Global Information Assurance Certification Paper

James G. McIntyre
Version 4.0
July 19, 2004

Table of Contents

Summary
This paper discusses the potential for utilizing computer technology for monitoring, analyzing, and collecting computer generated network and system alerts. It also details a design for a company with network and security requirements common to today's environment. Lastly, it provides an overview of a firewall rule set using common netfilter/iptables commands.

Assignment No. 1

The purpose of this assignment is to introduce or explore a particular technology or future technology that is not in common use in the Information Security Industry today, but that you feel will have significant impact on perimeter security, or defense in depth. Explain the problem it mitigates, how the technology works, what affect it will have on the Information Security industry, and the effect on personnel tasked with the day-to-day operations of perimeter technology. Page guideline, 5-10.

So you have implemented your grand plan for your network and system security. You have designed a tight perimeter and implemented layers of security for your defense-in-depth needs. Now the hard work begins, verify the perimeter security and the defense-in-depth works as designed. This must be done on all levels of the network, which include routers, servers, switches, and workstations. One must continuously monitor and analyze the network traffic and status of individual systems. Then a situation arises and an alert is generated by one of your many alert generators. Do you have the processes in place to collect the alert, correlate alerts, and perform some type of analysis in a timely fashion? Of course there is the problem of varying platforms generating alerts in different formats, different protocols, and stored in many locations.

The security alert generators I am referring to include diverse products such as Snort, (Caswell B), ACID (Danyliw), swatch (Swatch), and portsentry (Rawland). These tools are very effective at generating an alert for some specific problem or perceived problem. The amount of analysis required by these products to generate an alert very greatly. In Snorts case, the analysis can be quite extensive. Swatch on the hand performs simple pattern matching against a configuration file. But all of these tools have the same inherent problem, "What happens to the alert after it is generated?"

Some of the problems with alerts generated by these tools include:

1. Different format of alert message.
2. What is the priority of an alert?
3. Are there other alerts issued?
4. Is this the100th alert sent in the last minute?
5. Was the notification of the alert sent?
6. Did the notification go to the correct person(s)?

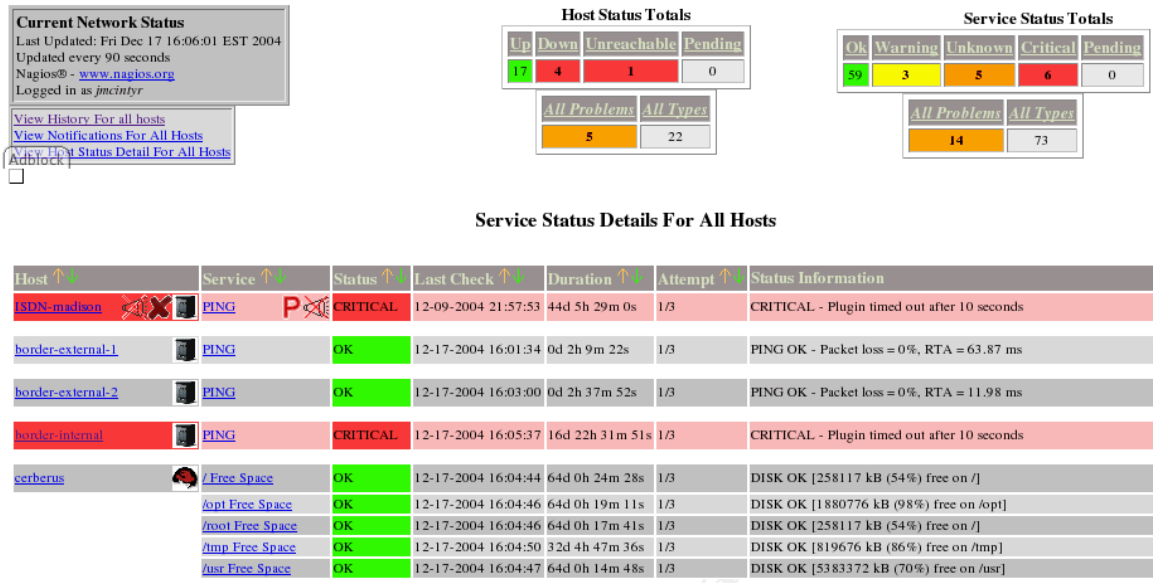
7. Did anyone respond?
8. Was the problem resolved and how?
9. Was information about the alert retained for later analysis?
10. How did this alert relate to other alerts?
11. Can we analyze this alert with others and develop some hypothesis of a larger problem?
12. Did this alert indicate a problem with the perimeter security?
13. Did this alert indicate a problem with the defense-in-depth?
14. How quickly can we gather the information we need to determine what is happening?
15. From one location can one easily access all of this data and perform some type of analysis in a timely manner?

The missing component to this process is the coordination and centralization of the alert data, analysis capabilities, and problem tracking. The pulling together of this data and being able to analyze it in either real-time or long-term planning has always been a problem. The alert generation tools were designed to fulfill the need of recognizing a possible problem and try to notify someone. We have been lacking the common collection, analysis console and notification follow-up. Obviously building this type of functionality into all of these tools is not a workable solution. To this end I would like to discuss the use of an Open Source tool known as Nagios. (Galstad)

Nagios is defined in its manual as a “Network Monitoring Application”. (Galstad). It is a framework for monitoring hosts, services, tracking state changes, collecting state changes, alert collection and storage, alert analysis, support personnel notification, and escalation procedures for problem notification. Nagios provides a web front end as its delivery method. Therefore it can very accessible to authorized personnel.

Nagios refers to a “host” as a router, switch, linux server, or perhaps a windows workstation. A “service”, in its standard use, would be cpu-utilization, network-utilization, DNS services functionality, file-system % utilization, or verification that a web server is responding to requests. The definition of what a service is in some respects left to the users creativity.

The “state” refers to a host or service availability or status. The state indicators include: up, down, ok, critical, or warning. A state of “up” could indicate the mail server was ping-able. The state of “warning” could indicate a file-system % utilization is between 65%-85%. The state of “critical” could indicate the network traffic utilization on the internal side the edge router is much higher than set limits.
When defining a service the user determines what a state of “critical” or “warning” means for that service. Also when a service is defined, it is associated with a specific host.



Current Network Status
Last Updated: Fri Dec 17 16:06:01 EST 2004
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *jmcintyr*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
17	4	1	0

All Problems	All Types
5	22

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
59	3	5	6	0

All Problems	All Types
14	73

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ISDN-madison	PING	CRITICAL	12-09-2004 21:57:53	44d 5h 29m 0s	1/3	CRITICAL - Plugin timed out after 10 seconds
border-external-1	PING	OK	12-17-2004 16:01:34	0d 2h 9m 22s	1/3	PING OK - Packet loss = 0%, RTA = 63.87 ms
border-external-2	PING	OK	12-17-2004 16:03:00	0d 2h 37m 52s	1/3	PING OK - Packet loss = 0%, RTA = 11.98 ms
border-internal	PING	CRITICAL	12-17-2004 16:05:37	16d 22h 31m 51s	1/3	CRITICAL - Plugin timed out after 10 seconds
cerberus	/ Free Space	OK	12-17-2004 16:04:44	64d 0h 24m 28s	1/3	DISK OK [258117 kB (54%) free on /]
	/opt Free Space	OK	12-17-2004 16:04:46	64d 0h 19m 11s	1/3	DISK OK [1880776 kB (98%) free on /opt]
	/root Free Space	OK	12-17-2004 16:04:46	64d 0h 17m 41s	1/3	DISK OK [258117 kB (54%) free on /]
	/tmp Free Space	OK	12-17-2004 16:04:50	32d 4h 47m 36s	1/3	DISK OK [819676 kB (86%) free on /tmp]
	/usr Free Space	OK	12-17-2004 16:04:47	64d 0h 14m 48s	1/3	DISK OK [5383372 kB (70%) free on /usr]

Figure 1. Example of “Service Status” monitoring by Nagios webpage.

Nagios’ use of color highlighting, red (critical), yellow (warning), and green (ok), helps one quickly identify problem host(s) or service(s). In this example several routers are not accessible. The user can select the service or host in question and drill down for more specific information about an alert. If desired, the user can drill down further and review the alert history.

Therefore, from one central interface, these questions could be answered relatively quickly:

1. Yes, an alert was generated for a host or service.
2. Has this service been having problems?
3. When did the problem begin?
4. When was the problem last seen?
5. What is the history of this service for the last 2 months?
6. Has the problem been occurring on other hosts?

From a support perspective, having one location for monitoring your entire IT environment has several positive aspects. First, there is one location support personnel are monitoring for possible problems. Access to this information is utilizing web pages so the information is easily accessible by authorized personnel. More importantly, it is possible to see multiple alerts for a specific host. Step back and one can identify alerts for several hosts. It may be possible to draw a correlation between multiple alerts given your wide enterprise view of your network. For instance, if you have a file system suddenly fill up, it might be helpful to know the cpu utilization for that host is very high, and the network traffic is higher than normal. Bringing together disparate alerts from several platforms, different alert tools and storing them in a common format is one of

Nagio's strong points.

So let's expand the standard use of service and extend it to the security monitoring area. Lets define a service as "swatch alerts" or "swatch - root login failures" or "Snort alert - Trojan horse identified" or "Portsentry – host added deny list" or "iptables alert – syn scan". The services you define can be as general or specific as the user desires. Given this new use of "services", the support person can be notified of security alerts with the other standard alerts. Now with our expanded use of services, additional data is available. Given our previous problem, now include a portsentry alert which denied new ip addresses and a firewall alert that identified a possible syn-attack. Now our analytical view has expanded greatly. Given this additional data, it may help determine if there is a perimeter or a defense-in-depth problem.

So Nagios has helped with the problem of alert collection in one central repository, provided an easy to use web front-end, a secure easily accessible front-end, and a framework for analyzing the generated alerts.

Additional unresolved problems with the alert generators includes tracking of the alert. Most of these tools do not have a notion of alert priority. For instance, a file system at 65% utilization is much less of a problem than 98% utilization. Nagios has a very well defined protocol for who and how notification is sent to the support personnel. Notification could include: email, pager, pop-ups, or cell phone. Of course the user can build their own special notification process and it is easily added it to Nagios. If a new notification process is required it only requires in one place versus multiple systems. Again, a central location for issuing notifications can greatly reduce the problems associated with each system issuing their own alert. Also, all information about the notification is tracked as to what when it was sent, to whom it was sent, and how was it sent.

Once the tech responds, their analysis and resolution can be retained in the Nagios database. But, what if the tech does not respond to the notification? Nagios has several escalation procedures for notification. If there is no response from the level 1 support, the next level of support is notified. The notification process can be the same as before, i.e. just an email. But perhaps for level 2 text messages are sent to multiple techs' cell phones. The user defines what escalation procedures they desire for each service. Again, all notifications or escalations are collected for later review.

At this point you have collected data about alerts for services or hosts, who was notified and how the problem was resolved. All of this information has been stored in the Mysql (Axmark D) database backend for Nagios. This information, if utilized, could be very helpful from a long or short term perspective. The alerts can be analyzed from several aspects. General reports of alerts can easily be generated based on time frames for hosts or services. Graphing tools allow one to graph trends or alert history. Again, quite a bit of information concerning the

security environment is in one common format and easily analyzed. Additional reporting formats and graphing could make the collected data even more useful for a security analyst. The user also has the option of writing analysis programs accessing the data stored in the Mysql database.

Much of what we have discussed depends on the actual collection of the alert. Nagios has two primary techniques for gathering information, “push or pull”. The technique used more often for is the “pull”, i.e. the Nagios server requests the information from a remote host. This can be accomplished via a snmp type call, a pre-written utility, or a user developed shell script. The “push” is accomplished with a pre-written utility or a user developed shell script. In this case the remote host sends the information to a special daemon on the Nagios server. Either technique seems to work well. Which technique used depends more of the application in question. For instance, the pre-written utility for portsentry to send an alert to the Nagios server is below. (Galstad)

```
#!/bin/sh
#
# arguments
# $1 = name of host in service definition
# $2 = name/description of service in service definition
# $3 = return code
# $4 = output
#
/bin/echo –e “$1\t$2\t$3\t$4\n” | /usr/local/nagios/bin/send_nsca_monitor
        –c /usr/local/nagios/etc/send_nsca.cfg
```

In conclusion, the combination of the well-known alert generators and a framework like Nagios will become more important to all sizes of shops. The number of hackers knocking at the perimeter door and tapping on the windows testing your environment is not going to decrease anytime soon. Therefore the number of disparate alerts you are generating will not be decreasing either.

