SANS GIAC GCFW Certification Assignment

Version: 4.0
Submitted by: Sandeep Singh Sandhu
Date: 16 October, 2004

Assignment 1: Intrusion Prevention Systems

Abstract

Organizations routinely deploy firewalls to protect their network perimeter to implement their security policy and its implied access controls. Intrusion detection systems monitor these defenses and alert security personnel if an attack is detected. These alerts need to be monitored by skilled personnel to take corrective action and prevent or contain an intrusion. On the other hand, attackers armed with automated tools to create a large scale disruption, conceal themselves and are successfully able to launch a variety of attacks at various layers of the network & hosts. Zero day attacks are a reality and to meet this challenge new technologies that stop and prevent intrusions are evolving from Intrusion Detection Systems.

'Intrusion Prevention Systems' attempt to provide an efficient, fast, accurate and automated response to any intrusion so that attack attempts are blocked or contained before they do any damage. We explore their design, characteristics and technologies to understand this complex & innovative approach for securing the network infrastructure. Their impact on network perimeter protection and on defense-in-depth is also discussed.

1.1 Introduction

Historically, Internet protocols have been developed in an open academic environment and did not have a strong focus towards security. As the Internet user base grew, and it became public, networks & hosts became vulnerable to attacks. In response, organizations began deploying firewalls to protect the perimeters of their networks from misuse and attacks. 'Choke points' have been created at the gateways to monitor and control traffic, these 'firewalls' have developed from simple packet-header examining filtering-routers to more advanced stateful connection tracking packet filters and application proxies.

Today's network infrastructure is comprised of large no of inter-networked hosts and devices which utilize multiple communication protocols. Organizations deploy elaborate access control mechanisms and multiple security layers in an attempt to control misuse of resources and prevent intrusions.

Due to newly discovered vulnerabilities in applications and unforeseen protocol weaknesses being discovered very frequently, merely configuring firewalls for stopping seemingly malicious network traffic was not enough. Attackers exploit weaknesses of protocols, services and applications to penetrate a firewall and compromise the site's security. In today's diverse and complex networks, it is not feasible to rectify and patch all systems to thwart attacks immediately by manual intervention whenever a vulnerability is discovered. Hence, Intrusion detection systems were developed to detect attacks in real time so that corrective action can be taken before an incident results in significant damage. They either use attack data signatures (fingerprints) or patterns analyzed from previous attacks to detect attacks.

1.2 Present Challenges

In order to better understand the requirements of an IPS we list out some common attacks observed at the network perimeter:

1) **Foot-printing** a network and its hosts (**network reconnaissance**), port scanning, stack fingerprinting, passive monitoring to map the victim network. This is done using various legitimate network protocols by using them in un-orthodox ways to extract information about the network devices, hosts, services and applications. E.g. TCP syn scans, tcp sequence number pattern analysis for inferring the network stack of different operating systems.
2) User account identification and **password guessing/cracking** to get a 'foothold' onto the system. This could either be done by sending dictionary words or random authentication data or hashes to obtain a valid login or by sniffing on the network and capturing authentication data for replaying it back to the host later. Some attacks also include obtaining hashed password files/databases which can then be decrypted offline at leisure to get the plaintext id/password.
3) **Session highjacking**: A client server session could be taken over by the attacker to establish a man-in-the middle attack or take-over the identity of the victim machine by exploiting vulnerability in the communication protocol to spoof network traffic for the session. The 'real' client or server is denied access for a sufficient time to enable the attacker achieve this goal. E.g. DNS redirection by reverse lookups or, TCP highjacking by TCP sequence number prediction.
4) **Denial of service**: DoS attacks are intended to prevent legitimate users, customers or clients of a site from successfully accessing it. An attacker could cause un-availability of a network's bandwidth or host's resources by creating a large amount of load on the system. DOS attacks include TCP syn flood attacks, fraggle attacks which exploit TCP/IP protocol implementation flaws in network software. Distributed denial of service (DDOS) attacks such as smurf or trinoo where numerous 'dumb' slave machines are employed to form an attack network used to amplify the flood of network attack traffic. Hosts can similarly be flooded by creating large 'core' dumps or temp files to consume the entire storage area; or, the system memory and process table may be exhausted by malicious programs to make it un-available for legitimate users.
5) **Data driven attacks**: The attacker may not subvert any network protocol but may simply inject harmful data to trigger a failure in the system component using that data. This failure is then used to gain access to the system. E.g. a buffer overflow could be caused by sending large amount of data to a program which does not do proper input size checking and validation before allocating space for the input data; this causes the program to overflow the memory space (buffer) allocated for input. By crafting executable code into the data it can be made to execute with the same privileges as the program providing any required access to the attacker. SQL injection and cross-site scripting similarly inject harmful data to a database or web server.
6) When any access is obtained on the system an attacker may change critical system files to subvert any monitoring agent from detecting the intrusion. System commands/files may be replaced with **Trojans** or backdoors which provide a convenient entry point for the attacker to regain control of the system at a later time. The kernel itself may be changed to render any monitoring agents ineffective. Complete sets of such tools called **rootkits** are available and constantly being developed by attackers. They change system configuration files and executable files such as 'ls', 'du' to conceal an intruder's presence.

Commonly deployed solutions to remedy these problems and their shortcomings are summarized below:

- **Integrity Checking & proactive vulnerability Analysis**: An integrity checker scans a file system and records each file's details and its cryptographic hash. This is stored into a database which can be archived. At a scheduled frequency another such “snapshot” of the file system is generated and compared with the earlier database to quickly verify if files have been tampered. Vulnerability analysis tools are very common today for detecting un-patched or mis-configured hosts or network components. These methods only work offline and cannot check for intrusions in real time to contain an attack in progress.

- **Firewalls**: The most common method to monitor traffic is by examining the TCP/IP packets header and allow/deny them based on the source or destination ip address, or ports. An attacker, to avoid detection, can most trivially spoof these. Proxy gateways operate at the application layer and intercept service requests from clients, translate these into new service requests to fetch service replies from the ‘real’ host. These replies are then relayed back to the client without either the server or client knowing there is a proxy between them. Attack methods have shifted to focus primarily on application layer vulnerabilities which can be exploited even with valid network traffic (SQL injection/cross-site scripting). Packet Filters cannot protect against protocol vulnerabilities or configuration errors. A big risk is from misconfiguration due to a poor understanding of the packet filter[1][23]. Attackers frequently conceal encrypted control connections within legitimate service traffic (http tunneling), which cannot be understood and proxied by packet filters or proxies. Some protocols do not lend themselves to proxying due to inherent protocol design problems.
- **Intrusion detection systems** [2]: Conventional IDS monitor network traffic and/or host system calls/logs and analyze them to detect known attack patterns. They alert security personnel if an attack is detected based on known fingerprints or patterns of attacks. Such ‘signature based’ detection systems fail to detect attacks which have not been recorded and analyzed earlier. Attackers are evading these methods by quickly developing attack tools when a new vulnerability is announced and attempt to exploit any systems that have not been patched. Malicious packets can be crafted (by fragmenting packets, sending un-ordered packets, flooding with garbage, etc. [3]) so that IDS miss out signature patterns from monitored network traffic. . E.g. the “whisker” tool exploited the weakness of incomplete network protocol analysis done by IDS, it was used to evade attack signature patterns. It used the http command “HEAD” instead of “GET” for attacking web servers, IDS signatures were setup to look for the ”GET” command, so scans were not detected. Signature based IDS can be overloaded with false attacks by triggering alarms all over the site so some intrusions go un-detected while others are investigated [4]. In such events it is very compelling for the administrators to simply shut off the network interface or “pull the plug”, which fulfills the attackers objective of denial of service. Attacks can be done very slowly (e.g. 2 scans per day) and from different sources so that tracking such attempts becomes difficult. These IDS generate a large no of false alarms; frequent false alarms have a tendency of being ignored by security personnel.

1.3 Intrusion Prevention Systems

The goal of deploying an intrusion prevention system is to prevent attacks from being successful against the site it protects. Their features & characteristics are as follows:

- IPS systems need to analyze and respond in real-time to stop an attack in progress, if the response is too slow or late, the hosts might already have been compromised and intrusions would not be prevented.
- An IPS should have very low false negatives and detect a wide range of known attacks so that almost no intrusion attempts evade detection. At the same time its responses should not cause a denial of service on its own services by blocking legitimate network traffic.

- IPS should be easily configurable and maintainable to avoid human errors since it will affect the usability and security of the entire site. It should easily integrate into the existing network for a quick deployment.
- Since training based expert system engines are the main decision makers in the system, these training periods should be minimal and allow manual fine-tuning to remove any inaccuracies due to the environment. Large organizations deploying IPS cannot afford to spend many weeks simply setting up the system.
- IPS should provide a comprehensive real-time 'view' of the entire organization's networks & hosts in a simple manner so it is easy to understand by the personnel monitoring it.
- Detect and report internal and external abnormal activity and any misuse of systems/resources.
- Alert personnel and respond to intrusion attempts by trapping the attacker or modifying the firewall to block the attack in progress.
- Today's high speed networks are able to generate huge amounts of network traffic this should not overload an IPS and prevent it from functioning.

