets such as those from IANA reserved addresses; loopback addresses and unallocated address. Management of the router is performed via console port.

External Firewall/VPN

GIACE uses a NetScreen-208 running NetScreen ScreenOS 4.0.3r2. This stateful inspection firewall appliance provides integrated security solution and simplified maintenance and administration. Its strong support for VPN with encrypted speed of 200 Mbps or greater makes it ideal for site-to-site and remote access VPN termination point. Other features such as Denial of Service protection, User authentication, Traffic management via Quality of Service, and High Availability makes this firewall a strong candidate as the second line of defense. As performance is one of the key concerns, defense at the application layer is deployed further downstream near or on the application servers themselves.

The WebAuth feature on the firewall is used to authenticate users before any access to the Internet is granted. This prevents unauthorized traffic flow such as those associated with trojan/spyware.

Logging and SNMP alerts are provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server.

External Servers Network

The external servers network consists of publicly accessible services including the GIACE website, External DNS+NTP, and External Mail server. Since budget is a major concern, open-source solution is selected whenever possible.

Web server

The GIACE website is hosted on a hardened Red Hat 9 running Apache 2.0.47 based on cost for performance and security measure. Appshield 100s appliance provides an application level firewall protection for the web server. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the External DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker. STEALTH software provides functionality similar to that of Tripwire, however, with one significant distinction – file hashes are offloaded to a remote control domain through SSH.

External DNS+NTP server

This server is hosted on a hardened Red Hat 9 running BIND 9.2.2 and NTP 4.1.1. The external DNS server is one part of the split DNS implementation where only the publicly accessible servers on the external network are resolvable. Zone transfers is limited to the ISP's DNS. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the External DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

External SMTP server

The mail server is hosted on a hardened Red Hat 9 running Postfix 2.0 patch level 13. The external SMTP server act as a relay between the Internet and the internal mail server. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the External DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Snort IDS

Intrusion Detection is provided through Snort 2.0 running on hardened Red Hat 9. Deployed strategically on the company's four network-segments, these IDS

sensors provide excellent intrusion detection coverage considering the cost. Each Snort server has two network cards with one configured to run in "stealth" mode with and IP address. The second interface is assigned an IP address to send the intrusion information to the Intrusion Monitor in the Management Network through MySQL TCP port 3306 with SSL. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

No sensor is deployed between the border router and the external firewall due to the lack of value it provides. Cost of responding to the large number intrusion attempts is not justifiable through business impact. In addition, such deployment can overwhelm the overall IDS solution should attackers flood the network with false attacks to trigger the signature detection. Lastly, Snort is not effective when it comes to encrypted traffic.

Internal Firewall

The Internal firewall is hosted on a hardened Red Hat 9 running iptables-1.2.8 provides a second layer of firewall defense through stateful packet filtering. This open source firewall has excellent logging capabilities and provides network level control around access to the Users, Internal Servers and Management Network segments.

Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Internal Servers Network

Internal DNS+NTP server

This server is hosted on a hardened Red Hat 9 running BIND 9.2.2 and NTP 4.1.1. The internal DNS server is the other part of the split DNS implementation where only internal systems on the intranet are resolvable. Zone transfers is not allowed and DNS query is forwarded to the External DNS+NTP server for which the internal server is non-authoritative. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Internal Mail server

The mail server is hosted on a hardened Red Hat 9 running Postfix 2.0 patch level 13. The internal server provides email service to users through SMTP and POP3 access. Administrative access is provided through OpenSSH 3.6.1. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Database server

The database server is hosted on a hardened Red Hat 9 running MySQL 4.0.13. This server is the main asset of the company's operation and contains the fortune cookies sayings along with customer, supplier, and partner information. Controlled SQL query access to the database is limited to just the web server on the external site and the Management Workstation in the Management Network segment. Administrative access is provided through OpenSSH 3.6.1 from the Management Workstation. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Management Network

Intrusion Monitor+STEALTH control.domain server

This server is hosted on a hardened Red Hat 9 running MySQL 4.0.13, Apache 2.0.46, PHP 4.3.2, ACID 0.9.6b23, and STEALTH 1.11. This server provides a consolidated view of all Snort alerts and STEALTH file integrity monitoring. Administrative access is provided through OpenSSH 3.6.1 from the Management Workstation. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Syslog+SNMP server

This server is hosted on a hardened Red Hat 9 running Syslogd, Swatch 3.0.8, and SNMPd configured to use version 2. This server provides a consolidated view of all SNMP alerts and Syslog information. Administrative access is provided through OpenSSH 3.6.1 from the Management Workstation. Logging is provided through local facility with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Management Workstation

This machine run on a hardened Red Hat 9 and provides a centralized management facility for system administration. Administrative access is provided through OpenSSH 3.6.1 to the selected administrators and individual requiring SQL query access to the main Database. Logging is provided through remote syslog to the Syslog+SNMP server in the Management Network with NTP time synchronization through the Internal DNS+NTP server. Additional layer of security is provided through STEALTH 1.11 File Integrity Checker.

Backup Facility

Backups are performed through locally installed DVD-ROM writers on a daily, weekly, and monthly basis depending on the criticality of the data involved. The DVDs are stored off-site located not too far from the company's physical facility.

Defense-in-Depth

The layered approach provides multiple levels of security control to protect an asset. Deployment of a variety of security technologies and platforms create a more effective defense against an intrusion. An attacker must find ways to piece together various exploits for each particular platform or technologies in order to be successful.

Enabling of filtering at the border router provides a basic level of protection against network-based attacks. The stateful firewalls provide a more advice level of network-layer attack protection. Using two different firewall vendors for the layered protection provides an extra level of security as the strengths and weaknesses of each vendor are complementary in controlling access to the various security zones. Weakness in NetScreen firewall such as being unable to block non-IP or ARP protocols are mitigated through filtering at the border router and the NetFilter firewall.

Application layer protection is provided through the Appshield 100s device, STEALTH File Integrity Checker, centralized logging, and intrusion detection devices.

Assignment #2: Security Policy and Tutorial

Security Policy / Configuration for Border Router

Router management:

First, enter privileged mode and give it a name.

```
Router > enable
Router # config term
Router (config)# hostname gborder
```

Assign password to console

```
gborder (config)#line console 0
gborder (config)#login
gborder (config)# password "consolepassword"
```

Password protect privileged mode with md5 hash

```
gborder (config)#enable secret “secretpassword”
gborder (config)#service password-encryption
```

Set console idle timeout

```
gborder (config)#line console 0
gborder (config)#exec-timeout 5 00
```

Deny remote logins

```
gborder (config)#access-list 10 deny any
gborder (config)#line vty 0 4
gborder (config)#access-class 10
```

Disable services that are not needed

```
gborder (config)#no snmp-server (no need at this time)
gborder (config)#no cdp run
gborder (config)#no ip source-route
gborder (config)#no service tcp-small-servers
gborder (config)#no service udp-small-servers
gborder (config)#no ip finger
gborder (config)#no ip bootp server
gborder (config)#no ip http server
```

```
gborder (config)#no ip dns server
gborder (config)#no service config
gborder (config)#no boot network
```

Send trap to SNMP server

```
gborder (config)#snmp-server community secret RW 11
gborder (config)#snmp-server enable traps snmp
gborder (config)#snmp-server host 999.999.999.6 traps version 2c
gborder (config)#access list 11 permit 999.999.999.6 0.0.0.255
gborder (config if)#ip access-group 11 in
```

Send log to Syslog server

```
gborder (config)#logging host 999.999.999.6
gborder (config)#logging trap emergencies
gborder (config)#logging trap alerts
gborder (config)#logging trap critical
gborder (config)#logging trap errors
gborder (config)#logging trap debug
gborder (config)#default logging buffered
```

Synchronize router clock with External NTP server

```
gborder (config)#ntp server 999.999.999.3
```

Serial interface configuration

```
gborder (config)#interface serial 0
gborder (config if)#no shutdown
gborder (config if)#no ip proxy-arp
gborder (config if)#no ip directed-broadcast
gborder (config if)#no ip unreachable
gborder (config if)#no ip redirect
gborder (config if)#no ipx network
gborder (config if)#no cdp enable
gborder (config if)#no decnet routing
gborder (config if)#no appletalk routing
gborder (config if)#ip access-group in_from_Internet in
```

Ethernet interface configuration

```
gborder (config)#interface ethernet 0
gborder (config if)#no shutdown
gborder (config if)#no ip proxy-arp
gborder (config if)#no ip directed-broadcast
```

```
gborder (config if)#no ip unreachable
gborder (config if)#no ip redirect
gborder (config if)#no ipx network
gborder (config if)#no cdp enable
gborder (config if)#no decnet routing
gborder (config if)#no appletalk routing
gborder (config if)#ip access-group in_from_Firewall in
```

Access Control List:

Ingress

Block all IANA reserved, loopback, broadcast, multicast, traffic without IP, traffic appearing from GIACE Internet address space, and auto configure addresses.

```
gborder (config)#ip access-list extended in_from_Internet
gborder (config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any log
gborder (config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
gborder (config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
gborder (config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
gborder (config-ext-nacl)# deny ip 255.0.0.0 0.255.255.255 any log
gborder (config-ext-nacl)# deny ip 224.0.0.0 15.255.255.255 any log
gborder (config-ext-nacl)# deny ip 0.0.0.0 any log
gborder (config-ext-nacl)# deny ip 999.999.999.0 0.0.0.31 any log
gborder (config-ext-nacl)# deny ip 169.254.0.0 0.0.255.255 any log
gborder (config-ext-nacl)# deny ip 240.0.0.0 7.255.255.255 any log
gborder (config-ext-nacl)# deny ip 248.0.0.0 7.255.255.255 any log
gborder (config-ext-nacl)# deny ip 192.0.2.0 0.0.0.255 any log
```

Block unallocated IP addresses

```
gborder (config-ext-nacl)# deny ip 1.0.0.0 0.255.255.255 any
gborder (config-ext-nacl)# deny ip 2.0.0.0 0.255.255.255 any log
gborder (config-ext-nacl)# deny ip 5.0.0.0 0.255.255.255 any log
etc..
```

Block ICMP types

```
gborder (config-ext-nacl)# deny icmp any any echo-request
gborder (config-ext-nacl)# deny icmp any any time exceeded
```

Allow remaining inbound packets to the GIACE Internet address space and allow the NetScreen firewall to perform the remaining protocol control.

```
gborder (config-ext-nacl)# permit ip any 999.999.999.0 0.0.0.31 log
```

Block everything else

```
gborder (config-ext-nacl)# deny ip any any log
```

Egress (partial)

Block packets leaving GIACE network with internal source address just in case the NetScreen firewall is leaky.

```
gborder (config)#ip access-list extended in_from_Firewall
gborder (config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
```

Allow remaining outbound packets from the GIACE Internet address space and allow the NetScreen firewall to perform the remaining protocol control.

```
gborder (config-ext-nacl)# permit ip 999.999.999.0 0.0.0.31 any
```

Block everything else

```
gborder (config-ext-nacl)# deny ip any any log
```

Security Policy for NetScreen Firewall

Rule ordering on the NetScreen firewall is important as it affects security filtering and performance. The rules are processed from top to bottom and abide by the principle of least privilege.