Most environments continue to become more complex with network hardware, servers, online applications, and security tools. Consequently, supporting the environment from a security perspective is more complex. Most environments are also limited by the number of support staff and resources for maintaining the security for the organization. Given all of these problems our need for utilizing computers to assist in the collection, presentation, notification is a priority if we are to keep up with all the security demands. Having tools that quickly and easily present many aspects of a networks status is necessary from a response perspective. Also, with a simplified interface a “lower level” support person could become an integral part of the security monitoring structure. Again, having all this security data in one location and easily analyzed can help improve our perimeters and our defense-in-depth defenses.

Assignment No. 2

GIAC Enterprises, GIAC-E, is a small business that markets fortune cookie Fortunes to customers worldwide. GIAC employs fifty people with the majority located in or near its head office and the remainder located in or near the four regional satellite offices geographically distributed around the world. All of GIAC Enterprises sales are done via the Internet. Page guideline, 5-10.

Define the network security architecture for GIAC Enterprises. Your architecture should consider access requirements and restrictions for:

Customers
Suppliers
Partners
GIAC Enterprises employees on the internal network
GIAC Enterprises remote users
The General public

Briefly describe how each of the groups will interact with GIAC Enterprises including required ports, protocols, and any restrictions that will be required.

In designing your architecture, you must include the following components:

Filtering Router(s)
Firewall(s)
VPN(s)
Network based IDS sensor(s)
An IP addressing scheme

You may optionally include additional components if they are appropriate to your design.

You must include a network diagram showing the location and IP of all security components. Provide the specific brand and version of each component. Your design should use layered security (defense-in-depth) to capitalize on the respective strengths of each component while being flexible enough to choose components based on technical, budgetary, and political constraints of the company.

You must include a discussion of how the architecture adheres to the principle of Defense-in-Depth. You must state for each component:

What is its purpose?
What security function does it perform?
How does its placement affect that function?
What are the security weaknesses and strength of that component?

How do you mitigate these weaknesses using Defense-in-Depth?
What technical, budgetary, or political factors influenced the decision to use it?

The network can be as complex or as simple as you like as long as it meets the functional requirements that you define according to the guidelines given above. The important thing is not how elaborate your network is, but that the design actually works to meet the access and security requirements of the company.

Corporate Overview

GIAC Enterprises is a world-renowned company known for their creative and thoughtful Fortune Cookie Fortunes. This is a very competitive market, so security is a primary concern to the company. The company's sales model is based solely on the Internet for sales and marketing. Providing corporate network access on a 24x7 basis to their remote sales force and customers is imperative due to their world wide sales. Even though GIAC Enterprises, GIAC-E, is a company of 50 employees, they are very computer savvy. They have developed a sophisticated environment for access by their employees, partners, suppliers, and customers. Their corporate office is located in the United States with 4 International Regional offices. Primary computer support is provided from the corporate office.

When determining the network layout, equipment utilized, and software solutions the management stressed cost containment. To assist in reaching that goal, Open Source software packages were utilized whenever possible. If one was not available or did not meet the requirements, a commercial package was acceptable. The PC manufacturer, DELL, provides all servers and workstations. A single source of support was determined to be desirable by the company. Most of the servers use Red Hat Enterprise Linux Release 3 operating system. The remainder of the systems are some version of MS Windows.

Corporate Network Access

GIAC Enterprises has developed a strict Computer Security Policy for corporate employees and authorized business partners. All employees are required to adhere to this policy which they are required to review and sign yearly. A monthly review of assigned users-ids is performed to verify its status. This process includes employees, business partners and customers. A strict password policy for all users is also in force. Some of the restrictions include: 9 character, minimum 1 number, minimum 1 special character, non-dictionary terms, and changed on a quarterly basis. Previous passwords are not permitted.

Authorization to access the GIAC-E systems is based on the users relationship to the company. The company has defined the access levels as:

Customers (Companies or individuals that purchase bulk online fortunes)

Suppliers (Companies that supply fortune cookie fortunes)
Partners (International companies that translate and resell fortunes)
Employee internal network
Employee remote access
General Public

Customer

Access Requirements

As with any company, the customers are one of your most important assets. Providing them with the most up to date product information, availability, and order status is imperative. Therefore, access to this information is provided via the corporate web site. A secured web server provides sales, marketing, customer information, financial, order, and product information. The secured web server is accessible by any browser that supports HTTP and SSL protocols.

GIAC-E definition of Customer is: “Any Company or individual that purchases bulk online fortunes”. A company and an individual will have very similar access, but additional access is provided to Company’s due to their sales volume. Customers whom are identified as “Individuals” may only purchase the “Fortunes” by providing a credit card. Companies in “good” status may place orders by providing a purchase order otherwise a corporate credit card must be provided. All credit card purchases are authorized online and at time of entry. Once the order has been authorized, the customer has immediate access to the “Fortunes”. Downloading of purchased “Fortunes” can be accomplished via the secured web site. For security reasons, the Customer has only 24 hours to access the purchased “Fortunes”. After that time, they must contact Customer Service and be re-authorized for access.

Provided Services

Services provided for this level of user include:
Order entry and status
Customer Information
Download of purchased “Fortunes”
Product and Marketing information - Monthly Specials
Customer Support
Secured Email to GIAC Enterprises employees and/or customer service

Required Ports & Protocols

Table 1: Required Ports & Protocols for Customers

Port	Protocol	Description
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
80	HTTP	HyperText Transfer Protocol
		Protocol to transfer herptext requests and information between servers and browsers.
443	HTTPS/SSL	HyperText Transfer Protocol Secured, Secure Sockets Layer
		SSL is the protocol that provides encrypted communication beneath the application protocol HTTP.

Security Architecture

The externally accessible web server the users connect to is located in the corporate DMZ. In actuality, this web server is a reverse proxy server, i.e. SQUID (Chadd A). As requests are received from the internet, they are processed by SQUID and passed to the real web server located on the internal lan.

The web server provides both encrypted and non-encrypted pages. The encrypted pages are only accessible via a browser that supports HTTPS. The web servers are all Apache v.1.3 with current patches. The internally developed web applications that reside on the server access an Oracle Data Base server also located on the internal lan. All incoming and outgoing SMTP mail is pre-processed by a server located in the DMZ. Once the email has been analyzed by the pre-processor it is then forwarded to the Exchange server located on the internal lan.

Suppliers

Access Requirements

GIAC Enterprises is the world's largest reseller of Fortune Cookie Fortunes. They have suppliers from all over the world providing these "Fortunes". All suppliers have the same sales model as GIAC-E, 100% Internet sales. Most purchased "Fortunes" are in the English language. But, some are written in the local language of the supplier.

Most purchases are accomplished online with the Fortunes being immediately available for download. These orders are performed online, usually only providing a purchase order. Some smaller suppliers require a corporate credit card at the placement of the order. The download of the Fortunes is usually done from the customers secured web site. A few suppliers prefer GIAC-E to secure FTP the Fortunes from the supplier's server.

Only the Purchasing Department is allowed to place the orders and download the "Fortunes". The only authorized browser allowed for use by GIAC-E is Firefox (MozillaFoundation). A specifically configured version has been setup for use by the Purchasing Department at GIAC-E for placing orders.

Provided Services

Services provided for this level of user include:
Purchase Order Status
Supplier Information
Download of purchased "Fortunes"
Customer Support
Secured Email to GIAC Enterprises employees and/or customer service

Required Ports & Protocols

Table 2: Required Ports & Protocols for Suppliers.

Port	Protocol	Description
22	SSH	Secure Shell
		Application that provides secured encrypted communication between 2 untrusted hosts and over an untrusted network.
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
80	HTTP	HyperText Transfer Protocol
		Protocol to transfer herptext requests and information between servers and browsers.
443	HTTPS/SSL	HyperText Transfer Protocol Secured, Secure Sockets Layer
		SSL is the protocol that provides encrypted communication beneath the application protocol HTTP.

Security Architecture

Refer to the *Security Architecture* for the Customer section.

Partners

Access Requirements

GIAC-E defines a Partner as an "International Company that translates and resells the translated Fortunes". Given this, their access requirements are very similar to those of a "Customer". Providing them with the most up to date product information, availability, and order status is imperative. Therefore, access to this information is provided via the corporate web site.

GIAC-E Partners purchase bulk online fortunes. Companies in "good" status may place orders by providing a purchase order. Otherwise, all purchases are authorized against a corporate credit card. All credit card purchases are authorized online and at time of entry. Once the order has been authorized, the customer has immediate access to the "Fortunes". Downloading of purchased "Fortunes" can be accomplished via the secured web site. For security reasons, the Partner has only 24 hours to access the purchased "Fortunes". After that time, they must contact Customer Service and be re-authorized for access.

Provided Services

Services provided for this level of user include:

- Order entry, status
- Customer Information
- Download of purchased "Fortunes"
- Product and Marketing information - Monthly Specials
- Customer Support
- Secured Email to GIAC Enterprises employees and/or customer service

Required Ports & Protocols

Table 3: Required Ports & Protocols for Partners.

Port	Protocol	Description
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
80	HTTP	HyperText Transfer Protocol
		Protocol to transfer herptext requests and information between servers and browsers.
443	HTTPS/SSL	HyperText Transfer Protocol Secured, Secure Sockets Layer
		SSL is the protocol that provides encrypted communication beneath the application protocol HTTP.

Security Architecture

Refer to the *Security Architecture* for the Customer section.

GIAC-E Internal Network Employees

Access Definition

GIAC-E has 2 types of users on their corporate network, local corporate users and regional users. The regional users are connected via a VPN connection from their regional office to the corporate network. This VPN connection allows the regional office users to access the corporate network via an encrypted tunnel. Like most users, their access requirements are to the web server and the exchange server.

Since GIAC-E is primarily a marketing company all users require some network connectivity. The users fall into four groups, support for the external sales force, human resources, management, or technical support.

Provided Services

Services provided for this level of user include:

- Sales Support/Development Internal & DMZ located web servers
- Customer Support Internal & DMZ located web servers
- Purchasing of “Fortunes”
- Marketing Development

- Secured internal email
- Pre-processed incoming email
- Oracle Data Base Access

Required Ports & Protocols

Table 4: Required Ports & Protocols for Internal Employees.

Port	Protocol	Description
22	SSH	Secure Shell
		Application that provides secured encrypted communication between 2 untrusted hosts and over an untrusted network. (OpenBSD)
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
50	Ipsec-ESP	VPN IPSec protocol with IKE and ESP – Encapsulating Security Payload – Internet Key Exchange
		Used by the cisco routers to establish and encrypt the VPN connection.
80	HTTP	HyperText Transfer Protocol
		"A TCP based Protocol to transfer hypertext requests and information between servers and browsers." http://www.hyperdictionary.com/dictionary/http (Hyperdictionary)
443	HTTPS/SSL	HyperText Transfer Protocol Secured, Secure Sockets Layer
		SSL is the protocol that provides encrypted communication beneath the application protocol HTTP. http://www.hyperdictionary.com/dictionary/secure+sockets+layer (Hyperdictionary)

Security Architecture

The corporate office based employees have direct access to the internal lan based web server. They do not utilize the DMZ when accessing this server. Like all users they must login to access the web server. There also a number of test web servers and test database servers the users and developers have access to.

The Technical Support group utilizes the secure shell protocol to access many of the servers. Using the secure shell utility verifies for the user the destination machine and encrypts traffic.(OpenBSD) No use of the telnet protocol is permitted.

The regional office users access the reverse proxy server similarly to any external user. If they are located at the regional office their traffic will be passed through the corporate VPN.

GIAC-E Employees Remote Users

Access Requirements

At GIAC-E there are two types of remote employees, sales support and technical support. This discussion is limited to only sales support. The sales force has been supplied with a corporate laptop for company use only. As with most sales people this has little meaning. The laptops have been configured with personal firewalls and virus protection. Every effort is made to restrict the user from changing the configuration, but changes have been made. Laptops are required to be updated by PC support staff on a regular basis. Access to the Internet is assumed to be through a local ISP or some other unsecured network connection. Given these restraints, all of their access is limited to the web server that provides sales/marketing information, customer/order information, and web mail access.

Provided Services

Services provided for this level of user include:

- Order Entry/Status
- Customer Information
- Sales Development/Training
- Product and Marketing information - Monthly Specials
- Internal Sales Support
- Secured Email to GIAC Enterprises employees and/or customer service
- Exchange Secured Web Mail Access

Required Ports & Protocols

Table 5: Required Ports & Protocols for Remote Users.

Port	Protocol	Description
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
80	HTTP	HyperText Transfer Protocol
		Protocol to transfer herptext requests and information between servers and browsers.