1.3.1 Architecture

An IPS is broadly composed of the following components[2]:

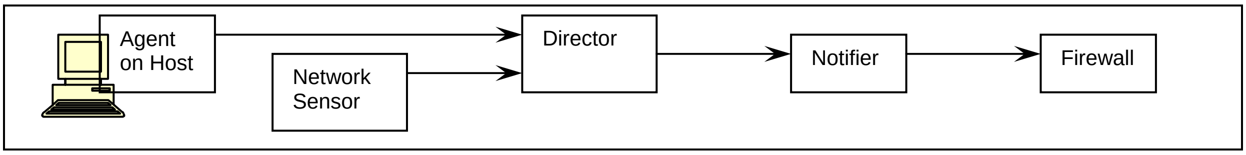
1) **Agent**: They run on each host & network being monitored and report relevant data to the director. System logs and calls are obtained from hosts. Network packet headers data or random samples of entire packet contents are collected by network 'sensors'.
2) **Director**: The director receives incoming reports from 'agents', stores them and filters them to obtain relevant knowledge of the events occurring at the hosts and the network. This is then sent to the analysis engine to determine if an intrusion has occurred. The analysis engine may use advanced expert systems to make decisions of whether an intrusion is in progress or not.
3) **Notifier**: A notifier takes any output from the director about an intrusion and either informs/alerts personnel or takes action to stop the intrusion. This action needs to be taken before the attack is complete so that firewall rule changes are effected immediately before the attack succeeds and a host is compromised. This can be done in many ways:

   a.} Generate a response action to stop and close the attack by isolating the host being attacked and "jailing" the attacker/attacking code.

   b.} Limit rate of data flow or access (throttling) to make the attack fail.

   c.} Containment by limiting access to resources by modifying the environment to stop the intrusion in progress.

   d.} Increase the audit logging level and closely monitor the specific anomalies and generate alerts for human intervention.



1.3.2 Open Source Intrusion Prevention Systems

An inline IPS is constructed from the usual IDS components and integrating these with a firewall. The firewall has a static rule set for maintaining the organization's security policy under normal circumstances. An IDS monitors the network and hosts and generates alerts if an attack or an anomaly is detected. This alert is then analyzed in real-time by the IPS to generate a new rule in response to the attack. This rule is then added to the firewall. The new rule makes the firewall now block the intrusion attempt.

Snort Inline utilizes such an approach. A Snort IDS is used to detect intrusions from network traffic[16]. Its alerts are translated by snort-inline to generate IPTables rules. The IPTables stateful packet filter is then run to block the relevant network traffic.

1.3.3 Commercial Intrusion Prevention Systems

There are now several commercial solutions available which provide integrated host and network based real-time intrusion detection and prevention solutions. Some products such as Juniper Networks Netscreen IDP[22], NFR Sentivist IPS, Top Layer's Attack Mitigator™ IPS, Proventia G200 are inline network based IPS, NAI McAfee Entercept provides a host based IDS. Most products employ sophisticated AI & pattern based/signature-based detection solutions to get accurate and complete alerts from the IDS. Most are flexible enough to be deployed as only IDS or as inline IPS. Some products incorporate vulnerability scanners to give a complete solution for maintaining & verifying the site's security configuration and compliance to the security policy. Vendors provide support for signature updates similar to anti-virus updates. Some IPS also offer the "packet scrubbing" feature where malicious or non-standard network traffic will be cleaned up before being forwarded internally to other hosts. These solutions also provide GUI interfaces for managing agents defining security policies, etc. and provide template rule bases for ease of use. Numerous aids are provided to get custom reports to analyze or export logs for analysis. Alerts can be reported using emails, console popup, mobile SMS, pagers, SNMP traps, etc. These systems integrate with authentication systems and a wide range or operating systems and services to provide a comprehensive view of network activity.

An important feature when deciding on an IPS is the network traffic handling capability at which these devices can successfully detect and stop all attacks - 100 megabit and gigabit network monitoring rates have been claimed by some vendors.

1.4 Advanced Intrusion Detection for an IPS

An Intrusion prevention system (IPS) is a logical extension of an IDS. But conventional signature based IDS do not provide assurance that an IPS will achieve its intended goals. If an IPS is expected to take automated decisions to block intrusions then better methods are required to detect intrusions otherwise legitimate services may be denied access due to false alarms.

To understand how intrusion detection has improved to provide more accurate detection we discuss some advanced intrusion detection technologies in detail.

Non-signature based IDS are designed to perform intelligent intrusion detection and use advanced artificial intelligence based expert systems for improving their accuracy (lower false positives) and completeness (lower false negatives). They try to detect intrusions more intelligently by constructing information 'models' and then applying AI to solve these problems to get optimal decisions. Such IDS may use one or more of a combination of the following information 'models' to detect intrusions[2]:

A. **Anomaly modeling**: This IDS monitors normal usage patterns of hosts and networks and triggers alerts when certain limits or thresholds defining 'normal' behavior are breached. In order to detect deviations, "normal behavior" needs to be defined; this can be described as: [2]

   - Actions of users and processes conform to statistically predictable pattern
   - Actions of users and processes do not include sequences of commands to subvert the security policy of the system.

- The actions conform to a set of specifications describing actions that the processes are allowed to do or not allowed to do

There are 3 types of statistical modeling employed to implement anomaly detection:

e.} **Threshold metrics** – Events are monitored to determine if they cross a threshold min or max, e.g. max limit of TCP SYN requests per second.

f.} **Statistical moments** – The mean/deviation/or other statistic of observed events are monitored, these are compared to 'moments' obtained earlier from "learning" data of a normal environment. Recent-most events are given more weights (importance) while determining these means or standard deviations calculations, so that the system adapts to changing environments. e.g. average number of logins in 1 hour.

g.} **Markov model** - Data on sequences of events which change the state of the whole system are collected for normal traffic. These event sequences are then grouped and analyzed; they are given probabilities of occurrence under normal conditions. During system monitoring, if the occurrence of particular sequences of events observed are different from the 'normal state' the system detects an intrusion. Its effectiveness depends on adequacy of training data from which the system learns which events-transitions sequences are 'normal'. E.g. the 'ls' unix command is supposed to show file system listings, if system call traces of the 'ls' program show file 'writes', this would be previously unknown activity for 'ls', which triggers an alert due to this anomaly.

B. **Misuse detection**: This type of IDS monitors the environment and detects if events are occurring in known patterns/sequences which are recognized to violate the security of a system. A rule set is created and provided to the system which contains knowledge of known vulnerabilities. These systems often use expert systems and artificial intelligence techniques to improve their detection capabilities. E.g. a large amount of network traffic is received for a ftp daemon and soon the CPU usage by the ftp service jumps to 99%, it could indicate a compromise of the ftp service and would trigger an alert.

C. **Specification based detection**: This type of IDS attempts to determine whether a sequence of events/instructions violates a specification of how a system should execute, if the specification is violated an intrusion is triggered. It requires that specification of key programs be written. E.g. if the Unix mailer service 'Sendmail' open a network connection to an unknown non-standard port and starts accepting traffic on that port no, it indicates behavior which is not included in the specification of the activities Sendmail does and triggers an alert.

The information models described above interpret system logs and events in different ways to detect intrusions. But these need to be analyzed intelligently to get accurate alerts. Many different AI algorithms and expert system techniques have been used to make intelligent decisions for detecting intrusions, some of these are:

- **Neural Networks**: They simulate the workings of biological neurons networked together. Inputs or stimulus received by the digital neural network is processed by these neurons and an output is obtained. This network is 'trained' on data to adapt their behavior towards detecting anomalies accurately. Their successful use in pattern recognition has led to widespread use in similar areas.
- **Fuzzy sets** [5]: By using fuzzy set theory and classifying event-transition clusters, they attempt to detect anomalies. To begin with, the system mines "fuzzy association sets" from the data. It consists of:
  - Events occurring during a time period, in a particular order, may have other events between them.
  - Samples of normal traffic is taken an associations learnt.

- While monitoring real traffic, such events and its associations are obtained and compared to the earlier 'learnt' data.

  Based on this comparison, decisions are made whether the system is under attack or not. If the event associations are not similar to 'normal system state' event associations learnt earlier, the system is alerted as being under attack.

  These similarity comparisons are much better than conventional “exact match” comparisons since they give much lesser false alarms.

  To deploy fuzzy based IDS, certain dependable and relevant features must be selected from the system. These features are monitored and events are obtained when properties of these features change. This can be further improved by using a technique called "genetic algorithms".

- **Genetic algorithms** [6]: GA optimization on information models helps create better working event association sets for normal or abnormal system states so that the results of comparisons are much more accurate. Solutions to the data interpretation problem are developed by evolving solution sets in a manner similar to natural evolution and then evaluating these. After a few generations and mutations an optimal solution can be expected to be reached.
- **Graph based pattern matching** [7]: The GRIDS project describes setting up agents on many network nodes to form one large distributed monitoring system. Events occurring on nodes are logged by the agents and are represented as graphs, pattern matching done on these graphs have successfully indicated intrusions or anomalies such as virus infections, etc.

Integrating host based monitoring with network based monitoring is critical for these techniques to succeed in detecting previously unknown attacks. They examine API calls, memory management (i.e. buffer overflow attempts), how the application interacts with the operating system, and how the user is suppose to interact with the application.

These methods require providing training data to the IDS for ‘learning’ to distinguish between ‘normal expected behavior and anomalous behavior of the network or system. Detailed discussion of these techniques is beyond the scope of this paper and would require detailed study of complex AI & operations research concepts.

1.5 Defense in Depth using an IPS

An IPS solution mitigates several problems identified earlier in the following manner:

- Host intrusion and tampering: By employing host based agents to monitor system events, any un-authorized changes can be detected and alerted in real-time by advanced AI algorithms used by IPS.
- Application vulnerabilities: An IPS using expert systems will be able to detect abnormal network and system usage if an attacker attempts to exploit an unknown bug in a service. Such attacks will not only be thwarted but also provide alerts to help identify and fix those new vulnerabilities.
- Network scans and fingerprinting are accurately detected as anomalous activity.
- Signature evasion: An IPS using a hybrid of signature detection and anomaly or misuse detection will be able to protect against a wide range of attacks which are both known and unknown. To evade signature detection an attacker would require use of un-conventional techniques which would show up as abnormal behavior in the detection system and hence reveal the attack.

An IPS provides layered security by utilizing multiple techniques to protect the site - packet filtering at the network protocol level, proxying for application protocols, host based detection for application vulnerabilities and expert systems for correlating all events centrally to perform better analysis of the site.