In addition, the concept of zoning allows grouping of policy to be explicit between various network segments.

Allow everyone HTTP and HTTPS access to the GIACE website. This rules is placed at the top for efficiency, as the majority of the traffic will flow through it. The IP is statically mapped to the real External Appshield 100s appliance IP address:

Zones: Internet to Global and Users to Global

Source	Destination	Service	Action	Options
Any	MIP(999.999.999.4)	HTTP		
Any	MIP(999.999.999.4)	HTTPS		

Allow the web server to perform direct SQL query against the MySQL database where all vital information resides.

Zone: External Servers to Management

192.168.1.200/255.255.255.255	192.168.3.200/255.255.255.255	MySQL		

Allow the DNS query to the External DNS server. Responses to the queries are small enough that there is no need to enable TCP rule for DNS. The IP is statically mapped to the real External DNS server IP address:

Zone: Internet to Global

Any	MIP(999.999.999.3)	DNS_query		

Allow the External DNS server to perform DNS query to the Internet:

Zone: External Servers to Internet

192.168.1.5/255.255.255.255	Any	DNS_query		

Allow Zone transfers between the External DNS and the ISP DNS servers.

Zones: Internet to External Servers

ISP_DNS	MIP(999.999.999.3)	DNS_tcp		

Allow mail flow between the Internet and the External SMTP server. The IP is statically mapped to the real External SMTP server IP address:

Zones: Internet to External Servers and vice-versa

Any	MIP(999.999.999.5)	MAIL		
192.168.1.25/255.255.255.255	Any	MAIL		

Allow mail flow between the External SMTP server and the Internal Mail server. Implementation of static routing table eliminates the for additional IP mapping.

Zones: Externals Servers to Internal Servers and vice-versa

192.168.1.25/255.255.255.255	192.168.3.25/255.255.255.255	MAIL		
192.168.3.25/255.255.255.255	192.168.1.25/255.255.255.255	MAIL		

Allows outbound HTTP, HTTPS, and FTP traffic from the Users segment. Individual user will need to authenticate through the WebAuth server before the traffic can be passed. DNS lookup is handled through the Internal DNS server.

Zone: Users to Internet

Source	Destination	Service	Action	Options
192.168.2.0/255.255.255.0	Any	HTTP		 
192.168.2.0/255.255.255.0	Any	HTTPS		 
192.168.2.0/255.255.255.0	Any	FTP		 

Allow Internal DNS server to forward DNS query to the External DNS server.

Zone: Internal Servers to External Servers

192.168.3.5/255.255.255.255	192.168.1.5/255.255.255.255	DNS query		

Allow Internal NTP server to synchronize its time with the External NTP server. All GIACE servers are time synchronize through NTP service to ensure effective use of centralized logging.

Zone: Internal Servers to External Servers

192.168.3.5/255.255.255.255	192.168.1.5/255.255.255.255	NTP		

Allow External NTP server to synchronize its time with public NTP server on the Internet. Lockdown to a specific destination is a more secure option, however, introduce additional overhead and control of the destination is left to that of the server's configuration.

Zone: External Servers to Internet

192.168.1.5/255.255.255.255	Any	NTP		

Allows management of the GIACE border router through NTP synchronization with the External NTP server and directing SNMP and Syslog traffic to the Syslog+SNMP server in the Management segment. The IPs are statically mapped to the real External NTP server and Syslog+SNMP server IP address respectively.

Zones: Internet to External Servers and Internet to Management

999.999.999.1	MIP(999.999.999.3)	NTP		
999.999.999.1	MIP(999.999.999.6)	SYSLOG		
999.999.999.1	MIP(999.999.999.6)	SNMP		

Allow the External IDS sensor to dump its alert directly to the Intrusion Monitor through a MySQL connection protected with SSL.

Zone: External Servers to Management

192.168.1.253/255.255.255.255	192.168.4.50/255.255.255.255	MySQL		

Allow SSH connectivity from the Management segment to the External Server segment for administration. Lockdown to specific IPs is an option but was not implemented to reduce administration cost.

Zone: Management to External Servers

192.168.4.0/255.255.255.0	192.168.1.0/255.255.255.0	SSH		

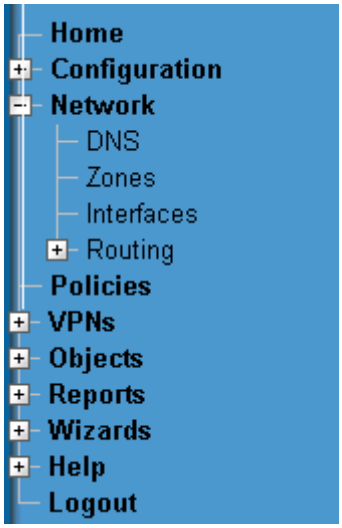
Allow Syslog and SNMP traffic to flow in the reverse direction to the Syslog+SNMP server.

Zone: External Servers to Management

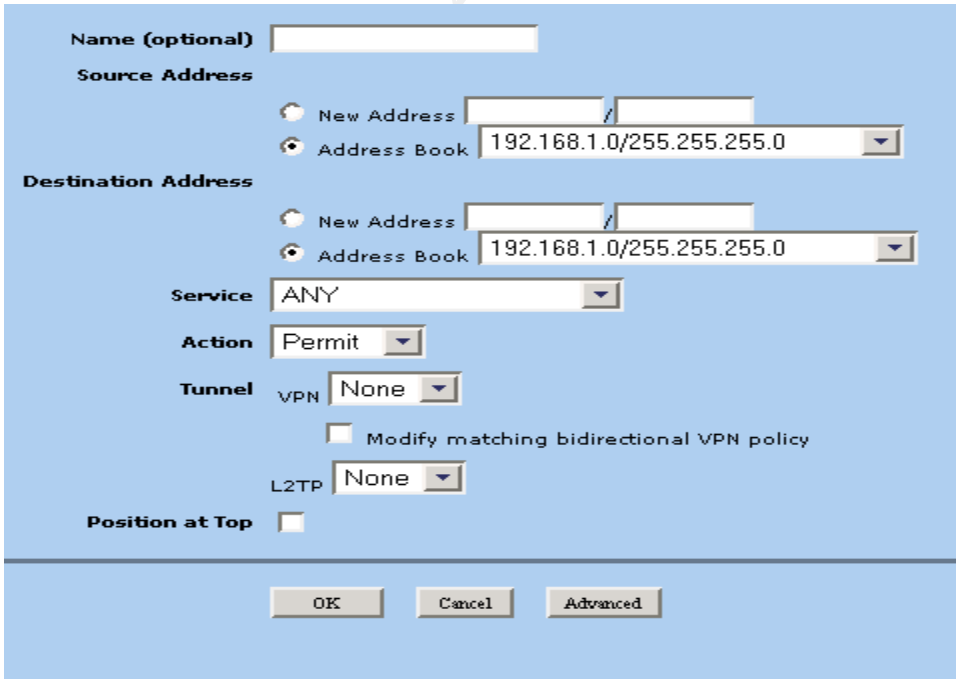
192.168.1.0/255.255.255.0	192.168.4.150/255.255.255.255	SYSLOG		
192.168.1.0/255.255.255.0	192.168.4.150/255.255.255.255	SNMP		

Tutorial for implementing policy on the NetScreen Firewall

Firewall rules are easily implemented through the web interface. Once login, the administrator is presented with a graphical user interface with the complete menu listed on the left hand side similar to the one below.