443	HTTPS/SSL	HyperText Transfer Protocol Secured, Secure Sockets Layer
		SSL is the protocol that provides encrypted communication beneath the application protocol HTTP.

Security Architecture

Refer to the *Security Architecture* for the Customer section.

General Public

Access Requirements

GIAC-E provides a minimal website for the general public. The information available through the corporate website includes: GIAC-E Corporate information, product examples, customer support email, and local philanthropic activities.

Provided Services

Services provided for this level of user include:

- Non-secure email access to customer service
- Unsecured web access

Required Ports & Protocols

Table 6: Required Ports & Protocols for General Public.

Port	Protocol	Description
25	SMTP	Simple Mail Transfer Protocol
		This protocol supports the transfer of email between the customers email server and the GIAC-E email server. All incoming & outgoing email is processed by an email-preprocessor located in the DMZ.
80	HTTP	HyperText Transfer Protocol
		Protocol to transfer unencrypted hypertext requests and information between servers and browsers.

Network Design

Overview

As an International Company whose sales model is internet-only sales, GIAC-E must provide access to their computer systems and yet provide the maximum protection to their whole computing environment. If their systems are unavailable the company stops making money.

For clarity purposes the network design has been divided into two pieces, the Corporate Office Layout and the Regional Office Layout. Each piece has its own purpose, requirements and security problems.

Corporate Office Overview

The Corporate Office provides most of the computing services for the company. All purchasing is performed out of this office. All servers that support other corporate functions are also located here. Some servers exist at the regional offices but for only local office support type functions.

The description has been broken into six functional areas: firewalls, DMZ, NMS, server, VPN, and Users. Each area provides certain services to the user base and also provides some overall layer of security.

Most of the servers specified in this document are utilizing Red Hat Enterprise V.3. At installation time, only software necessary for supporting the primary services was loaded. Each server only runs a very minimal number of services. This helps secure a server from several aspects. All of these servers are kept at a very current level of patch management and under a support contract. After configured for the application, security hardening was utilized via Bastille-Linux (Beale) and the Center for Internet Security Linux Hardening tool(CIS:Linux). Both tools are excellent sources for securing a box and learning more about security.

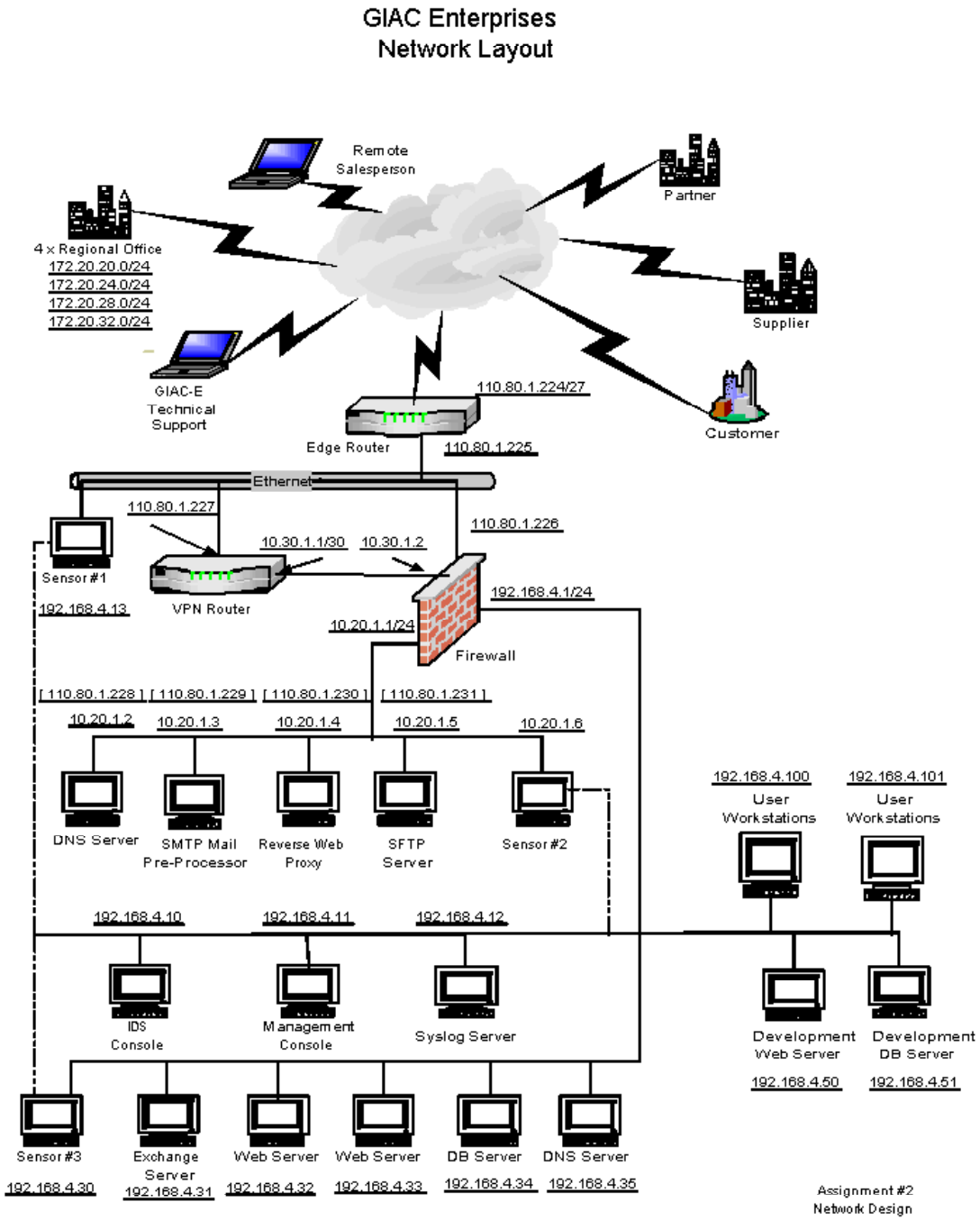


Figure 2: GIAC Enterprises Network Layout

Firewall – External Router Segment Overview

The components included in this segment include the external router and the firewall. These two systems provide the primary security function for the network. All traffic into and out of the network must pass through these two systems. Only specific services are allowed for use on the internal network. The external router provides the first level of protection by reducing the Internet noise that must be processed by the 2nd level, the firewall. Extensive rules have been developed for both devices.

Component – External Router

Purpose:
This router connects the T1 from the Internet to the local network. A serial interface card and an ethernet card are installed in the router. It is the initial restriction point for incoming traffic based on the source or destination ip address, service requested, or origination of session.

Security Function:
The ACL's defined on this device can reduce the amount of external noise from ever entering the local network. Reflexive ACL's are defined for each interface, serial and ethernet. Rules are generally defined from the perspective of traffic coming into the router, but outgoing rules can also be built. (SANSInstitute2.2). Therefore, most rules are defined for traffic coming into the router from the Internet and for traffic coming into the router from the local network.

Security weakness and how resolved by Defense-in-Depth:
Weaknesses of the router include the current Cisco IOS vulnerabilities and its inability to analyze VPN traffic. All routers are kept at a very current level of the IOS. A second problem deals with the VPN traffic coming from the regional offices. At this point, all traffic will be encapsulated and encrypted and the pack payload can not be examined. It may include traffic for services or computers that are not permitted. The regional office VPN router should not allow unauthorized service access, but again layers. The corporate VPN router and the firewall help resolve this problem. All external VPN traffic is sent to the corporate VPN router, which decrypts the traffic and applies its defined ACL rules. The permitted traffic is then passed to the firewall.

Strengths:
One of the major strengths is its ability to very quickly process the ACL rules against the incoming traffic. Extensive rules can be defined at this level to provide multiple layers of security. As with all corporate routers, they have been hardened using the Center for Internet Security Cisco Benchmarks and Audit Tool.(CIS:Cisco) <http://www.cisecurity.org/bench_cisco.html>

Device description:
The 1721 router was selected from the Cisco line. It is a midrange router with sufficient capabilities to meet our growth.
Cisco 1721, IOS 12.3, 1 serial and 1 ethernet interface, $2700

Component – Firewall

Purpose:
Analyze incoming/outgoing traffic and determine its validity based on defined network security rules. The valid traffic is passed to the appropriate destination interface. Depending on the type of blocked traffic, it may be logged before being thrown away.

Security Function:
The firewall provides the primary layer of security. All incoming/outgoing traffic can be analyzed for validity from the external router, VPN router, DMZ, or the internal lan network. This device also provides other functions such as NAT-ing. Incoming traffic destined for the web server and certain other systems have their destination address translated to an internal address before delivery. On return, the internal address is translated back to the original external destination address. For instance, the external ip address of the DNS server is 110.80.1.228 and the actual internal address is 10.20.1.2. This obfuscation of the address may add some complexity to externally mapping the internal network.

The firewall is a Linux based machine using netfilter/iptables as its firewall product. (NetfilterCoreTeam) For each interface a specific set of rules are defined for validating traffic. The rules include validation of: source and destination addresses, source and destination ports, protocol types, session status, packet flags, and other miscellaneous packet characteristics. The rules also perform NAT-ing type functions. This allows for the substitution of external ip addresses for internal ip addresses and visa versa. The functionality of this device is described in greater detail in Assignment 3.

Security weakness and how resolved by Defense-in-Depth:
Possibly vulnerable to the newest Linux or netfilter hack. Every effort is made to keep this machine as current as possible for all operating system products. Only the minimally required services have been started. The Linux operating system has been hardened using the Bastille Hardening System for Linux. REF:www.bastille-linux.org. Also, the Center for Internet Security Linux Benchmark & Audit Tool was utilized.(CIS:Linux) Lastly, the change-auditing tool Tripwire, an Open Source tool, is also installed and run on a regular basis.(TripwireInc.)

Defense-in-Depth:
The extensive reflexive ACL rules defined on both the external and VPN routers provide a previous layer of security. Much of the noise from the Internet has

been eliminated. Also downstream, all of the internal servers are also running some type of firewall.

Strengths:
The strengths of Linux and IP Tables are its cost, configurability and flexibility. Only required services are installed and running. Only the minimally required software is actually loaded on the system. No system tools for development or other tools a hacker may find useful need to be installed. The configurability and ease of installation of Linux makes it very appealing. The Open Source firewall utility, netfilter, is also very well respected. It does require some time in learning the product but it has an extensive feature set. One area the product excels, is the logging of packet information to the system log. (Ziegler). Based on extensive logging criteria options a log entry can be created which contains extensive information about the packet structure.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 80 gig hdd, and costs $1300. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, and costs $350.

VPN Segment Overview

In order to support the Regional Offices, a VPN has been setup that connects them to the Corporate Office. Due to the performance requirements of supporting a VPN, a router only supporting the VPN was installed. The Regional Offices have a dual function router due to their small number of employees. Utilizing a VPN on the router is easier to support for the Corporate IT staff versus installing a VPN client on all remote workstations and laptops.

Component – VPN Router

Purpose:
Establish the VPN connection between the corporate office and the regional offices.

Security Function:
Establish a secure connection between the regional vpn router and the corporate vpn router. Perform all encryption and decryption of packets. Analyze packets for network security based on the defined acl's. The regional vpn routers should only allow authorized and necessary traffic for transport over the vpn.

Security weakness and how resolved by Defense-in-Depth:
Both routers may be vulnerable to a current Cisco router hack. All router IOS's are kept as current as possible. If the regional router is compromised it maybe used to route otherwise unauthorized traffic across the VPN. To this end,

extensive rules on the corporate vpn router have been defined to alleviate this problem. And of course, there is the next layer of security, the firewall.

Strengths:
The VPN allows the regional users to have access to any authorized services in the corporate lan without having a local pc based vpn solution. Given the location of the regional offices, support for a pc base vpn product could be expensive.

Device description:
The 1721 router was selected from the Cisco line. It is a midrange router with sufficient capabilities to meet our growth.
Cisco 1721, IOS 12.3, 2 ethernet interfaces, VPN module, $3000

Component – IDS Sensor

Purpose:
Collect and perform limited analysis of all network traffic for the VPN router and the Firewall.

Security Function:
GIAC-E has selected the Open Source Tool Snort for all of the network intrusion analysis.(Caswell B) Although, the primary function of this sensor is data collection, it can also be used for real-time analysis of traffic. For example, the installed sniffer Ethereal is available for viewing traffic in real-time.(Ethereal). It is an excellent product and again Open Source. All of the traffic collected by the sensor is retrieved on a 30 minute basis by the Central IDS Server. The data is retained on this server for at least a week.