Use of IPS systems will impact day-to-day security operations considerably since it would no longer require highly skilled personnel to analyze and take corrective action on IDS alerts. The advanced detection methods and automated response would let the network 'take care of itself' in most cases.

This will also have a significant impact on the design of firewalls. An IPS creates a firewall with a 'feedback' control system which makes it an adaptive 'dynamic' setup. Future firewalls will have an IPS as a necessary aspect of the whole setup similar to an anti-virus setup which has become an accepted norm for all MS Windows hosts. While implementing an intrusion prevention system each organization needs to evaluate its requirements and deploy an IPS accordingly.

1.6 Conclusion

These Intrusion prevention systems deployed to protect the network will have a significant impact on perimeter security by providing credible automated real-time defenses to the network infrastructure. Expert systems used to detect intrusions will provide greater resistance to a determined attacker and provide more assurance to organizations deploying such solutions that the risk to their infrastructure is lowered.

Assignment 2: Security Architecture & Design of the Firewall

Abstract

GIAC enterprises is a small business employing 50 people who sell Fortune cookie sayings to its customers (individuals & large companies) over the Internet from their e-commerce portal. Most employees work at the head office. Some sales employees are also located at the 4 regional satellite locations; they need remote access to use the portal, file server & email communications.

A screened subnet architecture firewall has been designed to implement perimeter access controls according to the organization's security policy while enabling GIAC employees to utilize services from the Internet as well as host an e-commerce solution online. It implements layered security and industry best practices while providing flexibility of use.

2.1 Business Requirements

GIAC enterprises hosts and provides information to various types of users, which have different access rights to this information. Before defining the access controls, the access requirements of each group of users needs to be elaborated:

1) Customers: They are companies or individuals who login online to the e-com infrastructure and buy fortune cookies to be used by their businesses. They would be able to access the portal and buy fortune cookies by placing orders online. They are able to identify themselves using static id-passwords and track their orders online.
2) General public: The public website hosts company information for the general public to advertise its products, services and any other general information about the business. This excludes any internal company information or customers data.
3) Suppliers: These are companies/individuals who sell ideas for making the fortune cookies. They identify themselves online and utilize the e-commerce portal for selling their services to GIAC enterprises. They do not have access to any individual customer data but are given inventory related information by the company's customer order fulfillment unit. They also communicate with the finance unit for settlement of bills for their services.
4) Partners: These are companies located in different geographic locations with whom GIAC enterprises has partnered to translate fortune cookies & resell them. They access the e-com portal for placing and tracking orders to get cookie feeds.

***GIAC employees** - they can further be segregated into the following groups:*

5) System Administrators (3 users): They would have privileged system access for administering the hosts/servers and the network equipment.
6) Sales (25 users): These users could either be mobile and work remotely from the field or work from the 4 regional offices or they work at the head office. They access the portal from the Internet for placing sales orders & use email.
7) Finance (4 users), The finance unit is responsible for accounting, monitoring and controlling the finances of the company. They also generate various MIS and reports for use by the rest of the organization.
8) Helpdesk & Order fulfillment (9 users): They access all customer orders and track inventory; they coordinate orders to be fulfilled by suppliers and requests by partners. They also update the public website for specific product related information. They would respond to customer queries on status of their orders.
9) Developers, Website Designers & Researchers (6 users): They develop code for the systems and conduct research for creating better fortune cookies for ever changing customer requirements. They would not have access to the customer data or any other 'real' data but would have read access to various aggregated information such as portal performance, order related MIS or trends, or any other reports about the portal to improve the services offered by GIAC enterprises.
10) Senior Management (4 users): They would have read access to all data and would be able to communicate to all units to control and direct their activities.

2.2 Firewall Design

The firewall architecture for GIAC enterprises is a screened subnet architecture[10] with three layers of protection: an external facing packet filtering 'screening' router, a DMZ (de-militarized zone) with bastion hosts and application proxy servers, and finally an interior packet filtering 'screening' router. All externally visible services (e.g. web server and the DNS server) are hosted by bastion hosts on the DMZ. Internet services used internally are filtered by application proxies i.e. web proxy server and the email relay. There is no single vulnerable point that will compromise the internal network. This architecture provides multiple layers of security while keeping the design simple and flexible for future extensions.

2.2.1 Description

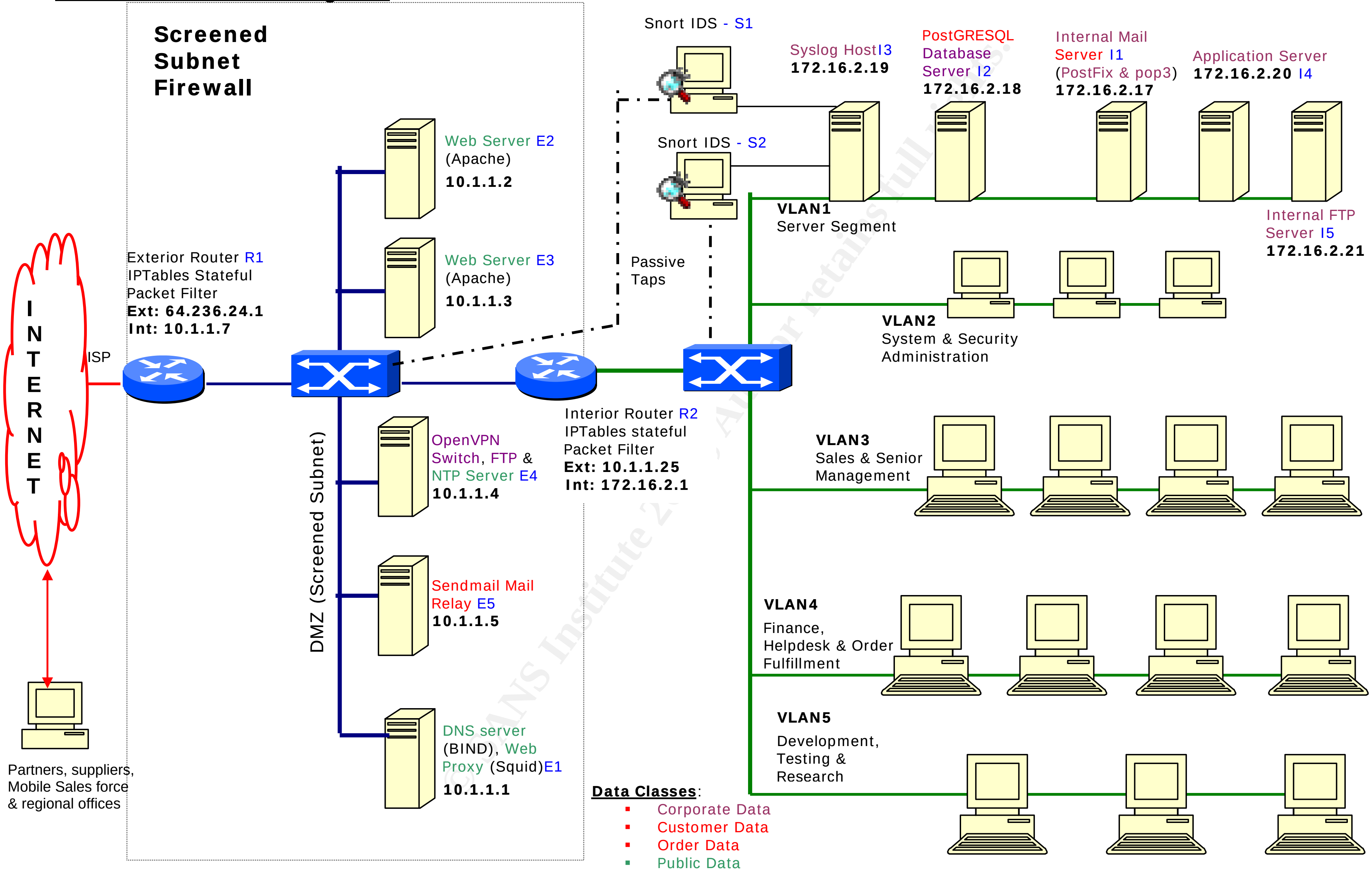
As shown in the network diagram, the firewall consists of an exterior stateful packet filter which connects to the ISP directly and is the only connection to the public internet. Network traffic is then intercepted by the application level proxies and hosts at the screened subnet (DMZ) comprising of bastion hosts for web server, DNS server, Proxy server, VPN switch and email relay. Traffic destined to the internal network is then once again filtered by the interior stateful packet filter router which is the only connection to the internal network. Hosts on the DMZ (de-militarized zone) can access the Internet via exterior router. No network traffic is permitted directly between the interior packet filter and the exterior packet filter. The organization's data and applications are hosted on an internal VLAN for servers. Only web servers are allowed access to the application servers. The database cannot be accessed by any host at the DMZ, and must be accessed by the application server.

The internal network hosts all employees' desktops and the critical mail server, application server, database server, and the file server. It also segregates network segments by creating separate Virtual LANs for Servers, administrators, sales, senior management, finance, order-fulfillment and customer helpdesk units. The workstations are a mix of Windows 2000 and Redhat Fedora 2 Linux. The organization connects to the Internet via a T1 line.