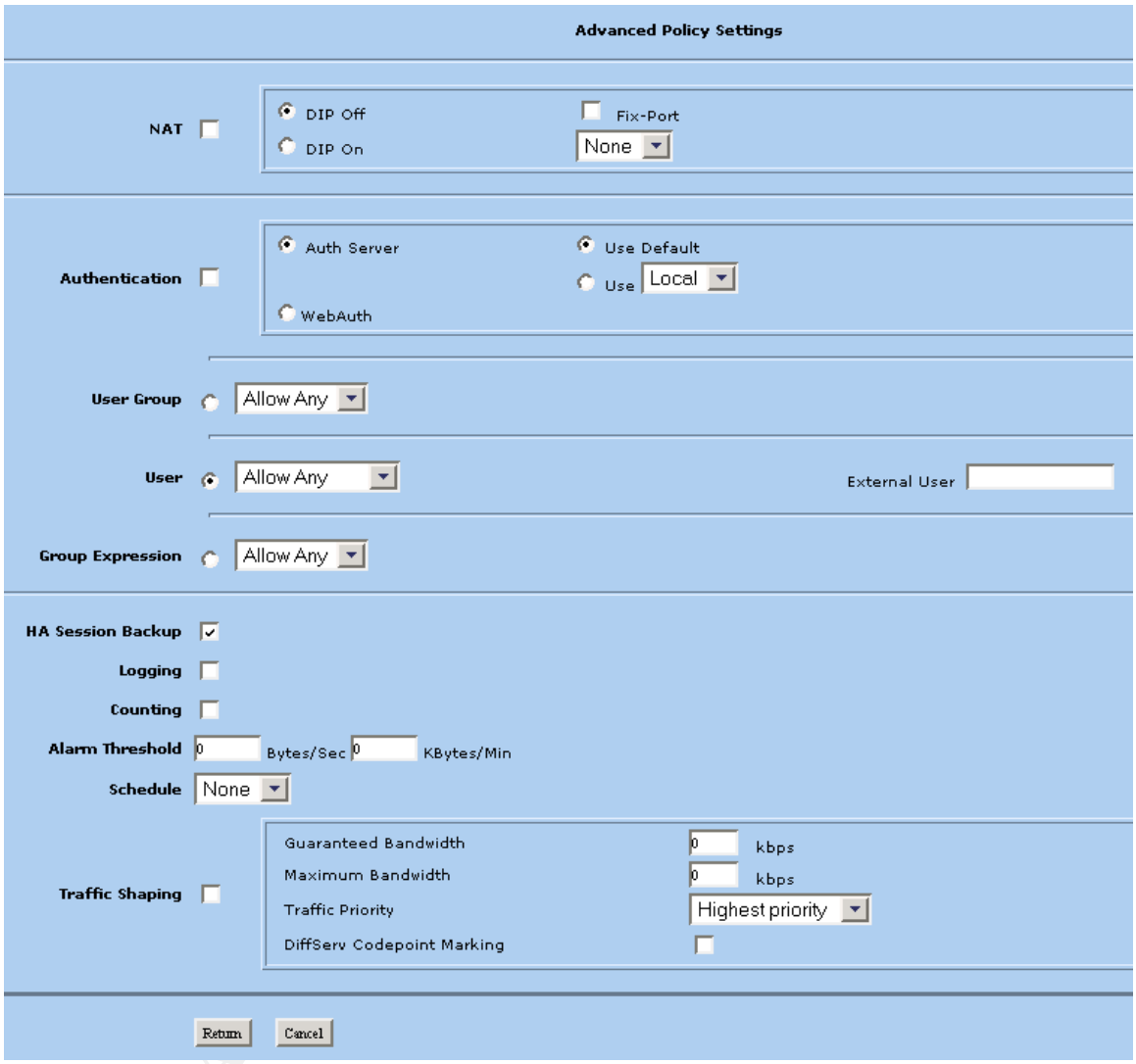


Clicking on the Policies, selecting the traffic flow from the defined zones and click on new would bring up the following screen on the right hand side. In this screen, the user would define the source, destination, service/protocol, permit/deny and click on OK for a simple rule.



Ticking the “Position at Top” option would put this rule on top of all the existing rules.

Other advanced options are available by clicking on the “Advanced” button at the bottom. This will bring up the following screen.



Policy-base NAT option above provides a greater flexibility in defining the source and direction where the translation would takes place.

User based authentication provides an addition layer of protection to the existing IP based authentication.

HA Session Backup is only applicable when the NetScreen device is participating in a NS Redundancy Protocol cluster.

Logging enable connection loggings of all traffics applicable to this rule.

Counting would track the total number of bytes of traffic applicable to this rule and record the information in historical graphs.

Alarm Threshold is useful in detecting DoS type of attacks.

Scheduling allows the rule to come into effect at specific interval during the day.

Traffic Shaping provides greater control over bandwidth usage.

Select the Return button to return to previous screen. Clicking OK at this point would save the rules. Sanity checks are performed on the inputs to ensure a proper rule can be created. An error message will pop-up if there is a problem with the rule. Note that some error messages can be quite cryptic.

Security Policy for NetScreen VPN

NetScreen 208 appliance can support both site-to-site and remote access VPN applications with encryption speed support greater than 200 Mbps. GIACE architecture requires the deployment of remote access VPN for remote employees connecting into the GIACE network. Using NetScreen Remote VPN client, remote employees are setup to establish dialup-to-LAN VPN using XAuth v6 and auto IKE and pre-shared key. Default lifetime for SA is used with Phase 1 SA expiring after the first 8 hours and Phase 2 SA expiring after 60 minutes. XAuth is set to expire after 10 hours. SHA-1 algorithm is used for the AH and 3DES for ESP. Aggressive mode (no ID protection) is used during IKE Phase 1 negotiations with pre-defined email address as its IKE ID and through standard proposal: preshared key; Diffie-Hellman group 2; 3DES; SHA. Phase 2 is set to operate in Quick mode with Replay Protection through standard proposal: Diffie-Hellman group 2; ESP; 3DES;SHA. Once connected the remote user is assigned a virtual IP address in the Users network segment and have the same privilege as users connecting locally within the GIACE office.

Configuration of the NetScreen VPN is generally as followed:

Security Zone and Tunnel:

Network > Interface > Edit (for Ethernet connected to the Users segment) set
    Zone Name: Users
    IP Address/Netmask: 192.168.2.1/24

Network > Interface > Edit (for Ethernet connected to the Internet segment) set
    Zone Name: Internet
    IP Address/Netmask: 999.999.999.3/30

Network > Interface > Tunnel IF New set

Tunnel Interface Name: tunnel.1
Zone Name: Users
Unnumbered: (select)
Interface: (same as Users segment)

Addresses:

Creating objects that the VPN user can connect to once the VPN has been established.

Creating objects for XAuth User.

Services:

Creating services that the VPN user can access on the objects once the VPN is up.

VPN:

VPNs > AutoKey Advanced > Gateway > New:
- Gateway Name: To_user
- Security Level: Custom
- Remote Gateway Type:
  - Dynamic IP Address: (select)
    - Peer ID: (pre-defined email address for IKE ID)
- Preshared Key: (set to something secret)
- Outgoing Interace: (Ethernet connected to the Internet)
- > Advanced:
  - Security Level: Custom
  - Phase 1 Proposal: pre-g2-3des-sha
  - Mode (Initiator): Aggressive
  - Enable XAUTH: (select)
    - Local Authentication: (select)
    - User: (select)
    - Name: (username)

VPNs > Autokey IKE > New:
- VPN Name: corp_user
- Security Level: Custom
- Remote Gateway:
  - Predefined: (select)
    - To_user
- > Advanced:
  - Security Level: Custom
  - Phase 2 Proposal: g2-esp-3des-sha

Replay Protection: (select)
Bind to: Tunnel Interface: (select)
Tunnel.1
Proxy-ID: (select)
Local IP/Netmask: 192.168.2.0/24
Remote IP/Netmask: Remote user's IP
Service: (defined services)

Routes:

Network > Routing > Routing Table > trust-vr New:
Network Address/Netmask: 0.0.0.0/0
Gateway: (select)
Interface: (Ethernet connected to Internet)
Gateway IP Address: ISP gateway

Network > Routing > Routing Table > trust-vr New:
Network Address/Netmask: Remote user's IP
Gateway: (select)
Interface: tunnel.1 (Internet)
Gateway IP Address: 0.0.0.0

Policies:

Creating policies around access controls for the VPN user.

Configuration for the NetScreen-Remote client is generally as followed:

NetScreen-Remote:

Options > Global Policy Settings:
Allow to Specify Internal Network Address: (select)

Options > Secure > Specified Connections
Add a new connection
Service type name
Connection Security: Secure
Remote Party ID Type: IP Address
IP Address: 192.168.2.0
Connect using Secure Gateway Tunnel: (select)
ID Type:
IP Address: 999.999.999.3
> Connection Policy
Security Policy
Aggressive Mode: (select)
My Identity

Pre-shared Key
Enter Key: (set to the same secret)
Internal Network IP Address: Remote user's IP
ID Type: (the same pre-defined email address for IKE ID)
Authentication (Phase 1) > Proposal 1:
Encrypt Alg: Triple DES
Hash Alg: SHA-1
Key Group: Diffie-Hellman Group 2
Key Exchange (Phase 2) > Proposal 1:
Encapsulation Protocol (ESP): (select)
Encrypt Alg: Triple DES
Hash Alg: SHA-1
Encapsulation: Tunnel
Save.

Security Policy for NetFilter Firewall

The NetFilter firewall was selected to protect the Management and Internal Server Networks. This provides a second layer defense through stateful packet filtering and comprehensive logging of traffic. Static routing is implemented for packet routing between networks rather than using NAT.

Rule ordering on the NetFilter firewall is also important as it affects security filtering and performance. The rules are processed from top to bottom of the chain and abide by the principle of least privileged.

Through the startup rc.firewall script inclusion in /etc/rc.d/rc.local, the firewall policies is automatically loaded during bootup. Changes to the script can be loaded during machine uptime by running the rc.firewall script.

The rules below are heavily adopted by a sample NetFilter rule set written by Chris Brenton, SANS Firewall Track instructor. A copy of this sample rule set is attached in the Appendix.

```
# Flush all old rules on restart

iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Default policies to drop all packets

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Declare environment variables allowing change in IPs without having to change the firewall rules.

```
ROUTER=999.999.999.1
ALL_INT_NETS=192.168.0.0/16
USERS_NET=192.168.2.0/24
MGT_NET=192.168.4.0/24
SYSLOG_SNMP=192.168.4.150
INTRUSION_MON=192.168.4.50
IDS1=192.168.1.253
IDS2=192.168.2.253
IDS3=192.168.3.253
EXT_DNS=192.168.2.5
EXT_SMTP=192.168.2.25
INT_DNS=192.168.3.5
INT_MAIL=192.168.3.25
MySQL_DB=192.168.3.200
MGT_WKS=192.168.4.99
WEB_SRV=192.168.1.200
```

```
# --------FORWARD CHAIN--------
#
# Allow all established state matches through

iptables -A FORWARD -m state --state ESTABLISH,RELATED -j ACCEPT

# Log ,and drop where applicable, illegal or suspicious packets.

iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j LOG --log-prefix " SFScan "
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j DROP

iptables -A FORWARD -p tcp --tcp-flags ALL SYN,ACK -d $MGT_NET -j LOG --log-prefix "
SAScan to MGT_NET"
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,ACK -d $INTERNAL_NET -j LOG --log-
prefix " SAScan to INTERNAL_NET "

iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j LOG --log-prefix " FSscan "

iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j LOG --log-prefix " NUScan "
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP

iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j LOG --log-prefix " XMAS"
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP

iptables -A FORWARD -p icmp -f -j LOG --log-prefix " ICMPFRAG "
iptables -A FORWARD -p icmp -f -j DROP

# Block and log all spoofed source IP packets

iptables -A FORWARD -p tcp -s 172.16.0.0/12 -d 0/0 -j LOG --log-prefix " 172.16 spoof "
iptables -A FORWARD -p tcp -s 172.16.0.0/12 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 10.0.0.0/8 -d 0/0 -j LOG --log-prefix " 10.0 spoof "
iptables -A FORWARD -p tcp -s 10.0.0.0/8 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 0.0.0.0/32 -d 0/0 -j LOG --log-prefix " 0.0 spoof "
iptables -A FORWARD -p tcp -s 0.0.0.0/32 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 127.0.0.0/8 -d 0/0 -j LOG --log-prefix " 127.0.0.1 spoof "
iptables -A FORWARD -p tcp -s 127.0.0.0/8 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 224.0.0.0/8 -d 0/0 -j LOG --log-prefix " 224.0.0.0 spoof "
iptables -A FORWARD -p tcp -s 224.0.0.0/8 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 169.254.0.0/16 -d 0/0 -j LOG --log-prefix " Autoconf spoof "
iptables -A FORWARD -p tcp -s 169.254.0.0/16 -d 0/0 -j DROP
iptables -A FORWARD -p tcp -s 240.0.0.0/4 -d 0/0 -j LOG --log-prefix " 240.0.0.0 spoof "
iptables -A FORWARD -p tcp -s 240.0.0.0/4 -d 0/0 -j DROP