Security weakness and how resolved by Defense-in-Depth:
Again, this machine is Linux so it has the weaknesses previously mentioned. The interface collecting the network traffic does not have an ip address assigned so it would be very had to direct traffic to this device. Also, the interface used to copy all collected traffic is moved over a second interface that is directly connected internal network. This second interface is only active when needed.

Strengths:
This system cannot be attacked directly from the outside due to the lack of an ip address. The interface used for moving the collected traffic inside is only active for a brief period of time. Again, all the previously stated techniques for hardening a Linux server have also been used for all IDS sensors.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 2 nic, 80 gig hdd, and costs $1360. The operating system

includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, and costs $350. The Open Source packages Snort v 2.3 and Ethereal v 10.8 are also installed.

DMZ Segment Overview

The DMZ is a segment of the internal network that is dedicated to servers that are externally addressable from the Internet. Since these systems are addressable from the outside they are prime targets for attack. To increase the overall security of each server, they only have one primary function. This narrows the focus for the security hardening required. Also, if a server is compromised the permitted traffic to the internal network is also still limited by the firewall.

The access to this segment is only via the firewall. Given this only access path, very restrictive rules can be written for the segment and specifically for each machine. For instance, only internal HTTP traffic can initiate a session with the HTTP proxy server. Inbound HTTP sessions from the internet are not allowed to access this server.

Component – External DNS Server

Purpose:
Resolve DNS queries for the externally address space assigned to GIAC-E and resolve internal queries from the internal DNS server.

Security Function:
The DNS server is configured to answer external non-recursive queries for the GIAC-E external address space. External queries for internal addresses are blocked. Internal network queries are accepted from other machines located in the DMZ and the internal DNS server. The internal DNS server resides on the internal network and resolves all queries for internal lan. If a query is not resolved by it, the query is sent to the DMZ DNS server for resolution. Zone transfers are only permitted to GIAC-E ISP's DNS servers, which function as secondary servers for GIAC-E domain. A netfilter/iptables firewall is also configured on this server.

Security weakness and how resolved by Defense-in-Depth:
May be susceptible to the Linux, iptables, and Bind vulnerabilities. For DNS traffic to reach this server it would have been accepted by the router and the firewall. This server is also running a netfilter firewall and specifically configured for a limited number of services. It has been hardened in the standard fashion detailed earlier. As with the other servers located in the DMZ, only necessary services are installed and running.

Strengths:
Placing the DNS server behind the Firewall allows for several layers of protection as previously mentioned. GIAC-E also has a separate internal DNS

server for resolving queries concerning the internal network layout. The new Bind features allow for the DMZ DNS server to know about the internal DNS server but can not query its database. But the internal DNS server can query the external DNS server for all name resolution. The only ports open for incoming traffic are ports 53, bind, and port 22, ssh.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 80 gig hdd, and costs $1300. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Bind 9.2.4, and costs $350.

Component – SMTP Mail Pre-Processor

Purpose:
To protect the internal Exchange Mail server from smtp type intrusions and insecure or unusual email traffic.

Security Function:
This pre-processor protects the internal Exchange server from several aspects. First, all GIAC-E incoming and outgoing smtp type traffic is handled by this server. The product that is in use is SMTPD.(Widdowson) Its only function is to receive email and place it in a queue for later processing by the scanner utility.

The functions of the scanner utility include: verification of source email address, verification of destination email address, restricting of attachment types, analysis for spam characteristics, and virus scan of the attachments. The scanner is a home grown utility. Virus scanning is performed by the McAfee Linux Virus Scanner engine. GIAC-E has an Enterprise License with McAfee so the Linux engine is part of the package. Once the email is analyzed and determined valid and clean, it is placed on another queue for delivery to Exchange server.

Security weakness and how resolved by Defense-in-Depth:
Possibly susceptible to Linux, Iptables, and SMTPD vulnerabilities. The SMTPD utility is quite limited in its capabilities. It consists of a enough smtp commands to receive email. The email must then be processed by two other utilities before delivery to the internal mail server.

Strengths:
The primary strength of this server is the path an email must take to be delivered to the internal exchange server. It is processed by three programs and placed on two different queues before its actual transmission to the internal server. Therefore, an exploit must be processed by three different programs before final delivery to Exchange. This design does not allow an external smtp server to

connect directly to the Exchange server.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 80 gig hdd, and costs $1300. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, sendmail 9.2.4, SMTPD 1.3, and costs $350. The scanner utility was developed by McIntyre & Associates, Inc.

Component – Reverse Proxy Web Server

Purpose:
Provide a secure front end to GIAC-E production web application server.

Security Function:
The reverse proxy server receives all incoming web traffic from the internet and pre-processes the request before passing it on to the internal web server. This pre-processing includes security checks, address checks, url, and url path validation. Once the internal web server has processed the request it is passed back to this reverse proxy server and forwarded out to the original requestor. This process protects the internal web server from direct connection to the outside.(SANSInstitute2.3)

Security weakness and how resolved by Defense-in-Depth:
May be susceptible to linux, iptables, and proxy server attacks. One negative aspect of the current version of the Squid proxy server is the SSL connection for an incoming user is terminated on this server. The traffic from this server to the internal web server will be unencrypted. This is of some concern to the management. The internal network is utilizing switches to reduce the possibility of sniffing. If this proves unworkable, a ssl tunnel program will be utilized to encrypt the traffic between servers until the feature is available in Squid.(Kumar)

Strengths:
External users/servers cannot access the internal web server directly. The proxy server is more hardened than an Apache web server. Also, by having the web server on the internal network, the database server it accesses can also be located internally. Therefore the database server is only accessible from the internal lan.

Device description – Technical specs – Budgetary stuff
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 80gig hdd, and costs $1300. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Squid v 2.5, and costs $350.

Component – Proxy Server

Purpose:
Act as a web proxy server for all internal users requesting external web access.

Security Function:
The Open Source tool Squid was selected as the proxy server of choice. Utilizing this tool extensive acl's can restrict the destination of all http/https traffic. The Acl's can be defined for a specific url and/or url path, and url's containing questionable words. Restrictions can be set for whom has access to the Internet. In GIAC-E case, the user must provide their windows userid and password.

Security weakness and how resolved by Defense-in-Depth:
May be susceptible to the Linux, Iptable, and Squid vulnerabilities. There is a support issue of configuring the users browser configuration to utilize a proxy. The firewall could also be configured to re-route all web traffic to the proxy server in the future.

Strengths:
Help restrict unauthorized use of the internet by requiring a login and password. The proxy server can also cache some web pages thereby improving response and limiting outbound traffic. Extensive analysis of the http request and url is performed. Using the squid product for both proxy servers helps reduce support costs by having to be knowledgeable about only one product. Lastly, additional products to support pop-up and banner ads blocking is available for SQUID. (SANSInstitute2.3). This feature may be installed at some later point it time.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 80gig hdd, and costs $1300. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Squid v 2.5, and costs $350.

Component – IDS Sensor

Purpose:
Collect all traffic in the DMZ for later analysis by the IDS Server. Please refer to the previous discussion on functionality and configuration.

NMS Segment Overview:

These servers support the ongoing security, reliability, and overall network

performance monitoring function. Logs collected from all servers and networking devices are stored on these servers. Performance statistics of servers and network devices is collected and analyzed. Network re-configurations are performed from these machines and backups are maintained.

Component – IDS Sensor

Purpose:
Collect traffic on the internal network segment and also functions as a sniffer for analyzing real-time network traffic. Please refer to the previous discussion of this device.

Component – IDS Server/Console

Purpose:
Act as a repository for all network traffic collected by the IDS sensors and analysis console for utilizing the IDS software.

Security Function:
This server utilizes the following Open Source tools: Snort (Caswell B), MySql (Axmark D), and Analysis Console for Intrusion Databases (ACID) (Danyliw). Using these tools all traffic is loaded into a database and analyzed for packet anomalies, unusual traffic patterns, and unusual traffic sequences. An extensive array of reports and web pages is generated concerning the analysis of the traffic. Also, alerts via email or pager can be configured to alert the on-call tech of any problems.

Security weakness and how resolved by Defense-in-Depth:
The analysis of the traffic is only as good as the Snort Rule set. It is downloaded as often as possible. A zero day attack probably will not be identified due to the lack of an appropriate rule. The rules are maintained and modified relatively quickly. The collected traffic contains sensitive and non-sensitive corporate data. Access to this system is restricted to Tech Support personnel only. The system is running an iptables firewall with very restrictive rules.

Strengths:

All traffic data is located in one database for analysis. From the information gathered, additional system security can be added to a particular server or firewall rules modified for increased network security. Extensive list of third party applications are available for the Snort IDS.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The higher level configuration includes: 2 cpu's (1ghz), 2gig memory, 1 nic, 4-80gig hdd, and costs $3000. The operating

system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Snort 2.5, MySql 4.1.8, ACID 0.9.6, and costs $350.

Component – Syslog Server

Purpose:
A central repository of system logs for all servers, network devices, and priority user workstations.

Security Function:
Most computer devices have the capability to generate logs containing important system or application status messages. In some instances the logs are reviewed by hand and in other cases by some special log analysis utility. These logs contain a wealth of information per the state of the machine, either from a systems, performance, security, or applications perspective. Many very good syslog analyzing utilities are available via Open Source (LogAnalysis.org) or commercially.

When the analysis is performed only on a single systems log it can lead one to a myopic view of their environment. If multiple systems collected and analyzed as a whole, a different perspective may be gained. For instance, analyzing network router logs, firewall logs, and the web server logs may provide a much clearer view of a particular problem or possible problem. To improve the usefulness of the logs as a whole, it is imperative that all devices use a common network timeserver. If system times vary greatly, much of the usefulness may be lost.

In order to improve the handling of logs from many machines, the Open Source utility syslog-ng is utilizied. (Balabit) This product can collect logs from multiple sources and parse them into different log files. (Ramsden)

Security weakness and how resolved by Defense-in-Depth:
The weakness with this process is many hours may have passed after an intrusion or problem has occurred. To help alleviate some of this problem, all Linux systems are running the Swatch utility in real-time against their logs.(Swatch) It does not have the analysis capabilities of Snort but it can identify some problems.

Strengths:
Collection of all traffic for later analysis can provide an enterprise wide view of who is accessing your systems and possibly what they have been doing.

Device description – Technical specs – Budgetary stuff
The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 4-80gig hdd, and costs $2500. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Swatch, syslog-ng, and costs $350.

Component – Network Management Console

Purpose:
Central location for managing, real-time monitoring, traffic collection, and configuration archiving for all network devices

Security Function:
This server was designed as the central location for storing all configurations for all router and network equipment. It also serves as a collect point and display of real-time statistics all network traffic. Utilizing the Open Source Tool MRTG, traffic statistics and device status information is collected continually and made available via a web interface. Using the Open Source tool flow-tools, network traffic can be analyzed. (Fullmer)

Additional information collected from this station including acl use statistics. This information can be quite helpful when it comes to evaluating the acl's from a performance or a security perspective.

Security weakness and how resolved by Defense-in-Depth:
This server is not addressable from the external network. It has limited services running and a netfilter firewall is installed and running.

Strengths:
Having all traffic data in one location helps from an analysis and monitoring perspective.

Device description:
The higher level configuration includes: 2 cpu's (1ghz), 2gig memory, 1 nic, 4-80gig hdd, and costs $3000. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, flow-tools, MRTG, and costs $350.

Server Segment Overview

This is where all production servers are located. These include the Exchange mail server, internal DNS server, web application server, database server, and other windows servers. The windows Domain controller and print servers have been hardened using the Center for Internet Security Windows Server Benchmarks.(CIS:Windows) All linux servers are running a netfilter firewall. Each server is hardened specifically to support its primary application.

User Segment Overview

This is the location of all user workstations, development web and database servers. Every effort has been made to separate this traffic from the production

servers.

Regional Office Network Design Overview

The regional offices are small in size. The largest supports six employees. The corporation wants to keep costs down, but security is still important. To this end, a VPN has been established between each regional office and the corporate office. It was determined supporting these VPN links was less costly than supporting VPN software for each user.