2.2.2 The Guiding Principles

The main principles [2] guiding this firewall's design are listed below:

Firewall Network Diagram


Screened Subnet Firewall
I N T E R N E T
ISP
Exterior Router R1
IPTables Stateful Packet Filter
Ext: 64.236.24.1
Int: 10.1.1.7
Partners, suppliers, Mobile Sales force & regional offices
DMZ (Screened Subnet)
Web Server E2 (Apache) 10.1.1.2
Web Server E3 (Apache) 10.1.1.3
OpenVPN Switch, FTP & NTP Server E4 10.1.1.4
Sendmail Mail Relay E5 10.1.1.5
DNS server (BIND), Web Proxy (Squid)E1 10.1.1.1
Snort IDS - S1
Snort IDS - S2
Passive Taps
Interior Router R2
IPTables stateful Packet Filter
Ext: 10.1.1.25
Int: 172.16.2.1
Syslog HostI3 172.16.2.19
PostGRESQL Database Server I2 172.16.2.18
Internal Mail Server I1 (PostFix & pop3) 172.16.2.17
Application Server 172.16.2.20 I4
Internal FTP Server I5 172.16.2.21
VLAN1 Server Segment
VLAN2 System & Security Administration
VLAN3 Sales & Senior Management
VLAN4 Finance, Helpdesk & Order Fulfillment
VLAN5 Development, Testing & Research
Data Classes:
▪ Corporate Data
▪ Customer Data
▪ Order Data
▪ Public Data

1. Principle of least privilege: A subject should be given only those privileges that it needs in order to complete its task. This is enforced by applying strict packet level checks to allow only the specific packets which are necessary, no extra access permissions have been given nor are they kept ambiguous in the configuration.
2. Fail-safe defaults: Unless a subject is given explicit rights to an object, it should be denied access to that object. All components of the firewall are configured with a default deny stance if an unspecified access is attempted.
3. Principle of economy of mechanisms: Security mechanisms should be as simple as possible. The architecture keeps the security mechanisms simple – each router has only two interfaces, traffic can flow in only 2 directions and it is simple and intuitive to configure them to let traffic pass through to the other side or not. This is unlike firewalls which are a single device of three interfaces where the network traffic can be routed in 6 different directions, such a mechanism is more difficult to configure and may lead to errors in configuration.
4. Principle of complete mediation: All accesses to objects must be checked to make sure they are allowed. All network traffic is checked and controlled by at least one component – exterior packet filter, proxies, or interior filter which ensures complete mediation.
5. Principle of open design: Security of a mechanism should not depend on the secrecy of its design or implementation. None of the firewall components rely on hiding their presence to ensure their effectiveness. They are equally effective in blocking attacks and performing their intended functions even if an attacker knew the architecture.

These best practices have been used to design the firewall and to create the firewall rule base for implementing the GIAC's security policy & access controls.

2.2.3 Firewall Components

Software	Version	Cost	Service	Security Strengths	Security Weaknesses
IPTables	1.2.11	$ 0	Stateful Packet Filtering External Router, Interior Router	▪ Choke point forces traffic through it where it can be controlled & monitored ▪ Connection tracking for stateful inspection ▪ NAT for concealing network architecture ▪ Detailed protocol level logging to alert attacks	▪ Cannot examine application-level access
Snort	2.2.0	$ 0	Network IDS at DMZ and internal network	▪ Network traffic anomalies detected ▪ Updated signatures for all known attacks	▪ Host anomalies are not monitored
SQUID (port 8080, 8443)	2.5 stable 6	$ 0	HTTP(S) proxy	▪ Application level filtering ▪ Permits Access controls to be implemented for web usage.	▪ Encrypted tunneled traffic escapes monitoring
BIND (port 53 TCP & UDP)	9.2.3	$ 0	Primary DNS server	▪ Hardened name-server to protect against DNS exploits ▪ Does not serve the internal hosts to avoid data-leakage from DNS	▪ No application level filtering to protect against unknown application-level attacks
Sendmail (port 25)	8.13.1	$ 0	DMZ Mail Relay	▪ SMTP application-level filtering ▪ Virus/Trojan removal before it reaches the email server ▪ Alerts application-level attacks	▪ Cannot protect against encrypted Trojan/virus
Postfix (port 25)	2.1.4	$ 0	Main email server	▪ Separate software from mail relay to avoid similar vulnerability affecting both servers.	▪ Cannot protect against encrypted Trojan/virus
OpenVPN (port 5000)	1.6.0	$ 0	SSL based VPN	▪ Uses proven and secure TLS 1 protocol for secure access. ▪ VPN access is further restricted to only few bastion hosts and	▪ Could open up the DMZ to attack by using OpenVPN tunnel to bypass the exterior

				the email server.	packet filter.
Syslog (port 514)	1.4.1	$ 0	Linux System logging	▪ Placed on internal network to protect logs	▪ Failure to function would remove all traces of an attacker's activity
SpamAssassin	2.64	$ 0	Spam checking mail-filter	▪ Protects email servers from spam which can be used for spreading viruses	▪ Cannot protect against encrypted Trojan/virus
Clam	0.75	$ 0	Anti-virus checking on emails	▪ Protects from viruses being delivered via emails	▪ Cannot protect against encrypted Trojan/virus or user's carelessness

2.2.4 Filtering Routers

The architecture ensures that all network traffic between the public Internet, the DMZ, and the internal network flows through two packet filtering routers. *IPTables* stateful packet filters control network access and permit, drop, or change IP packets to implement the firewall rule base. The routers act as 'choke-points' to ensure that illegitimate network traffic is kept out of the network and any permitted traffic is forced through the application level proxies.

The firewall packet filters are setup on Dell PowerEdge 750 servers with Linux 2.6.7 kernel that includes a very powerful firewalling & NAT architecture called Netfilter. Its main component is *IPTables*, a generic structure which enable firewall rules for NAT*ing* and stateful connection-tracking packet filtering.

These features of IPTables allow network access to be finely controlled. The external router performs DNAT (destination network address translation) for all externally visible services so that internal network details are disguised and it appears as if one device is hosting all these services (http, SMTP, dns, vpn, https).

The IPTables packet filter configuration commands are given below. The firewall rule-base is implemented using these commands to deny or permit packets [8] [9] [11] [12]

```
#Rule F3: Block source routing
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
#Don't respond to broadcast pings
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
#Kill timestamps
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
#Enable SYN Cookies
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
#Kill redirects
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
#Enable bad error message protection
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
#Log martians (loopback address packets)
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
#Reduce timeouts for DoS protection
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 2400 > /proc/sys/net/ipv4/tcp_keepalive_time
echo 0 > /proc/sys/net/ipv4/tcp_window_scaling
echo 0 > /proc/sys/net/ipv4/tcp_sack
```

A custom chain 'antidos' uses the limit-checking feature of iptables for checking tcp/icmp rate limits for detecting and preventing network denial of service attacks.

```
$IPT -N ANTIDOS
echo __Custom ANTIDOS Chain: syn flood stopper
#$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 5/s -p tcp --syn -j ACCEPT
$IPT -A ANTIDOS -m limit --limit 6/s -i $INET_FACE -p tcp --syn -j LOG --log-prefix "SYN
FLOOD: "
$IPT -A ANTIDOS -m limit --limit 6/s -i $INET_FACE -p tcp --syn -j DROP
echo __port scanner
#$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 4/s -p tcp --tcp-flags SYN,ACK,FIN,RST RST -j
ACCEPT
$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 5/s -p tcp --tcp-flags SYN,ACK,FIN,RST RST -j
LOG --log-prefix "port scanner: "
$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 5/s -p tcp --tcp-flags SYN,ACK,FIN,RST RST -j
DROP
echo __ping flood stopper
#$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 4/s -p icmp --icmp-type echo-request -j ACCEPT
$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 5/s -p icmp --icmp-type echo-request -j LOG --
log-prefix "PING FLOOD: "
$IPT -A ANTIDOS -i $INET_FACE -m limit --limit 5/s -p icmp --icmp-type echo-request -j DROP
echo __RULE F6: ALLOW ESTABLISHED CONNECTIONS ----
$IPT -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -t filter -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

The connection tracking stateful feature of iptables checks if a packet is already part of an existing session and allows it if so. It protects against TCP session highjacking.

```
$IPT -t filter -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#permit packets from localhost to localhost for normal system functions
$IPT -t filter -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
echo __Add custom filters for denial of service attempts
$IPT -t filter -A INPUT -p tcp -i $INET_FACE -j ANTIDOS
$IPT -t filter -A INPUT -p icmp -i $INET_FACE -j ANTIDOS
$IPT -t filter -A FORWARD -p tcp -i $INET_FACE -j ANTIDOS
$IPT -t filter -A FORWARD -p icmp -i $INET_FACE -j ANTIDOS
```

Malformed packets used to overload network devices are stopped here.

```
echo __RULE F3: FILTER FOR MALFORMED TCP PACKETS  ----
# Block TCP Packets With All Flags set (Christmas Tree) and other weird combinations
$IPT -t filter -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT -t filter -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
$IPT -t filter -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
$IPT -t filter -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPT -t filter -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
$IPT -t filter -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
echo __Reject packets with syn-ack in initial request
$IPT -t filter -A INPUT -p tcp --tcp-flags SYN,ACK SYN,ACK -m state --state NEW -j DROP
$IPT -t filter -A FORWARD -p tcp --tcp-flags SYN,ACK SYN,ACK -m state --state NEW -j DROP
```

IPTables checks new tcp packets for SYN and no ACK since connection is being tracked any discrepancy will apply the filter match and drop it. FTP data connections can similarly be tracked using the special ftp tracking module.

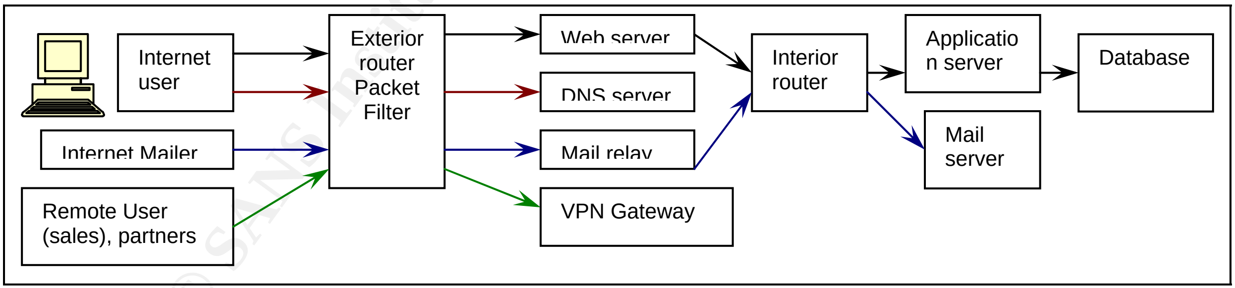
```
echo __RULE: F4, log and drop packets with no syn in first packet
$IPT -t filter -A INPUT -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "INP req
without SYN: "
$IPT -t filter -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -t filter -A FORWARD -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "FW req
without SYN: "
$IPT -t filter -A FORWARD -p tcp ! --syn -m state --state NEW -j DROP
echo __RULE F6: Allow selected ICMP types and drop the rest.
# Drop incoming pings (can be disabled)
$IPT -t filter -A INPUT -p icmp --icmp-type echo-request -j DROP
$IPT -t filter -A INPUT -p icmp --icmp-type 0 -j ACCEPT
$IPT -t filter -A INPUT -p icmp --icmp-type 3 -j ACCEPT
$IPT -t filter -A INPUT -p icmp --icmp-type 11 -j ACCEPT
$IPT -t filter -A FORWARD -p icmp --icmp-type echo-request -j DROP
$IPT -t filter -A FORWARD -p icmp --icmp-type 0 -j ACCEPT
$IPT -t filter -A FORWARD -p icmp --icmp-type 3 -j ACCEPT
$IPT -t filter -A FORWARD -p icmp --icmp-type 11 -j ACCEPT
echo __All broadcasts are stopped here
$IPT -t filter -A INPUT -m pkttype --pkt-type broadcast -j DROP
$IPT -t filter -A FORWARD -m pkttype --pkt-type broadcast -j DROP
```

For network traffic directed towards web, dns, vpn and mail services/ports on the router IP address, re-direct them to be forwarded to the correct DMZ bastion hosts by performing Destination Network address translation (DNAT). The packet then gets filtered by respective chains. Here DNAT is also used to do simple round-robin load-balancing for the web server.