# Pass SYSLOG and SNMP TRAP traffic unlogged for performance
```

```
iptables -A FORWARD -m state --state NEW -p udp -s $ALL_INT_NETS --sport 1024:65535 -
d $SYSLOG_SNMP --dport 514 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s $ALL_INT_NETS --sport 1024:65535 -
d $SYSLOG_SNMP --dport 162 -j ACCEPT

# Pass MySQL traffic from IDS sensors unlogged for performance and value as these
traffic are encrypted with SSL.

iptables -A FORWARD -m state --state NEW -p tcp -s $IDS1 --sport 1024:65535 -d
$INTRUSION_MON --dport 3306 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $IDS2 --sport 1024:65535 -d
$INTRUSION_MON --dport 3306 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $IDS3 --sport 1024:65535 -d
$INTRUSION_MON --dport 3306 -j ACCEPT

# Pass DNS and NTP queries to the Internal DNS and NTP server unlogged for
performance

iptables -A FORWARD -m state --state NEW -p udp -s $ALL_INT_NETS -d $INT_DNS --
dport 123 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s $ALL_INT_NETS -d $INT_DNS --
dport 53 -j ACCEPT

# Allow the Internal NTP server to synchronize with the External NTP server

iptables -A FORWARD -m state --state NEW -p udp -s $INT_DNS -d $EXT_DNS --dport 123 -
j ACCEPT

# Allow the Internal DNS server to forward DNS Queries to the External NTP server

iptables -A FORWARD -m state --state NEW -p udp -s $INT_DNS -d $EXT_DNS --dport 53 -j
ACCEPT

# Allow the Internal Mail server to communicate with the External SMTP server

iptables -A FORWARD -m state --state NEW -p tcp -s $INT_MAIL -d $EXT_SMTP --dport 25
-j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $EXT_SMTP -d $INT_MAIL --dport 25
-j ACCEPT

# Allow people on the Users Network access to the Internal Mail server

iptables -A FORWARD -m state --state NEW -p tcp -s $USERS_NET -d $INT_MAIL --dport
25 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $USERS_NET -d $INT_MAIL --dport
110 -j ACCEPT
```

Pass SSH, HTTP and HTTPS from Management Network unlogged for performance and value as the traffic will be encrypted (exception of HTTP).

```
iptables -A FORWARD -m state --state NEW -p tcp -s $MGT_NET -d 0/0 --dport 22 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $MGT_NET --sport 1024:65535 -d 0/0 --dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $MGT_NET --sport 1024:65535 -d 0/0 --dport 443 -j ACCEPT
```

Allow SQL query to the MySQL database from the Management workstation and the Web server but log the access as the database is the crown jewel that needs the most protection.

```
iptables -A FORWARD -m state --state NEW -p tcp -s $MGT_WKS -d $MySQL_DB --dport 3306 -j LOG --log-prefix " SQL query from MGT "
iptables -A FORWARD -m state --state NEW -p tcp -s $MGT_WKS -d $MySQL_DB --dport 3306 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s $WEB_SRV -d $MySQL_DB --dport 3306 -j LOG --log-prefix " SQL query from WEB "
iptables -A FORWARD -m state --state NEW -p tcp -s $WEB_SRV -d $MySQL_DB --dport 3306 -j ACCEPT
```

#Log and drop the rest

```
iptables -A FORWARD -s 0/0 -j LOG --log-level info --log-prefix " DROP_FORWARD "
```

--------INPUT CHAIN--------
#
Define the access policy for all traffic destined to the firewall itself.

Ignore NetBIOS, RPC, DHCP/BOOTP, broadcasts and SMB traffic

```
iptables -A INPUT -s 0/0 –d 0/0 --dport 135:139 -j DROP
iptables -A INPUT -s 0/0 –d 0/0 --dport 445 -j DROP
iptables -A INPUT -s 0/0 –d 0/0 --dport 67:68 -j DROP
iptables -A INPUT -s 0/0 -d 255.255.255.255 -j DROP
```

Allow established sessions through

```
iptables -A INPUT -m state --state ESTABLISH,RELATED -j ACCEPT
```

Allow firewall to be managed via SSH

```
iptables -A INPUT -m state --state NEW -p tcp -s $MGT_NET -d 192.168.4.1 --dport 22 -j
ACCEPT

#Log and drop the rest

iptables -A INPUT -s 0/0 -j LOG --log-prefix " DROP_INPUT "


# --------OUTPUT CHAIN--------
#
# Define the access policy for all traffic originating from the firewall itself.

# Allow SYSLOG out from Localhost

iptables -A OUTPUT -m state --state NEW -p udp -d $SYSLOG_SNMP --dport 514 -j ACCEPT

# Allow established sessions through

iptables -A OUTPUT -m state --state ESTABLISH,RELATED -j ACCEPT

# Drop all outbound unreachable traffic

iptables -A OUTPUT -p icmp -s 0/0 -d 0/0 -j DROP

# Allow Localhost NTP out to the Internal NTP server

iptables -A OUTPUT -m state --state NEW -p udp -d $INT_DNS --dport 123 -j ACCEPT

#Log and drop the rest

iptables -A OUTPUT -s 0/0 -j LOG --log-prefix " DROP_OUTPUT "
```

Assignment #3: Verify the Firewall Policy

Plan the validation:

In an effort to validate GIACE infrastructure is secured as per designed set by the policy, an independent audit on the External NetScreen firewall was requested. Paper work such as authorization letter and Non-disclosure agreement were signed and completed without complication. Two independent auditors were hired to perform the audit during GIACE regular maintenance window of 6 hours on Saturday and Sunday morning between 1AM to 7AM. Running the scan during this time period will ensure that downtime can be kept to minimal which translates to acceptable business impact should problem arises. Members of various technical team will be on stand-by incase of potential problem. Change management process is followed and communication of the audit schedule is communicated to the various stakeholders.

Technical approach:

The review will have the following scopes:

- Documentation with respect to architecture, network design, data flows and firewall policies.
- Firewall rules. It is important to note that the one implied policy that the Firewall must enforce is the "principle of least privilege". Validation of this policy required a comprehensive audit, which fall not too far from a vulnerability assessment. In the interest of saving time and cost, shortcut will be taken when possible.

Nmap will be run from the following segments:

- Internet – a hub is required to be insert between the border router and the firewall.
- External Servers network
- Users network

Internet

From the Internet-Nmap will be used to check for response to ping and open port (TCP&UDP) on the following components:

1) Firewall/VPN (.2)
2) External DNS+NTP server (.3)
3) External SMTP server (.5)
4) Web server (.4)

Next NMAP will be used to scan the following machines for ping response and open port (TCP&UDP)

5) External DNS+NTP server (.3)
6) Syslog+SNMP server (.6)

using IPs from the following components (the machine will need to be taken off-line temporarily):

1) Border Router (.1)

External Servers

From the External Servers network-Nmap will be used to check for response to ping and open port (TCP&UDP) to the following components:

1) Firewall (.1.1)

Next NMAP will be used to scan the following machines for ping response and open port (TCP&UDP)

1) 192.168.2.253 (representative for the Users network)
2) 192.168.3.5;.25;.200;.253 (representative for the Internal Servers network)
3) 192.168.4.50;.99;.150;.253 (representative for the Management network)
4) 999.999.999.1 (representative for the Internet)

using IPs from the following components (the machine will need to be taken off-line temporarily):

2) External DNS+NTP server (.5)
3) External SMTP server (.25)
4) Appshield 100s (.15)
5) Apache Web Server (.200)
6) Snort IDS (.253)

Users

From the Users network-Nmap will be used to check for response to ping and open port (TCP&UDP) to the following components:

1) Firewall (.2.1)
2) 192.168.1.5;.15;.25;.200;.253 (representative of the External Servers network)
3) 999.999.999.1 (representative for the Internet)

Note that the audit method described above cannot determine if the firewall is passing traffic through (i.e. Leaky firewall rules) when it shouldn't to a non-existent/responsive host. Therefore, a network sniffer (tcpdump) will be deployed on the destination network to address this problem.

Risks and considerations:

GIACE cannot afford pro-longed period of downtime as the loss in revenue would heavily impact its business.

There is a slight risk that a malformed packet might bring down a network device or the firewall itself. This should be easily remedy by having the proper staff on standby to restart the device, process, or machine.

There is also a possibility that logging devices might outgrow existing disk space. Administrators will need to set a disk space usage threshold and keep a tight check on these devices to ensure that it will not a problem.
There is also a possibility at a real attack might take place during the audit. As such, GIACE staffs must keep the information regarding the audit to a need to know basis and conscientiously monitors all logs and IDS. The audit will cease until further notice if the threat from a real attack begins to impact the business.

It is expected that bandwidth will not be an issue as the audit does not include firewall performance.

Cost and Effort Level:

Based on current approach, 18 different scans are required at a cost of approximately 0.5hr/each. Additional 1 hour of preparation time in between scans.

Task	Hours required	Cost@$150/hr
Documentation Review	8	
Conduct Nmap scan from each firewall interface	16	
Analysis of findings	8	
Reporting	16	
Total	**48**	**$7,200**

Conduct the validation:

To start, ping will be used to determine if the IP_address is responsive through the system ping command:

Ping IP_address

For TCP, Syn scans was used with the following options:

Nmap –vv –sS –n –P0 IP_address

-vv for detailed verbose
-sS for TCP SYN or "half-open" scan
-n to not do any reverse DNS resolution
-P0 to not ping host before scanning

In order to preserve scan time, only the default 1601 ports will be scanned. This covers all privileged ports (<1024) and ports listed in the Nmap services file.

When testing the firewall rules, Ack scans will be used in addition to the Syn scans with the following options:

Nmap –vv –sA –n –P0 IP_address

-sA for Ack scan to test firewall statefulness.

For UDP, UDP scans was used with the following options:

Nmap –vv –sU –n –P0 IP_address

-sU for UDP scans

In order to preserve scan time, only the default 1468 ports will be scanned.

Lastly, Nmap with the –g option was used selectively to make sure that the firewall does not allow tunneled scans through:

-g 53 with UDP scans masquerading as DNS servers

-g 80 with TCP scans masquerading as HTTP servers.

The results confirmed that the firewall does filter these types of packets.

Results of scans from the Internet segment:

First ping the Firewall:

```
PING 999.999.999.2 (999.999.999.2): 56 octets data

--- 999.999.999.2 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss
```

Good. Firewall is not responding to ICMP echo-request. Similar results were obtained for the External DNS+NTP, SMTP, and web servers.
For the TCP scans:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 999.999.999.2
All 1601 scanned ports on  (999.999.999.2) are: filtered

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1612 seconds
```

Good. Nothing interesting. Similar result for TCP scans against the DNS server.
Proceeding to UDP scans:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 999.999.999.2
 (The 1467 ports scanned but not shown below are in state: closed)
Port       State       Service
500/udp    open        isakmp

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1767 seconds
```

Again nothing interesting as port 500 is used by the VPN. Similar results were found when scanning the External SMTP and web server with UDP scans.

Proceeding to UDP scans against the DNS server:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 999.999.999.3
Interesting ports on  (999.999.999.3):
(The 1467 ports scanned but not shown below are in state: closed)
Port       State       Service
53/udp     open        domain

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1760 seconds
```

Good. This is as expected. Moving on to TCP against External SMTP:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 999.999.999.5
Interesting ports on  (999.999.999.5):
(The 1600 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1379 seconds
```

Good as expected. Now the ACK scan:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sA -n -P0 999.999.999.5
Interesting ports on  (999.999.999.5):
(The 1600 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     UNfiltered       smtp

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 859 seconds
```

Good. Firewall is performing stateful inspection. Proceeding to the web server:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 999.999.999.4
Interesting ports on  (999.999.999.4):
(The 1599 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 947 seconds
```

Good as expected. Now the ACK scan:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sA -n -P0 999.999.999.4
Interesting ports on  (999.999.999.4):
(The 1599 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     UNfiltered       http
443/tcp    UNfiltered       https

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 638 seconds
```

Again, as expected and state are being enforced by the firewall. Next, using the border router IP of (.1), scans were performed against the NTP server. As expected, nothing to show in the TCP results. The result for UDP scans below:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 999.999.999.3
Interesting ports on  (999.999.999.3):
(The 1466 ports scanned but not shown below are in state: closed)
Port       State       Service
53/udp     open        domain
123/udp    open        ntp

# Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1453 seconds
```

The rules are working as expected. Moving on, TCP scans against the Syslog+SNMP server yield no results. Results from the UDP scans below:

```
# nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 999.999.999.6
Interesting ports on  (999.999.999.6):
(The 1466 ports scanned but not shown below are in state: closed)
Port       State       Service
```

162/udp open snmptrap
514/udp open syslog

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1529 seconds

No surprise here either and validation of the firewall rules from the Internet is complete.

Results of scans from the External Servers segment:

First ping the Firewall:

PING 192.168.1.1 (192.168.1.1): 56 octets data

--- 192.168.1.1 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

Good. Firewall is not responding to ICMP echo-request. Similar results obtained for the Users, Internal Servers, Internet, and Management networks. For the TCP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 192.168.1.1
All 1601 scanned ports on (192.168.1.1) are: filtered

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1645 seconds

Good. Nothing interesting. Proceeding to UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 192.168.1.1
All 1468 scanned ports on (192.168.1.1) are: filtered

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1638 seconds

As expected as the firewall does not host any service on this segment. Next up is TCP scanning from the External DNS+NTP server (.5):

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 –S 192.168.1.5 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1601 scanned ports on (192.168.2.253) are: filtered

All 1601 scanned ports on (192.168.3.5) are: filtered

All 1601 scanned ports on (192.168.3.25) are: filtered

All 1601 scanned ports on (192.168.3.200) are: filtered

All 1601 scanned ports on (192.168.3.253) are: filtered

All 1601 scanned ports on (192.168.4.50) are: filtered

All 1601 scanned ports on (192.168.4.99) are: filtered

All 1601 scanned ports on (192.168.4.150) are: filtered

All 1601 scanned ports on (192.168.4.253) are: filtered

All 1601 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 10569 seconds

This is as expected and Ack scans yield the same results and firewall working as expected. For the UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 –S 192.168.1.5 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1468 scanned ports on (192.168.2.253) are: filtered

All 1468 scanned ports on (192.168.3.5) are: filtered

All 1468 scanned ports on (192.168.3.25) are: filtered

All 1468 scanned ports on (192.168.3.200) are: filtered

All 1468 scanned ports on (192.168.3.253) are: filtered

All 1468 scanned ports on (192.168.4.50) are: filtered

All 1468 scanned ports on (192.168.4.99) are: filtered

Interesting ports on (192.168.4.150):
(The 1466 ports scanned but not shown below are in state: closed)

Port	State	Service
162/udp	open	snmptrap
514/udp	open	syslog

All 1468 scanned ports on (192.168.4.253) are: filtered

All 1468 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 12732 seconds

As expected the DNS server is allow to send SNMP trap and Syslog to the Syslog+SNMP server in the Management segment. Although UDP scans to the Internet yields all ports are filtered, tcpdump sniffer placed outside of the firewall did capture packet outbound for NTP (123) and DNS (53). This is working as expected.

Next up is TCP scanning from the External SMTP server (.25) with the following changes to the Nmap command:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 –S 192.168.1.25 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1601 scanned ports on (192.168.2.253) are: filtered

All 1601 scanned ports on (192.168.3.5) are: filtered

Interesting ports on (192.168.3.25):
(The 1600 ports scanned but not shown below are in state: filtered)
Port State Service
25/tcp open smtp

All 1601 scanned ports on (192.168.3.200) are: filtered

All 1601 scanned ports on (192.168.3.253) are: filtered

All 1601 scanned ports on (192.168.4.50) are: filtered

All 1601 scanned ports on (192.168.4.99) are: filtered

All 1601 scanned ports on (192.168.4.150) are: filtered

All 1601 scanned ports on (192.168.4.253) are: filtered

All 1601 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 11367 seconds

This is as expected and Ack scans yield port 25 on 192.168.3.25 results as UNfiltered and firewall working as expected. Although TCP scans to the Internet yields all ports are filtered, tcpdump sniffer placed outside of the firewall did capture packet outbound for SMTP(25). This is working as expected. For the UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 –S 192.168.1.25 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1468 scanned ports on (192.168.2.253) are: filtered

All 1468 scanned ports on (192.168.3.5) are: filtered

All 1468 scanned ports on (192.168.3.25) are: filtered

All 1468 scanned ports on (192.168.3.200) are: filtered

All 1468 scanned ports on (192.168.3.253) are: filtered

All 1468 scanned ports on (192.168.4.50) are: filtered

All 1468 scanned ports on (192.168.4.99) are: filtered

Interesting ports on (192.168.4.150):
(The 1466 ports scanned but not shown below are in state: closed)
Port State Service
162/udp open snmptrap
514/udp open syslog

All 1468 scanned ports on (192.168.4.253) are: filtered

All 1468 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 13336 seconds

As expected the DNS server is allow to send SNMP trap and Syslog to the Syslog+SNMP server in the Management segment.

Next up is TCP scanning from the Appshield 100s (.15) with the following changes to the Nmap command:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 –S 192.168.1.15 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1601 scanned ports on (192.168.2.253) are: filtered

All 1601 scanned ports on (192.168.3.5) are: filtered

All 1601 scanned ports on (192.168.3.25) are: filtered

All 1601 scanned ports on (192.168.3.200) are: filtered

All 1601 scanned ports on (192.168.3.253) are: filtered

All 1601 scanned ports on (192.168.4.50) are: filtered

All 1601 scanned ports on (192.168.4.99) are: filtered

All 1601 scanned ports on (192.168.4.150) are: filtered

All 1601 scanned ports on (192.168.4.253) are: filtered

All 1601 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 10291 seconds

This is as expected and Ack scans yield port 80, and 443 on 999.999.999.1 as UNfiltered and firewall working as expected. Although TCP Syn scans to the Internet yields all ports are filtered, tcpdump sniffer placed outside of the firewall did capture packet outbound for HTTP(80) and HTTPS(443). This is working as expected. For the UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 –S 192.168.1.15 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1468 scanned ports on (192.168.2.253) are: filtered

All 1468 scanned ports on (192.168.3.5) are: filtered

All 1468 scanned ports on (192.168.3.25) are: filtered

All 1468 scanned ports on (192.168.3.200) are: filtered

All 1468 scanned ports on (192.168.3.253) are: filtered

All 1468 scanned ports on (192.168.4.50) are: filtered

All 1468 scanned ports on (192.168.4.99) are: filtered

Interesting ports on (192.168.4.150):
(The 1466 ports scanned but not shown below are in state: closed)
Port State Service
162/udp open snmptrap
514/udp open syslog

All 1468 scanned ports on (192.168.4.253) are: filtered

All 1468 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 14256 seconds

As expected the DNS server is allow to send SNMP trap and Syslog to the Syslog+SNMP server in the Management segment.

Next up is TCP scanning from the Apache Web Server (.200) with the following changes to the Nmap command:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 –S 192.168.1.200 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1601 scanned ports on (192.168.2.253) are: filtered

All 1601 scanned ports on (192.168.3.5) are: filtered

All 1601 scanned ports on (192.168.3.25) are: filtered

Interesting ports on (192.168.3.200):
(The 1600 ports scanned but not shown below are in state: filtered)
Port State Service
3306/tcp open mysql

All 1601 scanned ports on (192.168.3.253) are: filtered

All 1601 scanned ports on (192.168.4.50) are: filtered

All 1601 scanned ports on (192.168.4.99) are: filtered

All 1601 scanned ports on (192.168.4.150) are: filtered

All 1601 scanned ports on (192.168.4.253) are: filtered

All 1601 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 10733 seconds

This is as expected and Ack scans yield port 3306 on 192.168.3.200 as UNfiltered and firewall working as expected as the web server cannot communicate directly to the Internet. The only allowable communication is to Asset Database in the Internal Server segment. For the UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 –S 192.168.1.200 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1468 scanned ports on (192.168.2.253) are: filtered

All 1468 scanned ports on (192.168.3.5) are: filtered

All 1468 scanned ports on (192.168.3.25) are: filtered

All 1468 scanned ports on (192.168.3.200) are: filtered

All 1468 scanned ports on (192.168.3.253) are: filtered

All 1468 scanned ports on (192.168.4.50) are: filtered

All 1468 scanned ports on (192.168.4.99) are: filtered

Interesting ports on (192.168.4.150):
(The 1466 ports scanned but not shown below are in state: closed)

Port	State	Service
162/udp	open	snmptrap
514/udp	open	syslog

All 1468 scanned ports on (192.168.4.253) are: filtered

All 1468 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 13849 seconds

As expected the DNS server is allow to send SNMP trap and Syslog to the Syslog+SNMP server in the Management segment.

Next up is TCP scanning from the Snort IDS (.253) with the following changes to the Nmap command:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 –S 192.168.1.253 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1601 scanned ports on (192.168.2.253) are: filtered

All 1601 scanned ports on (192.168.3.5) are: filtered

All 1601 scanned ports on (192.168.3.25) are: filtered

All 1601 scanned ports on (192.168.3.200) are: filtered

All 1601 scanned ports on (192.168.3.253) are: filtered

Interesting ports on (192.168.4.50):
(The 1600 ports scanned but not shown below are in state: filtered)

Port	State	Service
3306/tcp	open	mysql

All 1601 scanned ports on (192.168.4.99) are: filtered

All 1601 scanned ports on (192.168.4.150) are: filtered

All 1601 scanned ports on (192.168.4.253) are: filtered

All 1601 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 10368 seconds

This is as expected and Ack scans yield port 3306 on 192.168.4.50 as UNfiltered and firewall working as expected as the IDS only communicate with the Intrusion Monitor in the Management segment. For the UDP scans:

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 –S 192.168.1.253 –e eth0 192.168.2.253, 192.168.3.5,25,200,253, 192.168.4.50,99,150,253, 999.999.999.1
All 1468 scanned ports on (192.168.2.253) are: filtered

All 1468 scanned ports on (192.168.3.5) are: filtered

All 1468 scanned ports on (192.168.3.25) are: filtered

All 1468 scanned ports on (192.168.3.200) are: filtered

All 1468 scanned ports on (192.168.3.253) are: filtered

All 1468 scanned ports on (192.168.4.50) are: filtered

All 1468 scanned ports on (192.168.4.99) are: filtered

Interesting ports on (192.168.4.150):
(The 1466 ports scanned but not shown below are in state: closed)

Port	State	Service
162/udp	open	snmptrap
514/udp	open	syslog

All 1468 scanned ports on (192.168.4.253) are: filtered

All 1468 scanned ports on (999.999.999.1) are: filtered

 # Nmap run completed at [DATE] -- 10 IP address (10 host up) scanned in 13682 seconds

As expected the DNS server is allow to send SNMP trap and Syslog to the Syslog+SNMP server in the Management segment.

Results of scans from the Users segment:

First ping the Firewall:

PING 192.168.2.1 (192.168.2.1): 56 octets data

--- 192.168.2.1 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sS -n -P0 192.168.2.1
All 1601 scanned ports on (192.168.2.1) are: filtered

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1762 seconds

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sU -n -P0 192.168.2.1
All 1468 scanned ports on (192.168.2.1) are: filtered

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1883 seconds

Good. Firewall is not responding to ICMP echo-request, TCP or UDP. Similar results obtained for the Internet, and External Servers networks. Since authentication with WebAuth is required prior to any access can pass through the firewall, all the scans returned as filtered including Ack scans.

 # nmap (V. 3.00) scan initiated [DATE] as: nmap -vv -sA -n -P0 IP_address
All 1601 scanned ports on (IP_address) are: filtered

 # Nmap run completed at [DATE] -- 1 IP address (1 host up) scanned in 1698 seconds

Once authenticated with the WebAuth server, traffic was allowed out to the Internet only for HTTP, HTTPS, and FTP. Access for HTTP and HTTPS traffic to External Servers segment from the Users segment were denied.

Evaluate the results:

Evaluation of the results consists of comparing the scans results to that of the rules sets defined on the NetScreen firewall. Overall, there was no significant finding. However, there might have been an oversight on the rule allowing HTTP and HTTPS access to the Web server. It was quite strange that machines on the Users segment and subsequently the Internal Servers, and Management segment were not able connect to the Web servers on the External Servers segment. It would make sense to allow employees on the Users network HTTP and HTTPS access to the web server. In applying a secure design approach, not allowing HTTP and HTTPS access from the Internal and Management segment is quite appropriate.

Recommendations:

1) Create rules allowing HTTP and HTTPS access to the Web server from Zone: Users to Zone: External Servers:

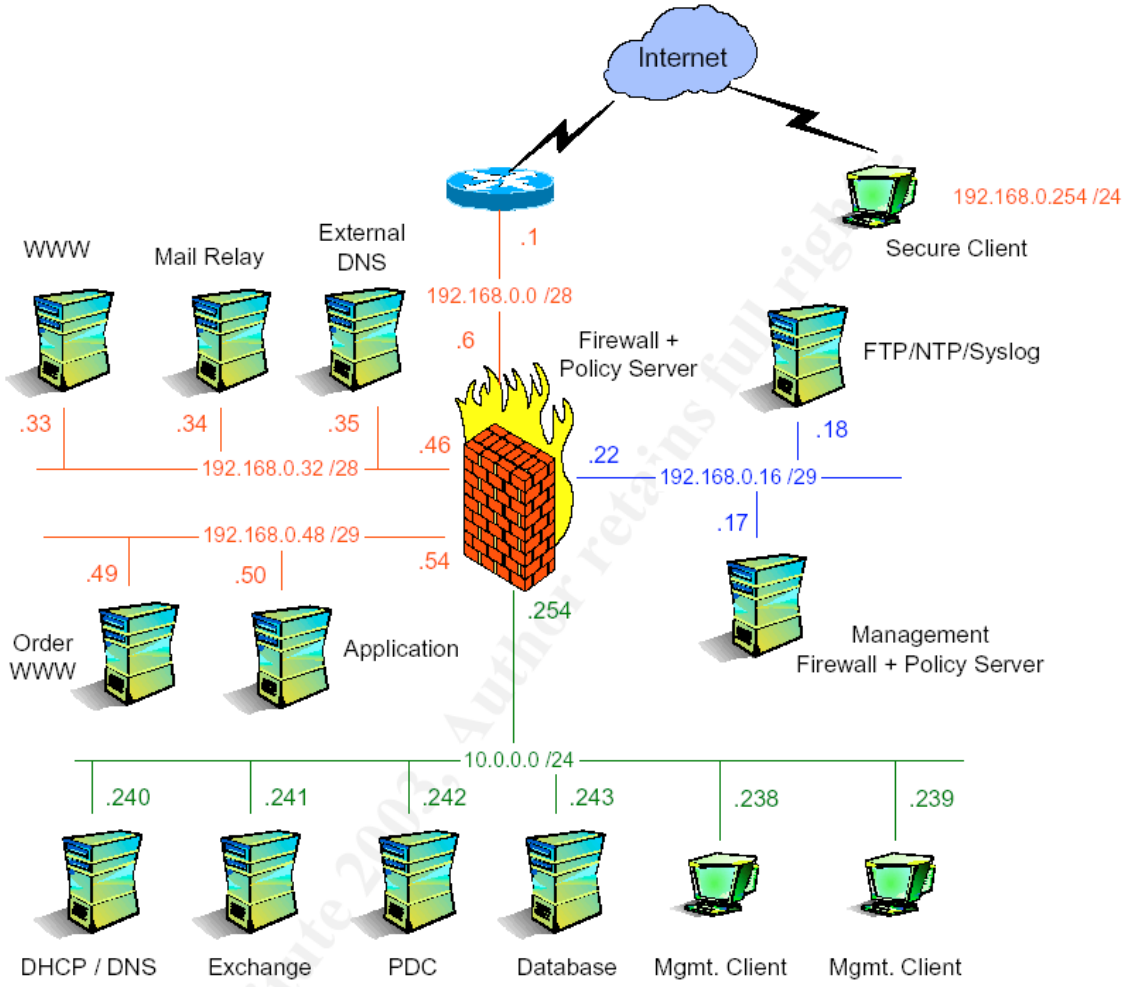
Zone: Users to External Servers

Source	Destination	Service	Action	Options		
Any	MIP(999.999.999.4)	HTTP		 	Edit	Clone
Any	MIP(999.999.999.4)	HTTPS		 	Edit	Clone

2) Consider tightening the rules around outbound NTP access for the External DNS+NTP server. Rather than allowing outbound NTP to ANY, perhaps limit the destination to 2 or 3 public NTP servers.

Assignment #4: Design under Fire

This section is devoted to the security analysis, from the perspective of a malicious hacker, of a network design from Wolfgang Gottschalk's (GCFW Analyst #405) Practical. The link to the complete document is http://www.giac.org/practical/GCFW/Wolfgang_Gottschalk_GCFW.pdf and the network diagram is included for reference below:



Attack against the Firewall

In this design, Checkpoint VPN-1/FireWall-1 NG FP3 HF1 based on RedHat 7.3 install using installation and hardening guidelines from Checkpoint and CIS, respectively. RedHat Linux is running kernel version 2.4.18-5 with all applicable patches dated Feb. 2003.

The following vulnerabilities were found using Google and Securityfocus.com:

1) bugtraq id 7161 – Malicious, remotely supplied syslog message containing escape sequences can cause unpredictable behavior if the log is viewed on the console.

2) bugtraq id 7159 – Denial of service attack through remote syslog on the Checkpoint firewall.

Checkpoing firewall software accepts syslog connections from multiple devices. This is not en-able by default.

In the first vulnerability, malicious traffic containing escape characters can cause unpredictable behavior when the log is being viewed using the command-line utility.

In the second vulnerability, an attacker can cause a high CPU utilization by sending excessive data to the syslog daemon.

Attack against the syslog daemon on the Checkpoint firewall is fairly straightforward as highlighted by the researcher who discovered the vulnerability:

To exploit the first vulnerability, send some special escape sequences through netcat to the Firewall on udp port 514:

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

Viewing the log through command line yields:
Mar 14 13:29:30 linuxbox 19: 00:01:04: Test^G^[[2J^[[2;5m^[[1;31mHACKER
ATTACK ^[[2;25m^[[22;30m^[[3q

As you can see the log is quite messy and hard to comprehend. The standard practice of using the Log Viewer or Smartview Tracker to view the log is not affected by this vulnerability. Checkpoint planned to fix this problem on the command line in future release of their software.

To exploit the second vulnerability, send large amount of random data through netcat to the Firewall on udp port 514:

[Reference 1]
[evilhost]# cat /dev/urandom | nc -u <replace with IP of firewall> 514

Use netstat and ps to check for the effect of the attack. After a short while, the syslog daemon should crash and restart is required. This issue has been addressed in NG FP3 HF2. However, syslog should ideally be placed on a dedicated host that is properly protected and not the Firewall.

From the Internet side, this attack would not succeed against Wolfgang's design. The syslog are passed to a dedicate syslog server that is well protected on the service segment. Since there is no IDS in place, this type of attack can easily be overlook without additional detection techniques.

A Distributed Denial of Service attack:

The compromise mechanism described below is purely theoretical, as no known exploit has been publicized todate. However, given the severity and the exploit potential, it will be only a matter of time before someone will exploit this hole. TFN2K variants are quite large in number and modifications to the source code to accomplish the techniques below are quite realistic.

Using spoofed source, mass emails containing an exploit for Microsoft HTML converter described at http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-023.asp were sent to a large number of Cable/DSL subscribers. As Upon execution, the exploit would covertly install a Remote Access Trojan variant of TFN2K that would connect to a website encrypted with SSL to register it IP. An attacker would connect to this website to retrieve the IP list of compromised machines. Compromised machines would remains idle waiting for further instruction on UDP port 12345.

Execution of the attack is very similar to the client/server scenarios presented by TFN2K functionality. To begin the attack, the cracker would send a udp packet using spoofed source to all the servers listening on port 12345. Within the udp packet is the instruction to execute the attack on the server and the pre-programmed password that the server will accept.

The command line instruction is almost a duplicate to that of TFN2K. The command to attack would ensure that the TFN2K server would send flood of:

UDP packets to port 53 destined for the External DNS server
TCP Syn packets to port 25 destined for the Mail Relay
TCP Syn packets to port 80 and 443 destined for the Web server.

The ratio of decoys to real is around 1000.

Countermeasures against the attack described about can be implemented through:

1) Implement SYNDefender on the Checkpoint Firewall either through Relay or Gateway mode.
2) Implement ingress filtering.
3) Implement committed access rate (CAR) feature in Cisco IOS to limit the number of SYN packets.
4) Implement HTTP reverse proxy. Similar techniques are available for DNS and SMTP but are not widely deployed and as such will cause compatibility problems if implemented.

Review of Wolfgang's design revealed that ingress filtering is implemented on the border router. However, logging of all deny traffic might become a problem during a denial of service attack. Since there are no IDS in place near real-time detection might not occurs unless additional monitoring is placed on the logs.

An attack plan to compromise an internal system:

Application server:

The target of attack is the application server as this is the gateway to the core of the business, the internal database server. Given that the existing design by Wolfgang has addressed security at the network layer quite appropriately, it is therefore best to attack the area that received less attention – that of application layer. Given that there is no IDS in place, the attack will likely not get noticed unless the web, application, and database server logs are monitored by someone who is quite knowledgeable of the application design.

Assessment tools for application layer testing are virtually non-existence in the public domain and almost all are within the commercial domain at this time. Some of the current offers are from Sanctum Inc, Kavado, SPI Dynamics, and Application Security Inc. On the public domain side, WebScarab Project through the OWASP organization is still under development.

As direct access to the server is not allowed, attempts to attack will be based on trial and errors of application manipulation. To start, the attacker will create an account as a customer using fake or stolen information. This is the only mean to attack the database in this design. The following techniques will be attempted in order to learn and determine weakness in the application.

1) Attempts debug options to gain information or access into the business logic and application information. By sending debug probes to find vulnerabilities or backdoor in the developed code.

2) Manipulation of hidden field will be attempted in order to determine if pricing or quantity can be adjusted to the attacker's advantage. Checking form for editable fields, manipulate and re-post to the "Order server"
3) Application buffer overflow to determine if the attack can yield the attacker the privilege of the application. By changing the form limit on input and sending long parameters to achieve memory corruption in the application
4) Cross-site scripting to determine if the attack can yield the attacker the account info belonging to either another customer, partner or supplier. By implanting a script into a dynamic web page to run on the victim's side and in essence, establish a virtual-hijack of the victim's session.
5) Cookie poisoning to determine if the attack can yield the attacker other valuable information belonging to either another customer, partner or supplier. By changing the contents of cookies from what was originally set.
6) Test for application 3rd party mis-configuration to gain control of the application by looking for configuration errors (such as default password) in the web and application server.
7) Sabotage the database with arbitrary commands to determine if data manipulation can be taken advantage of through SQL commands injections.
8) Test for data-encoding weakness to gain control of the application and /or server through different data encoding standards such as Unicode, UTF-8, and UTF-16.
9) Test for protocol tunneling weakness to gain control of the application and/or server through nested commands.
10) Test for known vulnerability to gain control of the application or server.
11) Forceful browsing to gain valuable information that is left unprotected on the server through guessing the file and directory names.

Network security countermeasures are limited by their lack of capability to deter or stop application layer attack. Countermeasures against these types of attacks:

1) Train developers on the techniques of secure programming and apply the practices in developing the application.
2) Embed security control and failure detection in the application to prevent and detect application layer attacks.
3) Employ intrusion detection systems that are application aware. Some of which might need to be developed in-house especially when it come to proprietary applications.
4) Deploy attack prevention device such as AppShield or InterDo to detect and prevent generic application layer attacks.

Appendix

Example Rule Base for NetFilter Firewall by Chris Brenton