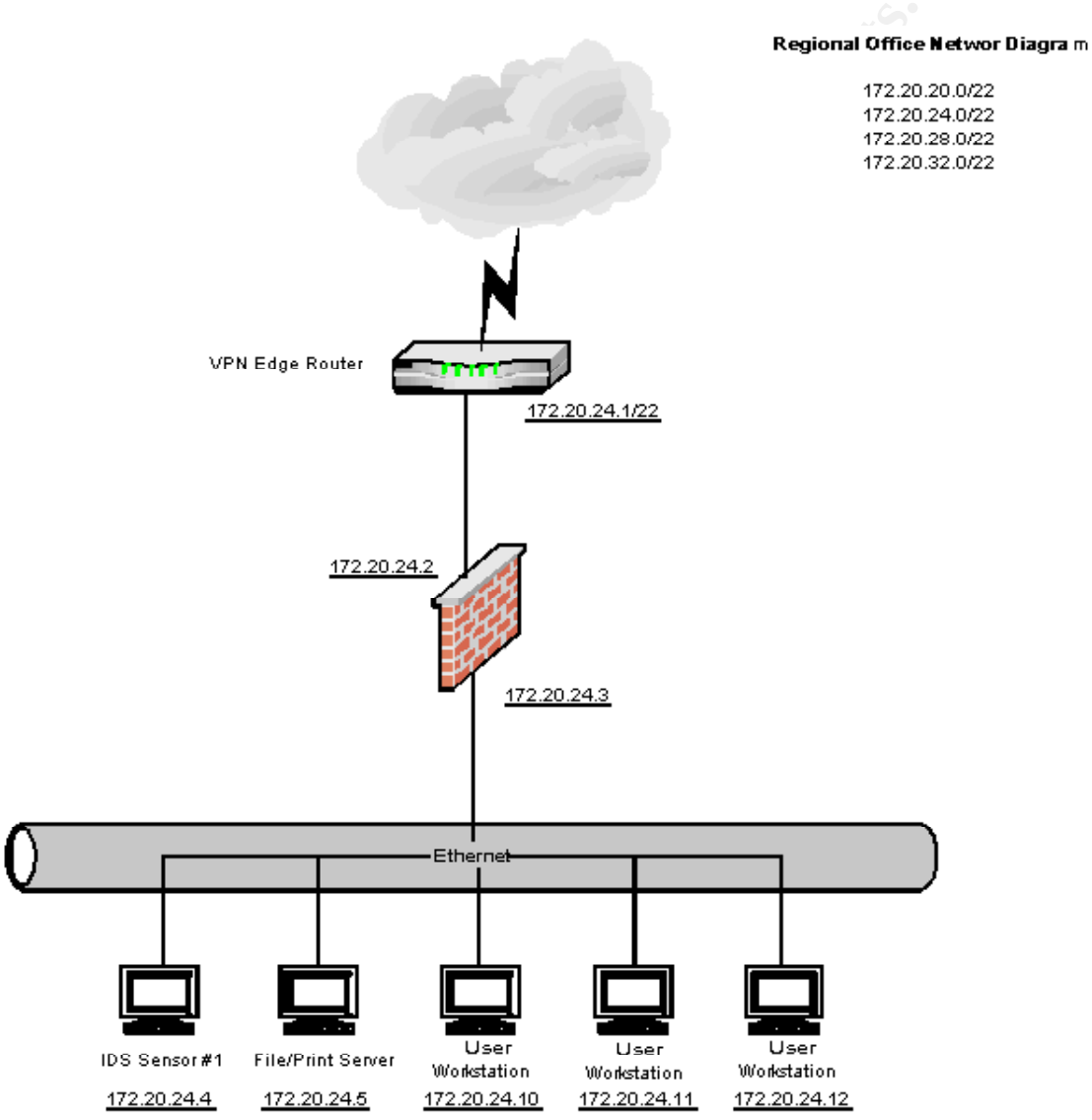


Figure 3. Regional Office Network Diagram

Firewall - VPN Segment Overview

These two servers provide all the network security for each location.

Component – VPN Router

Purpose:
Provides connectivity to the local internet ISP for the office network and VPN connectivity.

Security Function:
Provides the first and last line of defense for the network. It also provides the termination point for the corporate VPN. Only traffic destined for the corporate office should be sent over the VPN. All normal internet traffic should be routed directly to the internet.

Security weakness and how resolved by Defense-in-Depth:
May be susceptible to Cisco IOS and VPN vulnerabilities. The Cisco reflexive acl's do provide a statefull inspection of all traffic. The Linux firewall will be the next layer of protection. Also, the router logs are initially collected by the firewall and eventually passed to the corporate NMS server for analysis.

Strengths:
The Cisco router has been extensively hardened using the process mentioned earlier.

Device description:
The 1721 router was selected from the Cisco line. It is a midrange router with sufficient capabilities to meet our growth.
Cisco 1721, IOS 12.3, 1 serial and 1 ethernet interface, VPN bundle $3700

Component – Firewall

Purpose:
Provide a second level of network security for the internal network.

Security Function:
This Linux server utilizes the iptables firewall. It provides statefull inspection of all packets and also collects syslogs from the VPN router.

Security weakness and how resolved by Defense-in-Depth:
May be susceptible to Linux and iptables vulnerabilities. The router provides the last defense against any unusual traffic going to the internet. Having a remote Linux firewall may prove to be a support problem. If it does have a hardware replacement could be a problem. Discussed alternatives include a boot diskette based firewall or a hardware firewall.

Strengths:

Netfilter firewall provides extensive capabilities to monitor and analyze traffic. Real-time analysis of traffic is also available with a Linux firewall using a utility such as tcpdump or ethereal.

Device description:
Two levels of servers have been configured and both utilize the DELL Poweredge SC1420 platform. The lower level configuration includes: 1 cpu (1ghz), 1gig memory, 1 nic, 4-80gig hdd, and costs $2500. The operating system includes Red Hat Linux Enterprise V.3.1, iptables 1.2.11, Swatch, ethereal, tcpdump, and costs $350.

Assignment No. 3

Provide a rule base for the primary firewall defined in Assignment 2. A detailed description should be given of what each rule accomplishes and how it supports the security stance of the company. Include a discussion of the order of the firewall policy and why the order is, or is not, important. The firewall policy must accurately reflect the business needs and security considerations for all groups. Page guideline, 2-5.

Overview:
The firewall selected for this assignment is the corporate linux firewall. It was selected due to its extensive firewall and routing functions it must provide. It has four ethernet interfaces with traffic going in many directions. The rule base below is not a working rule set, just a close one. I believe all commands that would be required for a working set have been identified.

The selected firewall is netfilter. I intermix the terms netfilter and iptables in the documentation. Netrfilter refers to a set of hooks inside the linux kernel and ip tables is a generic table structure for the rule set. (NetfilterCoreTeam) This firewall was used on all linux servers and has proven to be very functional and well hardened. It is a standard package on most linux.

The original base for the rule set was acquired over four years and I do not know whom the original author to be. I believe it was originally available from the netfilter web site.(NetfilterCoreTeam)

The security policies reflected in the rules below should suffice the business needs & group needs as specified in Assignment 2. The default rule policy is drop. Therefore only traffic with a specific ACCEPT rule is allowed through.

The standard /etc/init/iptables was modified to execute this script rather than just a source of iptables commands. I have found this format easier to support and far more capable.

The following commands dynamically load modules that are required to the functionality of iptables. This section must be performed first and extends through to the message “done do first”.

The depmod command creates a dependency file, which is required by the kernel to be able to dynamically load the necessary iptables modules. (Linux)

```
/sbin/depmod –a
```

Verify the connection tracking modules are loaded and if not load them. Many kernels already contain these modules.(BLFS)

```
    /sbin/modprobe ipt_LOG
        /sbin/modprobe ipt_mac
        /sbin/modprobe ipt_state
        /sbin/modprobe ip_tables
        /sbin/modprobe iptable_filter
    /sbin/modprobe iptable_nat
        /sbin/modprobe ip_conntrack
    /sbin/modprobe ip_conntrack_ftp
        /sbin/modprobe ip_nat_ftp
    /sbin/modprobe ipt_REJECT
```

The following are miscellaneous variable definitions used throughout this script.

```
        IPTABLES="/sbin/iptables"         # $IPTABLES binary location
#
# --------------------------------------------------------------------
# Interface Declarations
#
        LOOPBACK="127.0.0.0/8"       # Loopback interface
        LAN_IF="eth0"                  # LAN internal interface
        EXT_IF="eth1"                  # External INTERNET interface
    VPN_IF="eth2"               # VPN interface
    DMZ_IF="eth3"               # DMZ interface
#
        LAN_IF_ADDR="192.168.4.1"     # internal interface addr
        EXT_IF_ADDR="110.80.1.226"    # external interface addr
        VPN_IF_ADDR="10.30.1.2"       # VPN interface addr
        DMZ_IF_ADDR="10.20.1.1"       # DMZ interface addr
#
        INT_LANS="192.168.4.0/24"     # All internal lans
        VPNSUBNET="172.20.0.0/19"     # Regional office VPN's
    EXTROUTER="110.80.1.225"          # external router
#
# --------------------------------------------------------------------
# IP Network Declarations

#
        ANY="any/0"                              # anywhere
        CLASS_A="10.0.0.0/8"                     # Class A Private Network
        CLASS_B="172.16.0.0/12"                        # Class B Private Network
        CLASS_C="192.168.0.0/16"                 # Class C Private Network
        CLASS_D_MULTICAST="224.0.0.0/4"   # Class D Multicast Network
        CLASS_E_RESERVED_NET="240.0.0.0/5"     # Class E Reserved
Network
        BROADCAST_SRC="0.0.0.0"                  # Broadcast Source
```

```
Address
        BROADCAST_DEST="255.255.255.255"        # Broadcast Dest.
Address
#
# --------------------------------------------------------------------
# Port Declarations
#
        PPORTS="0:1023"
        UPORTS="1024:65535"
    NETBIOS="137:139"
        WINYACK=”445”
    TRACEROUTE=”33434:33500”
    BOOTP="67:68"
    NTP="123"
    WHOIS="43"
#
# --------------------------------------------------------------------
# Internal servers
#
        INT_WWW="192.168.4.32"                  #WWW
        INT_DNS="192.168.4.35"              #DNS Server
        INT_DBS="192.168.4.34"              #Data Base Server
        SYSLOGHOST="192.168.4.12"           # syslog server
#
# --------------------------------------------------------------------
# External servers
#
        DNS1="204.117.214.9"                #ISP DNS 1
        DNS2="204.117.214.10"               #ISP DNS 2
        NTPSERVER1="198.82.162.213"             # NTP
        NTPSERVER2="129.6.15.28"        # time.nist.org
        EXT_MAIL="110.80.1.229"                 #SMTP server
        EXT_DNS="110.80.1.228"              #DNS server

# --------------------------------------------------------------------
#   DMZ servers
#
        DMZ_MAIL="10.20.1.3"                #SMTP server
        DMZ_DNS="10.20.1.2"                 #DNS server
        DMZ_WEB ="10.20.1.4"                #WEB server
        DMZ_PROXY=10.20.1.5                 #Outgoing Web Proxy Server
```

(Ziegler)

```
# --------------------------------------------------------------------
```

The following variable specifies ip address(es) or ranges from which traffic will

not be accepted. All incoming traffic source address will be tested against these addresses.

```
  BANNED_ADDR="24.184.185.220 24.184.185.246 61.218.64.76
190.100.7.255  \     255.255.255.255 64.76.208.0/24 63.167.46.98
38.114.21.104 38.114.21.13"

# ------------------------------------------------------------------------
```

The following code starts building the iptables rules/chains. First all rules are flushed or removed from the three default chains - INPUT, OUTPUT, and FORWARD. The INPUT chain defines traffic from any interface and destined for the firewall itself. The OUTPUT chain defines traffic originating from the firewall and destined for any interface. The FORWARD chain defines traffic originating from outside the firewall and not destined for the firewall, i.e. traffic that is just passing through. The mangle table contains rules for specialized packet routing flags. (Ziegler p90) The nat table contains rules for source and destination address/port translation. For a more detailed description of iptables, please see the many books or the iptables/netfilter website. (NetfilterCoreTeam)

```
#
# ------------------------------------------------------------------------
#
```

The following rules flush (-F) and delete (-X) any previously defined rules in the default chains.

```
        $IPTABLES -F INPUT
        $IPTABLES -F OUTPUT
        $IPTABLES -F FORWARD
        $IPTABLES -X
        $IPTABLES -t mangle -F
        $IPTABLES -t nat -F
        $IPTABLES -t mangle -X
        $IPTABLES -t nat -X

#------------------------------------------------------------------
```

Set the default policy to DROP for the default chains. Now only traffic that matches a rule will be passed to its final destination.

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

# ------------------------------------------------------------------------
```

Better explaination as to 1 in following rules. ??????

```
        # Still allow unlimited traffic on the loopback interface
        # This allows local apps to function but denys all other
        # traffic on all other interfaces
        # Inserting these rules forcefully at rule #1 may be overkill,
```

```
        # but how many previously theoretical exploits are now possible!