```
## 3) TRAFFIC WITH DESTINATION AS INET_IP
## TABLE: NAT, CHAIN: PREROUTING
echo '__DNAT packets to web, dns, openvpn & mail services'
$IPT -t nat -A PREROUTING -p tcp -d $INET_IP --dport 80 -i $INET_IFACE -j DNAT --to-
destination 10.1.1.2-10.1.1.3
$IPT -t nat -A PREROUTING -p udp -d $INET_IP --dport 53 -i $INET_IFACE -j DNAT --to-
destination 10.1.1.1
$IPT -t nat -A PREROUTING -p tcp -d $INET_IP --dport 53 -i $INET_IFACE -j DNAT --to-
destination 10.1.1.1
$IPT -t nat -A PREROUTING -p tcp -d $INET_IP --dport 25 -i $INET_IFACE -j DNAT --to-
destination 10.1.1.5
$IPT -t nat -A PREROUTING -p udp -d $INET_IP --dport 5000 -i $INET_IFACE -j DNAT --to-
destination 10.1.1.4
## TABLE: FILTER, CHAIN: INPUT
echo __RULE F1: FILTER FOR BAD PACKET ADDRESSES (RFC 1918)----
$IPT -t filter -A INPUT -i $INET_IFACE -s 0.0.0.0/8 -j DROP
$IPT -t filter -A INPUT -i $INET_IFACE -s 192.168.0.0/16 -j DROP
$IPT -t filter -A INPUT -i $INET_IFACE -s 172.16.0.0/12 -j DROP
$IPT -t filter -A INPUT -i $INET_IFACE -s 10.0.0.0/8 -j DROP
echo '__Rule F1: Prevent external packets from using loopback address (martian packets)'
```

```
$IPT -t filter -A INPUT -i $INET_IFACE -s $LO_IP -j DROP
$IPT -t filter -A INPUT -i $INET_IFACE -d $LO_IP -j DROP
echo __RULE F15: Allow administrators to login via ssh from dmz interface
$IPT -t filter -A INPUT -p tcp -i $DMZ_IFACE --dport 22 -s $LAN_IP -j ACCEPT
echo '__RULE F2, F5, F16: is redundant, but kept as fail-safe in case other rules fail'
echo __RULE F29: DROP EVERYTHING ELSE
$IPT -t filter -A INPUT -i $INET_IFACE -d $INET_IP -j DROP
$IPT -t filter -A INPUT -i $DMZ_IFACE -d $DMZ_IP -j DROP
## 4) TRAFFIC WITH SOURCE AS INET_IP
#packet with source as the router ip follows the filter-output, mangle postrout, nat-postrout.
## TABLE: FILTER, CHAIN: OUTPUT
echo __RULE F26: let ntp queries go to the dmz ntp server
$IPT -t filter -A OUTPUT -p tcp -o $DMZ_IFACE -s 10.1.1.7 -d 10.1.1.4 --destination-port 123 -
j ACCEPT
$IPT -t filter -A OUTPUT -p udp -o $DMZ_IFACE -s 10.1.1.7 -d 10.1.1.4 --destination-port 123 -
j ACCEPT
echo __DNS requests to the dns server
$IPT -t filter -A OUTPUT -p tcp -o $DMZ_IFACE -s 10.1.1.7 -d 10.1.1.1 --destination-port 53 -j
ACCEPT
$IPT -t filter -A OUTPUT -p udp -o $DMZ_IFACE -s 10.1.1.7 -d 10.1.1.1 --destination-port 53 -j
ACCEPT
echo __RULE 19: ALLOW SYSLOG LOGGING FROM EXT ROUTER
$IPT -t filter -A OUTPUT -p udp -o $DMZ_IFACE -s 10.1.1.7 -d 172.16.2.19 --destination-port
514 -j ACCEPT
echo __RULE F5: stop netbios microsoft broadcasts
$IPT -t filter -A OUTPUT -p tcp --sport 137:139 -o $INET_IFACE -j DROP
$IPT -t filter -A OUTPUT -p udp --sport 137:139 -o $INET_IFACE -j DROP
echo __RULE 32: drop everything else going from internet interface
$IPT -t filter -A OUTUT -o $INET_IFACE -s $INET_IP -j DROP
echo '__# TABLE: MANGLE, CHAIN: POSTROUTING'
# NONE
echo '__## TABLE: NAT, CHAIN: POSTROUTING'
# NONE
echo '__5) TRAFFIC FORWARDED BY ROUTER'
# packet with source and dest different from its ips
## TABLE: FILTER, CHAIN: FORWARD
echo '__Rule F1: Prevent external packets from using
__ loopback address (martian packets)'
$IPT -t filter -A FORWARD -i $INET_IFACE -s $LO_IP -j DROP
$IPT -t filter -A FORWARD -i $INET_IFACE -d $LO_IP -j DROP
echo '__RULE F1: Deny RFC 1918 private ip address packets,
__ loopback address, zero address'
$IPT -t filter -A FORWARD -i $INET_IFACE -s 192.168.0.0/16 -j DROP
$IPT -t filter -A FORWARD -i $INET_IFACE -s 172.16.0.0/12 -j DROP
$IPT -t filter -A FORWARD -i $INET_IFACE -s 10.0.0.0/8 -j DROP
$IPT -t filter -A FORWARD -i $INET_IFACE -s 0.0.0.0/8 -j DROP
echo __Check source address validity on packets going out to internet
$IPT -t filter -A FORWARD -s ! $DMZ_IP -i $DMZ_IFACE -j DROP
```

Valid protocols such as http(s), dns, vpn and smtp are permitted from external hosts.



```
echo __RULE F7:
$IPT -t filter -A FORWARD -p tcp -o $DMZ_IFACE -d 10.1.1.2 --destination-port 80 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -o $DMZ_IFACE -d 10.1.1.2 --destination-port 443 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -o $DMZ_IFACE -d 10.1.1.3 --destination-port 80 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -o $DMZ_IFACE -d 10.1.1.3 --destination-port 443 -j ACCEPT
echo __RULE F8:
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.1 --destination-port 80 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.1 --destination-port 443 -j ACCEPT
echo __RULE F9:
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.1 --destination-port 21 -j ACCEPT
echo __RULE F10:
$IPT -t filter -A FORWARD -i $INET_IFACE -d 10.1.1.1 --destination-port 53 -j ACCEPT
echo __RULE F11:
$IPT -t filter -A FORWARD -o $INET_IFACE -s 10.1.1.1 --destination-port 53 -j ACCEPT
echo __RULE F12:
$IPT -t filter -A FORWARD -p tcp -i $INET_IFACE -d 10.1.1.5 --destination-port 25 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -i $INET_IFACE -d 10.1.1.5 --destination-port 113 -j REJECT
echo __RULE F13:
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.5 --destination-port 25 -j ACCEPT
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.5 --destination-port 113 -j ACCEPT
```

```
echo __RULE F14:
$IPT -t filter -A FORWARD -p udp -i $INET_IFACE -d 10.1.1.4 --destination-port 5000 -j ACCEPT
echo __RULE F16:
$IPT -t filter -A FORWARD -p tcp -i $INET_IFACE -d 10.1.1.4 --destination-port 21 -j DROP
echo __RULE F17:
$IPT -t filter -A FORWARD -p tcp -i $INET_IFACE -d 10.1.1.4 --destination-port 123 -j DROP
$IPT -t filter -A FORWARD -p udp -i $INET_IFACE -d 10.1.1.4 --destination-port 123 -j DROP
echo '__RULE F18: Specify the stratum 2 ntp servers[18] which are permitted access'
$IPT -t filter -A FORWARD -p tcp -o $INET_IFACE -s 10.1.1.4 -d 139.80.64.114 --destination-
port 123 -j ACCEPT
#use similar cmd to accept other allowed level 2 ntp servers, commands omitted for brevity --
$IPT -t filter -A FORWARD -p udp -o $INET_IFACE -s 10.1.1.4 -d 139.80.64.114 --destination-
port 123 -j ACCEPT
#use similar cmd to accept other allowed level 2 ntp servers, commands omitted for brevity --
echo __FILTER FOR DISALLOWED PROTOCOLS ----
# RULE F2: Block not permitted protocols ####
$IPT -t filter -A FORWARD -p tcp -m multiport --destination-port 22,23,21,20,118,110,69,70,79
-i $INET_IFACE -j DROP
$IPT -t filter -A FORWARD -p tcp -m multiport --destination-port 7,546,547,194,389,143,161,
569,137,138,139 -i $INET_IFACE -j DROP
echo __block disallowed udp protocols also
$IPT -t filter -A FORWARD -p udp -m multiport --destination-port
513,514,512,111,123,520,540,137,138,139 -i $INET_IFACE -j DROP
echo '__RULE F4: Block outgoing NetBios; stop windows machines from broadcasting to internet.’
$IPT -t filter -A FORWARD --sport 137:139 -o $INET_IFACE -j DROP
## TABLE: MANGLE, CHAIN: POSTROUTING
```

Packets for all requests originating from the DMZ are address translated (Source NAT) to make it appear as if the source of the traffic is the exterior router. This limits GIAC enterprises requirement of having public IP addresses for all DMZ hosts; and keeps internal network addresses hidden from public view to slow down attackers.