```
# This is a script I load off from rc.local at boot up. I use this instead of a save script
# as it makes making changes on the fly much easier than using the canned save/restore script.
#
# Note these are examples should get you started. For some more advanced examples check out:
# http://www.stearns.org/snort2iptables
#
# Flush all old rules on restart
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# --------FORWARD RULES--------
#
# Allow all state matches through
iptables -A FORWARD -m state --state ESTABLISH,RELATED -j ACCEPT

# Log some of the weird traffic patterns. Create a second rule with -j DROP to block them.
#
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,FIN -j LOG --log-prefix " SYNFINSCAN "
iptables -A FORWARD -p tcp --tcp-flags ALL SYN,ACK -d local.network.1.0/23 -j LOG --log-prefix "
SYNACK "
iptables -A FORWARD -p tcp --tcp-flags ALL FIN -j LOG --log-prefix " FINSCAN "
iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j LOG --log-prefix " NULLSCAN "
iptables -A FORWARD -p tcp --tcp-flags ALL FIN,PSH,URG -j LOG --log-prefix " NMAPXMAS "
iptables -A FORWARD -p icmp -f -j LOG --log-prefix " ICMPFRAG "
iptables -A FORWARD -p tcp -s 0/0 --sport 31789 -d 0/0 --dport 31789 -j LOG --log-prefix "
HACKATACK "
iptables -A FORWARD -p tcp -d 0/0 --dport 12345:12346 -j LOG --log-prefix " NETBUS "
iptables -A FORWARD -p tcp -d local.network.1.0/23 --dport 1080 -j LOG --log-prefix "
PROXYSCAN "
iptables -A FORWARD -p tcp -d local.network.1.0/23 --dport 8080 -j LOG --log-prefix "
PROXYSCAN "
iptables -A FORWARD -p tcp -d 0/0 --dport 110 -j LOG --log-prefix " POP3SCAN "
iptables -A FORWARD -p udp -s 0/0 -d 0/0 --dport 500 -j LOG --log-prefix " IPSEC "


# Block all spoofed source IP packets
iptables -A FORWARD -i eth0 -p tcp -s 192.168.0.0/16 -d local.network.1.0/23  -j DROP
iptables -A FORWARD -i eth0 -p tcp -s 172.16.0.0/12 -d local.network.1.0/23 -j DROP
iptables -A FORWARD -i eth0 -p tcp -s 10.0.0.0/8 -d local.network.1.0/23 -j DROP
iptables -A FORWARD -i eth0 -p tcp -s 0.0.0.0/32 -d local.network.1.0/23 -j DROP
```

```
iptables -A FORWARD -i eth0 -p tcp -s 255.255.255.255/32 -d local.network.1.0/23 -j DROP
iptables -A FORWARD -i eth0 -p tcp -s 127.0.0.0/8 -d local.network.1.0/23 -j DROP

# Banned IP addresses
# I use a for loop for adding banned addresses as it makes the file easier to manage.
# Keeping them in numeric order would probably be helpful but I keep forgetting. ;)
        for ADDRESS in 61.177.70.237 200.151.29.0/24 217.97.92.128/25 61.163.229.154
203.253.193.236 207.21.247.4 200.128.0.0/16 200.158.0.0/16 213.23.97.14 24.207.211.131
210.97.5.252 207.228.64.4; do
iptables -A FORWARD -i eth0 -s $ADDRESS -d 0/0 -j LOG --log-level info --log-prefix " BADGUY  "
iptables -A FORWARD -i eth0 -s $ADDRESS -d 0/0 -j DROP
done

# All high volume rules in both directions
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.6/32 --
dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s local.network.1.0/23 -d local.network.1.10/32 --
dport 514 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p udp -s local.network.2.3/32 -d 0/0 --dport 53 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.4/32 --
dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.1.133/32 --dport
27001:27050 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.1.134/32 --dport
27001:27050 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.1.130/32 --dport 53 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.1.131/32 --dport 53 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.7/32 --
dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.3/32 -d 0/0 --dport 25 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s local.network.1.0/23 -d local.network.1.130/32 -
-dport 123 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s local.network.1.0/23 -d local.network.1.131/32 -
-dport 123 -j ACCEPT

# Ignore inbound NetBIOS probes due to high noise
iptables -A FORWARD -p udp -s 0/0 -d local.network.1.0/23 --dport 135:139 -j DROP
iptables -A FORWARD -p tcp -s 0/0 -d local.network.1.0/23 --dport 135:139 -j DROP
iptables -A FORWARD -p tcp -s 0/0 -d local.network.1.0/23 --dport 445 -j DROP

# Requestedinbound services
```

```
iptables -A FORWARD -m state --state NEW -p tcp -s 168.103.43.49/32 --sport 1024:65535 -d
local.network.2.10/32 --dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.2/32 --
dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.0/28 --dport 25 -j LOG --
log-level info --log-prefix " MAIL  "
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.0/28 --dport 443 -j LOG --
log-level info --log-prefix " SSL  "
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.6/32 --dport 443 -j DROP
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.0/28 --dport 443 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.2.3/32 --dport 53 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.3/32 --
dport 25 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.10/32
--dport 80 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 --sport 1024:65535 -d local.network.2.10/32
--dport 443 -j ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.10/32 --dport 22 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.2.10/32 --dport 25 -j
ACCEPT