        $IPTABLES  --insert INPUT 1 -i ! lo -j DROP
        $IPTABLES  --insert OUTPUT 1 -o ! lo -j DROP
        $IPTABLES  --insert FORWARD 1 -j DROP

# ------------------------------------------------------------------------
Allow all loopback traffic from the firewall to the firewall.
        $IPTABLES -A INPUT  -i lo -j ACCEPT
        $IPTABLES -A OUTPUT -o lo -j ACCEPT

# ------------------------------------------------------------------------
```

The following commands define chains of rules permitting traffic destined for the firewall or traffic originating from the firewall. To help limit the number of rules in a chain, one has been defined for each ethernet interface path. This technique eases the burden of defining and the understanding of each chain. In some respects a chain can function like a subroutine, similar to a normal programming language. Only in this language once a packet match is made, processing for that packet terminates, i.e. accept or drop.

The chain-naming format I use is “interface source – interface destination”. Given the chain fw-lan, these rules define the traffic originating from the firewall(fw) and destined for the internal lan. Conversely, lan-fw, defines rules for traffic originating on the internal lan and destined for the firewall (fw). The other abbreviations include: dmz - indicates traffic to/from the DMZ ethernet interface, ext – traffic to/from the external internet interface, and vpn – traffic to/from the VPN interface. All chain names must be defined before use.

```
        $IPTABLES -N fw-lan
        $IPTABLES -N lan-fw
        $IPTABLES -N lan-if-udp-junk
#
        $IPTABLES -N fw-ext
        $IPTABLES -N ext-fw
        $IPTABLES -N ext-if-udp-junk
#
        $IPTABLES -N fw-dmz
        $IPTABLES -N dmz-fw
        $IPTABLES -N dmz-if-udp-junk
#
        $IPTABLES -N fw-vpn
        $IPTABLES -N vpn-fw
        $IPTABLES -N vpn-if-udp-junk

# ------------------------------------------------------------------------
```

The following commands define chains of rules for permitting traffic that is

“passing through” the firewall. For example, ext-dmz refers to traffic from the external internet interface (ext) and destined for the dmz. Conversely, dmz-ext refers to traffic from the dmz to the external internet interface.

```
        $IPTABLES -N ext-dmz
        $IPTABLES -N dmz-ext
#
        $IPTABLES -N ext-lan
        $IPTABLES -N lan-ext
#
        $IPTABLES -N vpn-lan
        $IPTABLES -N lan-vpn
#
        $IPTABLES -N vpn-dmz
        $IPTABLES -N dmz-vpn

# ------------------------------------------------------------------------
Define packet filtering chain for permitting ICMP Packets.
        $IPTABLES -N int-if-icmp
        $IPTABLES -N ext-if-icmp

# ------------------------------------------------------------------------
Define packet filtering chain for checking tcp flag settings.
        $IPTABLES -N badtcpflags

# ------------------------------------------------------------------------
Define packet-filtering chain for permanently banned addresses as defined
earlier by the BANNED_ADDR variable.
        $IPTABLES -N banned

# ------------------------------------------------------------------------
The chains defined below are used for logging messages and final action for the
packet. Each chain has a specific log message header that will appear in the
log with the actual packet in question. This can help tremendously for debugging
or intrusion analysis. Please refer to the actual rule definition further down in this
paper for actual functionality.
        $IPTABLES -N sanity
        $IPTABLES -N sanity-blocked
        $IPTABLES -N tcpflags-blocked
        $IPTABLES -N blocked

#DONE DO FIRST
# ------------------------------------------------------------------------
As described earlier, the INPUT chain defines rules for traffic coming from any of
the interfaces and destined for the firewall. Rules are defined based on the
source of the incoming traffic i.e. allowed traffic is  dependent on the source of
```

the traffic. These rules are order specific.

First we verify the tcp packet flags. The first rules analyzes traffic from the external internet interface $EXT_IF (eth1). If there were unusual settings, the packet would be dropped. Next based on the ethernet interface of the arriving packet pass it to the special chain for verification.

```
        $IPTABLES -A INPUT –p tcp -j badtcpflags
        $IPTABLES -A INPUT -i $EXT_IF -d $EXT_IF_ADDR -j ext-fw
#
        $IPTABLES -A INPUT -i $VPN_IF -d $EXT_IF_ADDR -j vpn-fw
#
        $IPTABLES -A INPUT -i $DMZ_IF -d $EXT_IF_ADDR -j dmz-fw
#
        $IPTABLES -A INPUT -i $LAN_IF  -j lan-fw
#
```

At this point the packet has not matched any ACCEPT rule, so we log it and finally drop it from existence. We have specified a limit for the number of times this message that can be generated in a 1 second time frame, just in case.

```
        $IPTABLES -A INPUT -j LOG  --log-prefix "INPUT block:" -m limit
        --limit 1/second
        $IPTABLES -A INPUT -j DROP

# ------------------------------------------------------------------------#
```

The following chains are part of the default chain FORWARD. These rules define what traffic will be passed from one interface to another. Again, the chains have been broken down by logical path. The lan-ext chain defines rules for traffic leaving the lan and destined for the internet. This group of commands should be kept together.

```
        $IPTABLES -A FORWARD -j badtcpflags
        $IPTABLES -A FORWARD -i $LAN_IF    -o $EXT_IF -j lan-ext
        $IPTABLES -A FORWARD -i $LAN_IF    -o $LAN_IF -j lan-lan
        $IPTABLES -A FORWARD -i $LAN_IF    -o $VPN_IF -j lan-vpn
        $IPTABLES -A FORWARD -i $LAN_IF    -o $DMZ_IF -j lan-dmz

        $IPTABLES -A FORWARD -i $DMZ_IF    -o $EXT_IF -j dmz-ext
        $IPTABLES -A FORWARD -i $DMZ_IF    -o $VPN_IF -j dmz-vpn
        $IPTABLES -A FORWARD -i $DMZ_IF    -o $LAN_IF -j dmz-lan

        $IPTABLES -A FORWARD -i $VPN_IF    -o $EXT_IF -j vpn-ext
        $IPTABLES -A FORWARD -i $VPN_IF    -o $DMZ_IF -j vpn-dmz
        $IPTABLES -A FORWARD -i $VPN_IF    -o $LAN_IF -j vpn-lan
```

At this point the packet has not matched any ACCEPT rule, so we log it and

finally drop it from existence. We have specified a limit on the number of these messages that can be generated in a 1 second time from, just in case.

```
        $IPTABLES -A FORWARD -j LOG  --log-prefix "FORWARD block:" -m
limit
        --limit 1/second
        $IPTABLES -A FORWARD -j DROP
#
# ------------------------------------------------------------------------
```

The following chains are part of the default chain OUTPUT. These rules define traffic generated on the firewall which can be sent to each interface. Again, the chains have been broken down by logical path. The fw-ext chain defines rules for traffic from the firewall (fw) and destined for the internet (ext).

```
        $IPTABLES -A OUTPUT -o $EXT_IF -j fw-ext
        $IPTABLES -A OUTPUT -o $LAN_IF -j fw-lan
        $IPTABLES -A OUTPUT -o $DMZ_IF -j fw-dmz
        $IPTABLES -A OUTPUT -o $VPN_IF -j fw-vpn

#
```

At this point the packet has not matched any ACCEPT rule, so we log it and finally drop it from existence. We have specified a limit for the number of times this message that can be generated in a 1 second time frame, just in case.

```
        $IPTABLES -A OUTPUT -j LOG  --log-prefix "OUTPUT block:" -m limit
        --limit 1/second
        $IPTABLES -A OUTPUT -j DROP
#
# ------------------------------------------------------------------------
                INPUT CHAIN RULES
# ------------------------------------------------------------------------
```

ext-fw

The following rules pertain to the traffic originating from the external interface and destined for the firewall itself.

```
        CHAIN="ext-fw"
```

At this point the packet should be structurally verified. If this packet is part of an established session, accept it and we are done.

```
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Before we do any real processing of the packet, verify it does not come from mars or is on our bad boy list, i.e. BANNED_ADDR.

```
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
```

If icmp traffic jump to the icmp rule chain. The packet will either be dropped or accepted. Control will not return to this point.

```
        $IPTABLES -A $CHAIN -p icmp -j ext-if-icmp
```

If udp traffic, go analyze it for being junk. If so we do not want to log it.

```
        $IPTABLES -A $CHAIN -p udp  -j  ext-if-udp-junk
```

Allow ssh traffic destined for the firewall. An entry will be made in the state table and the proceeding traffic will be accepted by the previous "ESTABLISHED,RELATED" rule.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```

Reject remaining traffic but first log it with an appropriate header for later analysis. Specifying the chain name in the log header helps from a diagnostic perspective and security analysis.

```
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

vpn-fw

In an effort to reduce the length of this document, previously defined rules will not be restated.

The following rules pertain to the traffic originating from the vpn interface and destined for the firewall itself.

```
        CHAIN="vpn-fw"
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
        limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

lan-fw

The following rules pertain to the traffic originating from the lan interface and destined for the firewall itself.

```
        CHAIN="lan-fw"
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Even though the traffic is originating from the internal lan, it is best to still verify

the addresses.

```
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
        limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
```

dmz-fw

The following rules pertain to the traffic originating from the dmz interface and destined for the firewall itself.

```
        CHAIN="dmz-fw"
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
        limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
                OUTPUT CHAIN RULES
# ------------------------------------------------------------------------
```

fw-ext

The following rules pertain to the traffic originating from the firewall and destined for the external interface.

```
        CHAIN="fw-ext"
```

At this point the packet should be structurally verified. If this packet is part of an established session accept it.

```
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Before we do any real processing of the packet, verify it does not come from mars or is on our bad boy list, i.e. BANNED_ADDR.

```
        $IPTABLES -A $CHAIN -j banned
```

We will allow any type of icmp packet the firewall builds.

```
        $IPTABLES -A $CHAIN -p icmp -j int_if_icmp
```

If udp traffic, go analyze it for being junk. We should not generate any of this but check it anyway.

```
$IPTABLES -A $CHAIN -p udp  -j  ext-if-udp-junk
```

Allow ssh traffic originating from the firewall interface.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```

Allow all the traceroute ports. Sequential ports can be specified as show below.

```
$IPTABLES -A output-lan-if -p udp --dport $TRACEROUTE -j ACCEPT
```

Allow whois to get to the outside.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport $WHOIS  -j
ACCEPT
```

Reject remaining traffic but first log it with an appropriate header for later analysis.

```
$IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
--limit 1/second
$IPTABLES -A $CHAIN -j DROP
```

accept outgoing reply to incoming ping----

```
iptables -A OUTPUT      -p icmp -m icmp --icmp-type echo-reply  -j ACCEPT
```

```
# ------------------------------------------------------------------------
```

fw-vpn

The following rules pertain to the traffic originating from the firewall and destined for the vpn interface.

```
CHAIN=”fw-vpn”
$IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A $CHAIN -p icmp -j ACCEPT
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
$IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
$IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
--limit 1/second
$IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

fw-lan

The following rules pertain to the traffic originating from the firewall and destined for the external interface.

```
CHAIN=”fw-lan”
```

```
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -p icmp -j ACCEPT
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```
Allow outgoing smtp traffic to the internal smtp server.
```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW –d $INT_MAIL --dport
        smtp –j ACCEPT
```
Allow system and kernel logging to central logging host.
```
        $IPTABLES -A $CHAIN -p udp -d $SYSLOGHOST --dport syslog -j
ACCEPT
        $IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
        limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
```
fw-dmz

The following rules pertain to the traffic originating from the firewall and destined for the external interface.
```
        CHAIN=”fw-dmz”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -p icmp -j int_if_icmp
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
        limit
        -- limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
                FORWARD CHAIN RULES
# ------------------------------------------------------------------------
```
The following rules pertain to the traffic originating from any interface and destined for another interface, ie the traffic is just passing through. The first rule refers to all packets originating on the lan interface (LAN_IF) and destined for the external internet interface (EXT_IF). Therefore, depending on the source and destination interface select the correct chain to use.