```
echo __TABLE: NAT, CHAIN: POSTROUTING
# SNAT local subnet use ports starting from 2048 to allow 1024 to 2047 to be used by the os
$IPT -t nat -A POSTROUTING -s $DMZ_IP -o $INET_IFACE -j SNAT -p tcp --to-source $INET_IP:2048-
32000
echo __RULE 29: LAST OF ALL, WHAT IS LEFT GETS LOGGED AND DROPPED ----
$IPT -A INPUT -j LOG
$IPT -A FORWARD -j LOG
$IPT -A OUTPUT -j LOG
$IPT -A INPUT -j DROP
$IPT -A FORWARD -j DROP
$IPT -A OUTPUT -j DROP
```

The interior packet filter is also configured with similar deny rules as the exterior packet filter to act as a second layer of defense in case the DMZ hosts or the exterior router are compromised.

2.2.5 Firewall Proxies

GIAC enterprises needs to provide its employees email and web access. Sendmail serves as a proxy for SMTP. It acts as a mail relay between the public Internet and the internal mail server to protect it from application-level attacks and remove harmful content such as viruses, attachments, spam, and MIME-type based filtering to block executable content before forwarding mails to/from the *real* mail server *‘PostFix (version 2.1.14)’,* hosted in the internal network. Different types of software are used for the internal and the external mail server to avoid the same vulnerability effecting both mail servers at once.

A SQUID web proxy cache filters is deployed at the DMZ on a bastion host. GIAC employees access HTTP(s) and FTP services from the Internet via the proxy server which performs application level filtering for all traffic. GIAC’s security policy prohibits certain objectionable sites to be accessed form the internal network, this can be enforced by configuring different access controls at the squid proxy server. The proxies also prevent trojans/virus/rootkits from establishing direct connections with Internet hosts by masquerading as a legitimate protocol. SQUID is a full-featured web proxy cache. It is easy to use, with a large no of config options, supports all major caching protocols, has extensive access controls and has good logging features supporting the common log format. Good logging helps in misuse detection.

2.2.6 Domain Name Service (DNS)

The DNS service is an important component of any site since mail server and users from the Internet would require it to query DNS records. It is important to be secured since it has been the target of many known denial of service and man-in-the-middle attacks. For simplicity and due to the small size of the organization, an internal DNS server is not used. All internal hosts are identified by fixed IP addresses. *Bind* software on a bastion host on the DMZ provides the primary DNS server for the site. For the public Internet it appears as if the exterior router itself provides the DNS service, since it is internally DNAT*ed* (destination NAT) to re-direct queries to the DNS server. An external entity can be used to host the secondary DNS server. DNS MX records point to the mail server using the setting:

```
giac.com    IN    MX    5     www.giac.com.
```

2.2.7 VPN

Remote users (sales/partners) connect into the Internet and require POP3 and SMTP to access internal emails and require an internal ftp server for sharing files with other employees at GIAC enterprises. An SSL based VPN gateway is setup using OpenVPN[20] on a bastion host at the DMZ. It uses a UDP based SSL tunnel to implement a VPN. This tunnel terminates at the DMZ and enables remote users to access SMTP, POP3 and ftp services which would otherwise be unavailable from the Internet as the packet filtering routers would block them. Windows 2000 Laptops of the remote work force using VPN are protected by anti-virus and personal firewall. They would connect to the public Internet and then use start OpenVPN to use these internal services. OpenVPN uses TLS to create virtual TUN/TAP devices at the remote user's desktop for access to the DMZ network (TUN is a virtual IP interface which enables tunneling of TCP/IP protocols, TAP is a virtual Ethernet adapter which can tunnel any network protocol).

Regional offices: At each regional office, due to the low no of employees (5 sales personnel each); any kind of network infrastructure would be expensive to maintain. Hence, all sales employees would connect directly to the Internet from their hardened laptops configured with personal firewalls and anti-virus and use OpenVPN to get access to internal services at GIAC enterprises.

2.2.8 Syslog Server

The syslog server is central to intrusion detection and monitoring of the firewall. It is hosted on the internal network so that host logging is secured to the maximum possible extent. It functions as the central log repository for recording log system and application log messages[21]. A write-once media (DVD+R with "*growisofs*") could be used to make regular copies of the logs as indestructible evidence in the event of the syslog server also being compromised. By providing central log storage, intrusion attempts at various network hosts can be correlated and detected more quickly.

2.2.9 Network based IDS

Two Snort sensors have been deployed at the DMZ and internal network to function as a network IDS. A large no of recently updated signatures and scripts are available [31], but only those signatures are deployed which are relevant to the services hosted to minimize false alarms. It provides customizable alerting features and rules are simple to define. Snort is initially configured to log all alarms, which are then manually analyzed to understand normal activity. Irrelevant alarms are disabled to minimize false alerts. Each network sensor is configured with 2 interfaces – one for passively gathering data from a network 'Tap' and the other to connect to the syslog server to log alerts. This prevents the IDS from being a possible victim in an attack.

2.2.10 Other Services

The OpenVPN gateway at the DMZ also hosts the NTP server for synchronizing time across the site - this is important for correlating events in logs. An NTP client on the host first synchronizes itself with the NTP stratum 2 servers and then other internal hosts can be synchronized using this internal NTP service. It is restricted to connect to pre-defined stratum 2 NTP servers[18]. An ftp service (Wu-Ftpd) for partners/suppliers & sales files sharing is also hosted at this server. Apache with OpenSSL is used as the portal web server, a J2EE application server enforces application-level access controls for querying/modifying the PostgreSQL database.

2.3 IP addressing scheme

Note: The private intranet address space from RFC1918 is used for the DMZ and the internal network.

IP Address	Hosts	Remarks
64.236.24.1	Exterior Packet Filtering Router (R1)	This IP address is chosen randomly, any correlation with actual internet IP address is purely coincidental and un-intentional
10.1.1.1 to 10.1.1.8 (10.1.1.0/27)	Exterior router internal interface, DMZ hosts – VPN, Web proxy, Web servers, mail relay, snort; and interior router interface on the DMZ	DMZ network: R1, E1 to E6, and R2
172.16.2.1 to 172.16.2.64	Interior router internal interface and all internal hosts & devices	Internal corporate network: R2, I1 to I6, all internal devices
10.1.1.9 to 10.1.1.34	OpenVPN virtual addresses	

2.4 Defense in depth

2.4.1 Placement of Components

Layered security concepts are implemented at several areas of the design. Services, which face the Internet, are all placed in the DMZ so that if compromised these remain isolated form the network. The database is placed on the internal network and kept inaccessible from any DMZ host or Internet; it can only be accessed by the application server which in turn is accessible only from the web servers. The highly visible web-server does not store any data and is placed on the DMZ so if compromised it still cannot reach the internal network. The actual mail server is placed on the internal network unreachable directly from the Internet. All SMTP traffic is intercepted by the mail relay on the DMZ and filtered before forwarding. If the email services are compromised it only affects the mail relay on the DMZ bastion host. The OpenVPN tunnel also ends up at the DMZ so all access by VPN too gets filtered by the interior packet filtering router to protect the internal network if a sales user's remote workstation is compromised by a Trojan or virus. Creation of a DMZ or screened subnet provides an ideal location where traffic can be monitored by an IDS

2.4.2 Mitigating Security Weaknesses using defense-in-depth

Individual security weaknesses of components are mitigated by employing complementary protection mechanisms to add more layers of security. Inherent weaknesses in packet filtering to stop attacks are mitigated firstly by using two packet filters with strict rules and secondly by forcing network traffic to bastion hosts which intermediate access at the application and protocol levels using web-proxy and the email relay. DMZ network is a switched network to prevent traffic sniffing.

2.4.3 Host hardening

Hardening the host machines is a critical step and must be done as per a pre-defined checklist for each type of service and operating system. Some common steps such as deploying SELinux, running daemons under chroot jails, eliminating un-necessary services, removing redundant users and running file integrity checkers are essential. A full discussion on hardening is beyond the scope of this paper. Very good resources are available for performing host hardening for Linux/Unix systems such as the Linux security and optimization guide [13], CISecurity's Linux benchmark tests[14] and CERT Unix checklists[15] which are highly recommended.

2.5 Factors influencing the choice of components

The decision to use popular open-source software for building the firewall was driven by both economic and technical reasons. IPTables, Squid, Bind, Sendmail, Snort, Apache provides advanced state-of-the-art features and being open-source have been widely deployed & tested in different environments. These popular open source software have quick development,debugging & test cycles, which reduces probability that severe flaws go un-detected. The large no of excellent books, manuals, how-tos and tutorials available for them make them easy to configure, administer and debug.

2.6 Administration

All administration and monitoring of hosts are done via OpenSSH 3.9p1 and sftp. This provides a secure protocol for the highly sensitive administrative activities. Administrators authenticate SSH sessions using public-private keys which are changed periodically. A central CA may be deployed using OpenSSL and OpenLDAP for site-wide PKI.

2.7 Supporting security mechanisms

A firewall is not complete and ready for use by installing and configuring the software alone, several supporting processes are required to ensure it is effective. These are:

1) A system integrity checker be used, e.g. AIDE (Advanced intrusion detection environment) [17] is a file integrity checking program used here for intrusion detection to determine if system binaries or configuration files have been altered. The AIDE database backup copy be kept offline from the host itself.
2) The site serves customers at diverse geographical locations and must be available 24x7. Adequate backups of data as well as software must be kept ready for immediate recovery in a contingency scenario [19]. All system logs and security logs must be stored and retained onsite and later moved to offsite storage location. Maintaining bootable & installable CDs for quick recovery.
3) Site must be monitored at all times. Network or IDS alerts and security incidents need to be responded immediately by a team as per a pre-defined escalation process set by management; logs should be preserved for use as evidence later.
4) Ensure adequate physical security of the infrastructure to protect all equipment; protect unattended terminals by setting up session timeouts.
5) Change control & patching process should be well defined and followed strictly; changes should be documented and tested out in non-production environments.
6) BIOS level passwords should be setup on a few security critical hosts so that any re-boot caused by malware gets stuck up and personnel are alerted.
7) The organization's infrastructure should not be used to harm others. All users must be made aware that misuse could result in strict and severe action.
8) Network & Firewall documentation: An updated, good, clean and detailed documentation of the network and its configurations should be available for ready reference for troubleshooting or recovery from network/service failure.
9) Testing process: To verify access controls are functional these must be validated periodically by using network vulnerability probing tools. Some tools are: Sara, Saint, Nessus, nmap. Probes should be run at Internet, DMZ and internal network.
10) The Switches should have port locks to allow only specific pre-defined devices to connect into the network based on their MAC address, it adds a layer of security.
11) The organization must not permit any access into the network bypassing the firewall, all wireless or dial up must be disallowed unless it is firewalled.

Note: All the equipment mentioned in this assignment was not available for testing.

Assignment 3: Firewall Security Policy

This section discusses the rule base for the firewall described in assignment 2. The business requirements have been described in assignment 2. Based on that discussion, an access control matrix is created to define access controls when an object (information) is accessed by a subject (users). The access control matrix is then elaborated to develop the firewall rule-base for the firewall architecture.

3.1 Data Classes

The information hosted by GIAC enterprises infrastructure can be classified into the following classes:

1. Public data: This includes product specifications, marketing promotional data, public data about the company and other relevant public news of interest to anyone visiting the company website.
2. Customer data: Any data related to a customer's order which the organization is obliged to protect.
3. Corporate data: Internal data of the company's operations/ finances/ performance/ processes which the company is obliged to protect or information if disclosed to un-authorized individuals could provide a competitive advantage to its competitors. This data would also include software developed and used by the organization, its research activities, product plans, future marketing plans & strategies, etc.
4. Order fulfillment data: Data about new fortune cookie requirements & inventory levels made available to the suppliers or provided by the partners or sales personnel. This is also part of the corporate data "class", but has been listed separately since partners and suppliers also use it.

3.2 Access Control Matrix:

The access permissions and security requirements for each group (subjects) for controlling access to each object is listed in the matrix below:

	Public Data	**Customer Order data**	**Corporate (internal) data**	**Order fulfillment data**
General Public	Read	No access	No access	No access
Customers	Read	Create, modify	No access	No access
Partners	Read	No access	No access	Create, modify
Suppliers	Read	No access	No access	Create, modify
Sales	Read	Read	Create, modify	Create, modify
Developers & Researchers	Create, modify	No access	Create, modify	No access
Finance, Order fulfillment & helpdesk unit	Create, modify	Read, modify	Create, modify	Create, modify
Administrators	Modify (DNS)	Read	Create, modify	Read
Senior Management	Read	Read	Create, modify	Create, modify

A matrix showing the network traffic accessing these services & data to and from each host/router is created to help in creating the firewall rule-base.

3.2.1 DMZ Access Matrix

The exterior packet filtering router controls access to/from the Internet, the DMZ bastion hosts and the interior router. This is represented as an access control matrix to validate the controls required from the exterior packet filtering router. Network requests (TCP or UDP) are shown as ‘Q’, and replies are shown as ‘A’.

From ⇨ To ⇩	Internet	Web Proxy/DNS	Web Server	OpenVPN, DMZ ftp & NTP	Mail Relay	Interior router
Internet		http Q, https Q, ftp Q, dns A/Q	http A, https A	ssl A, ntp Q	SMTP Q, SMTP A	-
Web Proxy/DNS	http A, https A, ftp A, dns Q/A		-	-	dns Q	http Q, https Q, ftp Q