# Requested outbound services
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.12/32 -d 0/0 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 129.6.13.136/32 -
-dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 153.2.228.50/32 -
-dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 153.2.224.50/32 -
-dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 153.2.228.50/32 -
-dport 443 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 153.2.224.50/32 -
-dport 443 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.3/32 -d 0/0 --dport 53 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.8/32 -d 0/0 --dport 22 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.0/28 -d 0/0 --dport 79 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.0/28 -d
local.network.1.130/32 --dport 53 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.0/28 -d
local.network.1.131/32 --dport 53 -j ACCEPT
```

```
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 209.140.49.4/32 -
--dport 50 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 0/0 --dport 43 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.2/32 -d 0/0 --dport 43 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.9/32 -d 0/0 --dport 43 -j
ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d
206.253.210.200/32 --dport 443 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.7/32 -d
206.253.210.200/32 --dport 443 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.3/32 -d 208.185.125.0/24
--dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.3/32 -d 216.33.22.0/24 --
dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.3/32 -d 65.200.202.0/24 -
--dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.9/32 -d 66.129.1.101/32 -
--dport 22 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d 66.129.1.101/32 -
--dport 50 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.4/32 -d
129.170.248.235/32 --dport 80 -j ACCEPT
iptables -A FORWARD -i eth3 -m state --state NEW -p tcp -s local.network.2.10/32 -d 0/0 --dport 25 -j
ACCEPT

# Let Service network machines do their thing
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.130/32 -d 0/0 --dport 53 -
j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p tcp -s local.network.1.130/32 -d 0/0 --dport 53 -j
ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.131/32 -d 0/0 --dport 53 -
j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p tcp -s local.network.1.131/32 -d 0/0 --dport 53 -j
ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p tcp -s local.network.1.131/32 -d 0/0 --dport 25 -j
ACCEPT

# Let external users access services
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.10/32 --dport 22 -j LOG --
log-level info --log-prefix " SSH_Admin1  "
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.10/32 --dport 22 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.132/32 --dport 80 -j
ACCEPT
```

```
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.131/32 --dport 25 -j LOG
--log-level info --log-prefix " MAIL  "
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.131/32 --dport 25 -j
ACCEPT

# Fix authentication delays
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d 0/0 --dport 113 -j LOG --log-level info --
log-prefix " IDENT  "
iptables -A FORWARD -m state --state NEW -p tcp -s local.network.1.0/23 -d 0/0 --dport 113 -j
ACCEPT
iptables -A FORWARD -p tcp -s 0/0 -d local.network.1.0/23 --dport 113 -j REJECT --reject-with tcp-
reset

# Allow NTP servers to talk to the off-site feeder servers
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.130/32 -d
216.152.230.3/32 --dport 123 -j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.131/32 -d
216.152.230.3/32 --dport 123 -j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.130/32 -d
130.207.244.240/32 --dport 123 -j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.131/32 -d
130.207.244.240/32 --dport 123 -j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.130/32 -d
128.59.35.142/32 --dport 123 -j ACCEPT
iptables -A FORWARD -i eth1 -m state --state NEW -p udp -s local.network.1.131/32 -d
128.59.35.142/32 --dport 123 -j ACCEPT

# Let internal users do anything
iptables -A FORWARD -i eth2 -m state --state NEW -s local.network.1.1/26 -d 0/0 -j ACCEPT

# Setup honeypot ports
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.134/32 --dport 21 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.134/32 --dport 23 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.134/32 --dport 25 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.134/32 --dport 80 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p tcp -s 0/0 -d local.network.1.134/32 --dport 31337 -j
ACCEPT
iptables -A FORWARD -m state --state NEW -p udp -s 0/0 -d local.network.1.134/32 --dport 31337 -j
ACCEPT


# Log all non-matches
```

```
iptables -A FORWARD -p tcp -d 0/0 --dport 21 -j LOG --log-prefix " FTPSCAN "
iptables -A FORWARD -p tcp -d 0/0 --dport 21 -j DROP
iptables -A FORWARD -p tcp -d 0/0 --dport 22 -j LOG --log-prefix " SSHSCAN "
iptables -A FORWARD -p tcp -d 0/0 --dport 22 -j DROP
iptables -A FORWARD -p tcp -d 0/0 --dport 25 -j LOG --log-prefix " BAD_MAIL "
iptables -A FORWARD -p tcp -d 0/0 --dport 25 -j DROP
iptables -A FORWARD -p tcp -d 0/0 --dport 80 -j LOG --log-prefix " WEBSCAN "
iptables -A FORWARD -p tcp -d 0/0 --dport 80 -j DROP
iptables -A FORWARD -p tcp -d 0/0 --dport 110 -j LOG --log-prefix " RPCSCAN "
iptables -A FORWARD -p tcp -d 0/0 --dport 110 -j DROP
iptables -A FORWARD -s local.network.2.0/28 -d 0/0 -j LOG --log-level info --log-prefix "
SYSTEMCRACKED  "
iptables -A FORWARD -s local.network.2.0/28 -d 0/0 -j DROP
iptables -A FORWARD -s 0/0 -j LOG --log-level info --log-prefix " DROP_FORWARD  "


# --------INPUT RULES--------
#
# Ignore internal bootp & NetBIOS traffic
iptables -A INPUT -i eth2 -s local.network.1.0/26 -d local.network.1.63/32 -j DROP
iptables -A INPUT -i eth1 -s local.network.1.128/27 -d local.network.1.159/32 -j DROP

iptables -A INPUT -m state --state ESTABLISH,RELATED -j ACCEPT

iptables -A INPUT -p udp -s 127.0.0.1/32 -d 127.0.0.1/32 --dport 123 -j ACCEPT

iptables -A INPUT -i eth3 -s local.network.2.0/28 -d local.network.2.15/32 -j DROP
iptables -A INPUT -s 0/0 -d 255.255.255.255 -j DROP

# Allow firewall to be managed
iptables -A INPUT -m state --state NEW -p tcp -s local.network.1.10 -d local.network.1.1 --dport 22 -j
ACCEPT

# Log all non-matched packets
iptables -A INPUT -s 0/0 -j LOG --log-level info --log-prefix " DROP_INPUT  "


# --------OUTPUT RULES--------
#
# Permit log entries to be submitted
iptables -A OUTPUT -p udp -s local.network.1.1/32 -d local.network.1.10 --dport 514 -j ACCEPT

# Permit State matches
iptables -A OUTPUT -m state --state ESTABLISH,RELATED -j ACCEPT

# Drop all outbound unreachables
```

```
iptables -A OUTPUT -p icmp -s local.network.1.249/32 -d 0/0 -j DROP

# Allow time sync
iptables -A OUTPUT -p udp -s 0/0 -d local.network.1.130 --dport 123 -j ACCEPT
iptables -A OUTPUT -p udp -s 0/0 -d local.network.1.131 --dport 123 -j ACCEPT

# Log all non-matched packets
iptables -A OUTPUT -s 0/0 -j LOG --log-level info --log-prefix " DROP_OUTPUT  "

# Don't think these are used
#iptables -A OUTPUT -m state --state NEW -s local.network.1.1 -d 0/0 -j ACCEPT
#iptables -A OUTPUT -m state --state NEW -s local.network.1.249 -d 0/0 -j ACCEPT
#iptables -A OUTPUT -m state --state NEW -s local.network.2.1 -d 0/0 -j ACCEPT
#iptables -A OUTPUT -m state --state NEW -s local.network.1.129 -d 0/0 -j ACCEPT
iptables -A OUTPUT -p udp -s 127.0.0.1/32 -d 127.0.0.1/32 --dport 123 -j ACCEPT



# --------POLICY RULES--------
#
# Define default policy for all chains
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Display the rules to the console when this script is run. This might make finding
# error messages a bit difficult so doing a:
# ./example-netfilter-rules > rules.txt
# is best as errors write to screen and rules write to the file
iptables -L –n
```

References

1) Dr. Bieringer, Peter. “checkpoint-fw1-ng-fp3-syslog-crash.txt”. URL: http://www.aerasec.de/security/advisories/txt/checkpoint-fw1-ng-fp3-syslog-crash.txt (July 11, 2003)

2) Briddell, Matt. “GCFW Practical Assignment.” GIAC Certified Firewall Analyst (GCFW #308). April 12, 2002. URL: http://www.giac.org/practical/Matt_Briddell_GCFW.zip (July 11, 2003)

3) Brokken, Frank B. “Stealth”. June 2003. URL: ftp://ftp.rug.nl/contrib/frank/software/linux/stealth/ (July 11, 2003)

4) Check Point Software Technologies LTD. “FireWall-1” URL: http://www.checkpoint.com/products/protect/firewall-1.html (July 11, 2003)

5) Cisco Systems, Inc. “Cisco IOS Software Releases 12.3 Mainline”. URL: http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/prod_command_reference_list.html (July 11, 2003)

6) Dittrich, David. “Distributed Denial of Service (DDoS) Attacks/tools”. June 23, 2003 URL: http://staff.washington.edu/dittrich/misc/ddos/ (July 11, 2003)

7) Fyodor “Nmap Stealth Port Scanner 3.30” June 2003 URL: http://www.insecure.org/nmap/index.html (July 11, 2003)

8) Gottschalk, Wolfgang. “GCFW Practical Assignment.” GIAC Certified Firewall Analyst (GCFW #405). April 27, 2002. URL: http://www.giac.org/practical/GCFW/Wolfgang_Gottschalk_GCFW.pdf (July 11, 2003)

9) Kavado, Inc. 2001. URL: http://www.kavado.com/ (July 11, 2003)

10) Kohlenberg, Toby. “GCFW Practical Assignment.” GIAC Certified Firewall Analyst (GCFW #342). November 30, 2002. URL: http://www.giac.org/practical/Toby_Kohlenberg_GCFW.zip (July 11, 2003)

11) Lasser, Jon. Beale, Jay “Bastille Linux” November 2002 URL: http://bastille-linux.org (July 11, 2003)

12) NetScreen Technologies, Inc. “NetScreen Firewall” URL: http://www.netscreen.com/ (July 11, 2003)

13) OWASP. ”The Open Web Application Security Project” 2001. URL: http://www.owasp.org/ (July 11, 2003)

14) Sanctum, Inc. 2003. URL: http://www.sanctuminc.com/ (July 11, 2003)

15) SecurityFocus. 1999. URL: http://www.securityfocus.com/ (July 11, 2003)

16) Snort "The Open Source Network Intrusion Detection System 2.0" April 2003 URL: http://www.snort.org (July 11, 2003)

17) SPI Dynamics, Inc. 2003. URL: http://www.spidynamic.com/ (July 11, 2003)