```
        $IPTABLES -A FORWARD -i $LAN_IF    -o $EXT_IF -j lan-ext
        $IPTABLES -A FORWARD -i $LAN_IF    -o $LAN_IF -j lan-lan
        $IPTABLES -A FORWARD -i $LAN_IF    -o $VPN_IF -j lan-vpn
        $IPTABLES -A FORWARD -i $LAN_IF    -o $DMZ_IF -j lan-dmz
```

```
        $IPTABLES -A FORWARD -i $DMZ_IF    -o $EXT_IF -j dmz-ext
        $IPTABLES -A FORWARD -i $DMZ_IF    -o $VPN_IF -j dmz-vpn
        $IPTABLES -A FORWARD -i $DMZ_IF    -o $LAN_IF -j dmz-lan

        $IPTABLES -A FORWARD -i $VPN_IF    -o $EXT_IF -j vpn-ext
        $IPTABLES -A FORWARD -i $VPN_IF    -o $DMZ_IF -j vpn-dmz
        $IPTABLES -A FORWARD -i $VPN_IF    -o $LAN_IF -j vpn-lan
        $IPTABLES -A FORWARD -i $EXT_IF    -o $DMZ_IF -j ext-dmz
#
        $IPTABLES -A OUTPUT -j LOG  --log-prefix "Forward block:" -m limit
        --limit 1/second
        $IPTABLES -A OUTPUT -j DROP
# ------------------------------------------------------------------------lan-ext
```

The following rules pertain to the traffic originating from the lan and destined for the external interface.

```
        CHAIN="lan-ext"
```

At this point the packet should be structurally verified. If this packet is part of an established session accept it.

```
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Before we do any real processing of the packet, verify it does not come from mars or is on our bad boy list, i.e. BANNED_ADDR.

```
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
```

If icmp traffic jump to the icmp rule chain. The packet will either be dropped or accepted. Control will not return to this point.

```
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
```

If udp traffic, go analyze it for being junk. Do not need to log all packets.

```
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
```

Allow ssh traffic originating from the firewall interface.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```

Allow outgoing smtp traffic to the internal smtp server.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW –s $INT_MAIL –d
        $DMZ_MAIL --dport smtp –j ACCEPT
```

Allow authentication for smtp servers.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW --dport auth
        -j ACCEPT
```

Allow outgoing http,https connections

```
$IPTABLES -A $CHAIN -p tcp –m multiport --dport www,https -j ACCEPT
```

Allow traceroute to anywhere.

```
$IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
```

Allow whois port.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport $WHOIS
-j ACCEPT
```

Reject remaining traffic but first log it with an appropriate header for later analysis.

```
$IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
--limit 1/second
$IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

lan-vpn

The following rules pertain to the traffic originating from the lan and destined for the vpn interface.

```
CHAIN=”lan-vpn”
$IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A $CHAIN -j sanity
$IPTABLES -A $CHAIN -j banned
$IPTABLES -A $CHAIN -p icmp -j ACCEPT
$IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
$IPTABLES -A $CHAIN -p tcp –m multiport --dport www,https -j ACCEPT
$IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
$IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
--limit 1/second
$IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

lan-lan

The following rules pertain to the traffic originating from the firewall and destined for the lan interface. Packets are confused, so send them on their way.

```
CHAIN=”lan-lan”
$IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A $CHAIN -j sanity
```

```
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
lan-dmz
```
The following rules pertain to the traffic originating from the lan and destined for the dmz interface.

```
        CHAIN=”lan-dmz”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j ACCEPT
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -p tcp –m multiport --dport www,https -j ACCEPT
        $IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
```

Allow internal dns calls only to the DMZ DNS only from the internal DNS server.

```
      $IPTABLES -A $CHAIN -p tcp -m state --state NEW –s $INT_DNS --sport
$UPORTS \
       -d $DMZ_DNS --dport domain -j ACCEPT
      $IPTABLES -A $CHAIN -p udp -m state --state NEW –s $INT_DNS --sport
$UPORTS \
       -d $DMZ_DNS --dport domain -j ACCEPT
```

Allow access to proxy server located in the DMZ. All users are expected to use the corporate proxy server for external web access.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  -d $DMZ_PROXY
        --dport $PROXY_PORT -j ACCEPT

        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
```
The following rules pertain to the traffic originating from the dmz and destined for the external interface.

CHAIN=”dmz-ext”

At this point the packet should be structurally verified. If this packet is part of an established session accept it.

```
$IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Before we do any real processing of the packet, verify it does not come from mars or is on our bad boy list, i.e. BANNED_ADDR.

```
$IPTABLES -A $CHAIN -j sanity
$IPTABLES -A $CHAIN -j banned
```

If icmp traffic jump to the icmp rule chain. The packet will either be dropped or accepted. Control will not return to this point.

```
$IPTABLES -A $CHAIN -p icmp -j int-if-icmp
```

If udp traffic, go analyze it for being junk.

```
$IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
```

Allow ssh traffic originating from the firewall interface.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j ACCEPT
```

Allow outgoing smtp traffic from the dmz smtp server.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW –s $DMZ_MAIL --
dport smtp –j ACCEPT
```

Allow authentication for smtp servers.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW --dport auth
-j ACCEPT
```

Allow outgoing http,https connections from the proxy server.

```
$IPTABLES -A $CHAIN -p tcp –m state –-state NEW –m multiport
        --dport www,https -j ACCEPT
```

Allow traceroute to anywhere.

```
$IPTABLES -A $CHAIN -p udp –m state –-state NEW --dport $TRACEROUTE
        -j ACCEPT
```

Allow whois port.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport $WHOIS
-j ACCEPT
```

Reject remaining traffic but first log it with an appropriate header for later analysis.

```
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
dmz-vpn
The following rules pertain to the traffic originating from the dmz and destined
for the vpn interface.
        CHAIN=”dmz-vpn”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j int_if_icmp
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -p udp --dport $TRACEROUTE -j ACCEPT
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
        --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
dmz-lan
The following rules pertain to the traffic originating from the dmz and destined
for the lan interface.
        CHAIN=”lan-dmz”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j int_if_icmp
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
Allow the DMZ mail server to send email inside.
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW –s $DMZ_MAIL –d
$INT_MAIL
                --dport smtp –j ACCEPT
Allow the reverse proxy server to call internal web server
        $IPTABLES -A $CHAIN -p tcp –m state –state NEW -d $INT_WEB –m
multiport
                --dport www,https -j ACCEPT
```

Allow system and kernel logging to central logging host.

```
        $IPTABLES -A $CHAIN -p udp -d $SYSLOGHOST --dport syslog -j
ACCEPT

        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit --limit 1/second
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

vpn-ext

The following rules pertain to the traffic originating from the vpn and headed to the internet. This traffic should not be coming thru here, it has its own interface to the internet. Just log it and quit.

```
        CHAIN=”vpn-ext”
```

Reject remaining traffic but first log it with an appropriate header for later analysis.

```
        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit --limit 1/second
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

vpn-dmz

The following rules pertain to the traffic originating from the vpn and destined for the dmz interface.

```
        CHAIN=”vpn-dmz”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j  ext-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```

The should allow the regional office access to the corporate web servers.

```
        $IPTABLES -A $CHAIN -p tcp –m state -—state NEW –m multiport –d
$DMZ_WEB
                --dport www,https -j ACCEPT

        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
                --limit 1/second
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```

vpn-lan
The following rules pertain to the traffic originating from the vpn and destined for the lan interface.

```
        CHAIN=”vpn-lan”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j int-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j  int-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
        $IPTABLES -A $CHAIN -p udp -d $SYSLOGHOST
                --dport syslog -j ACCEPT

        $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit
                --limit 1/second
        $IPTABLES -A $CHAIN -j DROP

# ------------------------------------------------------------------------
```

ext-dmz
The following rules pertain to the traffic originating from the external interface and destined for the dmz interface.

```
        CHAIN=”ext-dmz”
        $IPTABLES -A $CHAIN -m state --state ESTABLISHED,RELATED -j
ACCEPT
        $IPTABLES -A $CHAIN -j sanity
        $IPTABLES -A $CHAIN -j banned
        $IPTABLES -A $CHAIN -p icmp -j ext-if-icmp
        $IPTABLES -A $CHAIN -p udp  -j  ext-if-udp-junk
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW  --dport ssh -j
ACCEPT
```

Allow incoming smtp traffic from the outside.

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW –d $DMZ_MAIL --
        dport smtp –j ACCEPT
```

Skip the ident/auth packets. Most applications, smtp, will still work without it. Not responding would just delay the mail from arriving.(SANSInstitute2.3)

```
        $IPTABLES -A $CHAIN -p tcp -m state --state NEW --dport auth
                                                -j REJECT –reject-with tcp-reset
        $IPTABLES -A $CHAIN -p tcp –m state –state NEW –m multiport      \
                                --dport www,https –d $DMZ_WEB -j ACCEPT
```

Allow external dns calls only to the DMZ DNS. If tcp must be from our ISP DNS servers since they are providing secondary.

```
$IPTABLES -A $CHAIN -p tcp -m state --state NEW \
        -s $DNS1 -d $DMZ_DNS --dport domain -j ACCEPT
$IPTABLES -A $CHAIN -p tcp -m state --state NEW \
        -s $DNS2 -d $DMZ_DNS --dport domain -j ACCEPT
$IPTABLES -A $CHAIN -p udp -m state --state NEW --sport $UPORTS \
  -d $DMZ_DNS --dport domain -j ACCEPT

  $IPTABLES -A $CHAIN -j LOG --log-prefix "$CHAIN BLKD PCKT: " -m
limit --limit 1/second
  $IPTABLES -A $CHAIN -j DROP


# ------------------------------------------------------------------------
```

The following chain is used for analyzing udp type traffic coming from the lan. The internal lan is a noisy place and there is no need to log this traffic, just make it go away.

```
CHAIN="lan-if-udp-junk"
  $IPTABLES -A $CHAIN  -p udp --sport $NETBIOS  -j DROP
  $IPTABLES -A $CHAIN  -p udp --dport $NETBIOS  -j DROP
  $IPTABLES -A $CHAIN  -p udp --sport $WINYACK  -j DROP
  $IPTABLES -A $CHAIN  -p udp --dport $WINYACK  -j DROP
  $IPTABLES -A $CHAIN  -p udp --sport $BOOTP  -j DROP
  $IPTABLES -A $CHAIN  -p udp --dport $BOOTP  -j DROP
  $IPTABLES -A $CHAIN  -p udp --sport who  -j DROP
  $IPTABLES -A $CHAIN  -p udp --dport who  -j DROP

 If the port is not found, return to whence you came.
  $IPTABLES -A $CHAIN  -j RETURN

# ------------------------------------------------------------------------
```

The following chain is used for analyzing udp type traffic coming from the external internet interface. The internet is a noisy place and there is no need to log this traffic, just make it go away. This may need throw away some important security traffic. May need to log in the future.

```
  CHAIN="ext-if-udp-junk"
  $IPTABLES -A $CHAIN   -p udp --dport 161  -j DROP
  $IPTABLES -A $CHAIN   -p udp --dport $NETBIOS  -j DROP
  $IPTABLES -A $CHAIN   -p udp --dport $WINYACK  -j DROP
  $IPTABLES -A $CHAIN   -p udp --dport who  -j DROP
```

If the port is not found, return to whence you came.

```
  $IPTABLES -A $CHAIN -j RETURN

# ------------------------------------------------------------------------
```

The following commands check for stealth scans and unusual tcp flag settings. These rules are applied to all interfaces of a chain. (Ziegler p234-5) The basic format of the command is to first indicate the bits to test and then verify if the second set of bits are set. So the first rule specifies to test ALL bit-flags and if NONE are set then an error condition.

```
    CHAIN=badtcpflags
No flag bits set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags ALL NONE  -j tcpflags-blocked
Syn and Fin both set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags SYN,FIN SYN,FIN -j \
        tcpflags-blocked
Syn and Rst both set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags SYN,RST SYN,RST  -j \
        tcpflags-blocked
Fin and RST both set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags FIN,RST FIN,RST  -j \
        tcpflags-blocked
Fin set but ACK not set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags ACK,FIN FIN  -j tcpflags-
blocked PSH is set and ACK not set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags ACK,PSH PSH  -j tcpflags-
blocked URG set and ACK not set. Wrong.
        $IPTABLES -A $CHAIN -p tcp –-tcp-flags ACK,URG URG  -j tcpflags-
blocked If no match, return from whence we came.
        $IPTABLES -A $CHAIN -j RETURN
```

All of these commands above were purloined from Linux Firewalls, 2nd edition (Ziegler) for above commands. It is an excellent reference for anyone wanting to learn more about netfilter iptables.

```
# ------------------------------------------------------------------------
```

This chain logs packets that were determined to have unusual tcp flags set. The log record is sent to syslog with a special header. The packet is subsequently dropped. These are unusual packets so you probably want to know about them.

```
        $IPTABLES -A tcpflags-blocked -j LOG --log-prefix "tcp flags block: " -m
limit --limit 1/second
        $IPTABLES -A tcpflags-blocked -j DROP

# ------------------------------------------------------------------------
```