Web Servers	http Q, https Q	-		-	-	http Q, https Q
OpenVPN, DMZ ftp & NTP	ssl Q, ntp A	-	-		-	ssh A, ftp Q, ntp Q
Mail Relay	SMTP Q, SMTP A	dns A	-	-		SMTP Q, SMTP A
Interior router	-	http A, https A, ftp A	http A, https A	ssh Q, ftp A, ntp A	SMTP Q, SMTP A	

Other than these services, SSH for administration & syslog for logging will be permitted for all hosts.

3.2.2 Internal Network Access Matrix

Similarly, the matrix given below shows the access controls required to be implemented by the interior packet filtering router.

From DMZ ⇨ To ⇩	Web Proxy/DNS	Web Server	OpenVPN, DMZ ftp & NTP	DMZ Mail Relay	Exterior Router
Application Server	-	https 8443,8080	ntp	-	-
Database Server	-	-	ntp	-	-
Internal Mail server	-	-	Pop3, SMTP, ntp	SMTP	-
File server & patch distribution host	http, https, ftp	-	ftp, ntp	-	-
Administrator Workstations	http, https, ftp, ssh	http, https, ssh	Ssh, ntp	ssh	ssh
User workstations	http, https, ftp	http, https	ntp	-	-
Syslog Host	syslog	syslog	Syslog, ntp	syslog	syslog

We now apply these and the guiding principles described in assignment 2 to evolve the firewall rule base.

3.3 Firewall Rule Base

The firewall rules will apply to the external packet filtering router, bastion hosts, and the internal packet filtering router to implement access controls in accordance with the access matrix and the guiding principles described in assignment 2. Here, eth0 is the internet/DMZ facing external interface and eth1 is the inward facing interface.

Sl no	Source Address	Source Port	Protocol	Device & Interface	Destination address	Dest Port	State	Other Options	Decision
F1	127.0.0.0/8 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 0.0.0.0/8	any	tcp, udp, icmp	R1 eth0	any	any	-	-	DROP
	127.0.0.0/8 172.16.0.0/12 192.168.0.0/16, 0.0.0.0/8	any	tcp, udp, icmp	R2 eth0	any	any	-	-	DROP
All incoming packets with bogus source IP addresses (internal intranet IP address range - RFC 1918) should be denied all access at the external router. This is because IP packets with source address of private intranet address range are definitely spoofed and must be blocked. Similarly, all incoming packets with source IP address other than the DMZ be blocked at the interior router. All traffic except ssh and syslog from the exterior router to the interior router be denied at the interior router.									
F2	any	any		R1 eth0	any	22, 23, 21,20, 119, 110, 513, 514, and 512, 69, 111, 123, 520, 70, 540, 79, 7, 546, 547, 194, 389, 143, 161, 569	-	-	drop
Any incoming traffic for services not hosted for internet users by GIAC is blocked; this includes DHCP requests, echo, RPC, ntp, nntp, ftp, nfs, tftp, telnet, finger, ssh, uucp, gopher, rlogin, rsh, and rexec, RIP, IRC, LDAP, IMAP, snmp, MSN, refer to [19]									
F3	any	any	tcp, udp	R1 & R2 eth0	any	any	-	source routed	Log, drop
	any	any	tcp	R1&R2 eth0 and eth1	any	any	-	FIN, URG, PSH set	Log, drop
	any	any	tcp	R1&R2 eth0 and eth1	any	any	-	All options set set	Log, drop

	any	any	tcp	R1&R2 eth0 and eth1	any	any	-	NO options set	Log, drop
	any	any	tcp	R1&R2 eth0 and eth1	any	any	-	SYN, RST set	Log, drop
	any	any	tcp	R1&R2 eth0 and eth1	any	any	-	SYN, FIN set	Log, drop
Incoming packets with source routing should be denied access; source routed IP packets are most likely to have been spoofed. Bad TCP packets be denied access – e.g. fin,urg,psh set; all options set; no option set; syn & fin options set; syn,rst options set									
F4	any	any	tcp	R1 & R2 eth0 & eth1	any	any	new	No syn set	Log, drop
	any	any	tcp	R1 & R2 eth0 & eth1	any	any	established, related		accept
All outgoing or incoming tcp response packets which are not related to any previous requests should be denied access. This is done by checking the packet 'state' using the connection tracking feature of the firewall.									
F5	any	137:139	tcp	R1, R2	any	any			Drop
	any	137:139	udp	R1, R2	any	any			Drop
Drop all NetBIOS over IP packets from being forwarded over the router									
F6	any	any	icmp	R1&R2 eth0	any	any		Type 3 code 4	accept
	any	any	icmp	R1&R2 eth0	any	any		Type 4	accept
Disallow ICMP packets other than type 3 (for mtu discovery) and type 4 (source quench) since these may be part of a scan or a DOS attempt									
F7	any	any	tcp	R1 eth0	10.1.1.2 10.1.1.3	80, 443			accept
	10.1.1.2 10.1.1.3	80, 443	tcp	R1 eth0	any	any			accept
HTTP & HTTPS requests to the bastion host web server from the internet and its related responses from the web server be permitted. This is the most critical service and hence the first matching rule after filtering out the bogus connections.									
F8	10.1.1.1	any	tcp		any	80, 443			accept
	any	80, 443	tcp		10.1.1.1	any			accept
HTTP & HTTPS requests from the proxy server to internet & its response packets back to the proxy server should be permitted.									
F9	10.1.1.1	any,data	tcp	R1 eth0	any	ftp			accept
	any	ftp	tcp	R1 eth0	10.1.1.1	any + data			accept
FTP requests from the proxy server and its response packets back to the proxy server should be permitted.									
F10	any	any	tcp, udp	R1 eth0	10.1.1.1	53			accept
	10.1.1.1	53	tcp, udp	R1 eth0	any	any			accept
Incoming DNS requests to the DNS server and its responses be permitted									
F11	10.1.1.1	any	tcp, udp	R1 eth0	any	53			accept
	any	53	tcp, udp	R1 eth0	10.1.1.1	any			accept
DNS requests to the Internet from the DNS and Proxy server and its responses back should be permitted.									
F12	any	any	tcp	R1	10.1.1.5	25			accept
	10.1.1.5	25	tcp	R1	any	any			accept
	any	any	tcp	R1	10.1.1.5	113			reject
	10.1.1.5	113	tcp	R1	any	any			reject
Incoming SMTP requests to the bastion host mail server & its related responses should be permitted, reject IDENT requests									
F13	10.1.1.5	25	tcp	R1	any	any			accept
	any	any	tcp	R1	10.1.1.5	25			accept
Outgoing SMTP requests from the bastion host mail server and its responses should be permitted									
F14	any	any	udp	R1	10.1.1.4	5000			accept
	10.1.1.4	5000	udp	R1 eth1	any	any			accept
Incoming OpenVPN SSL requests to the VPN bastion host and its related responses from the VPN bastion host be permitted.									
F15	172.16.2.0/24	any	Tcp	R1 eth1	10.1.1.7	22	-	-	accept
	10.1.1.7	22	Tcp	R1 eth1	172.16.2.0/24	any			accept
Only ssh connection coming from the interior network should be permitted since only administration should have shell access.									
F16	any	any	Tcp	R1 eth0, eth1	10.1.1.4	ftp			Log, drop
	10.1.1.4	ftp	Tcp	R1 eth0, eth1	any	any			Log, drop
Explicitly deny FTP requests from the internet to the DMZ FTP server									
F17	any	any	tcp	R1 eth0, eth1	10.1.1.4	ntp			Log, drop
	10.1.1.4	ntp	tcp	R1 eth0, eth1	any	any			Log, drop
Explicitly deny NTP requests from the internet to the DMZ NTP server									
F18	10.1.1.4	any	tcp	R1 eth0, eth1	stratum2 servers	ntp			accept
	stratum2 servers	ntp	tcp	R1 eth0, eth1	10.1.1.4	any			accept
Allow internal NTP server to query and synchronize itself with the stratum 2 servers identified [18].									
F19	10.1.1.0/27	any	Tcp	R2	172.16.2.19	syslog	-	-	accept
	172.16.2.19	syslog	Tcp	R2	10.1.1.0/27	any	-	-	accept
All syslog connections coming from the DMZ network to the syslog host should be permitted									
F20	172.16.2.0/24	any	tcp	R2	10.1.1.2, 10.1.1.3	80, 443	-	-	accept
	10.1.1.2, 10.1.1.3	80, 443	tcp	R2	172.16.2.0/24	any	-	-	accept
All http & https requests routed to the web servers be allowed from the internal network (e-com portal)									
F21	172.16.2.0/24	any	tcp	R2	10.1.1.1	80, 443, 21			accept

	10.1.1.1	80, 443, 21	tcp	R2	172.16.2.0/ 24	any			accept
Permit HTTP requests from the internal network to the web proxy server									
F22	172.16.2.17	any	tcp	R2	10.1.1.5	25			accept
	10.1.1.5	25	tcp	R2	172.16.2.17	any			accept
Permit SMTP requests from the interior mail server to the DMZ relay mail server									
F23	10.1.1.0/24	any	tcp	R2	172.16.2.17	110			accept
	172.16.2.17	110	tcp	R2	10.1.1.0/24	any			accept
Accept any pop3 requests from the DMZ (openvpn addresses) to the internal POP3 mail server									
F24	10.1.1.0/24	any	tcp	R2	172.16.2.17	25			accept
	172.16.2.17	25	tcp	R2	10.1.1.0/24	any			accept
Accept any SMTP requests from the DMZ (openvpn addresses) to the internal SMTP mail server									
F25	172.16.2.0/24	any,data	tcp	R2	10.1.1.4	ftp	new		accept
	10.1.1.4	ftp	tcp	R2	172.16.2.0/24	Any + data	established		accept
Permit FTP requests from the interior network to the DMZ ftp server									
F26	172.16.2.0/24	any	tcp	R2	10.1.1.4	ntp			accept
	10.1.1.4	ntp		R2	172.16.2.0/24	any			accept
Permit NTP requests from the interior network to the DMZ NTP server for synchronizing with the time.									
F27	10.1.1.2 10.1.1.3	any		R2	172.16.2.20	8080, 8443			accept
	172.16.2.20	8080, 8443		R2	10.1.1.2 10.1.1.3	any			accept
Permit web servers at the DMZ to access the application servers for the e-com portal									
F28	172.16.2.0/24	any	tcp	R2	10.1.1.1/27	22			accept
	10.1.1.1/27	22	tcp	R2	172.16.2.0/24	any			accept
SSH requests from the administration VLAN to the Interior router, Bastion Hosts & exterior router be permitted; its responses should also be permitted.									
F29	any	any		R1, R2	any	any			Log, drop
Deny all other traffic not matching with any of the above accept rules									

3.4 Remarks

Under the ‘decision’ field in the Rule Base there exist two options for stopping packets:

- **REJECT** option stops the packet, and sends a message back to the sender that the packet was not accepted.
- **DROP** option simply drops the packet, and no message will be sent back to the sender.

DROP is used often since REJECT may reveal the firewalls existence when the firewall is being scanned it will inform the port scanner which ports are not available on the target system.

By using DROP the sender will not get any reply for those ports that are scanned.

In some cases a **REJECT** with reply becomes important to maintain the requirements of internet standards or when the external hosts request is genuine and a timeout would degrade the service, e.g. IDENT reject when contacting an SMTP server.

The firewall rule base supports the security stance of the company by implementing layered access control (2 stateful packet filters) and preventing any direct access from any host into the Internet.

The rule base has not been listed in any specific order; it lists rules for the exterior router and then for the interior router. The IPTables scripts add rules to the chains in order of service criticality, vulnerability risk level, or as per the expected traffic rate for each type of service. Initially all packets may be logged, these logs can be manually examined and later rules can be re-ordered as per traffic, with high traffic filtered out first to improve filtering performance.

References

[1] D Brent Chapman, “Network (In)Security through packet filtering”, 1993-07-20 http://csrc.nist.gov/secpubs/pktfilt.ps

[2] Matt Bishop: “Computer Security” (Pearson Education Inc), 2003

[3] Thomas H. Ptacek, “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection”, January 1998, http://secinf.net/info/ids/idspaper/idspaper.html

[4] Rebecca Bace and Peter Mell, “Intrusion Detection Systems: NIST Special Publication on Intrusion Detection Systems”, 2000.

[5] Susan M. Bridges, Rayford B. Vaughn, “Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection”, 16-oct-00, http://csrc.nist.gov/nissc/2000/proceedings/papers/005slide.pdf

[6] Adhitya Chittur, “Model Generation for an Intrusion Detection System Using Genetic Algorithms”, 27-Nov-2001, http://www1.cs.columbia.edu/ids/publications/gaids-thesis01.pdf

[7] Steven Cheung, Rick Crawford, Mark Dilger, Jeremy Frank, Jim Hoagland, Karl Levitt, Jeff Rowe, Stuart Staniford-Chen, Raymond Yip, Dan Zerkle, Department of Computer Science, University of California at Davis, “The Design of GrIDS: A Graph-Based Intrusion Detection System”, January 26, 1999

[8] Netfilter.org, “IPTables Man. page”, 2002 http://iptables-tutorial.frozentux.net/other/iptables.html

[9] Oskar Andreasson, “Iptables Tutorial 1.1.19 “, 2001-2003 http://www.jollycom.ca/iptables-tutorial/iptables-tutorial.html

[10] Zwicky, Cooper & Chapman, “Building Internet Firewalls”, 2003

[11] Comer, “Internetworking with TCP/IP Volume 1”, 2001

[12] “Well known TCP/IP port numbers” http://www.webopedia.com/quick_ref/portnumbers.asp

[13] Ted Nackad, “Securing and Optimizing Linux: The Ultimate Solution”, 2001-06-10 http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-The-Ultimate-Solution-v2.0.pdf

[14] Centre for Internet Security, “CIS Level-1 Benchmark and Scoring Tool for Linux”, 2003 http://www.cisecurity.org/bench_linux.html , 2004

[15] CERT, “Unix Security Checklist”, ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist

[16] William Metcalf, HoneyNet Project, “Snort Inline”, 2004 http://snort-inline.sourceforge.net/FAQ.html

[17] Rami Lehti, “AIDE Manual” http://www.cs.tut.fi/~rammer/aide/manual.html

[18] David L Mills, “Public NTP Secondary (stratum 2) Time Servers”, 30-Aug-2004 http://www.eecis.udel.edu/~mills/ntp/clock2b.html

[19] CERT, Recover from a system compromise http://www.cert.org/tech_tips/root_compromise.html

[20] James Yonan, “OpenVPN How-to”, 2002-2004, http://openvpn.sourceforge.net/howto.html

[21] Setting up a remote log server using syslog, http://www.linuxsecurity.com/feature_stories/remote_logserver-3.html

[22] Netscreen, “Intrusion detection& prevention System: Architecture”, http://www.juniper.net/products/intrusion/architecture.html

[23] Rain Forest Puppy, “A look at whisker's anti-IDS tactics”, http://www.wiretrip.net/rfp/