The following chain determines how to handle icmp packets. Please refer to the TCP/IP Illustrated, Volume 1 (Stevens p69) for more detail on icmp. Another very good reference is Linux Firewalls (Ziegler p171-5).

```
        CHAIN=ext-if-icmp
If the icmp packet is fragmented, this is not good. Please log it for later analysis.
```

(Ziegler)

```
        $IPTABLES -A $CHAIN -p icmp --fragment  -j LOG
                                                –log-prefix “icmp fragmented: “
        IPTABLES -A $CHAIN -p icmp --fragment  -j DROP

        $IPTABLES -A $CHAIN -p icmp --icmp-type redirect  -j DROP
        $IPTABLES -A $CHAIN -p icmp --icmp-type echo-request –m state -
–state NEW
                -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type destination-unreachable
                -j DROP
        $IPTABLES -A $CHAIN -p icmp --icmp-type source-quench -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type time-exceeded -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type parameter-problem -j
ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type fragmentation-needed -j
ACCEPT
```

If the packet is not allowed by above, make it go away.

```
        $IPTABLES -A $CHAIN -j DROP
```

```
# ----------------------------------------------------------------------
```

These are the rules for internally generated icmp messages. We are a little bit more relaxed since it is from the inside.

```
        CHAIN=int-if-icmp

        $IPTABLES -A $CHAIN -p icmp --fragment  -j LOG
                                                –log-prefix “icmp fragmented: “
        IPTABLES -A $CHAIN -p icmp --fragment  -j DROP

        $IPTABLES -A $CHAIN -p icmp --icmp-type redirect  -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type echo-request –m state
                –state new -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type echo-reply -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type destination-unreachable
                -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type source-quench -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type time-exceeded -j ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type parameter-problem -j
ACCEPT
        $IPTABLES -A $CHAIN -p icmp --icmp-type fragmentation-needed -j
ACCEPT
```

If the packet is not allowed by above, make it go away.

```
        $IPTABLES -A $CHAIN -j DROP
```

```
# ------------------------------------------------------------------------
```
The following loop builds a banned chain from the addresses defined in $BANNED_ADDR. We block incoming and outgoing packets.

```
        for banned_addr in $BANNED_ADDR ; do
          $IPTABLES -A banned -p tcp -d $banned_addr -j blocked
          $IPTABLES -A banned -p tcp -s $banned_addr -j blocked
        done
```

If the address is not found, return to whence you came.

```
        $IPTABLES -A banned -j RETURN
```

```
# ------------------------------------------------------------------------
```
The following chain verifies the source ip address for validity. If not valid, we go to sanity-blocked for logging and dropping. Please refer to the definitions at the beginning for the actual addresses being analyzed.

```
        $IPTABLES -A sanity -s $CLASS_A -j sanity-blocked
        $IPTABLES -A sanity -s $CLASS_B -j sanity-blocked
#       $IPTABLES -A sanity -s $CLASS_C -j sanity-blocked
        $IPTABLES -A sanity -s $CLASS_D_MULTICAST -j sanity-blocked
        $IPTABLES -A sanity -s $CLASS_E_RESERVED_NET -j sanity-blocked
```

This list includes the loopback, multicast, & reserved addresses.
The following are based on reservations as listed by IANA. Please regularly check for the current status of each of these ranges on IANA's webpage, <http://www.iana.org/assignments/ipv4-address-space>. (IANA)

```
#
      $IPTABLES -A sanity   -s 0.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 1.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 2.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 5.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 7.0.0.0/8 -j sanity-blocked
         $IPTABLES -A sanity   -s 10.0.0.0/8 -j sanity-blocked    ????
      $IPTABLES -A sanity   -s 23.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 27.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 31.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 36.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 37.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 39.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 41.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 42.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 49.0.0.0/8 -j sanity-blocked
      $IPTABLES -A sanity   -s 50.0.0.0/8 -j sanity-blocked
#      $IPTABLES -A sanity   -s 72.0.0.0/5 -j sanity-blocked
#      $IPTABLES -A sanity   -s 80.0.0.0/4 -j sanity-blocked
      $IPTABLES -A sanity   -s 96.0.0.0/3 -j sanity-blocked
```

```
$IPTABLES -A sanity   -s 169.254.0.0/16 -j sanity-blocked
$IPTABLES -A sanity   -s 192.0.2.0/24 -j sanity-blocked
$IPTABLES -A sanity   -s 197.0.0.0/8 -j sanity-blocked
$IPTABLES -A sanity   -s 224.0.0.0/4 -j sanity-blocked
```

If the address is not found, return to whence you came.

```
$IPTABLES -A sanity -j RETURN
```

```
# ------------------------------------------------------------------------
```

This chain logs packets that were determined to be without sanity to syslog usinf a special header. The packet is subsequently dropped.

```
$IPTABLES -A sanity-blocked -j LOG --log-prefix "sanity BLOCKED
PACKET: "
        -m limit --limit 1/second
$IPTABLES -A sanity-blocked -j DROP
```

```
# ------------------------------------------------------------------------
```

This chain logs packets that were determined to be unauthorized traffic. The packet will be logged to syslog with a special header. The packet is subsequently dropped. A limit of three packets per minute has been set for logging this type of message.

```
$IPTABLES -A blocked -j LOG --log-prefix "BLOCKED PACKET: " -m limit --
limit 3/minute
$IPTABLES -A blocked -j DROP
```

```
#------------------------------------------------------------------------
```

The following commands perform pre-routing and post-routing manipulation of a packet. Destination/source addresses/ports can be modified by these rules. This help hide the true internal address of a server. Also, the internal server can change ip addresses without concern to the external access to the server.

The following rule translates the external mail address to the internal mail address. Port verification will be in later rules.

```
$IPTABLES -t nat -A PREROUTING  -i $EXT_IF -d $EXT_MAIL -j DNAT
        --to-destination $DMZ_MAIL
```

This rule changes the source ip address for packet from the internal mail server ip address back to its external address.

```
$IPTABLES -t nat -A POSTROUTING -o $EXT_IF  -s $DMZ_MAIL -j SNAT
        --to $EXT_MAIL
```

The following rules are for the DNS server.

```
$IPTABLES -t nat -A PREROUTING  -i $EXT_IF -d $EXT_DNS -j DNAT
        --to-destination $DMZ_DNS
$IPTABLES -t nat -A POSTROUTING -o $EXT_IF  -s $DMZ_DNS -j SNAT
        --to $EXT_DNS
```

The following rules are for the WEB server.

```
        $IPTABLES -t nat -A PREROUTING  -i $EXT_IF -d $EXT_WEB -j DNAT
                --to-destination $DMZ_WEB
        $IPTABLES -t nat -A POSTROUTING -o $EXT_IF  -s $DMZ_WEB -j
SNAT
                --to $EXT_WEB
```

The following rules are for the Internal Proxy server.

```
        $IPTABLES -t nat -A PREROUTING  -i $EXT_IF -d $EXT_PROXY -j
DNAT
                --to-destination $DMZ_PROXY
        $IPTABLES -t nat -A POSTROUTING -o $EXT_IF  -s $DMZ_PROXY -j
SNAT
                --to $EXT_PROXY
```

The following route will place the external firewall address as the source for the remainder of the outbound traffic. For the most part this should be limited to the internal lan ip addresses. We do not want our non-routable addresses to reach the outside.

```
        $IPTABLES -t nat -A POSTROUTING -o $EXT_IF -s $LANSUBNET -j
SNAT
                --to $EXT_IF_ADDR

# ------------------------------------------------------------------
#       "Remove initial fail-safe Blocking Rules"
        $IPTABLES -D INPUT 1
        $IPTABLES -D FORWARD 1
        $IPTABLES -D OUTPUT 1
        #
# ------------------------------------------------------------------
```

The following commands are for enabling kernel monitoring support. This can be accomplished here or from one of the system startup scripts. Many of these setting are useful for any Linux system that needs to be hardened. (Ziegler), (CIS:Linux)

```
        #
        # Enable packet forwarding
        echo 1 > /proc/sys/net/ipv4/ip_forward
        #
        # Enable TCP SYN Cookie Protection
        echo 1 >/proc/sys/net/ipv4/tcp_syncookies
        #
        # Enable always defragging Protection
        echo 1 > /proc/sys/net/ipv4/ip_always_defrag
        #
```

```
# Enable broadcast echo protection
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
#
# Enable bad error message protection
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
#
# Enable IP spoofing protection, turn on Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done
#
# Disable ICMP Redirect Acceptance
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
   echo 0 > $f
done
#
# Disable Source Routed Packets
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
  echo 0 > $f
done
#
# log packets with impossible addresses.
for f in /proc/sys/net/ipv4/conf/$EXT_IF/log_martians; do
  echo 1 > $f
done
#
```

List of References:

The following materials were consulted in the preparation of this document.

Axmark D, Larsson A, Widenius M. Mysql. Computer software, 2004.

Balabit, IT Ltd. Balabit Syslog-Ng. Vers. 1.6.2. Computer software. Balabit IT Ltd, January 2003.

Beale, Jay. Bastille Linux. Vers. 2.x. Computer software, 2004.

BLFS, Developement Team. "Beyond Linux from Scratch: Version 1.0 Chapter 4: Security". 2001-2003. <http://www.fr.linuxfromscratch.org/view/blfs-1.0/postlfs/firewall.html >.

Caswell B, Roesch M. Snort(Tm). Vers. 2.3.0 RC2. Computer software, 2004.

Chadd A, Colins R, Nordstrom H, Rousskov A, Wessels D. Squid Web Proxy Cache. Vers. 2.5 Stable7. Computer software, 2004.

CIS:Cisco. "Center for Internet Security". 2004. Benchmarks/Tools: CIS Level-1 Level 2 Benchmarks and Audit Tool for Cisco IOS Routers and PIX firewalls. <http://www.cisecurity.org/bench_cisco.html >.

CIS:Linux. "Center for Internet Security". 2003. Benchmarks/Tools: CIS level-1 Benchmark and Scoring Tools for Linux. <http://www.cisecurity.com/bench_linux.html >.

CIS:Windows. "Center for Internet Security". 2004. Benchmarks/Tools: Benchmarks and Scoring Tools for Windows XP Professional, Windsows Server 2003, Windows 2000 and Windows NT. <http://www.cissecurity.org/bench_win2000.html >.

Danyliw, Roman. "Analysis Console for Intrusion Databases". 2003. <http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html >.

Ethereal. Vers. 0.10.8. Computer software, 2004. Unix, Linux, Windows.

Fullmer, Mark. "Flow Tools Information". 2003.

<http://www.splintered.net/sw/flow-tools >.

Galstad, Ethan. "Nagios". 2004. <http://www.nagios.org >.

Hyperdictionary. 2003. <http://www.hyperdictionary.com/ >.

IANA. "Internet Assigned Numbers Authority". 2004. (July 2004). <http://www.iana.org/assignments/ipv4-address-space >.

Kumar, Rajeev. "Firewalling Http Traffic Using Reverse Squid Proxy." SysAdmin February 2004 2004: 21-26.

Linux. Linux Man Pages. Vers. 2.4. Computer software, 2003.

LogAnalysis.org. 2004. The System Log: Logging News and Information. <http://www.loganalysis.org >.

MozillaFoundation. Mozilla. Vers. 5.0 (x11; U; Linux i686; rv 1.7.3). Computer software, 2004.

NetfilterCoreTeam. 2004. Netfilter Firewalling, NAT and Packet mangling for Linux 2.4. (2004). <http://www.netfilter.org >.

OpenBSD. Openssh. Vers. 3.9. Computer software, 2004.

Ramsden, Corey. "Centralized Logging for Unix, Windows and Network Devices." SysAdmin December 2004 2003: 32-35.

Rawland, Craig. Portsentry 1.2. Vers. 1.2. Computer software, 2003.

SANSInstitute2.2. Track 2.2 - Firewalls, Perimeter Protection & Virtual Private Networks: Firewalls. Vol. 2.2: SANS Press, 2004.

SANSInstitute2.3. Track 2.3 - Firewalls, Perimeter Protection & Virtual Private Networks: Firewalls. Vol. 2.3: SANS Press, 2004.

Stevens, W. Richard. Tcp/Ip Illustrated Volume 1, the Protocols. Vol. 1. Reading, Massachusetts: Addison-Wesley Publishing Company, 2000.

Swatch. Vers. 3.1. Computer software, 2004.

TripwireInc. Tripwire. Vers. 2.2.1. Computer software, 2004.

Widdowson, Liam. "Spammer Deterrent". 2002. Obtuse SMTPD 2.0 with Spammer Deterrent 1.3. <http://sd.inodes.org >.

Ziegler, Robert L. Linux Firewalls 2nd Edition. Indianapolis, Indiana: New Riders Publishing, 2002.